

Introdução ao Framework ZK



UNIVERSIDADE ESTADUAL DE MARINGÁ
PRÓ-REITORIA DE RECURSOS HUMANOS E ASSUNTOS COMUNITÁRIOS - PRH
DIRETORIA DE RECURSOS HUMANOS - DRH
DIVISÃO DE TREINAMENTO E DESENVOLVIMENTO - TDE

Introdução ao Framework ZK



Criação e Adaptação:
Walter Marcondes Filho - wmfilho@uem.br

Revisão:
Giancarlo Lucca - glucca@uem.br

Junho de 2016



ÍNDICE

Apresentação.....	1
Introdução ao Framework ZK.....	2
O que é um Framework?.....	2
O que é o Ajax?.....	2
O que é o ZK?.....	3
Quem usa ZK?.....	5
Visão Geral e Arquitetura.....	5
Ambiente de Desenvolvimento.....	7
Navegadores Compatíveis.....	7
Servidores Web.....	7
Ambiente de Desenvolvimento Integrado.....	7
Sinergias em Tecnologia Web.....	8
Instalação do JDK.....	8
Instalação do Servidor Web.....	8
Instalação do IDE.....	9
Instalação do Framework ZK.....	9
Criação do Projeto.....	10
Criando um Projeto com Framework ZK.....	10
Estrutura do Projeto.....	11
Componentes.....	12
Estruturação dos Componentes.....	12
Window.....	12
Label.....	13
Button.....	14
Popup.....	15
Componentes Básicos para Entrada de Dados.....	16
Grid.....	17
Tabbox.....	18
Combobox.....	20
Radiogroup.....	21
Groupbox.....	21
Listbox.....	22
Menu / Toolbar.....	24
Hlayout / Vlayout.....	26
Hbox / Vbox.....	27
Borderlayout.....	28
Div.....	28
Framework ZK e Código Java.....	29
Zscript e Linguagem de Expressão.....	29

Attribute name.....	30
Classe Java com a Interface Gráfica.....	31
Referências.....	34

Apresentação

Esta apostila foi desenvolvida para cumprir uma dupla função: a primordial, será a de servir de suporte aos cursos ministrados pela TDE da Universidade Estadual de Maringá.

A outra função será a de servir como um guia de referência básica, fornecendo subsídios ao usuário que não conhece o Framework ZK.

Na diversidade de alternativas apresentadas pelo Framework ZK, nem todas as possibilidades ou recursos são explorados, visto que este trabalho tem por objetivo fornecer apenas subsídios básicos aos seus usuários.

Esperamos que esta apostila proporcione um melhor aproveitamento do treinamento e desejamos que faça bom proveito do Framework ZK para atividades de desenvolvimento web.

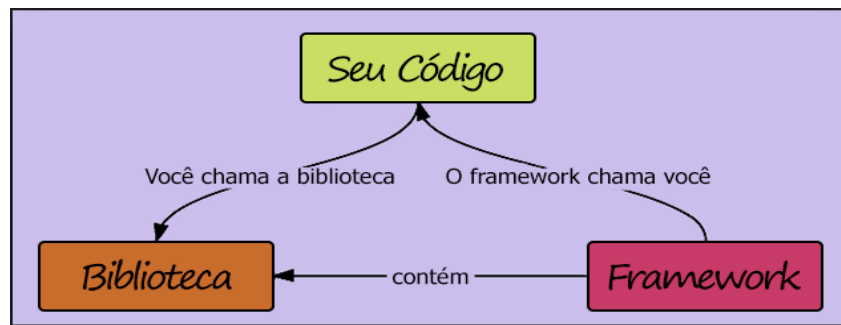
Atenciosamente,

Walter

Introdução ao Framework Zk

O que é um Framework?

É um conjunto de classes, interfaces e padrões dedicada a resolver uma grupo de problemas por meio de uma arquitetura flexível para permitir a construção de várias aplicações com pouco esforço, especificando apenas as particularidades de cada aplicação.



O que é o Ajax?

O AJAX (Asynchronous JavaScript and XML, requisição assíncrona de JavaScript e XML) é uma tecnologia utilizada atualmente, que está em bastante evidência porque ela torna seus aplicativos muito mais dinâmicos e com maiores capacidades de respostas. O Ajax basicamente utiliza JavaScript, XML e HTML dinâmica. Ajax não é uma linguagem de programação, mas uma nova forma de usar os padrões existentes.

No Ajax o Javascript faz uma solicitação ao servidores, no servidor nada muda com a utilização de Ajax, ele continua respondendo cada solicitação exatamente como fazia antigamente quando não se usava Ajax. Porém a resposta do servidor agora retornará apenas os dados que a página precisa, sem qualquer marcação ou apresentação. Uma característica que será notada pelo usuário que está acessando a página, é que grande parte da página não será alterada, mas sim apenas partes que necessitarão de atualização. Antigamente a página inteira era carregada, porém com Ajax esse paradigma muda.

A desvantagem da utilização do Ajax está relacionado ao tempo de codificação de Javascript e ao aprendizado do desenvolvedor web.

```
1  <head>
2  <script language="javascript" type="text/javascript">
3      var request = null;
4
5      function createRequest() {
6          try {
7              request = new XMLHttpRequest();
8          } catch (trymicrosoft) {
9              try {
10                 request = new ActiveXObject("Msxml2.XMLHTTP");
11             } catch (othermicrosoft) {
12                 try {
13                     request = new ActiveXObject("Microsoft.XMLHTTP");
14                 } catch (failed) {
15                     request = null;
16                 }
17             }
18         }
19     }
```

O que é o ZK?

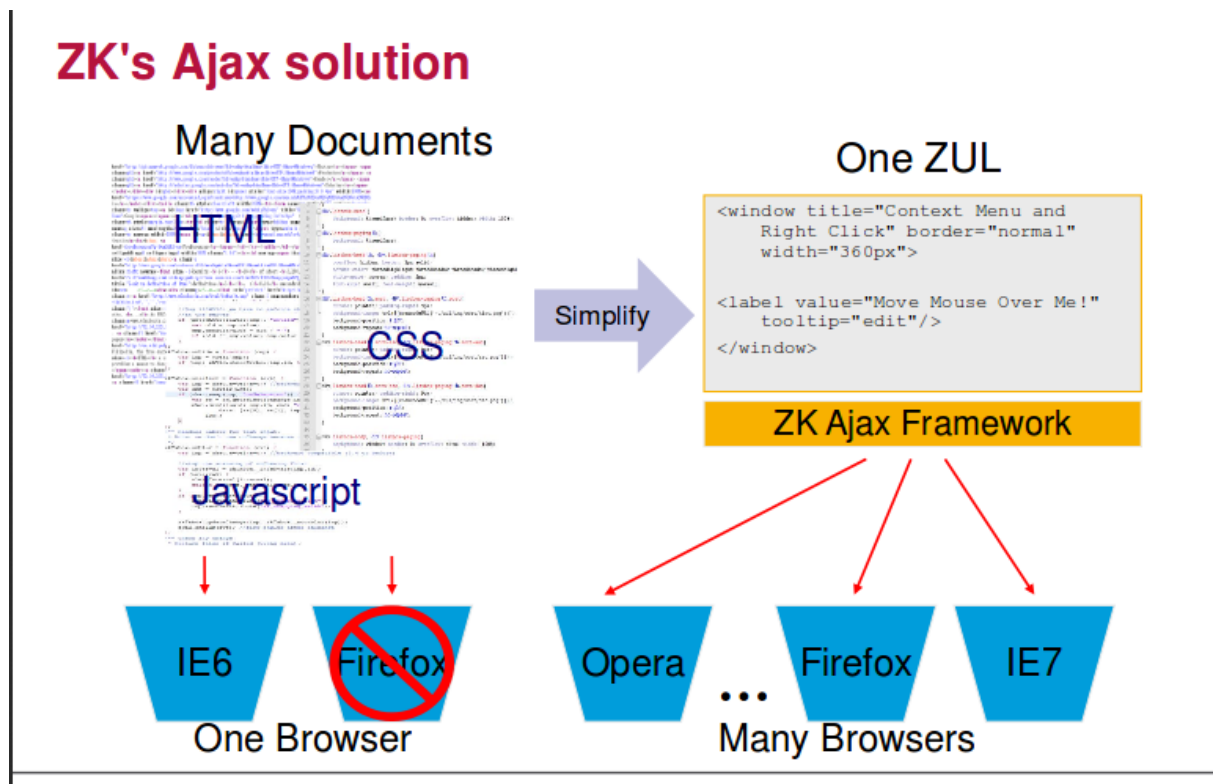
O ZK é um framework orientado a eventos para desenvolvimento web baseado em AJAX. Permitindo o desenvolvimento de interfaces ricas para aplicações web com pouca programação e um custo de desenvolvimento reduzido. Possui as seguintes características:

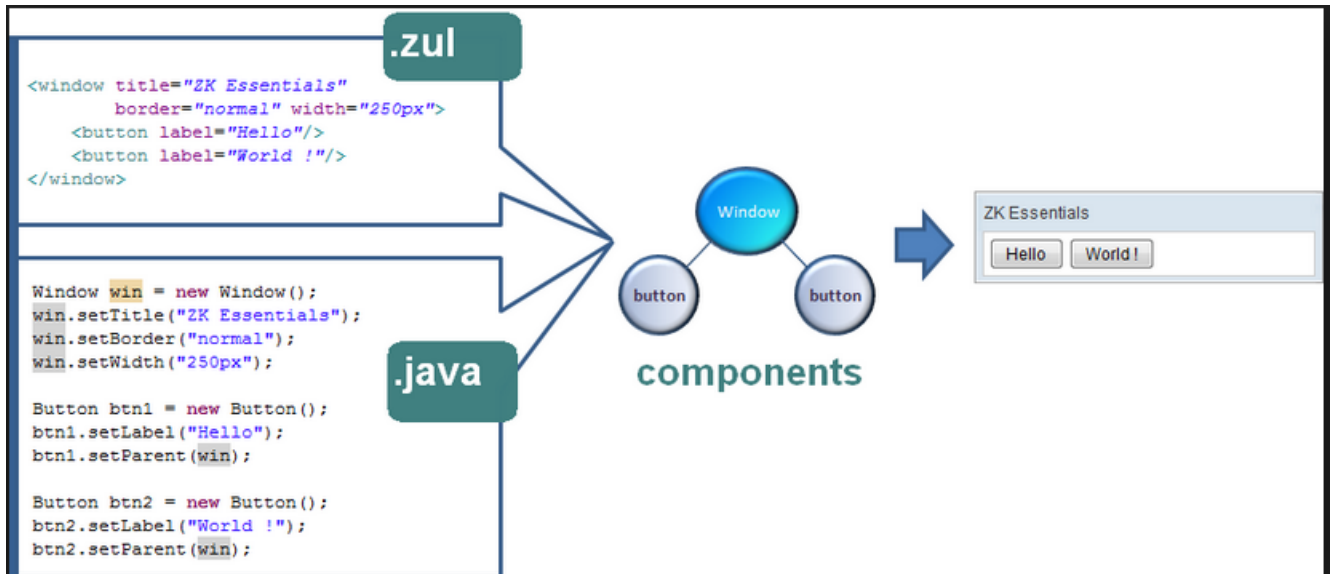
- Zk é um framework Open Source Ajax + Mobile, no entanto existem licenças pagas para componentes mais avançados;
- A comunidade de desenvolvedores ZK é extremamente ativa com mais de 20 traduções, mais de 100 artigos/blogs e mais de 100.000 linhas de código, mais de 1.500.000 downloads de mais de 190 países;
- Possui mais de 200 componentes prontos. Vários componentes de terceiros: JFreeChart, JasperReports, Google Maps, FCKeditor, Timeline, ExtJS, Dojo e mais;
- ZK estende o alcance de sua aplicação para os dispositivos móveis. ZK suporta Java Mobile, Android e vários navegadores móveis;
- Não é concebido para aplicações de gráficos vetoriais, tridimensionais e edição de fotos e vídeos;
- É concebido desde sua criação para ser seguro.

O ZK utiliza uma linguagem de meta-definição baseada em XML (ZUML) para definir a interface do usuário traduzindo para código HTML quando a página é solicitada pelo cliente.

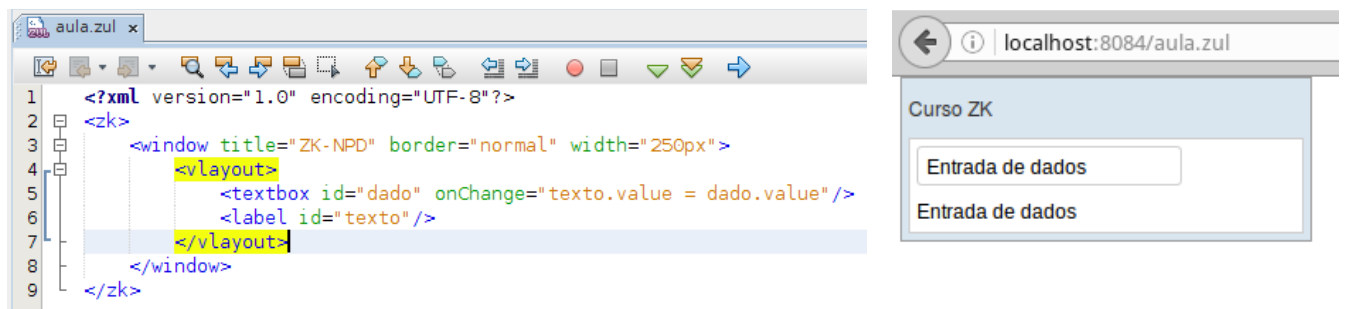
Foi desenvolvido para atender a camada de apresentação sem necessitar de nenhuma outra tecnologia para melhorar seu funcionamento. No entanto permite total integração com outras tecnologias como Hibernate, JDBC, Spring, JavaServer Faces e outras.

Portanto além de utilizar AJAX em background, se caracteriza por ser uma ferramenta para a camada de apresentação, que fornece um conjunto de componentes para Interface Gráfica do Usuário. Os componentes são declarados usando ZUML, em um arquivo ZUL gerando todo HTML, Javascript e CSS necessários.





Este exemplo abaixo, a entrada de dados do usuário no componente **textbox** é refletido no **label** abaixo instantaneamente quando o textbox perde o foco: A marcação declarada renderiza o programa.



Os componentes poderiam ser declarados totalmente em Java:

```

1 package Sistema;
2
3 import org.zkoss.zk.ui.Page;
4 import org.zkoss.zk.ui.GenericRichlet;
5 import org.zkoss.zk.ui.event.Event;
6 import org.zkoss.zk.ui.event.EventListener;
7 import org.zkoss.zul.*;
8
9 public class ExemploJava extends GenericRichlet {
10
11     public void service(Page page) {
12         final Window win = new Window("ZK-NPD", "normal", false);
13         win.setWidth("250px");
14
15         Vlayout vl = new Vlayout();
16         vl.setParent(win);
17
18         final Textbox dado = new Textbox();
19         dado.setParent(vl);
20
21         final Label texto = new Label();
22         texto.setParent(vl);
23
24         dado.addEventListener("onChange", new EventListener() {
25             @Override
26             public void onEvent(Event t) throws Exception {
27                 texto.setValue(dado.getValue());
28             }
29         });
30         win.setPage(page);
31     }
32 }

```


Quem usa ZK?

ZK capacita uma ampla variedade de empresas e instituições, que vão desde pequenas a grandes e várias indústrias. Empresas como a Barclays, Sun Microsystems, Swiss Re, Oracle, Alcatel-Lucent, State Grid, MMC, China Southern Power Grid), governos (EUA, Espanha, Japão, Austrália, China, Nova Zelândia) e dezenas de milhares de desenvolvedores em todo o mundo.



Visão Geral e Arquitetura

Cada objeto de interface é representado como um componente. Assim, compor componentes trata-se da montagem dos componentes. Para alterar a interface basta modificar os atributos e os eventos dos componentes.

Visão geral:

Usuario	Thiago Baesso Procaci	
Senha	
Peso:	83,02	
Data de Nascimento:	<div> <div>29/01/2009</div> <div>Gerar</div> <div>qwasdf</div> </div> <div> <div>Jan, 2009</div> <div> <div>Jan</div><div>Fev</div><div>Mar</div><div>Abr</div><div>Mai</div><div>Jun</div> <div>Jul</div><div>Ago</div><div>Set</div><div>Out</div><div>Nov</div><div>Dez</div> </div> <div> <div>Seg</div><div>Ter</div><div>Qua</div><div>Qui</div><div>Sex</div><div>Sáb</div><div>Dom</div> </div> <div> <div>29</div><div>30</div><div>31</div><div>1</div><div>2</div><div>3</div><div>4</div> <div>5</div><div>6</div><div>7</div><div>8</div><div>9</div><div>10</div><div>11</div> <div>12</div><div>13</div><div>14</div><div>15</div><div>16</div><div>17</div><div>18</div> <div>19</div><div>20</div><div>21</div><div>22</div><div>23</div><div>24</div><div>25</div> <div>26</div><div>27</div><div>28</div><div>29</div><div>30</div><div>31</div><div>1</div> </div> </div>	

Gera HTML, Javascript (incluindo CSS)

Código ZUML

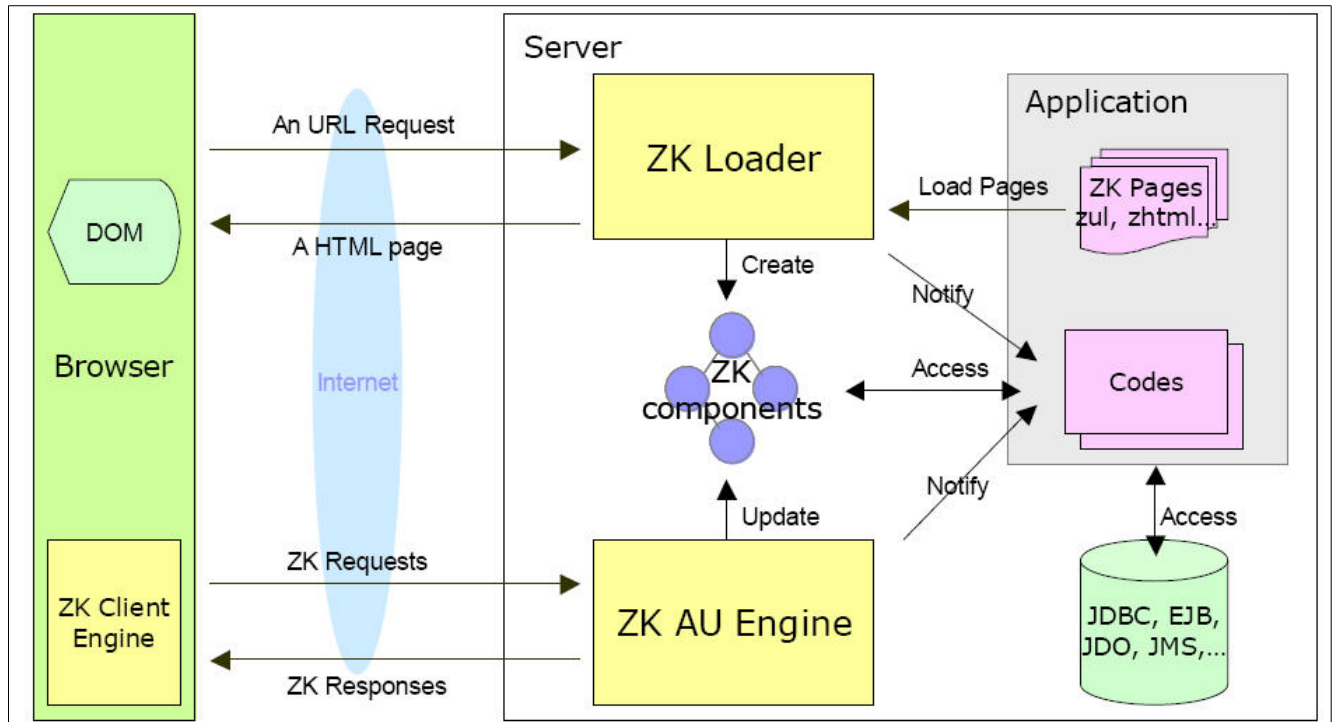
```
<grid width="600px">|
  <rows>
    <row>
      <label value="Usuario" width="150px" />
      <textbox value="Thiago Baesso Procaci" width="150px" />
    </row>
    <row>
      Senha
      <textbox type="password" value="123456789" width="150px" />
    </row>
    <row>
      Peso:
      <decimalbox format="###.##" value="83.02" width="150px" />
    </row>
    <row>
      Data de Nascimento:
      <hbox>
        <datebox id="db" width="150px" />
        <button label="Gerar" onClick="cpa.randomValue(); val.value=cpa.value;" />
        <vbox>
          <textbox value="qwasdf" width="150px" />
          <captcha id="cpa" length="6" width="150px" height="50px" />
        </vbox>
      </hbox>
    </row>
  </rows>
</grid>
```

Exemplo ZUML

Possibilidade de configurar atributos (como na decimalbox).

O input gerado assumirá valores nos moldes do atributo format

Arquitetura:



Estrutura de funcionamento do Framework ZK:

1. O cliente efetua sua solicitação por um endereço URL;
2. Um servidor web (Apache Tomcat, Jboss, GlassFish e etc) ativa o ZK Loader que interpreta a solicitação do cliente, gerando a página HTML, as folhas de estilo CSS, os códigos Javascript e a criação dos componentes ZK;
3. O ZK Loader envia a página HTML para o cliente e respectivamente envia os componentes ZK para o Atualizador Assíncrono (ZK AU Engine);
4. ZK AU Engine envia as atualizações dos elementos para o ZK Cliente Engine.

Como o ZK Cliente Engine se situa no lado do cliente para estabelecer o controle dos eventos AJAX no web browser, então ele se torna o responsável por receber eventos e atualizar a página no navegador do cliente.

Estrutura de funcionamento do AJAX:

1. Quando algum evento AJAX ocorre, devido a alguma ação do cliente, o ZK Cliente Engine envia a solicitação para o ZK AU Engine no lado do servidor;
2. Dessa forma, o ZK AU Engine recebe as solicitações e atualiza os componentes ZK e envia a resposta de volta para o ZK Cliente Engine.
3. Finalmente o ZK Cliente Engine atualiza o conteúdo com os elementos pertencente a página ativa no web browser.

Essas atividades se repetem enquanto o usuário estiver efetuando algum evento, as informações contidas na página serão atualizadas apenas nos elementos com os métodos dos eventos.

Ambiente de Desenvolvimento

Navegadores Compatíveis

Para acessar informações pela internet é necessário utilizar um programa web browser ou navegador web, o Framework ZK é compatível com os navegadores:

- Internet Explorer;
- Firefox;
- Chrome;
- Safari;
- Opera.



Servidores Web

Como o Framework ZK é utilizado para desenvolver aplicações para web na plataforma Java, ele precisa de um servidor web específico para executar, podendo ser:

- Servlet;
- Tomcat;
- Jboss;
- Jetty;
- Webshpere;
- Glassfish;
- WebLogic;
- Oracle WebLogic.



Ambiente de Desenvolvimento Integrado

IDE suportados:

- Eclipse;
- MyEclipse;
- NetBeans;
- IBM RAD;
- BEA Workshop;
- Oracle Jdeveloper.

Ferramentas:

- ZK Studio - Eclipse;
- Zats Minic.

Sinergias em Tecnologia Web

W3C Standards  HTML, HTML5, XHTML, XML, CSS, CSS 3, XUL	Testing Utilities  Selenium, JUnit, HP LoadRunner, Grinder, NeoLoad, ...	Reporting Libraries  JasperReport, JFreeChart, Flash Chart, Fusion Chart, BIRT
Java EE Compliance  Servlet 2.3/2.4/+, Portlet, JSP, JSF, JPA, EJB, JDO, CDI, JDK 1.4/5/6/+	Supporting Frameworks  Spring, ZK Spring, Grails, ZKGrails, Struts, Tiles, Liferay, JetSpeed 2, Hibernate, Oracle TopLink, Seam, Seasar...	Ajax UI Libraries  jQuery, Flash, CKEditor, SIMILE Timeline, SIMILE Timplot, Google Maps
Scripting Support  Java, Groovy, JavaScript, Ruby, Python, PHP, Scala, Clojure	Browser Support  IE 6/7/8/9, Firefox, Chrome, Safari, Mobile Safari, IE Mobile, Opera, Webkit, Konqueror, Mozilla	Mobile Platform Support  iPad, iPhone, Android, Blackberry, JavaME enabled mobile devices
Web Servers  Apache Tomcat, IBM WebSphere, IBM WebSphere Process Server, IBM WebSphere Portal, SAP NetWeaver, Spring tc Server, JBoss Application Server, BEA WebLogic, Oracle OC4J, Sun GlassFish, Jetty, Google App Engine, among other application server compliant with servlet 2.3 specification and above.		

Instalação do JDK

Java SE Development Kit (JDK) é um Kit de Desenvolvimento Java, ou seja, um conjunto de utilitários que permitem criar sistemas de software para plataforma Java. Ele contém todo o ambiente necessário para a criação e execução de aplicações Java, incluindo a máquina virtual Java (JVM), o compilador Java, APIs do Java e outras ferramentas utilitárias.

Efetue o download em <http://java.sun.com> ou no caso de usar o linux pode-se instalar o JDK por linha de comando:

Abra um terminal com o atalho **CTRL+ALT+T** e execute os seguintes comandos.

- Adicione o repositório do Web Upd8.
sudo add-apt-repository ppa:webupd8team/java
- Atualize a lista de pacotes disponíveis.
sudo apt-get update
- Finalmente, instale o pacote do JDK 7.
sudo apt-get install oracle-java7-installer
- Para verificar se a instalação funcionou, verifique a versão do JDK com o seguinte comando.
java -version
- Você também pode verificar a versão do compilador.
javac -version

Instalação do Servidor Web

O software Apache Tomcat é um servidor de aplicações Java para web, sendo distribuído como software open source.

Efetue o download no endereço <http://tomcat.apache.org>

Após instalado e ativado, verifique se está funcionando no navegador <http://localhost:8080>

Quando se instala o IDE Netbeans o Apache Tomcat pode ser instalado junto.

Instalação do IDE

O Framework ZK pode rodar em vários ambientes de desenvolvimento integrado, os mais utilizados são o Netbeans e o Eclipse.

Para a instalação do Netbeans pode-se efetuar o download pelo site www.netbeans.org, no caso do linux após baixar o “**netbeans-8.1-linux.sh**” faça a instalação por linha de comando:

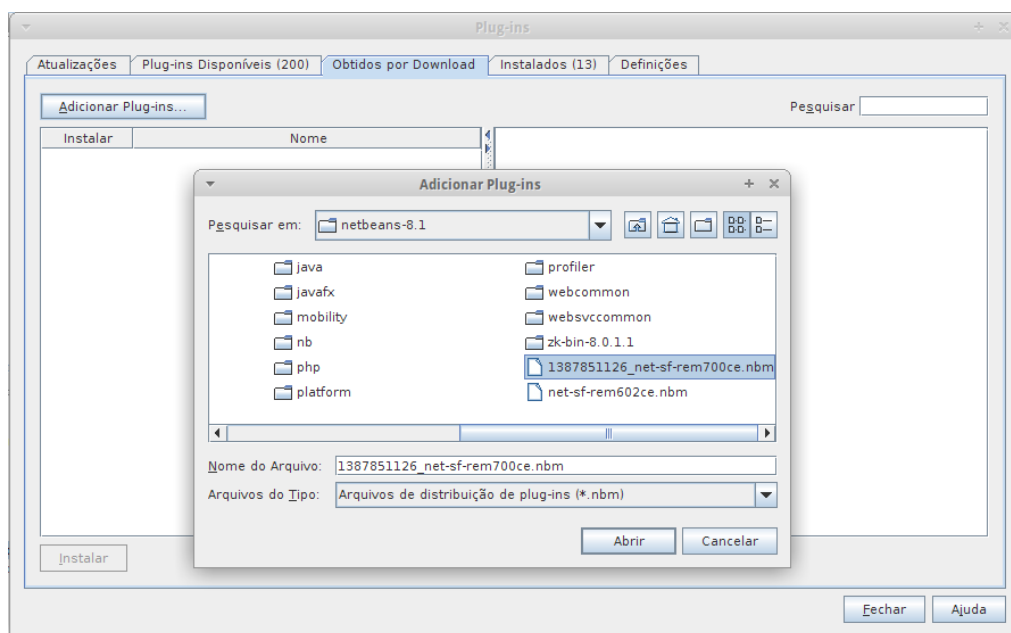
Abra um terminal com o atalho **CTRL+ALT+T** e execute os seguintes comandos.

- Execute o instalador.
./netbeans-8.1-linux.sh
- Adicione o servidor web **Apache Tomcat** em: forma de instalação / personalizar.

Instalação do Framework ZK

Para a instalação do Framework ZK no Netbeans, é necessário fazer um download do plugin **REM** que pode ser encontrado no site <http://sourceforge.net/projects/rem1>

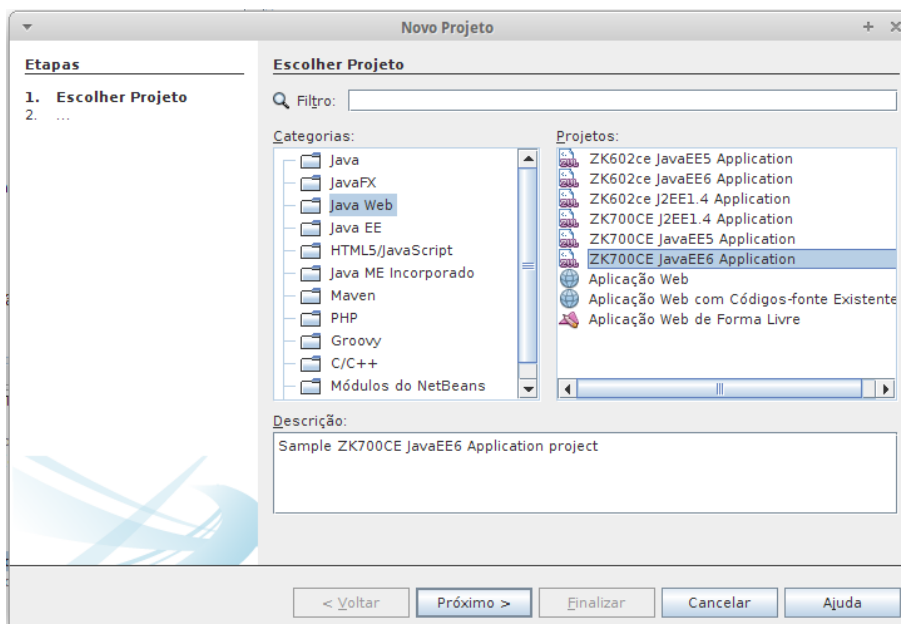
Em seguida abra o Netbeans selecione a opção Ferramentas/Plug-ins



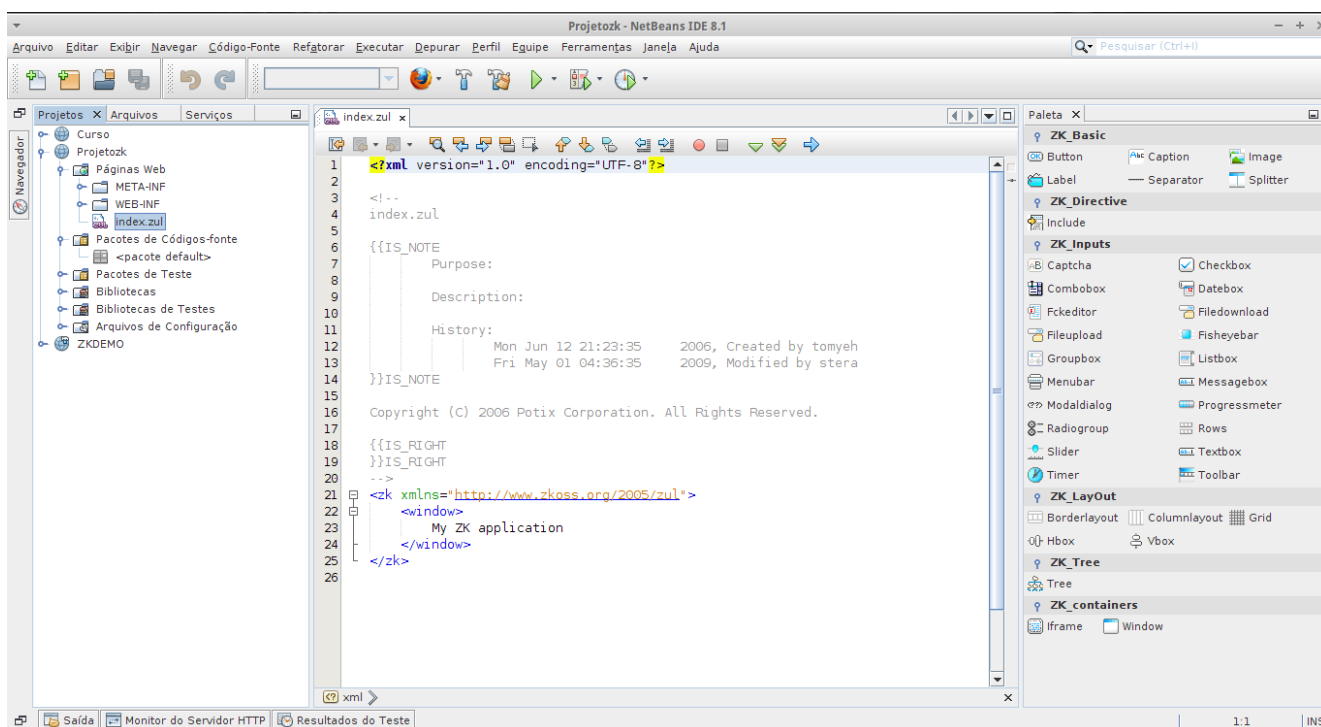
Criação do Projeto

Criando um Projeto com Framework ZK

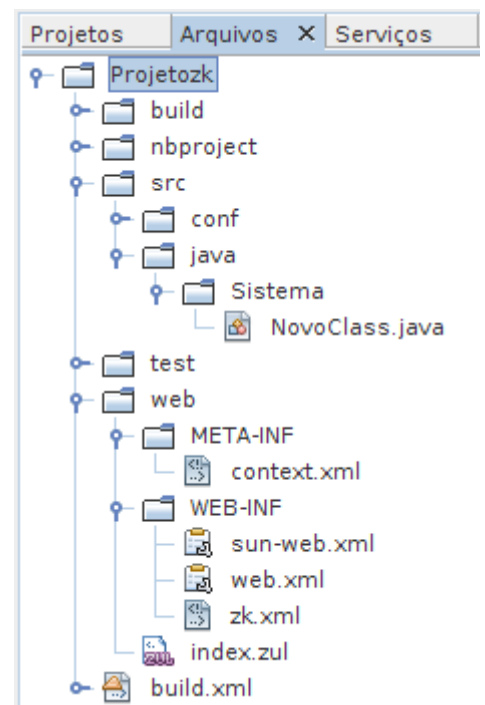
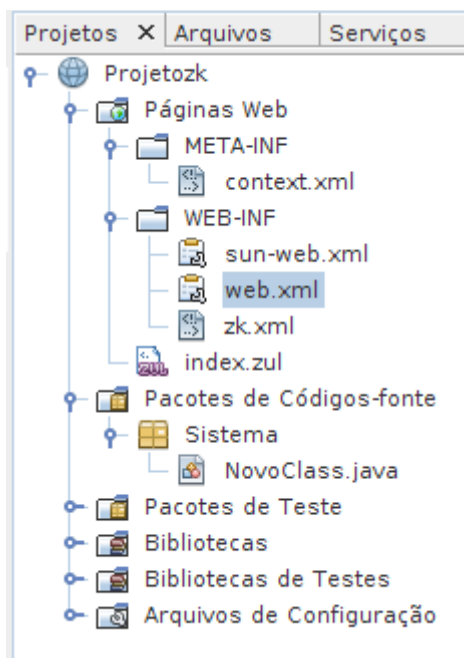
Acesse Arquivo / Novo Projeto



Após Finalizado o projeto ficará com a seguinte estrutura:



Estrutura do Projeto



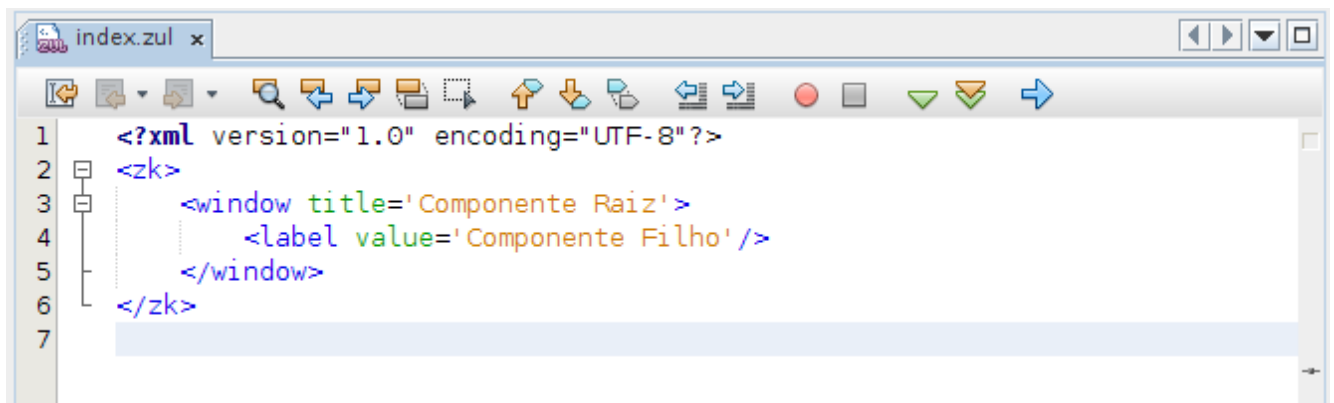
Componentes

Estruturação dos Componentes

Como uma estrutura de árvore, um componente tem no máximo um pai, enquanto ele pode ter vários filhos.

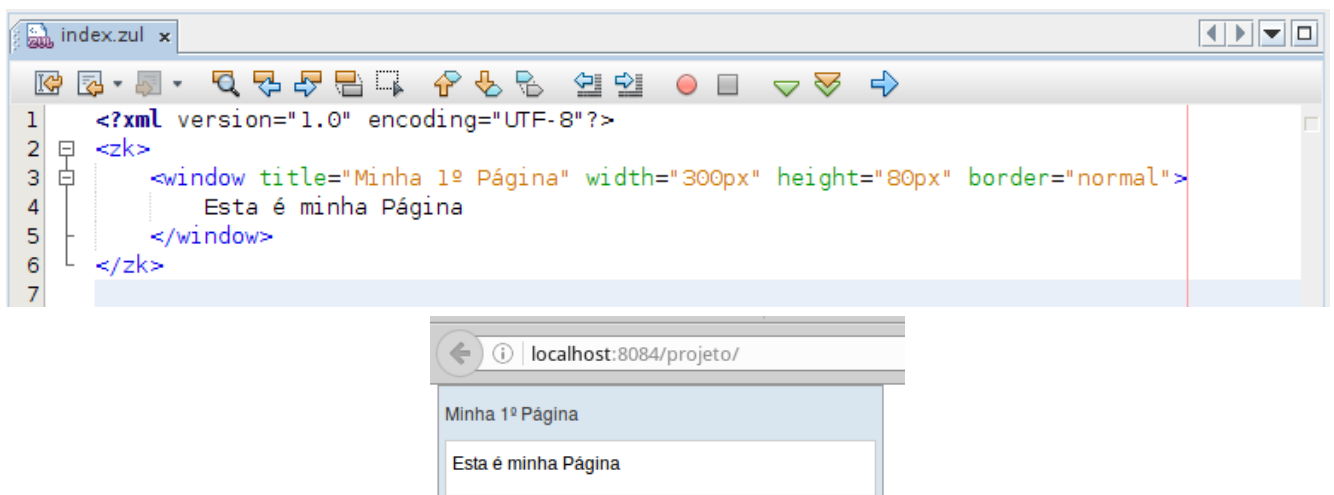
Alguns componentes aceitam apenas tipos específicos de componentes como filhos, como por exemplo, o componente **Grid** aceita apenas os componentes **Columns**, **Rows** e **Foot**.

Outros não permitem nenhum tipo de componente, como por exemplo, o componente **Textbox**. Um componente sem nenhum pai é chamado de componente raiz. Múltiplos componentes raízes são permitidos em cada página.



Window

Uma window é um suporte de espaço em uma janela do navegador, onde os componentes ZK podem ser anexados e desanexados. A window é uma coleção de componentes e apenas os componentes ligados a uma página estão disponíveis no cliente. Uma window é criada automaticamente quando o usuário requisita um recurso como uma página ZUL.



Observe que o endereço da aplicação é <http://localhost:8084/projeto/>. Não foi informado o nome do arquivo index.zul, este será ativado automaticamente devido à existência da referência do index.zul no arquivo web.xml.

Uma janela é apresentada no web browser. Note que é muito simples desenhar uma janela, basta apenas utilizar as tags **<window>** e **</window>**. Essas tags representam o componente window do framework ZK.

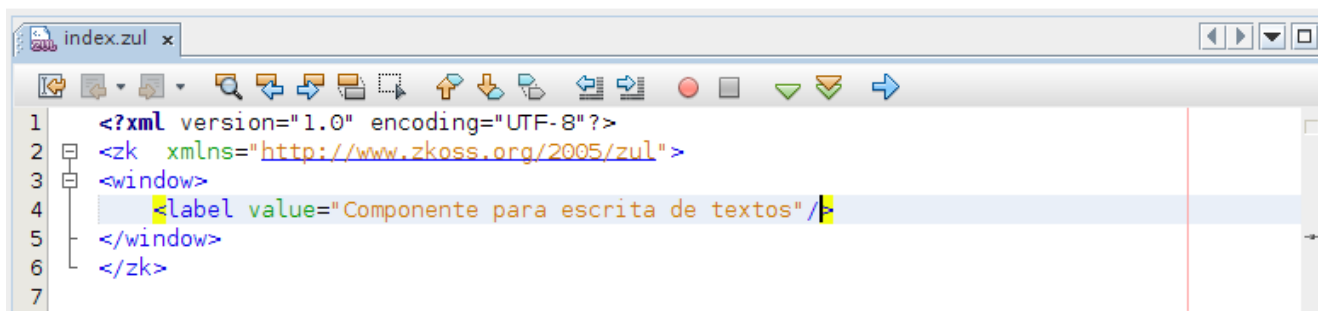
O Corpo da janela possui apenas a frase **“Esta é minha Página”**. Tudo que estiver entre as tags **<window>** e **</window>** faz parte do componente window.

A janela tem o título **“Minha 1ª Página”**, que é o conteúdo de **title**. Assim, **title**, **border**, **width** e **height** são alguns atributos do componente entre outros:

- **id**: O atributo ID é uma referência do ZK ao controle, é o nome do componente;
- **title**: apresenta o título na barra de título da janela;
- **width**: indica a largura em pixels ou %. Exemplo 50px ou 70%;
- **height**: indica a altura em pixels ou %. Exemplo 200px ou 100%;
- **border**: apresenta o estilo da borda. Pode ser **normal** ou **none**(padrão);
- **sizable**: valor booleano(true ou false) que indica se a janela pode ser redimensionada;
- **closeable**: valor booleano(true ou false) que habilita ou não o **X** no canto superior direito da janela para fechá-la;
- **mode**: armazena o conteúdo de um estilo para a janela. Pode ser **embedded** (incorporado/esquerda superior/não perde foco), **highlighted** (destacado/centraliza/ao perder o foco fica visível), **popup** (sobreposto/esquerda superior/ ao perder o foco não fica visível) e **overlapped** (sobreposto/esquerda superior/ao perder o foco fica visível).
- **position**: posiciona a janela no web browser nas seguintes posições: left, center, right / top, bottom.
- **ContentStyle**: pode-se personalizar a aparência da janela.(contentStyle="background:yellow" contentStyle="overflow:auto"

Label

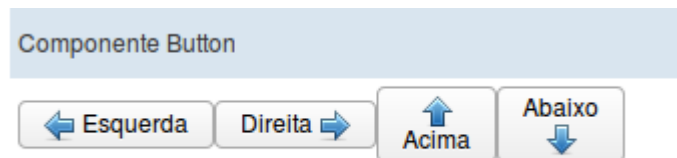
Representa um texto específico. É indicado com a tag **<label>**, possui o atributo **value** para receber o texto.



Button

Cria um botão que pode disparar **eventos** para acesso a **métodos** ou **referência** a um site. É indicado com a tag `<button></button>` permite ainda inserir figuras a sua estrutura com o atributo `image= "imagem.jpg"`.

```
<zk xmlns="http://www.zkoss.org/2005/zul">
  <window title="Componente Button">
    <button label="Esquerda" image="/img/esquerda.png"
      hoverImage="/img/esquerdaverde.png"/>
    <button label="Direita" image="/img/direita.png"
      hoverImage="/img/direitaverde.png" dir="reverse"/>
    <button label="Acima" image="/img/acima.png"
      hoverImage="/img/acimaverde.png" orient="vertical"/>
    <button label="Abaixo" image="/img/abaixo.png"
      hoverImage="/img/abaixoverde.png" orient="vertical" dir="reverse"/>
  </window>
</zk>
```



Possui os seguintes eventos:

- **onClick**: ao clicar no componente;
- **onFocus**: quando um componente recebe o foco;
- **onBlur**: quando um componente perde o foco.

onClick="win.gravarCadastro()" o evento onClick dispara o método gravarCadastro().

```
<zk xmlns="http://www.zkoss.org/2005/zul">
  <window id="win" title="Componente Button">
    <button label="Evento Alerta"
      onClick="alert(self.label)"/>
    <button id="bt1" label="Evento Mudando Label"
      onClick='bt1.setLabel("Você clicou no botão")'/>
    <button id="bt2" label="Evento Mudando Título da Janela"
      onClick='win.setTitle("Novo Título")'/>
    <button label="Evento Abrindo Página"
      href="index.zul"/>
    <button label="Evento Abrindo Página nova Aba"
      href="index.zul" target="_blank"/>
    <button label="Evento Redirecionando Página"
      onClick="Executions.sendRedirect("&quot;index.zul&quot;")"/>
    <button label="Redirecionando p/ nova Aba"
      onClick="Executions.sendRedirect
        ("&quot;http://www.google.com&quot;;&quot;_blank&quot;")"/>
  </window>
</zk>
```

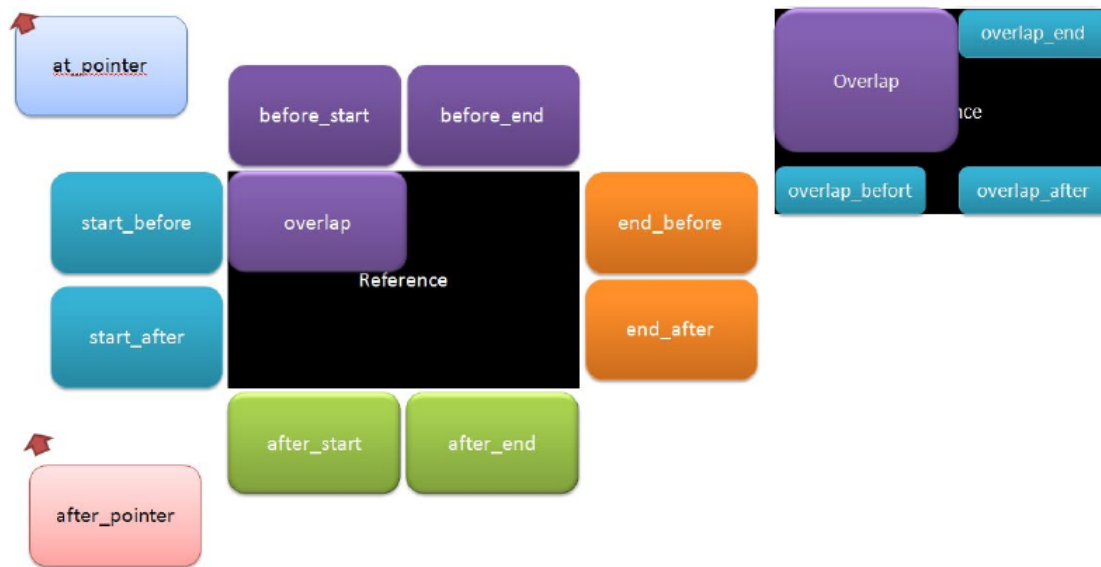
Popup

Um recipiente pode ser exibido como um popup. A janela popup não têm qualquer quadro especial. Popups podem ser exibidos quando um elemento é clicado.

São atributos do componente:

- **id**: O atributo ID é uma referência do ZK ao controle, é o nome do componente;
- **title**: apresenta o título na barra de título da janela;
- **width**: indica a largura em pixels ou %. Exemplo 50px ou 70%;
- **height**: indica a altura em pixels ou %. Exemplo 200px ou 100%;

Posicionamento do popup:



```
<zk xmlns="http://www.zkoss.org/2005/zul">
  <window title="exemplo de popup" border="normal" width="100%" height="100%">
    <popup id="pop" width="150px" height="100px">
      Meu popup
    </popup>
    <button label="Abrir Popup" width="500px"
      onClick='pop.open(self, "overlap_end")' />
  </window>
</zk>
```



Componentes Básicos para Entrada de Dados

O framework ZK possui diversos componentes para entrada de dados como **textbox**, **intbox**, **decimalbox**, **doublebox**, **combobox** e etc:

- **textbox**: recebe informações em forma de caracteres alfanuméricos, tag `<textbox/>`;
- **intbox**: recebe apenas valores inteiros, é indicado com a tag `<intbox/>`;
- **longbox**: recebe valores inteiros para uma faixa de valores maiores, tag `<longbox/>`;
- **decimalbox**: recebe valores decimais, tag `<decimalbox/>`;
- **doublebox**: recebe valores decimais para uma faixa de valores maiores, tag `<doublebox/>`;
- **datebox**: permite digitação ou escolha de data e hora, tag `<datebox/>`;
- **timebox**: recebe valores de hora, tag `<timebox/>`;

```

<zkg>
  <window title="Entrada de Dados" border="normal">
    Nome      <textbox      width="200px" />
    Senha     <textbox      width="50px"  />
    Idade     <intbox       width="50px"  />
    Salário   <decimalbox   width="80px"  />
    Aniversário <datebox/>
    E-mail    <textbox      width="150px" />
  </window>
</zkg>

```

Entrada de Dados

Nome Senha Idade Salário Aniversário E-mail

Um componente de entrada pode estar associado com uma restrição, para validar o valor digitado. Há uma validação padrão que pode lidar com muitas condições. Se não for suficiente, você pode implementar sua própria validação.

Tipos de **Constraint**:

- **no empty**: vazio não é permitido;
- **no negative**: números negativos não são permitidos;
- **no zero**: zero não é permitido;
- **no past**: data no passado não é permitido;
- **no today**: data de hoje não é permitido;
- **no future**: data no futuro não é permitido;

```

<zk>
<window title="Entrada de Dados com Validação" border="normal">
  Nome <textbox width="200px" constraint="no empty:Não pode ser vazio"/>
  <separator spacing="15px"/>
  Senha <textbox width="50px" type="password" maxlength="5"/>
  <separator spacing="15px"/>
  Idade <intbox width="50px" constraint="no negative,no zero:Não pode ser menor 0"/>
  <separator spacing="15px"/>
  Salário <decimalbox width="80px" format="R$###.00"/>
  <separator spacing="15px"/>
  Aniversário <datebox format="dd/MM/yyyy"/>
  <separator spacing="15px"/>
  E-mail<textbox width="150px"
    constraint="/.+@.+\. [a-z]+/: E-mail inválido" />

  <separator bar="true" spacing="30px"/>
  Email: <textbox type="email"/>
  <separator spacing="15px"/>
  WebSite: <textbox type="url"/>
</window>
</zk>

```

The screenshot displays a web form titled "Entrada de Dados com Validação". It contains several input fields with associated validation messages:

- Nome:** A text box with a red border and a red error message: "Não pode ser vazio".
- Senha:** A password text box with masked characters (dots).
- Idade:** An integer text box containing the value "0" with a red error message: "Não pode ser menor 0".
- Salário:** A decimal text box containing the value "R\$3578.52".
- Aniversário:** A date box containing the value "30/07/1971" with a calendar icon.
- E-mail:** A text box with a red border and a red error message: "E-mail inválido".
- Email:** A text box containing the value "wmfilho".
- WebSite:** A text box containing the value "google".

Grid

O componente é utilizado para alinhar os elementos em uma janela. Possui os elementos **<grid>**, **<columns>** e **<column>**, **<rows>** e **<row>**.

A tag **<columns>** possui a tag **<column>**, que corresponde a cada uma das colunas da grade. A tag **<rows>** possui a tag **<row>** que corresponde a linha da grade.

```

<zk>
  <window title="Grid" border="normal">
    <grid>
      <columns>
        <column width="100px"/>
        <column />
      </columns>
      <rows>
        <row>
          Nome
          <textbox width="200px" constraint="no empty:Não pode ser vazio"/>
        </row>
        <row>
          <label value="Senha"/>
          <textbox width="50px" type="password" maxlength="5"/>
        </row>
      </rows>
    </grid>
  </window>
</zk>

```

Tabbox

O componente é o container que controla os elementos de abas ou painéis. Possui os elementos **<tabbox>**, **<tabs>** e **<tab>**, **<tabpanel>** e **<tabpanel>**.

A tag **<tabs>** possui a tag **<tab>**, que corresponde a cada aba. A tag **<tabpanel>** possui a tag **<tabpanel>** que corresponde a área da aba.

```

<zk>
  <window title="Tabbox" border="normal" width="500px">
    <tabbox >
      <tabs>
        <tab label="Dados básicos"/>
        <tab label="Previlégio de acesso"/>
      </tabs>
      <tabpanel>
        Nome
        <textbox width="200px" />
      </tabpanel>
      <tabpanel>
        Acesso
        <textbox width="100px" />
      </tabpanel>
    </tabpanels>
  </tabbox>
</window>
</zk>

```

The first screenshot shows a window titled 'Tabbox' with two tabs: 'Dados básicos' and 'Previlégio de acesso'. The 'Dados básicos' tab is active, showing a text input field labeled 'Nome'.

The second screenshot shows the same window, but the 'Previlégio de acesso' tab is now active, showing a text input field labeled 'Acesso'.

<tabbox orient="vertical">

The screenshot shows a window titled 'Tabbox' with a vertical orientation. The 'Dados básicos' tab is selected, displaying a text input field labeled 'Nome'. The 'Previlégio' tab is visible below it.

<tabbox mold="accordion">

The screenshot shows a window titled 'Tabbox' with the 'accordion' mold. The 'Dados básicos' tab is selected, displaying a text input field labeled 'Nome'. The 'Previlégio' tab is visible below it.

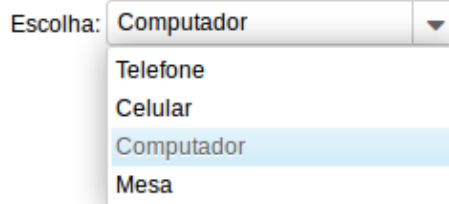
Combobox

É uma caixa de texto especial que permite ao usuário selecionar um item de uma lista. Possui os elementos `<combobox>` e `<comboitem>`.

A tag `<combobox>` possui a tag `<comboitem>`, que corresponde a cada item da lista.

```
<zk>
<window title="Combobox">
    Escolha:
    <combobox>
        <comboitem label="Telefone"/>
        <comboitem label="Celular"/>
        <comboitem label="Computador"/>
        <comboitem label="Mesa"/>
    </combobox>
</window>
</zk>
```

Combobox



```
<zk>
<window title="Combobox com figura">
    Escolha:
    <combobox value="Computador">
        <comboitem label="Telefone" image="/img/telefone.jpeg" />
        <comboitem label="Celular" image="/img/celular.jpeg" />
        <comboitem label="Computador" image="/img/computador.jpeg" />
        <comboitem label="Mesa" image="/img/mesa.jpeg" />
    </combobox>
</window>
</zk>
```

Combobox com figura



Radiogroup

É um componente que agrupa todas opções em um mesmo conjunto de informações, sendo que só uma pode ser selecionada.

Possui os elementos `<radiogroup>` e `<radio>`.

A tag `<radiogroup>` possui a tag `<radio>`, que corresponde a cada opção de escolha.

```
<zk>
  <window title="Radiogroup">
    Escolha:
    <radiogroup>
      <radio label="Telefone" checked="true"/>
      <radio label="Celular"/>
      <radio label="Computador"/>
      <radio label="Mesa"/>
    </radiogroup>
  </window>
</zk>
```

Radiogroup

Escolha: ☒ Telefone ☐ Celular ☐ Computador ☐ Mesa

Groupbox

É um componente que agrupa todas opções em um mesmo conjunto de informações, permitindo a seleção de várias opções.

Possui os elementos `<groupbox>` e `<checkbox>`.

A tag `<groupbox>` agrupa todas informações da tag `<checkbox>`, que corresponde a cada opção de escolha.

```
<zk>
  <window title="Groupbox / checkbox">
    Escolha:
    <groupbox mold="3d" width="300px">
      <checkbox label="Telefone" checked="true" />
      <checkbox label="Celular"/>
      <checkbox label="Computador"/>
      <checkbox label="Mesa"/>
    </groupbox>
  </window>
</zk>
```

Groupbox / checkbox

Escolha:

☐ Telefone ☒ Celular ☒ Computador ☐ Mesa

Listbox

Tem um funcionamento parecido com o combobox, difere na quantidade de informações mostradas, pode exibir um elemento ou uma lista com várias colunas. Permite utilizar uma grande variedade de atributos.

Possui os elementos `<listbox>`, `<listhead>` e `<listheader>`, `<listitem>` e `<listcell>`.

A tag `<listhead>` possui a tag `<listheader>`, que corresponde ao cabeçalho das colunas da lista. A tag `<listitem>` possui a tag `<listcell>` que corresponde a linha e coluna da lista, é o valor de célula.

```
<zk>
<window title="Listbox" border="normal" width="250px">
  <listbox>
    <listhead sizable="true">
      <listheader label="Carro" sort="auto" />
      <listheader label="km/h"/>
    </listhead>
    <listitem>
      <listcell label="Bugatti Veyron" />
      <listcell label="431,07" />
    </listitem>
    <listitem>
      <listcell label="SSC Tuatara" />
      <listcell label="442" />
    </listitem>
    <listitem>
      <listcell label="Saleen S7 Twin-Turbo" />
      <listcell label="399,12" />
    </listitem>
  </listbox>
</window>
</zk>
```

Listbox	
Carro	km/h
Bugatti Veyron	431,07
SSC Tuatara	442
Saleen S7 Twin-Turbo	399,12

mold= "select "

```
<zk>
<window title="Listbox Select" border="normal" width="400px">
  Escolha seu carro:
  <listbox mold="select">
    <listitem label="Bugatti Veyron" />
    <listitem label="SSC Tuatara" />
    <listitem label="Saleen S7 Twin-Turbo" />
  </listbox>
</window>
</zk>
```

Listbox Select	
Escolha seu carro:	Bugatti Veyron ▼

checkmark="true" multiple="true" mold="paging" pageSize="2"

```
<zk>
<window title="Listbox" border="normal" width="400px">
  <listbox checkmark="true" multiple="true" mold="paging" pageSize="2">
    <listhead sizable="true">
      <listheader label="Carro" width="300px" sort="auto" />
      <listheader label="km/h" width="70px" sort="auto" />
    </listhead>
    <listitem>
      <listcell label="Bugatti Veyron" />
      <listcell label="431,07" />
    </listitem>
  </listbox>
</window>
```

Listbox /checkmark/multiple/paging	
Carro	km/h
<input checked="" type="checkbox"/> Bugatti Veyron	431,07
<input checked="" type="checkbox"/> SSC Tuatara	442
<div> << < 1 / 2 > >> </div> <div>[1 - 2 / 3]</div>	

textbox spinner listbox

```

<listitem>
  <listcell label="Bugatti Veyron" />
  <listcell>
    <textbox width="100%" />
  </listcell>
</listitem>
<listitem>
  <listcell label="SSC Tuatara" />
  <listcell>
    <spinner inplace="true" value="240" width="100%"
      constraint="min 200 max 450: Mínimo 200 Máximo 450" />
  </listcell>
</listitem>
<listitem>
  <listcell label="Saleen S7 Twin-Turbo" />
  <listcell>
    <listbox mold="select">
      <listitem label="250" />
      <listitem label="350" />
      <listitem label="450" />
    </listbox>
  </listcell>
</listitem>

```

Carro	km/h
Bugatti Veyron	270
SSC Tuatara	254
Saleen S7 Twin-Turbo	350

Menu / Toolbar

Um componente Menu é muito parecido com o Button, ele é colocado em uma barra de menu. Quando o usuário clica o elemento do menu, um sub menu será exibido. Este componente é usado para disparar eventos para acesso a métodos ou referência a sites.

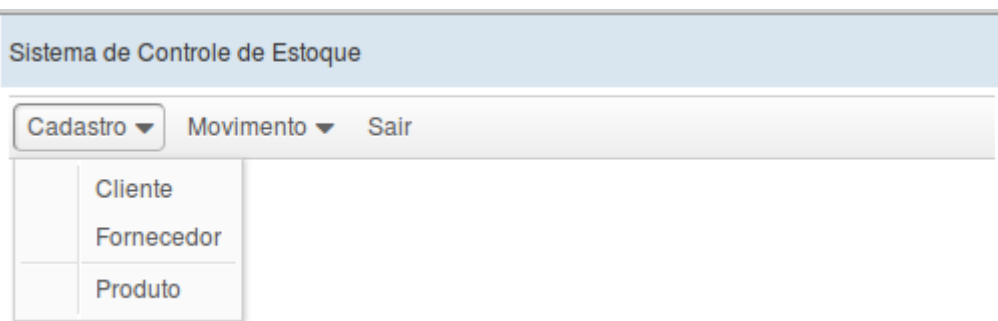
Possui os elementos **<menubar>**, **<menu>**, **<menupopup>** e **<menuitem>**.

A tag **<menubar>** cria uma barra de menu. A tag **<menu>** gera as opções principais da barra de menu. A tag **<menupopup>** cria uma barra de sub opção e finalmente a tag **<menuitem>** gera o último elemento para disparar os eventos.

```

<zk>
  <window id="win" title="Sistema de Controle de Estoque" width="500px" height="400px">
    <menubar>
      <menu label="Cadastro">
        <menupopup>
          <menuitem label="Cliente" onClick="alert(self.label)"/>
          <menuitem label="Fornecedor" onClick="alert(self.label)"/>
          <menuseparator/>
          <menuitem label="Produto" onClick="alert(self.label)"/>
        </menupopup>
      </menu>
      <menu label="Movimento">
        <menupopup>
          <menuitem label="Entrada" onClick="alert(self.label)"/>
          <menuitem label="Saída" onClick="alert(self.label)"/>
        </menupopup>
      </menu>
      <menuitem label="Sair" onClick="win.detach()" />
    </menubar>
  </window>
</zk>

```



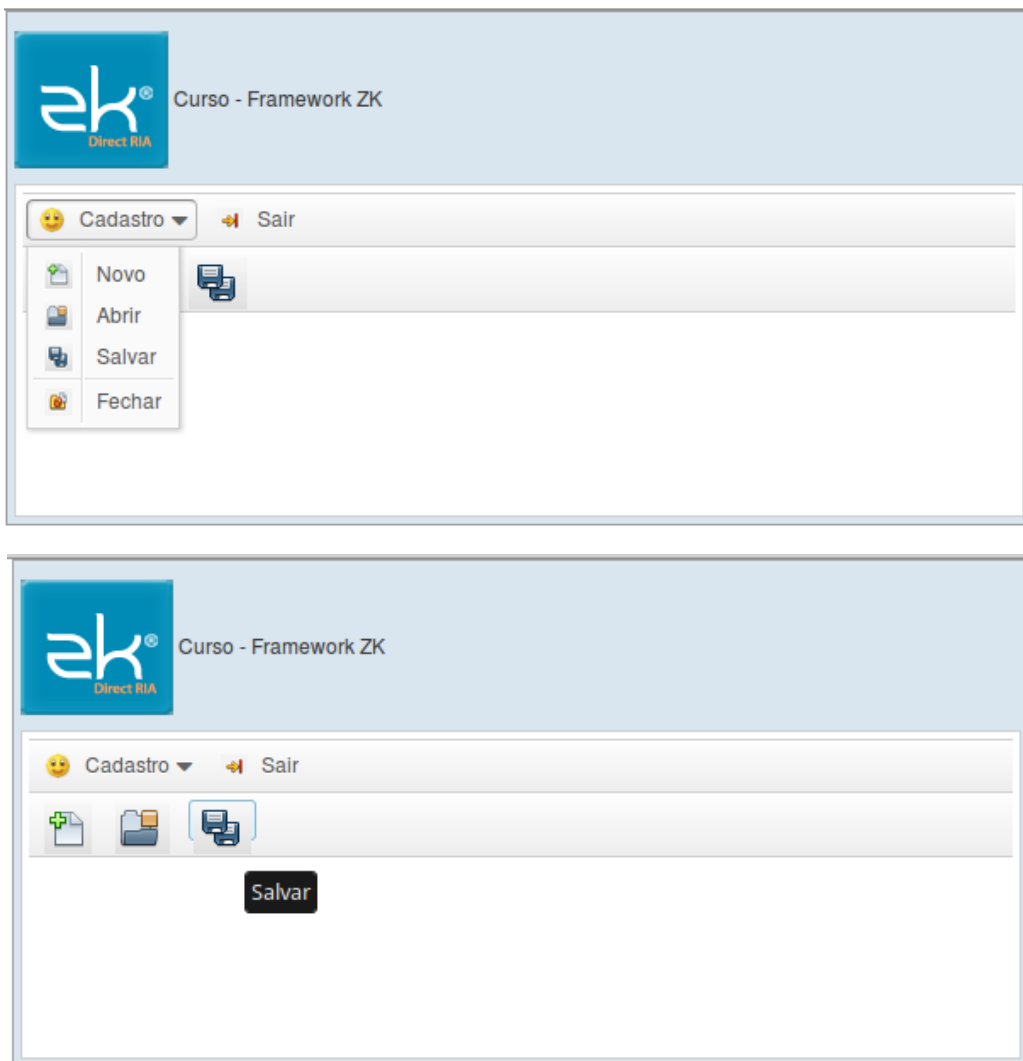
Um componente Toolbar é uma barra de ferramentas, é usada para colocar uma série de botões.

Possui os elementos **<toolbar>** e **<toolbarbutton>**.

```

<window id="win" title="Curso" width="100%" height="400px" border="normal">
  <caption label="Framework ZK" image="/img/ZK-Logo-old.gif" />
  <menubar id="menubar">
    <menu label="Cadastro" src="/img/m1.gif" >
      <menupopup>
        <menuitem src="/img/novo.png" label="Novo" onClick="alert(self.label)"/>
        <menuitem src="/img/abrir.png" label="Abrir" onClick="alert(self.label)"/>
        <menuitem src="/img/salvar.png" label="Salvar" onClick="alert(self.label)"/>
        <menuseparator/>
        <menuitem src="/img/fechar.png" label="Fechar" onClick="alert(self.label)"/>
      </menupopup>
    </menu>
    <menuitem label="Sair" src="/img/sair.png" onClick="win.detach()" />
  </menubar>
  <toolbar>
    <toolbarbutton image="/img/novo.png" tooltip="Novo"
      onClick="alert(self.tooltiptext)"/>
    <toolbarbutton image="/img/abrir.png" tooltip="Abrir"
      onClick="alert(self.tooltiptext)"/>
    <toolbarbutton image="/img/salvar.png" tooltip="Salvar"
      onClick="alert(self.tooltiptext)"/>
  </toolbar>
</window>

```



Hlayout / Vlayout

Hlayout e Vlayout são componentes simples de para o layout, que organizam outros componentes contidos em uma orientação horizontal ou vertical.

O hlayout e vlayout não suportam separador, alinhamento e posicionamento.

```
<zk>
  <window title="hlayout / vlayout" border="normal" width="500px" height="200px">
    <vlayout spacing="20px">
      <button label="Button 1"/>
      <button label="Button 2"/>
    </vlayout>
    <separator spacing="20px"/>
    <separator bar="true"/>
    <hlayout>
      <button label="Button 3"/>
      <space spacing="20px" />
      <button label="Button 4"/>
    </hlayout>
  </window>
</zk>
```



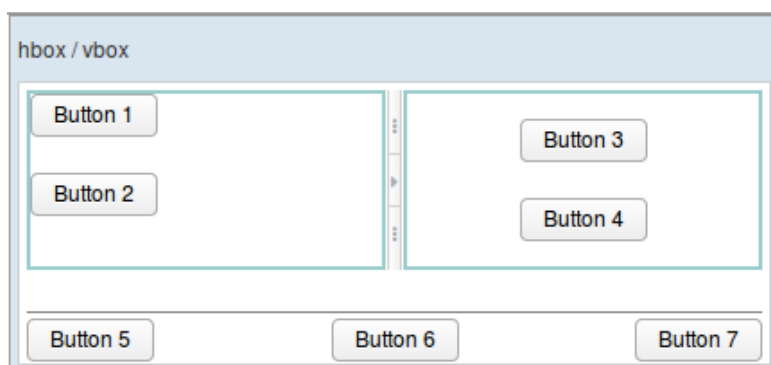
Hbox / VBox

O componente hbox é usado para criar uma caixa orientada horizontalmente. Cada componente colocado no hbox vai ser colocada horizontalmente em uma fileira.

O vbox tem o mesma função só que verticalmente.

O hbox e vbox são projetados para fornecer layout mais sofisticado, com divisor, alinhamento e posicionamento.

```
<zk>
<window title="hbox / vbox" border="normal" width="430px" height="200px">
  <hbox>
    <vbox width="200px" height="100px" spacing="20px"
      style="border:2px solid; border-color:#99CCCC">
      <button label="Button 1"/>
      <button label="Button 2"/>
    </vbox>
    <splitter collapse="after" />
    <vbox align="center" pack="center" width="200px" height="100px" spacing="20px"
      style="border:2px solid; border-color:#99CCCC">
      <button label="Button 3"/>
      <button label="Button 4"/>
    </vbox>
  </hbox>
  <separator spacing="20px"/>
  <separator bar="true"/>
  <hbox spacing="100px">
    <button label="Button 5"/>
    <button label="Button 6"/>
    <button label="Button 7"/>
  </hbox>
</window>
</zk>
```



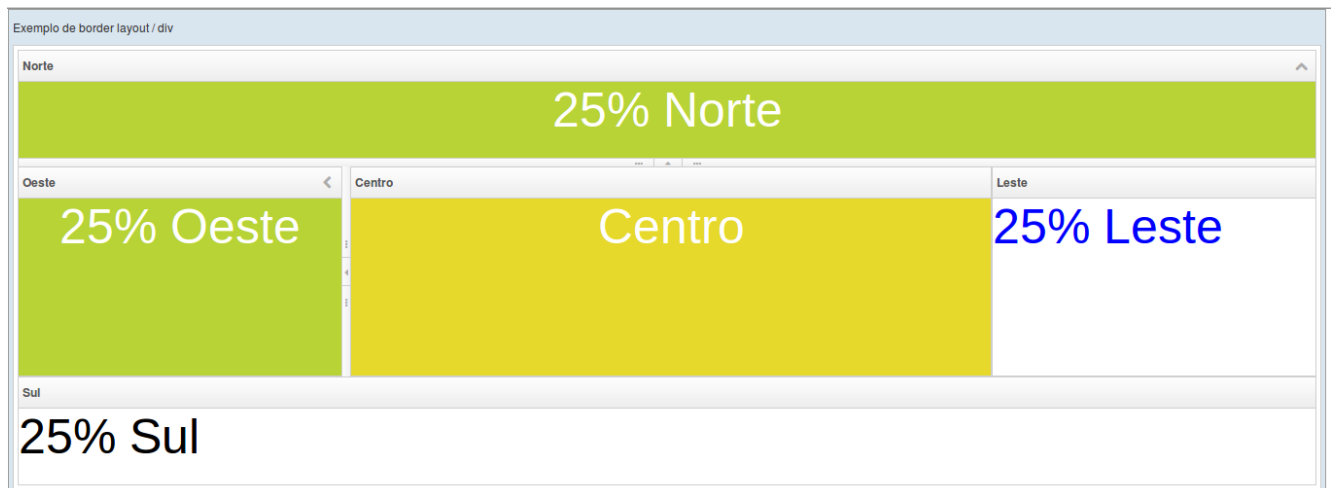
Borderlayout

O componente `borderlayout` cria uma área para posicionamento de tela alinhado. O componente pai é o `<borderlayout>`, e seus componentes filhos incluem `<north>`, `<south>`, `<center>`, `<west>`, e `<east>`.

Div

É um componente que gera um recipiente ou bloco para armazenar outros componentes, criando um layout mais sofisticado.

```
<borderlayout>
  <north title="Norte" size="25%" flex="true" maxsize="250" splittable="true" collapsible="true">
    <div align="center" style="background:#B8D335">
      <label value="25% Norte" style="color:white;font-size:50px" />
    </div>
  </north>
  <west title="Oeste" size="25%" flex="true" maxsize="250" splittable="true" collapsible="true">
    <div align="center" style="background:#B8D335">
      <label value="25% Oeste" style="color:white;font-size:50px" />
    </div>
  </west>
  <center title="Centro" flex="true">
    <div align="center" style="background:#E6D92C">
      <label value="Centro" style="color:white;font-size:50px" />
    </div>
  </center>
  <east title="Leste" size="25%">
    <label value="25% Leste" style="color:blue;font-size:50px" />
  </east>
  <south title="Sul" size="25%">
    <label value="25% Sul" style="color:black;font-size:50px" />
  </south>
</borderlayout>
```



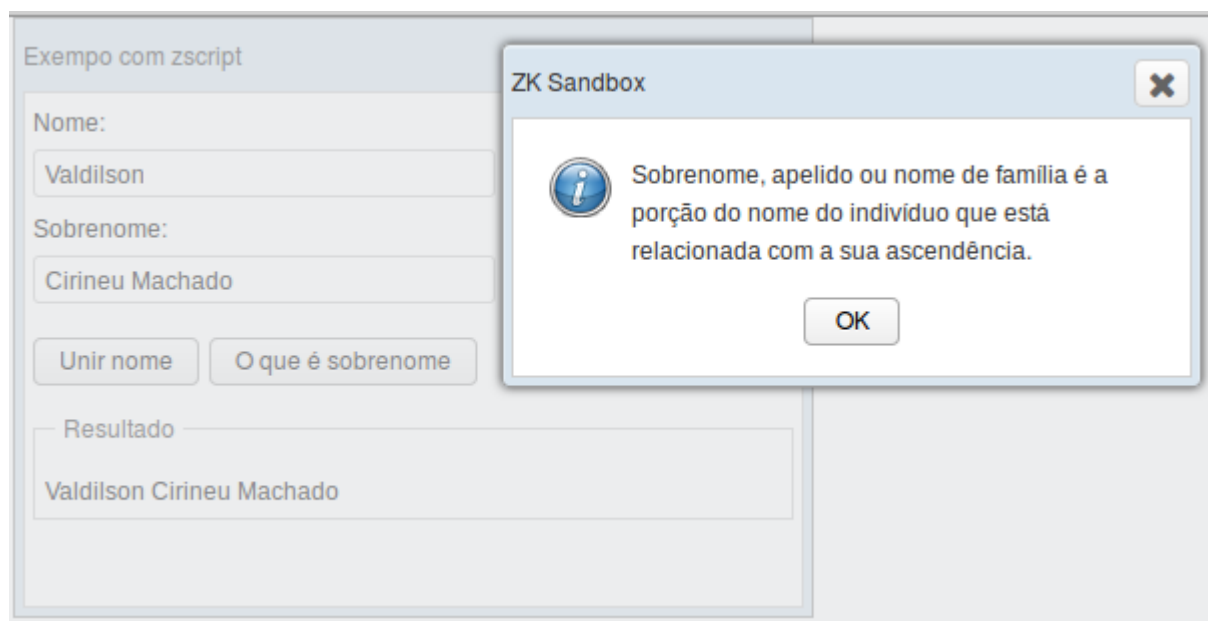
Framework ZK e Código Java

Zscript e Linguagem de Expressão

Para tornar mais fácil a criação de uma página web dinâmica, o documento ZUML permite embutir o código de `<zscript></zscript>`. O elemento `zscript` é utilizado para incorporar o código que será executado quando a página é carregada, enquanto o manipulador de eventos será executado quando o evento é recebido.

A linguagem padrão utilizada no `zscript` é o Java, mas pode-se utilizar: JavaScript, Ruby, Groovy e Python.

```
<vlayout>
  Nome:
  <textbox id="nm_nome" cols="30"/>
  Sobrenome:
  <textbox id="nm_sobrenome" cols="30"/>
  <separator/>
  <hlayout>
    <button label="Unir nome" onClick="unirNome()"/>
    <button label="O que é sobrenome" onClick="sobrenome()"/>
  </hlayout>
  <groupbox>
    <caption label="Resultado"/>
    <label value="" id="resultado"/>
  </groupbox>
</vlayout>
<zscript>
void unirNome(){
    resultado.setValue(nm_nome.getValue()+" "+ nm_sobrenome.getValue());
}
void sobrenome(){
    alert("Sobrenome, apelido ou nome de família é a porção do nome do indivíduo
        | que está relacionada com a sua ascendência.");
}
</zscript>
```

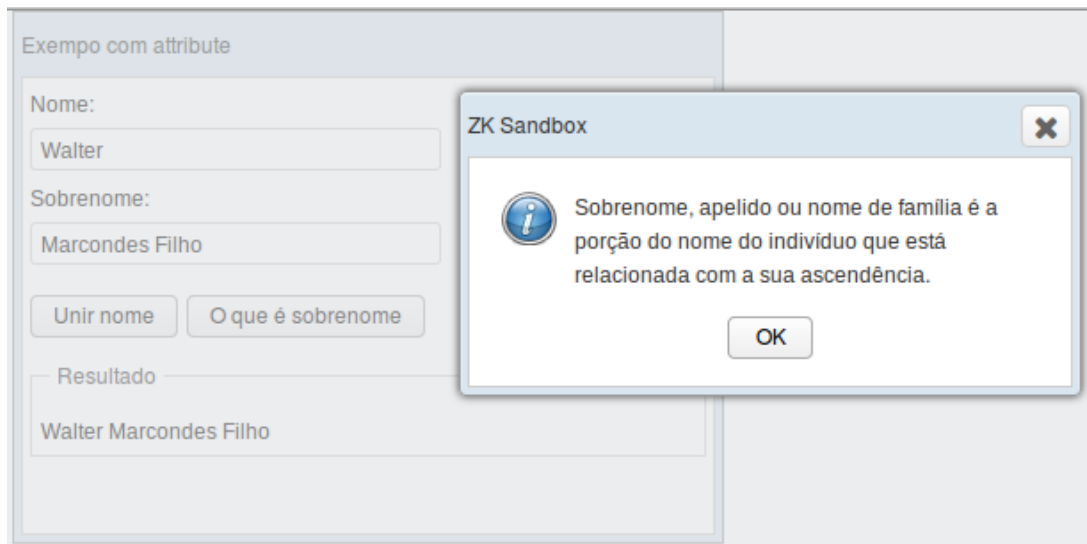


Attribute name

Ele define um atributo XML do elemento delimitador. O conteúdo do elemento é o valor do atributo, enquanto o **attribute name** especifica o nome do atributo.

```
<vlayout>
  Nome:
  <textbox id="nm_nome" cols="30"/>
  Sobrenome:
  <textbox id="nm_sobrenome" cols="30"/>
  <separator/>
  <hlayout>
    <button label="Unir nome">
      <attribute name="onClick">
        resultado.setValue(nm_nome.getValue()+" "+ nm_sobrenome.getValue());
      </attribute>
    </button>

    <button label="O que é sobrenome">
      <attribute name="onClick">
        alert("Sobrenome, apelido ou nome de família é a porção do nome do
        indivíduo que está relacionada com a sua ascendência.");
      </attribute>
    </button>
  </hlayout>
</vlayout>
<groupbox>
  <caption label="Resultado"/>
  <label value="" id="resultado"/>
</groupbox>
```



Classe Java com a Interface Gráfica

Um programador precisa desenvolver a parte de regras de negócios, o que garantirá maior segurança de acesso aos dados da aplicação. A solução do webdesigner com o layout e do programador com as regras de negócios permite uma maior interação com o usuário. O trabalho dos dois tipos de profissionais são distintos, assim é interessante separar o **código Java** com o **arquivo zul** para evitar dificuldades na manutenção.

Para evitar a mistura dos códigos com **<zscript>** ou **<attribute name>**, pode-se inserir uma referencia a uma **classe java controladora** tornando o sistema mais flexível a mudanças.

Arquivo do framework ZK: use.zul

```
<zk>
<window title="Separando Classe Java do ZK" id="win" use="sistema.ManterSobrenome"
border="normal" width="400px" height="300px">
  <vlayout>
    Nome:
    <textbox id="nm_nome" cols="30"/>
    Sobrenome:
    <textbox id="nm_sobrenome" cols="30"/>
    <separator/>
    <hlayout>
      <button label="Unir nome" onClick="win.unirNome()"/>
      <button label="0 que é sobrenome" onClick="win.sobrenome()"/>
    </hlayout>
    <groupbox>
      <caption label="Resultado"/>
      <label value="" id="resultado"/>
    </groupbox>
  </vlayout>
</window>
</zk>
```

Arquivo de controle Classe Java: ManterSobrenome.java

```
package sistema;

import static org.zkoss.zk.ui.util.Clients.alert;
import org.zkoss.zul.Label;
import org.zkoss.zul.Textbox;
import org.zkoss.zul.Window;
/**
 *
 * @author walter
 */
//CLASSE ManterSobrenome
public class ManterSobrenome extends Window {
  //DECLARAÇÃO DOS CAMPOS DO FORMULÁRIO USE.ZUL
  private Window win;
  private Textbox nm_nome;
  private Textbox nm_sobrenome;
  private Label resultado;

  //CONSTRUTOR DA CLASSE
  public ManterSobrenome() {
  }
```

```

//METODO EXECUTADO AO CRIAR O FORMULARIO ZK
public void onCreate() {
    //ASSOCIAÇÃO DAS VARIÁVEIS DA CLASSE COM OS COMPONENTES DO FORMULARIO ZK
    this.setWin((Window) getFellow("win"));
    this.setNm_nome((Textbox) getFellow("nm_nome"));
    this.setNm_sobrenome((Textbox) getFellow("nm_sobrenome"));
    this.setResultado((Label) getFellow("resultado"));
}

//METODO EXECUTADO AO CLICAR NO BOTAO
public void unirNome() {
    this.resultado.setValue(this.nm_nome.getValue() + " " +
                           this.nm_sobrenome.getValue());
}

//METODO EXECUTADO AO CLICAR NO BOTAO
public void sobrenome() {
    alert("Sobrenome, apelido ou nome de família é a porção do nome do"
          + " indivíduo que está relacionada com a sua ascendência.");
}

//***** GETTERS E SETTERS DA APLICAÇÃO *****/

public Window getWin() {
    return win;
}

public void setWin(Window win) {
    this.win = win;
}

public Textbox getNm_nome() {
    return nm_nome;
}

public void setNm_nome(Textbox nm_nome) {
    this.nm_nome = nm_nome;
}

public Textbox getNm_sobrenome() {
    return nm_sobrenome;
}

public void setNm_sobrenome(Textbox nm_sobrenome) {
    this.nm_sobrenome = nm_sobrenome;
}

public Label getResultado() {
    return resultado;
}

public void setResultado(Label resultado) {
    this.resultado = resultado;
}
}

```

Separando Classe Java do ZK

Nome:

Walter

Sobrenome:

Marcondes Filho


Unir nome

O que é sobrenome

Resultado

Walter Marcondes Filho

ZK Sandbox

 Sobrenome, apelido ou nome de família é a porção do nome do indivíduo que está relacionada com a sua ascendência.

OK

Referências

CHEN, Henry; CHENG Robbie. **ZK - Ajax without Javascript Framework**. New York: Potix Corporation, (2007)

STAUBLE, Markus; SCHUMACHER, Hans-Jurgern. **ZK - Developer's Guide** - Developing responsive user interfaces for web applications using Ajax, XUL, and the open source ZK rich web client development framework. Birmingham - Mumbai:Packt Publishing , (2008)

CAMARGO,Vicente Paulo. **Ria com Framework ZK** - Desenvolva Aplicações Ricas para a Internet (RIA), na plataforma Java, com o framework ZK, Eclipse e MySQL. Rio de Janeiro:Editora Ciência Moderna Ltda,2010

<https://www.zkoss.org/>

https://www.zkoss.org/wiki/ZK_Component_Reference

<https://www.zkoss.org/zksandbox/>

<https://www.zkoss.org/zkdemo/>