A arquitetura fornece um componente de localização de entidades, composto por um botão de localizar e a janela de localização:



O componente é todo customizado e composto pelos seguintes atributos:

* **title**: Título da janela de localização, caso não definida é configurado o valor associado a chave “LOCATOR\_TITLE”, na arquitetura tem o valor de “Localizar...”;
* **pageSize**: Quantidades de itens a serem exibidos por página, caso não seja definido é configurado para exibir 10 itens por página;
* **osPaging**: Define se deve exibir a página de sistemas operacionais (como na figura) ou não. Valor padrão é false;
* **actionLabel**: Define o nome do botão de Ação da janela de localização, caso não seja definido é configurado o valor associado a chave “LOCATOR\_ACTION\_LABEL”, na arquitetura tem o valor de “Selecionar”;
* **cancelLabel**: Define o nome do botão de Cancelar ação da janela de localização, caso não seja definido é configurado o valor associado a chave “LOCATOR\_CANCEL\_LABEL”, na arquitetura tem o valor de “Cancelar”;
* **emptyMessage**: Define a mensagem para exibição da lista quando vazia, mesmo se a mesma for definida no listbox personalizado, o emptyMessage do locatorButton tem prioridade sobre o listbox, caso não seja definido é configurado o valor associado a chave “INFO\_EMPTY\_LIST”, na arquitetura tem o valor de “Não existe dados para listagem”;
* **customListbox**: Atributo opcional para definir uma listbox personalizada para a janela de Localização, caso seja definido o componente irá buscar o arquivo .zul que deverá conter apenas a tag listbox com o mesmo configurado;
* **entityClass**: OBRIGATÓRIO. Define qual entidade o localizador irá trabalhar, deve apontar um método no compositor que retorna a classe da entidade;
* **key**: OBRIGATÓRIO. Define a chave para buscar as restrições que deve aplicar a busca/listagem, mesmo que não tenha nenhuma deve ser configurada uma chave de restrição;
* **composer**: OBRIGATÓRIO. Associa ao componente o compositor de controle da visão;
* **value**: OBRIGATÓRIO. “Binda” o valor ao componente, ou seja, quando selecionar uma entidade na janela de localização, o valor do mesmo irá ser lançado no atributo relacionado no compositor;
* **refreshComponentId**: Define o id do componente que deverá ser recarregado após a seleção de uma entidade;
* **image**: Imagem para ser exibida no botão de localização. Caso não seja definido nenhuma e mesmo assim não seja associado um label ao componente será exibida uma imagem padrão de localizar;
* **hoverImage**: Imagem para ser exibida no botão de localização quando o mouse estiver sobre o botão;
* **label**: Nome para ser exibido no botão de localização;
* **imageOrientation**: Define a orientação da imagem em relação ao texto, sendo:
  + **LEFT**: Imagem a esquerda do texto (padrão);
  + **RIGHT**: Imagem a direita do texto;
  + **TOP**: Imagem em cima do texto;
  + **BOTTOM**: Imagem a baixo do texto;

Para utilização do componente em um caso de uso é preciso que a entidade implemente ILocalizableEntity, o controlador implemente IlocalizableController (caso use o GenericController ou alguma de suas sub-classes não é necessário a implementação, pois o mesmo já o implementa) e a visão implemente ILocalizableView (caso use o ZK não é necessário a implementação, pois a super-classe SuperViewZK já o implementa).

A entidade também deve ter pelo menos um de seus atributos anotado com @SearchableField para ser considerado como campo de busca, e a entidade deve ser anotada com @VisibleName também.

**CSS do componente**

O CSS do componente pode ser sobrescrito e criado novos. Para alterar a borda da janela foi reconfigurado as classes do window, sendo utilizado: .locator-window .z-window-modal-tl; .locator-window .z.window-modal-tr; .locator-window .z-window-modal-hr; .locator-window .z-window-modal-hm; .locator-window .z-window-modal-header. E criado as seguintes classes:

* comboloc: Associada a todos os comboboxs criados na janela;
* buttonloc: Associada a todos os botões criados na janela;
* inputloc: Associada a todos os textboxs criados na janela;
* locator-window: Associada a janela de localização;
* searchButton: Associado aos botões do formulário de busca – buscar, listar, lipmar;
* listButton: Associado apenas ao botão de listagem;
* separatorloc: Aplicado a todos os separators da janela;
* listboxSeparator: Aplicado ao separador que fica entre a div do formulário de busca e o listbox;
* buttonsSeparator: Aplicado ao separador que fica entre o listbox e a div dos botões de ação;
* searchForm: Aplicado a div do formulário de busca que contém os inputs e botões relacionado a eles;
* searchInput: Aplicado a div que contém os botões de buscar, listar e limpar;
* listboxDiv: Aplicado a div que ficará o listbox;
* buttonsDiv: Aplicado a div que contém os botões de ação (Selecionar e Cancelar);
* searchButtonsDiv: Associado ao div que contém os botões de busca, listagem e limpar;
* searchInputs: Associado a todos os inputs do formulário de busca, tanto textbox quanto combobox;
* sf-field: Associado ao combobox de escolha do campo de busca;
* sf-operator: Associado ao combobox de escolha da condicional (operador);
* sf-value: Associado ao textbox de digitação do valor;
* sf-hgvalue: Associado ao textbox de digitação do segundo valor;

Caso seja utilizado o listbox padrão do componente ainda é aplicado os seguintes CSS sobre os headers do mesmo:

* headerloc: Aplicado a todos os headers do listbox;
* header + contador: Aplicado a cada coluna do header, sendo que a primeira coluna começa com 0 (header0) e já tem o css definido com o width em 80px;
* col- + nome do header: Aplicado a cada coluna do header, funciona similar ao header + contador, porém adiciona o nome do header em vez do índice da coluna.