

Mira-Editor

Ferramenta de prototipação auxiliar ao Mira Framework

Guia do Usuário

INF 2102 - Projeto Final de Programação

João Victor Magela

1421957

Sumário

1 - Instalação	2
2 - Configuração	2
3 - Uso	2
3.1 - Projeto.....	3
3.2 - Interface abstrata.....	4
3.2.1 - Obter Informações	5
3.3 - Interface Concreta.....	6
3.4 - Formulário de Propriedades	6
3.5 - Código Gerado.....	7
3.6 - Preview.....	8

1 - Instalação

O Mira-Editor é executado diretamente no navegador, através do link <http://tecweblab.github.io/mira-editor/>, portanto não é necessária a instalação de nenhum outro software.

2 - Configuração

Caso deseja-se obter os dados do *Synth*, é necessário iniciar o servidor através do comando `jruby -1.9 -S scriptserver` no prompt de comando do sistema operacional utilizado.

3 - Uso

O Mira-Editor é composto por seis camadas que conversam entre si. São elas:

- Projeto
- Interface Abstrata
- Interface Concreta
- Propriedades
- Código Gerado
- *Preview*

Observe nas figuras 1 e 2 como estão dispostas as camadas.

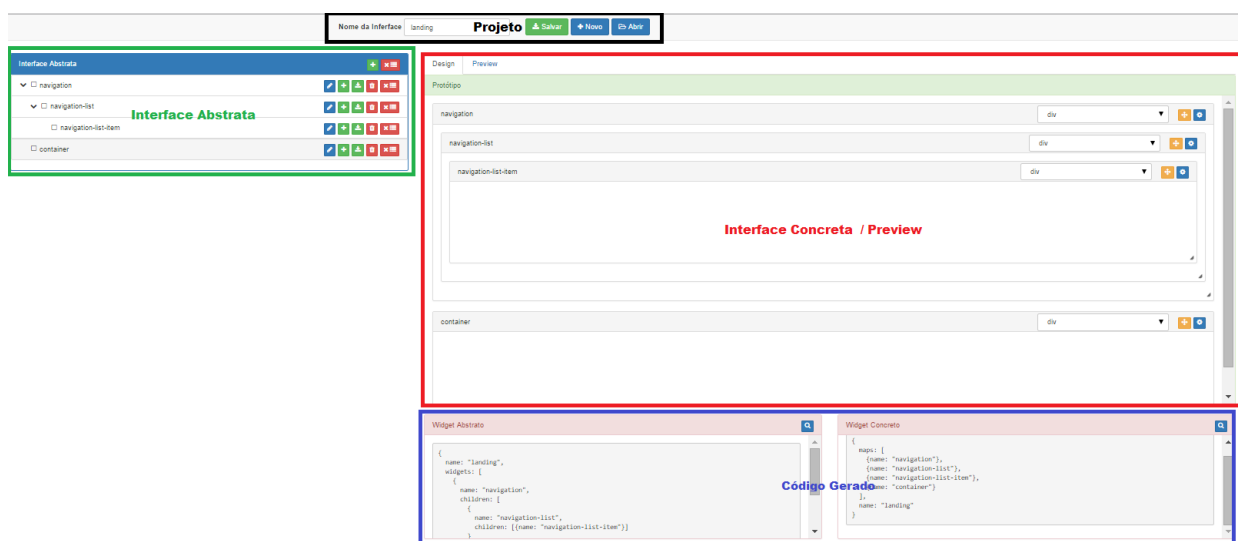


Figura 1 - Divisão em camadas do Mira-Editor

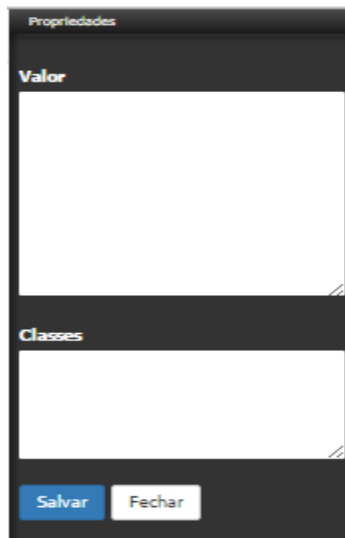


Figura 2 - Camada Propriedade que é acessível através da camada Interface Concreta.

3.1 - Projeto

Local onde define-se o nome do projeto e as operações “Salvar”, “Novo” e “Abrir”.



Figura 3 - Detalhes da camada Projeto.

1. **Nome:** Define o nome do projeto, que será também o nome da interface concreta e abstrata.
2. **Salvar:** Faz o download de um arquivo JSON contendo as informações do projeto.
3. **Novo:** Redefine o projeto para os valores padrões, ou seja, inicia um novo projeto.
4. **Abrir:** Permite ao usuário selecionar um arquivo JSON contendo as informações do projeto no qual deseja abrir.

3.2 - Interface abstrata

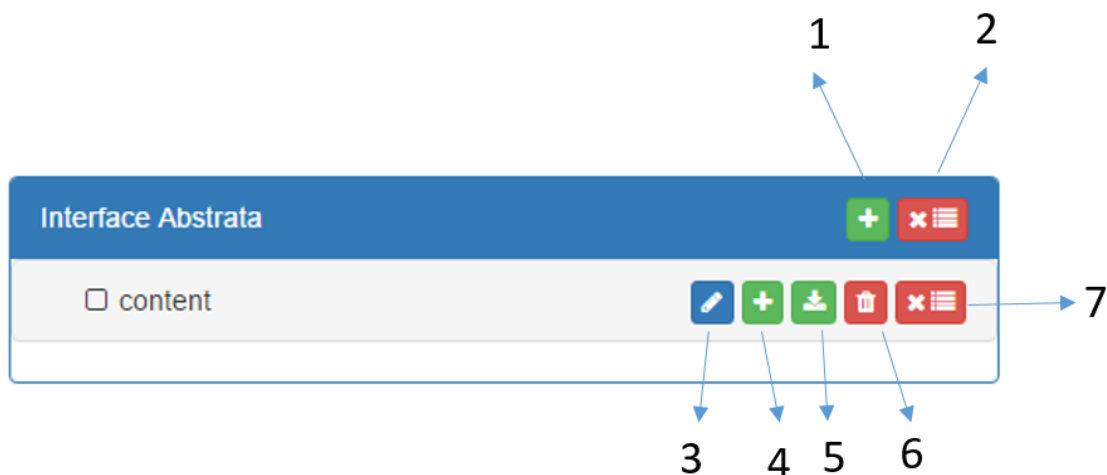


Figura 4 - Detalhes da camada Interface Abstrata.


Exibe a *Treeview* representando os elementos da interface abstrata. Podemos fazer as seguintes operações:


1. **Adicionar elemento à raiz:** Adiciona um elemento na raiz da árvore.
2. **Remover itens selecionado da raiz:** Remove todos os itens selecionados da raiz da árvore e consequentemente seus filhos.
3. **Editar Elemento:** Faz com o que o nome do elemento seja editável. Para confirmar a operação de edição, deve-se pressionar a tecla "Enter" ou "Tab".
4. **Adicionar filho:** Cria um novo filho para um determinado elemento. Gera-se um campo editável para que o usuário possa inserir seu nome. Caso deseja-se cancelar a operação, basta pressionar a tecla "Esc".
5. **Obter informações:** Exibe um modal que permite inserir as informações "Bind", "Datasource" e "Parse", como podemos ver na Figura 5. Para mais informações sobre obter informações, consulte o capítulo 3.2.1.
6. **Remover item:** Remove um determinado elemento e seus respectivos filhos.
7. **Remover filhos selecionados:** Remove os filhos selecionados e seus respectivos filhos.

3.2.1 - Obter Informações

Baixar Informações

Bind

Link  Processar Datasource

Parse  Processar Parse

Campos
Nome
id
nome
ponto
estado

Lista de atributos

Figura 5 - Detalhes da área Obter Informações.

Essa é uma importante funcionalidade do Mira-Editor, com ela podemos associar informações a um elemento abstrato. Um formulário é exibido e podemos inserir as seguintes informações:

- **Bind:** Expressão utilizada para referenciar o valor de um elemento abstrato a partir de um conjunto de dados. Aconselha-se utilizar a expressão `$data.[atributo com a informação desejada]`. Exemplo: `$data.name`.
- **Datasource:** Expressão ou link que permite o Mira obter as informações desejadas. Ex: `http://mira.tecweb.inf.puc-rio.br/api/futebol`
- **Parse:** Expressão para filtrar um conjunto de dados. A expressão segue a forma explicada no item *Bind*.

Para essa camada temos as seguintes operações:

1. **Processar Datasource:** Através da expressão o link informado, o Mira-Editor obter as informações e exibe os atributos disponíveis na tabela.
2. **Processar Parse:** Através da expressão informada o Mira-Editor obter as informações filtradas e exibe os atributos disponíveis na tabela.

Sempre que um *Datasource* ou *Parse* é processado corretamente, os atributos disponíveis são exibidos na lista de campos, como pode ser visto na figura 5.

3.3 - Interface Concreta

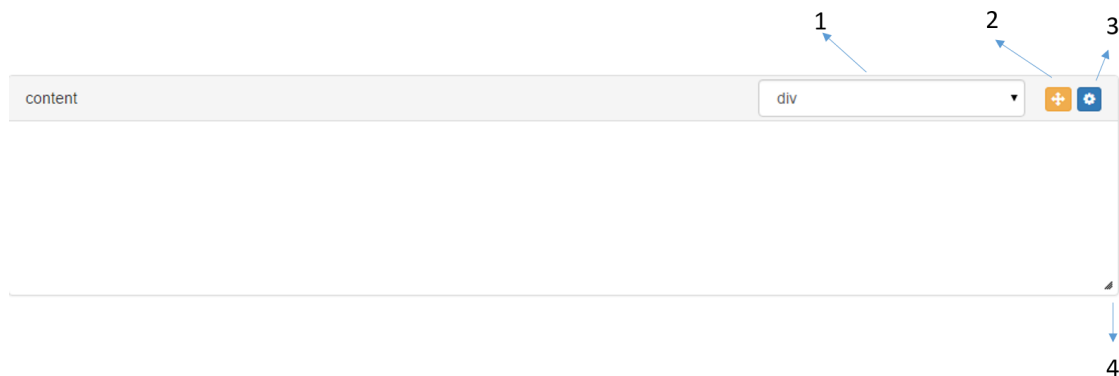


Figura 6 - Detalhes da camada Interface Concreta

Nessa camada podemos fazer o mapeamento dos elementos abstratos em elementos concretos. Inicialmente, cada elemento abstrato está associado ao elemento concreto DIV. Nessa camada temos as seguintes operações:

1. Alterar tipo do elemento: Permite ao usuário alterar o tipo do elemento concreto.
2. Arrastar elemento: Pressionando esse botão, pode-se arrastar o elemento para a posição desejada. Vale informar que o elemento concreto só pode ser arrastado dentro do conteúdo do seu pai. Caso não exista um pai, o protótipo se torna o pai.
3. Propriedades do elemento: Faz com que o formulário de propriedades surja, podendo alterar as propriedades do elemento concreto.
4. Redimensionar elemento: Permite alterar as dimensões do elemento.

3.4 - Formulário de Propriedades

Através desse formulário podemos alterar propriedades de um elemento concreto. Essas propriedades serão mapeadas no código gerado para interface concreta. Os campos “Valor” e “Classes” estão presentes no formulário, independentemente do tipo de elemento concreto. O primeiro define o valor que será associado ao elemento. Caso o valor seja oriundo do *Datasource*, então o valor no formulário será “\$bind”, caso contrário, o valor informado será exibido no *Preview*. O segundo campo são classes CSS que serão adicionadas no *Preview*.

3.5 - Código Gerado



Figura 7 - Código JSON para a interface abstrata e concreta.

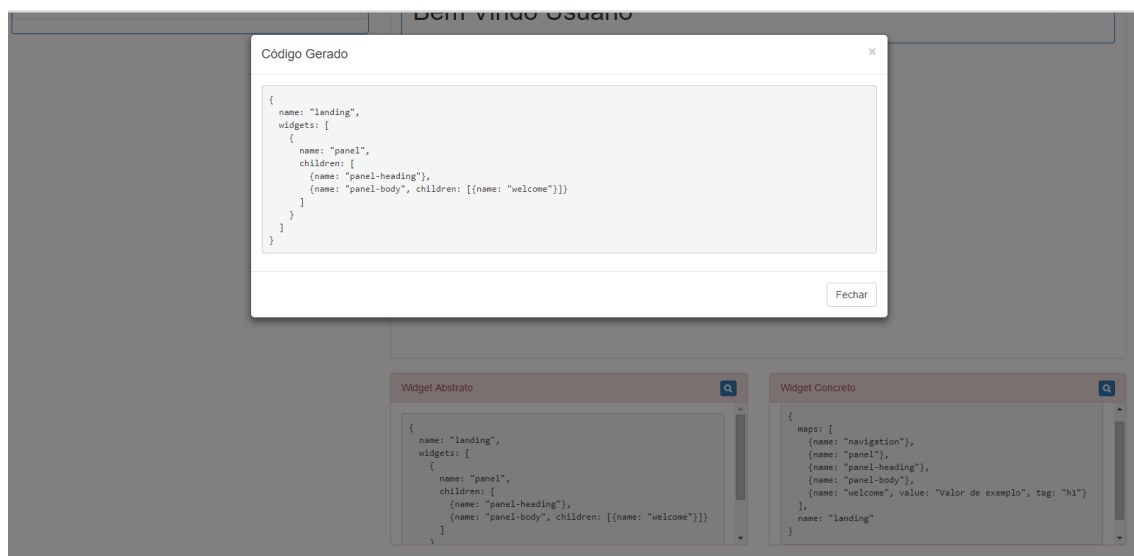


Figura 8 - Código gerado sendo exibido em um modal para melhor visualização.

Nessa camada temos a geração do código JSON para a interface abstrata e concreta. Esse código deve ser inserido na aplicação a ser executada pelo Mira. Nessa camada temos como única opção a visualização do código em um modal, facilitando assim a visualização, como visto na Figura 8.

3.6 - Preview

O *Preview* exibe como a aplicação será renderizada pelo Mira. O *Preview* considera os dados coletados no *Datasource*, com isso é possível observar se o trabalho feito até o momento está de acordo com o que o usuário realmente deseja.



Figura 9 - Preview gerado pelo Mira-Editor