Criando Componente para bloco de programação - Cliente

Passos para criar um componente para o bloco de programação do tipo cliente

Passo 1: Criar diretório

Criar um novo diretório que irá conter o arquivo JavaScript referente ao componente. Faz parte da boa prática de desenvolvimento criar arquivos referentes ao bloco de programação do tipo cliente no diretório **Códigos Fonte Cliente>js.**

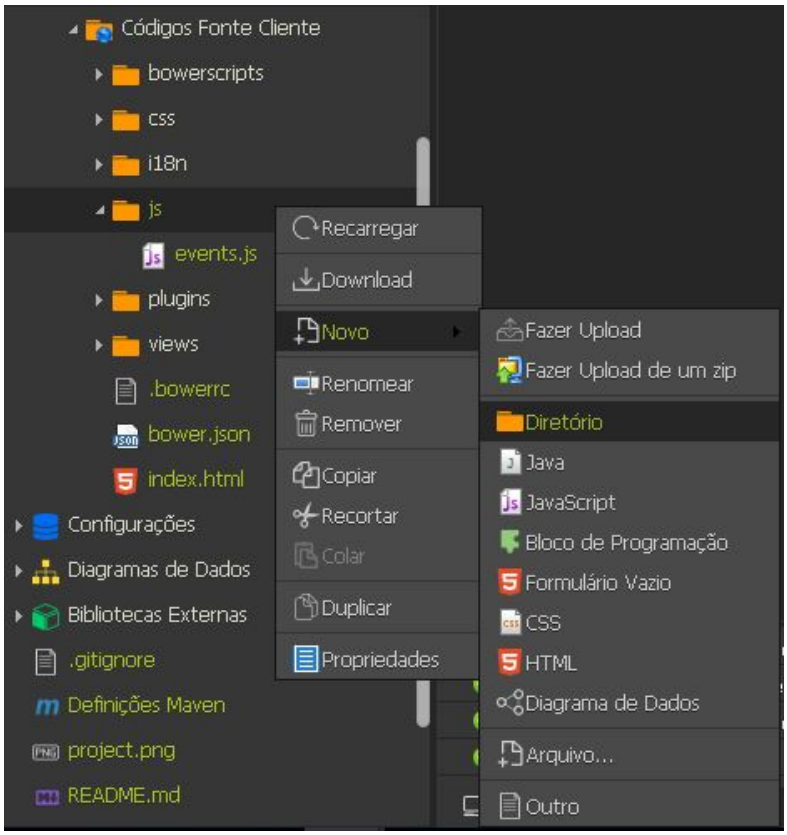

Figura 1 - Criação do diretório

Passo 2: Criar classe java

O diretório criado deve receber um novo arquivo do tipo JavaScript, **Códigos Fonte Cliente>js>Diretório>JavaScript** (**Figura 2**). Após selecionada a opção a IDE irá exibir a tela de escolha do modelo do arquivo. Para o caso da criação de componentes, o modelo correto é o **Nov a Função Cliente para Bloco de Programação** como pode ser visto na (**Figura 3**). O botão **Avançar** confirma a seleção do modelo e exibe a tela de definição as informações.

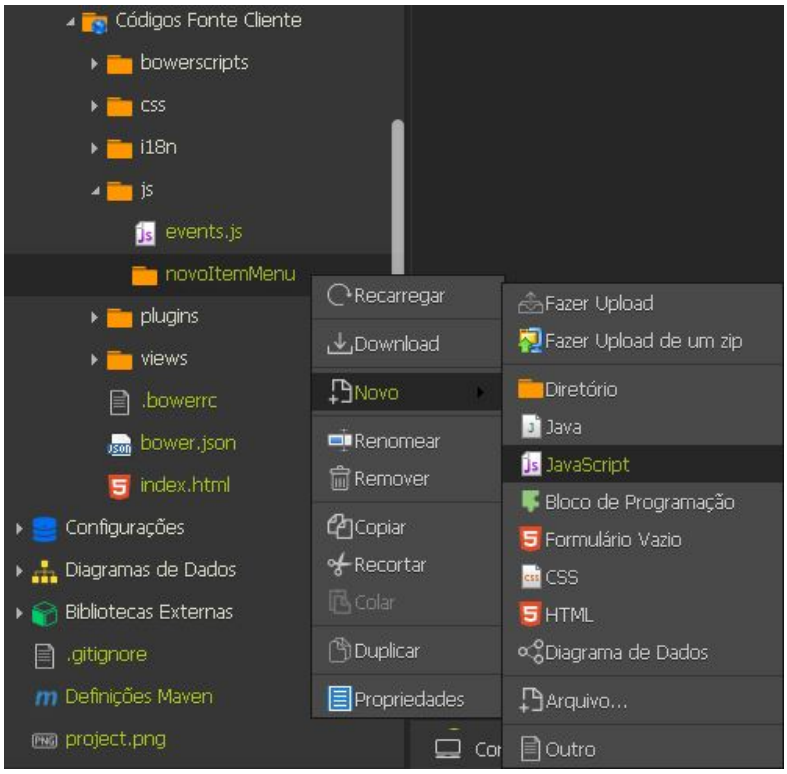

Figura 2 - Criação do arquivo JavaScript

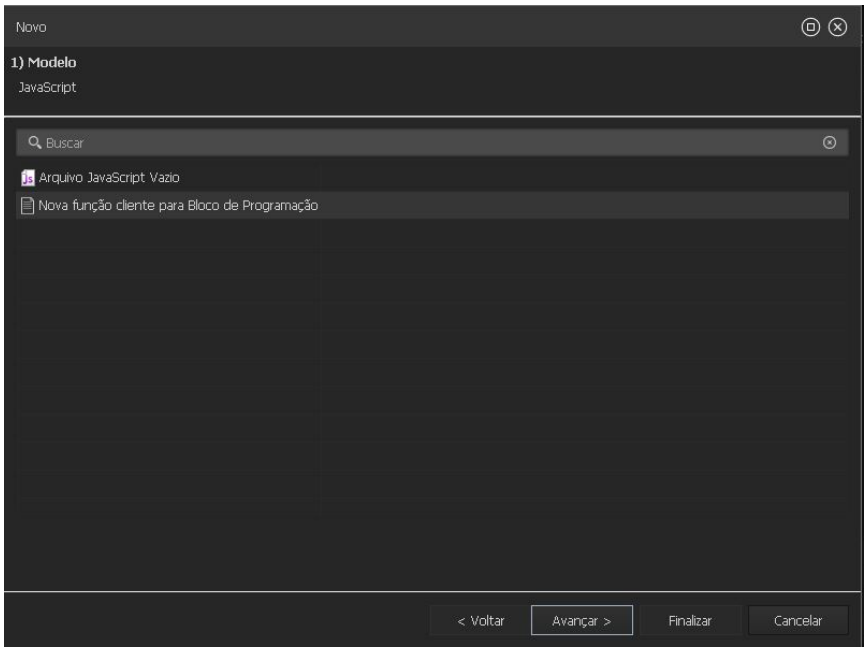

Figura 3 - Escolha do modelo do arquivo

Passo 3: Definir informações do componente

O componente a ser criado pode ter as seguintes informações definidas: Nome da classe, Nome da função, Nome reduzido da função, Descrição da função e Categoria. As alterações podem ser aplicadas no contexto da aplicação após a confirmação das informações através do botão **Finalizar**, como pode ser visto na (**Figura 4**).

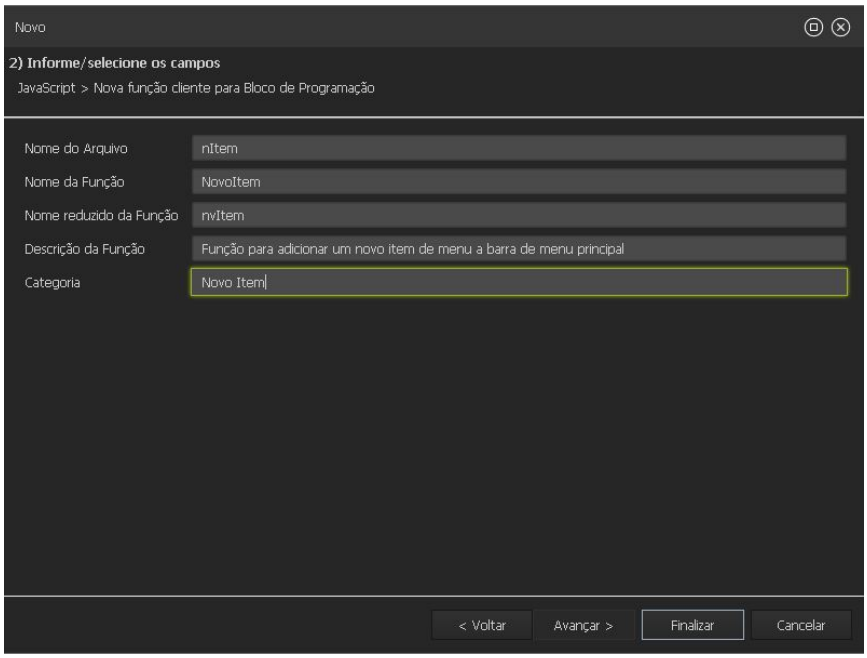

Figura 4 - Definindo informações sobre o componente

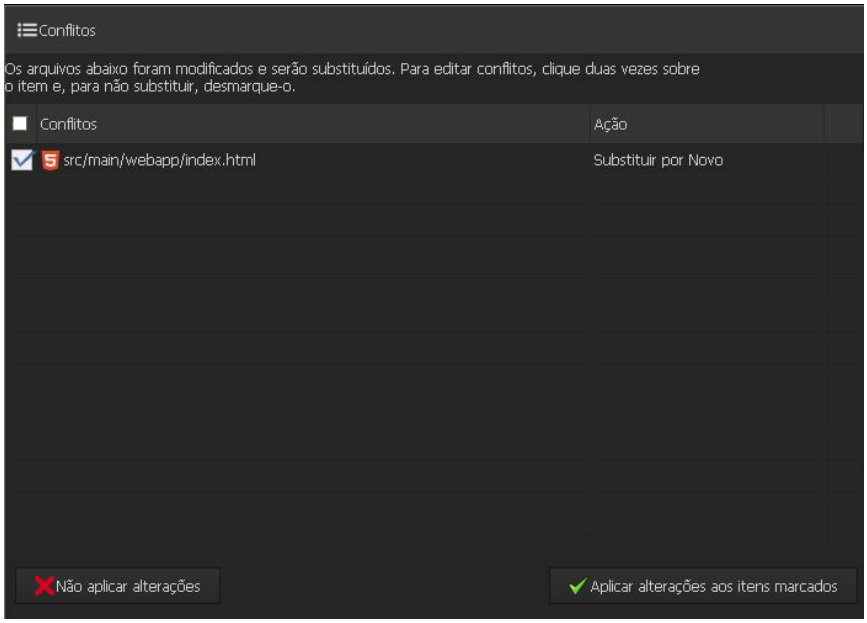

Figura 5 - Aplicando alterações realizadas

Passo 4: Acessar o arquivo JavaScritp criado

Após a aplicação das alterações, o arquivo JavaScript criado deve estar contido no diretório especificado com o seguinte padrão de nomenclatura: **nomeArquivo.cronapi.js**. O código base do arquivo criado para o componente vem com alguns parâmetros (anotações) tidos como padrão, como por exemplo: **@type**, **@name**, **@description**, **@multilayer**, **@param, @returns.** Esses parâmetros não são fixos, podendo ser alterados a depender da necessidade, ver (**Figura 6**).

Mais sobre
@type- Tipo do componente

@name - Nome da função(componente)

@description - Descrição sobre a funcionalidade do componente

@multilayer - Define se o componente criado estará disponível no tipo de bloco cliente e servidor(**True**) ou apenas do lado do cliente(**False**)

@param - Define qual o tipo do parâmetro de entrada da função(componente)

@returns - Define qual o tipo de retorno da função(componente)

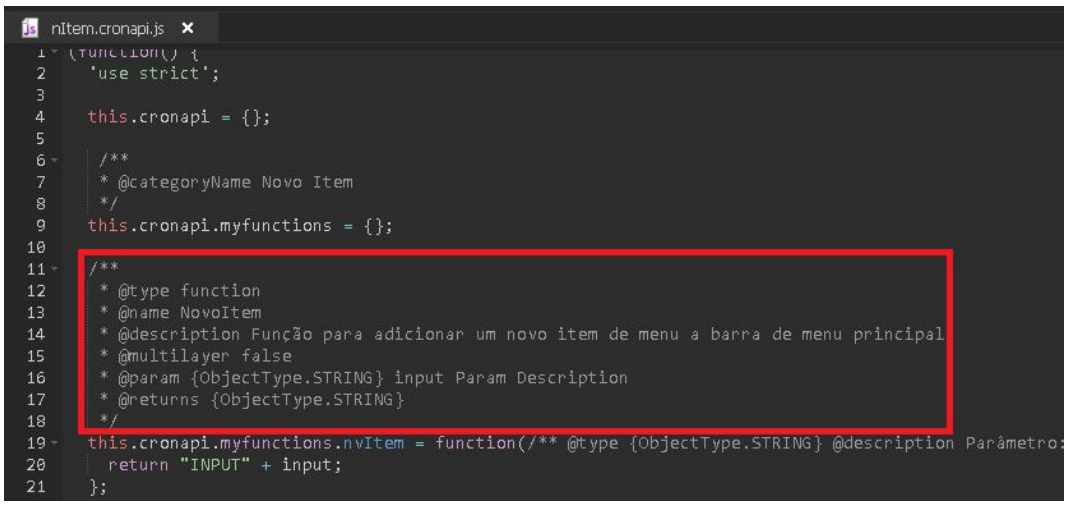

Figura 6 - Anotações padrão adicionadas junto a criação do aquivo JavaScript

Figura 7 - Alguns dos tipos de parâmetro de entrada e saída que podem ser utilizados

Passo 5: Adicionar código

Após a adequação dos parâmetros, o código referente a funcionalidade pode ser adicionado ao corpo da função. Para esse exemplo não é necessário que o componente tenha um retorno ou um parâmetro de entrada, portanto as anotações **@param** e **@returns** não se fazem necessárias, sendo excluídas do conjunto. O código que adiciona um novo item de menu a barra de navegação é o seguinte:

Exemplo

```
$($('.nav')[0]).append(' Menu Item Action Item ')
```

```
/**
 * @type function
 * @name NovoItem
 * @description Função para adicionar um novo item de menu a barra de menu principal
 * @multilayer false
 */
this.cronapi.myfunctions.nvItem = function() {

$($('.nav')[0]).append(' Menu Item Action Item ')

};
```

Figura 8 - Código adicionado ao corpo da função referente a funcionalidade do componente

Passo 6: Acessar a função

O último passo após a inserção do código no componente é a visualização desta função como disponível para utilização no bloco de programação do tipo cliente. Para componentes do tipo cliente, muitas vezes é necessário recarregar o projeto para que a nova função seja reconhecida (**Figura 9**).

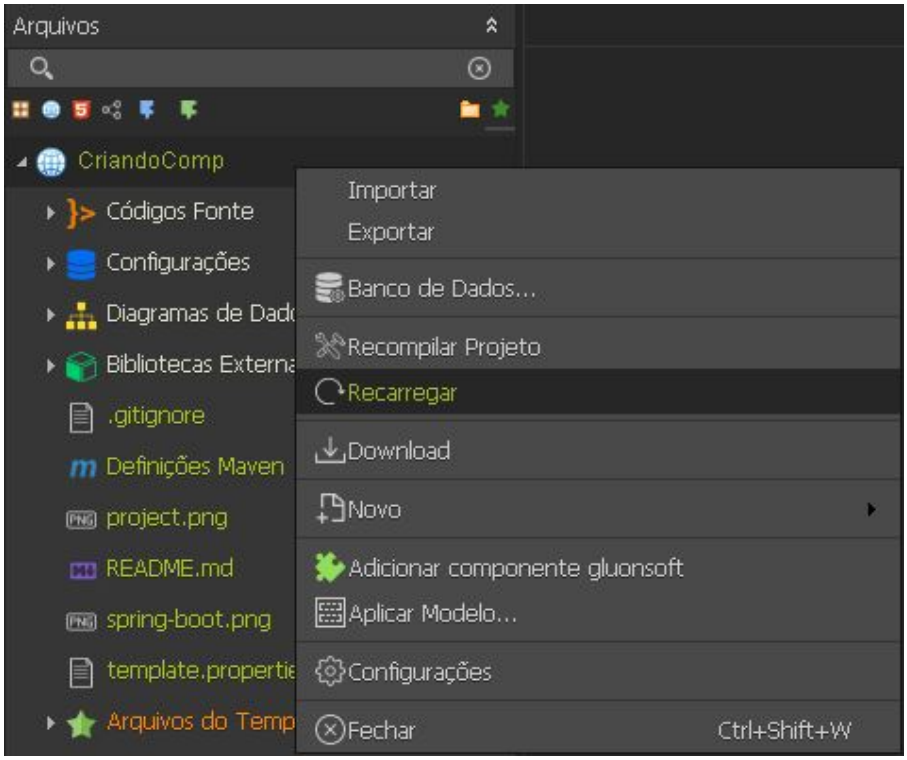

Figura 9 - Recarregando o projeto

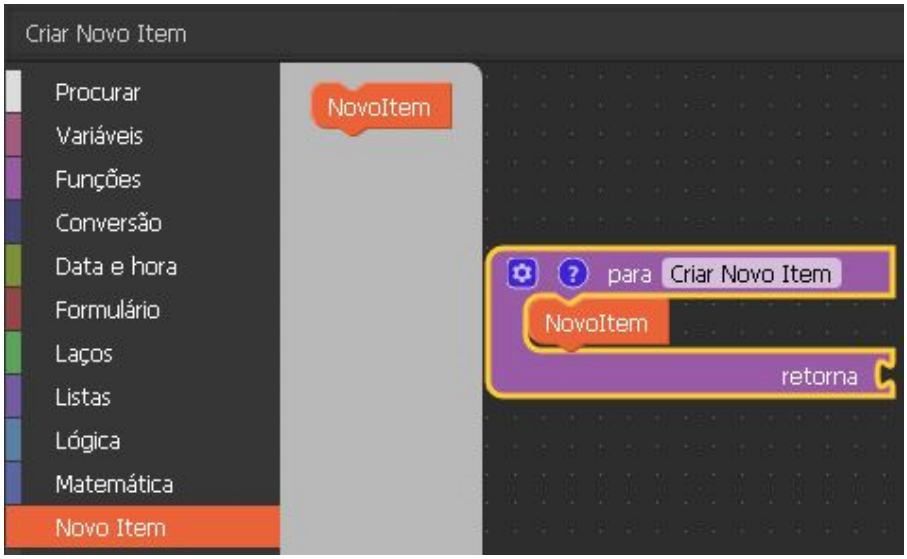

Figura 10 - Visualização e utilização do componente

Passo 7: Exibição de informações

É possível visualizar as informações definidas durante o desenvolvimento da função passando-se o ponteiro do mouse sobre o componente (**Figura 11**).

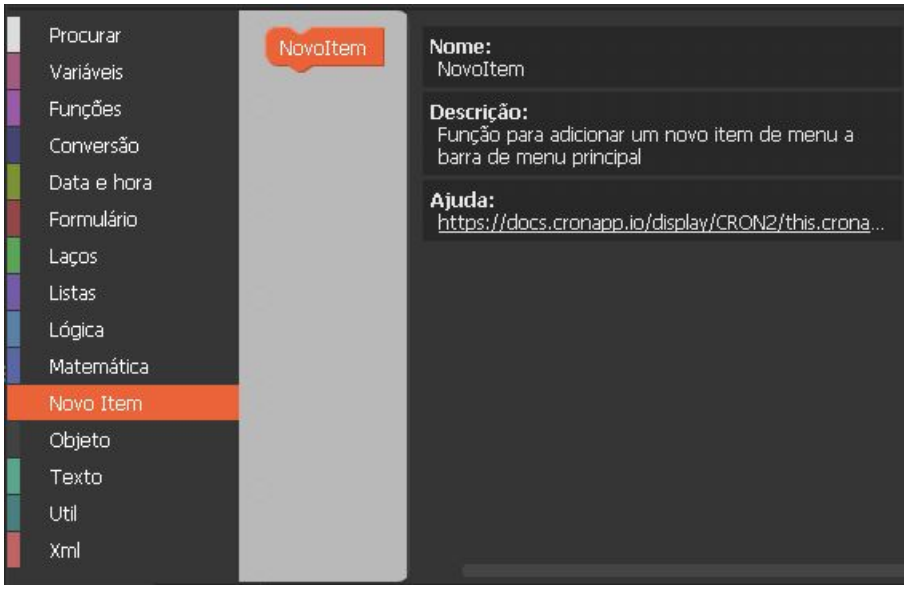

Figura 11 - Exibição das informações da função