

Fragmento

O componente **Fragmento** é utilizado para **renderizar o arquivo **Fragmento**** em outra página. A ideia da utilização do **Fragmento** é poupar tempo e esforço, já que os fragmentos são independentes e reutilizáveis, ou seja, basta criar uma vez e utilizar sempre que for necessário no projeto.

Caso esteja interessado em adicionar uma página interna ou externa ao Cronapp, acesse a documentação do componente [Frame \(web\)](#).



Figura 1- Componente **Fragmento** sendo utilizado para replicar os botões

Principais propriedades

Na tabela abaixo estão descritas as principais propriedades do componente.

Nome	Propriedade	Função
Fragmento	origin	Define qual arquivo Fragmento será utilizado.
Parâmetros	params	Campo destinado à informar os parâmetros que serão passados para o componente fragmento .
Identificador	id	Atributo que especifica um ID exclusivo para o componente. Por padrão, um ID é gerado automaticamente e atribuído ao componente.
Estilo	style	Altera o estilo geral do componente da forma <i>inline</i> .
Exibir	ng-show	Propriedade Angular usada para mostrar ou oculta o componente.
Repetir	crn-repeat	Propriedade que seleciona uma fonte de dados contida na tela e repete o componente baseado no número de itens dessa fonte de dados.
Segurança	cronapp-security	Propriedade do Cronapp que trata as permissões de segurança da aplicação. Usado para permitir ou não edição e visualização do componente com base nos perfis de selecionados.
Skin	crn-skin	Permite selecionar um skin que afetará apenas o componente selecionado.
Classes CSS	class	Adiciona classes CSS que já foram criadas.

Arquivo **Fragmento**

Para que o componente **Fragmento** funcione, primeiro temos que criar um arquivo do tipo **Fragmento**. Esse arquivo possui a extensão `*.component.html` e devem ficar nos diretórios:

- **Localização mobile:** `Fragmentos/mobile/`
- **Localização web:** `Fragmentos/web/`

Arquivos do tipo `Fragmento` abrem a janela do [Editor de views](#).

Exemplo

Nome em inglês

`fragment`

Nesta página

- [Principais propriedades](#)
- [Arquivo **Fragmento**](#)
- [Exemplo](#)
 - [Inserindo o componente **Fragmento** na view](#)
- [Passagem de parâmetros](#)
- [Drag and Drop da árvore de recursos](#)

Compatibilidade

- Formulário web
- Formulário mobile

Equivalente mobile

`Fragmento`

Botão do Componente

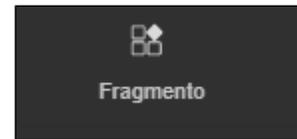


Imagem no Editor Visual



Assista sobre o tema no Cronapp Academy

Caso seja seu primeiro acesso ao Cronapp Academy, crie antes uma conta gratuita e matricule-se no curso abaixo.

- Aula: [Fragmentos Web](#)

Como exemplo de utilização, criaremos um fragmento que contém um botão de adicionar um produto ao carrinho.

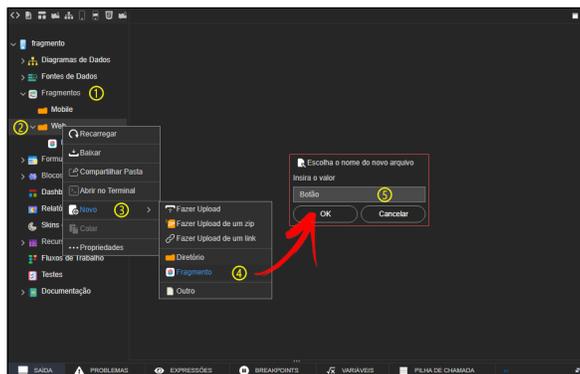


Figura 2 - Criando um Arquivo Fragmento

1. Na estrutura de arquivos clique com o botão direito no diretório **Fragments**.
2. Clique com o botão direito na pasta **Web**.
3. Clique em **Novo**.
4. Clique em **Fragmento**
5. Por fim, dê um nome ao arquivo, neste exemplo demos o nome de "Botão".

Agora iremos adiciona o botão no arquivo criado. É possível configurar esse botão da forma que quisermos, inclusive adicionando blocos de programação.

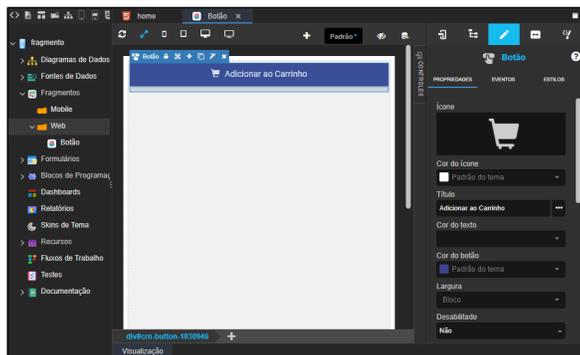


Figura 2.1 - Adicionando o botão

Inserindo o componente Fragmento na view

Com o arquivo Fragmento criado, é hora de adicionar o **componente Fragmento**, confira na figura 3.

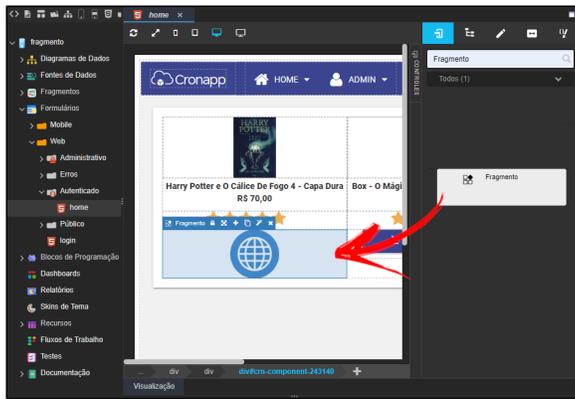


Figura 3 - Adicionando o componente Fragmento

Após inserir o componente na view, é hora de selecionar o arquivo Fragmento criado anteriormente. Clique nos '...' da propriedade **Fragmento** (destaque 1 da figura 3.1), em seguida selecione o Fragmento que deseja utilizar (destaque 2 da figura 3.1)

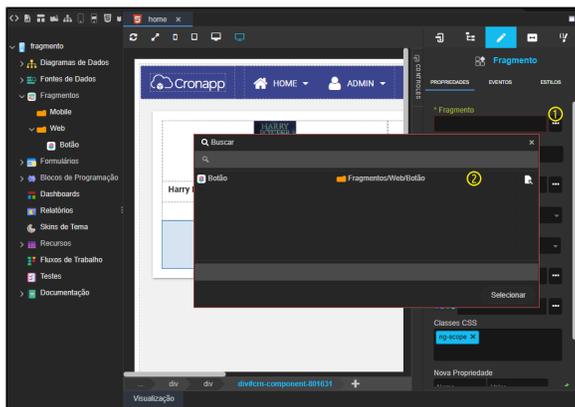


Figura 3.1 - Adicionando o arquivo Fragmento ao componente Fragmento

Na figura 3.2 é possível ver o que o Fragmento criado foi adicionado à página em edição.

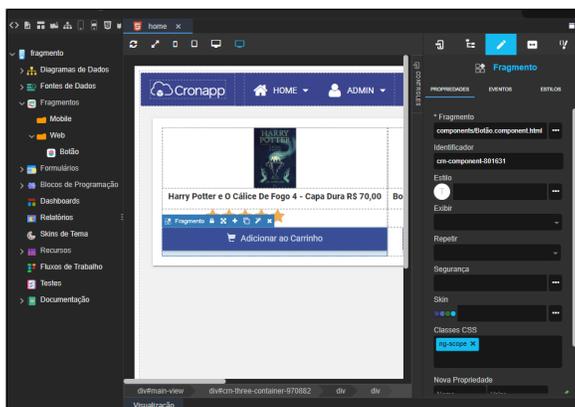


Figura 3.2 - Fragmento adicionado ao projeto

Passagem de parâmetros

Outra forma de criar o card do exemplo anterior é utilizando a propriedade **Parâmetros** do componente Fragmento. Essa propriedade permite passar parâmetros via *query string* da página que possui o componente fragmento para os campos dentro do arquivo fragmento. Esse fluxo é de mão única, logo, não é possível passar parâmetros do fragmento para a página que contém o fragmento.

Para este exemplo, vamos utilizar um [Componente visual fonte de dados](#) da página que contém o componente visual fragmento e passar os dados de cada registro (URL da imagem, preço e título do livro) via parâmetro, no arquivo fragmento vamos alimentar um componente [Imagem](#) e dois [Texto Simples](#) com esses dados.

Acesse o menu de contexto do arquivo fragmento e selecione a opção **Propriedades da página** (destaque 1 da figura 4) e, em seguida, clique no ícone "... " na propriedade **Parâmetros da tela** (destaque 2 da figura 4). Acesse o tópico "Propriedades e Eventos da página" da documentação [Editor de views](#), para mais detalhes sobre essa propriedade.

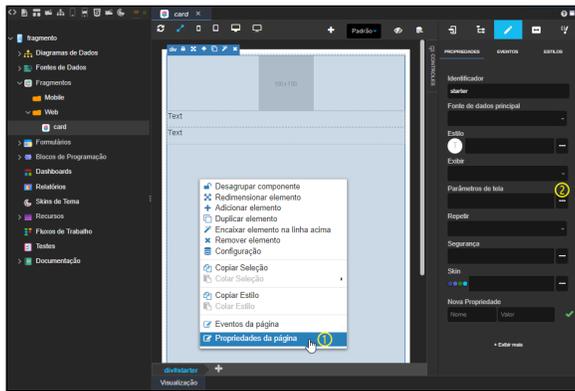


Figura 4 - Propriedades da página

Feito isso, uma janela será exibida para inserir os parâmetros *query string* que serão recebidos pelo fragmento.

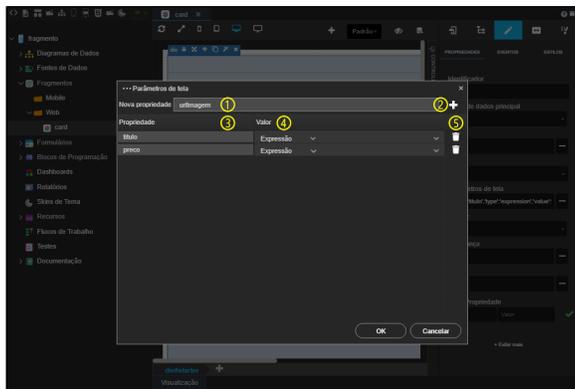


Figura 4.1 - Janela Parâmetros de tela

1. **Nova propriedade:** nome do novo parâmetro *query string* do fragmento.
2. **Adicionar:** insere o parâmetro na lista de propriedades.
3. **Propriedade:** coluna que informa os parâmetros já adicionados.
4. **Valor:** define o tipo do parâmetro, podendo ser: texto, numérico, data e hora, hora, lógico e expressão. O campo ao lado irá se adaptar com base no tipo selecionado.
5. **Excluir:** exclui uma propriedade da lista.

Para esse exemplo, incluímos os parâmetros "titulo", "preco" e "urlimagem". Não alteramos as colunas **Valor** (4), caso seja informado aqui, o valor do parâmetro será estático. Ao final da configuração, clique em **OK**.

Agora precisamos configurar os componentes visuais que serão alimentados pelos parâmetros que configuramos acima. Selecione o componente visual e clique na opção **Editar expressão** (destaque 1 da figura 4.2) da propriedade **Conteúdo** para os componentes **Texto Simples** e **Origem** para o componente **Imagem**, selecione a aba **Campo da tela** e escolha um dos parâmetro (figura 4.2).

Salve o arquivo fragmento.

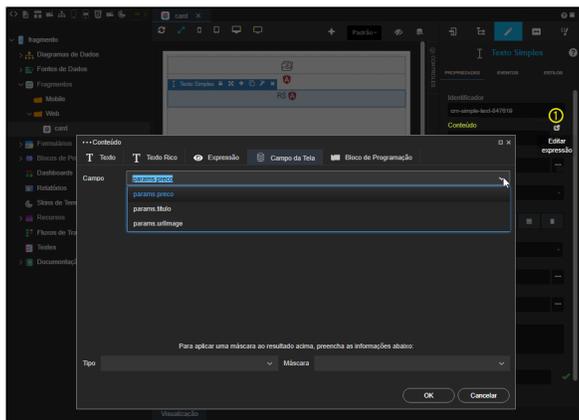


Figura 4.2 - Seleção de parâmetros

Na página que irá renderizar o Fragmento, arraste e configure o **Componente visual fonte de dados**, nesse exemplo selecionamos uma classe de Livros, e depois insira o componente visual **Fragmento**, da mesma forma como fizemos na figura 3.1.

Acesse a propriedade **Parâmetros** (destaque 1 da figura 4.3) do componente Fragmento e associe os campos da fonte de dados (coluna **Valor**) com os parâmetros do Fragmento (coluna **Propriedade**), como mostra a figura 4.3.

Acesse também a propriedade **Repetir** do componente Fragmento e selecione a Fonte de dados utilizada, dessa forma, o Fragmento será repetido para cada registro de livro contido na fonte de dados.

Atenção!

Nesta etapa é muito importante configurar a propriedade do **Componente visual fonte de dados**, **Postergar carga**, como **Não**, caso contrário a consulta não será realizada ao abrir a página.

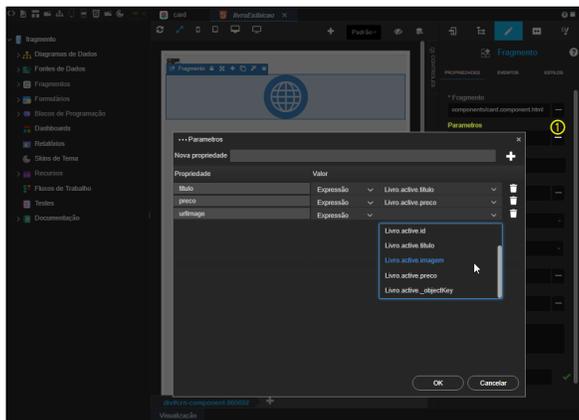


Figura 4.3 - Associação dos campos da fonte de dados com o fragmento

Após as configurações realizadas nos passos anteriores, podemos conferir o resultado na figura abaixo.

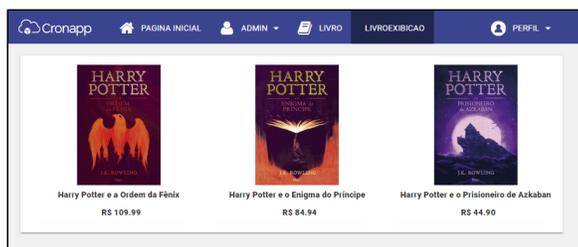


Figura 4.4 - Resultado da exibição do fragmento associado à fonte de dados

Drag and Drop da árvore de recursos

Outra forma de gerar o **componente Fragmento** é a partir do recurso de Drag and Drop (arrastar e soltar). Selecione um **arquivo do tipo Fragmento** da árvore de recursos e arraste-o para a área de edição. Para entender um pouco mais sobre o recurso, acesse o tópico "Drag and Drop da árvore de recursos" da documentação [Editor de views](#).

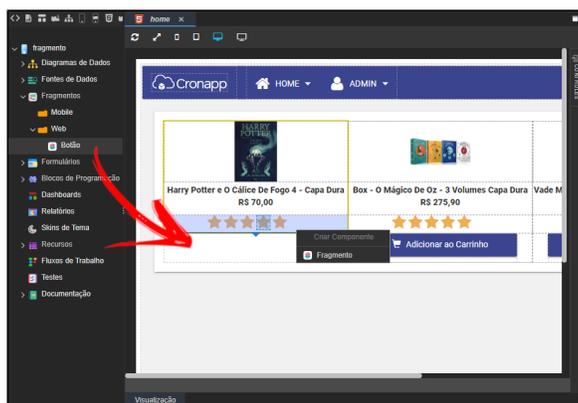


Figura 5 - Arrastando um arquivo Fragmento da árvore de recursos para o Editor de views

Confira o resultado na figura 5.1 da adição do componente na view via Drag and Drop.

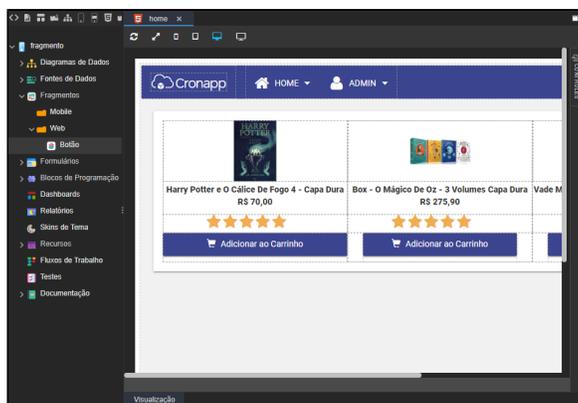


Figura 5.1 - Componente Fragmento gerado através do Drag and Drop