Editor de views

O Editor de *views* do Cronapp é responsável por modelar o *layout* das páginas *web* e das telas dos aplicativos *mobile*. Através da técnica *drag and drop* (arrasta e solta) é possível construir páginas e telas utilizando os diversos componentes visuais, sem a necessidade de manipulação no código HTML ou CSS gerado. O resultado é uma página HTML5 crua sem *scripts server-side* e que permite configurar os eventos cliente (JavaScript) e servidor (Java) de forma *Low code*.

Caso queira, também é possível criar ou alterar o conteúdo de uma página em modo *high code*, trabalhando diretamente com o código, e ainda manipular e personalizar os *templates* das páginas CRUD geradas pelo Diagrama de dados. As aplicações geradas pelo Cronapp são responsivas e seguem os padrões definidos pelo W3C.

Observação

Páginas criadas em modo *high code*, sem o Editor de views, não possuirão o mesmo padrão das páginas criadas pelo editor - sendo assim, algumas funcionalidades, como vincular com outras páginas HTML criadas em modo *low code* ou eventos de componentes podem não funcionar.

Acesso ao editor

Para abrir o editor de *views*, bastar dar um duplo clique nos arquivos contidos nos seguintes diretórios:

- Páginas da aplicação **Web** (Localização: `Formulários/Web/`).
- Telas da aplicação **Mobile** (Localização: `Formulários/Mobile/`).
- Fragmentos de páginas **Web** (Localização: `Fragmentos/Web/`).
- Fragmentos de telas **Mobile** (Localização: `Fragmentos/Mobile/`).

Para gerar uma nova página, clique no ícone (**+**) ao lado do diretório desejado, web ou mobile. A janela de seleção de **Modelos** será exibida.

Modelos são exclusivos para páginas web ou telas mobile, não é possível selecionar um modelo ao criar um Fragmento,

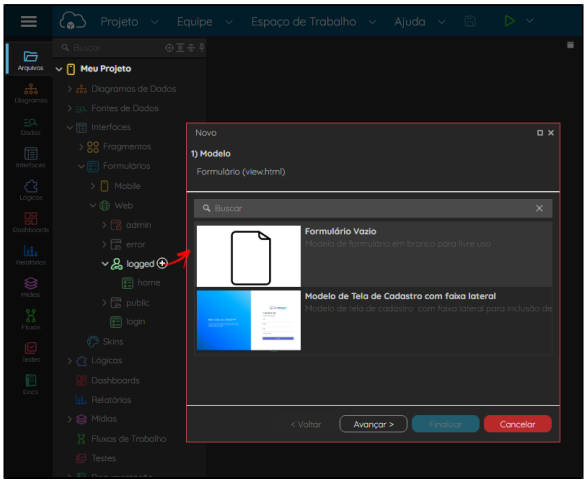

Figura 1 - Janela de seleção de modelos para páginas web

Na lista de modelos é possível selecionar uma página vazia ou com um layout diferente de alguma página do projeto modelo, como as páginas usadas ao logar na aplicação ou cadastrar um novo usuário.

Atualmente temos as seguintes opções:

- **Web**
  - **Formulário Vazio**: modelo de formulário em branco para uso livre.
  - **Cadastro com faixa lateral**: modelo de tela de cadastro com faixa lateral para inclusão de textos.

Nesta página

- Acesso ao editor
- Janela de edição
  - Área de design
  - Redimensionamento
  - Acessibilidade
  - Modal
    - Seleção do modal
  - Menu lateral
    - Controles
    - Hierarquia
      - Árvore DOM e subcomponentes
    - Propriedades dos componentes
      - Subaba Propriedade
      - Subaba Eventos
      - Subaba Estilos
    - Responsividade
    - Estilos
  - Subabas visualização e código
- Propriedades e Eventos da página
  - Parâmetros da tela
- Drag and Drop da árvore de arquivos
  - Arquivos aceitos no Editor web
  - Arquivos aceitos no Editor mobile

Conteúdo complementar

- Componentes visuais
  - Estilos - CSS
  - Eventos dos componentes visuais
  - Propriedades dos componentes visuais
  - Formatação de máscaras na camada servidor
  - Formatação de máscaras na camada cliente
- Internacionalização
- Responsividade
- Customização de templates

Componentes visuais

- **Cadastro com texto lateral**: modelo de Tela de Cadastro com texto lateral.
- **Login com faixa lateral**: modelo de tela de login com faixa lateral para inclusão de textos.
- **Login com texto lateral**: modelo de tela de login com frame lateral para inclusão de textos informativos.

- **Mobile**
  - **Formulário Vazio com Cabeçalho**: modelo de formulário mobile em branco com cabeçalho para uso livre.
  - **Formulário Vazio**: modelo de formulário mobile em branco para uso livre.

Após selecionar o modelo, clique em **Avançar**, informe o nome do novo arquivo (figura 1.1) e clique em **Finalizar**, o Editor de *views* será aberto em sequência.

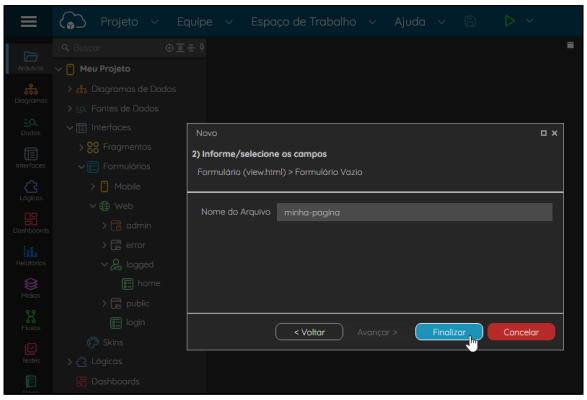

Figura 1.1 - Nome do novo arquivo

Janela de edição

O nosso editor visual é do tipo WYSIWYG (*What You See Is What You Get* ou “o que você vê é o que você obtém”, em português), ou seja, a página em modo de edição será semelhante ao modo em execução, facilitando muito o desenvolvimento. O menu lateral (item 6 da figura 1.2) possui cinco abas superiores: **Controle**, contém os componentes visuais; **Hierarquia**, exibe a estrutura hierárquica dos componentes na tela; **Propriedades**, responsável por personalizar os componentes; **Responsividade** e **Estilos** (CSS).

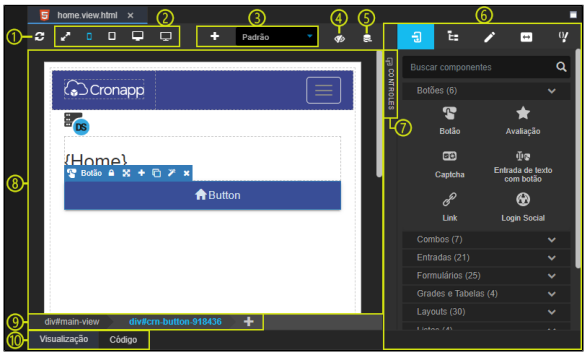

Figura 1.2 - Área de trabalho da view

1. Recarregar Área de Design.
2. Redimensionamento: redimensiona a tela para verificar a responsividade em diferentes equipamentos.
3. Modal: botão para criar modal e caixa de seleção dos modais.
4. **Exibir elementos ocultos:** exibe todos os elementos que serão renderizados em algum momento após uma ação feita pelo usuário.
5. **Exibir elementos não visuais:** exibe/oculta elementos que não serão renderizados ao executar o projeto, como ícones de fonte de dados.

- Componentes Web
- Componentes Mobile

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: Modal - Parte 1

6. Menu lateral**:** área que abrange as abas de controle, hierarquia, propriedade, responsividade e estilos.
7. **Botão Controles**: permite expandir ou recolher o Menu lateral (6), ampliando a área de design da página.
8. Área de design**:** área de confecção das páginas *web* ou telas *mobile*.
9. Árvore DOM: exibe a árvore DOM (*Document Object Model*) do componente selecionado na área de edição.
10. Abas visualização e código: exibe o conteúdo de forma visual ou o código-fonte HTML.

    É possível exibir a subaba **código** através do Modo Avançado, ou habilitando a opção **Sempre mostrar código fonte** a partir do menu do sistema **Espaço de Trabalho** > **Preferências**.

Área de *design*

Para criar e configurar o layout das páginas web ou telas mobile, os componentes visuais devem ser arrastados do menu lateral (aba Controles) para o centro do editor de views (ação *drag and drop*), enquanto o botão do mouse estiver pressionado, é possível definir o loca onde ele será "solto", uma linha com um seta azul (destaque 1 da figura 2) indica a posição em que o componente ficará. Também é possível reordenar os componentes em tela, utilizando o mesmo recurso de arrastar e soltar.

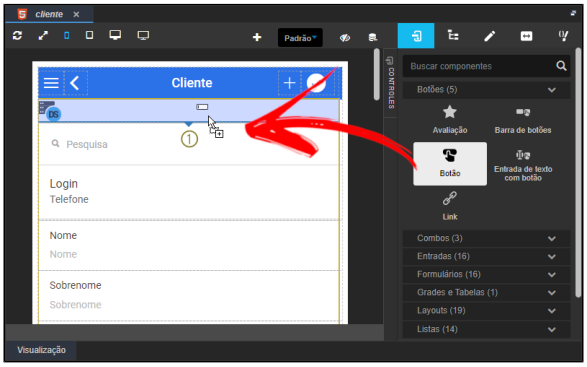

Figura 2 - Incluindo o componente visual Botão na tela

Cada componente em tela exibe um contorno pontilhado mostrando a área em que ocupa na tela. Ao clicar em um componente na área do editor, ele será selecionado e uma **Barra de opções** é exibida logo acima, mostrando seu nome e exibindo algumas opções de edição (destaque 1 da figura 2.1). Veja mais detalhes sobre a barra de opção em Componentes visuais.

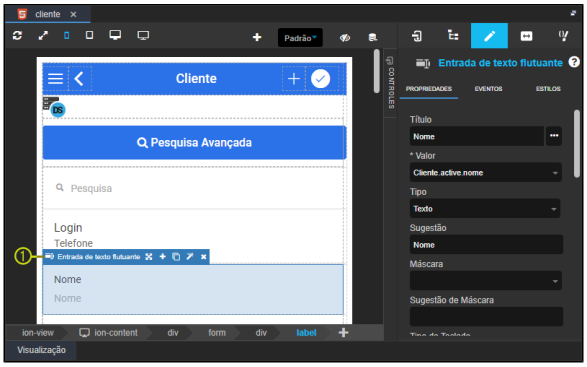

Figura 2.1 - Componente selecionado e destaque para a barra de opções

Redimensionamento

No menu superior da área da edição se encontra as ferramentas para redimensionamento da página, facilitando a visualização da página em diferentes tamanhos de tela.

Figura 3 - Itens do redimensionamento

1. **Recarregar área de design**: atualiza a área de design.
2. **Resetar:** ajusta a área de design do formulário baseando-se no tamanho da janela no browser.
3. **Área de redimensionamento:** redimensiona a área de design para mostrar como é seu comportamento conforme o dispositivo selecionado:
    - **Dispositivos pequenos, telefones**;
    - **Dispositivos médios, *tablets***;
    - **Dispositivos *desktops***;
    - **Dispositivos grandes, TVs**.

Acessibilidade

As aplicações desenvolvidas no Cronapp podem ser acessíveis para usuários com certo grau de deficiência motora e visual. Temas como o Material seguem os padrões da W3C e WCAG (*Web Content Accessibility Guidelines*) 2.0 e 2.1 na construção de suas aplicações. Além disso, o Cronapp utiliza o *Wave Web Accessibility*, ferramenta de avaliação de acessibilidade para atestar a conformidade com os padrões WCAG, garantindo que as aplicações geradas sigam esses padrões (Figura 4).

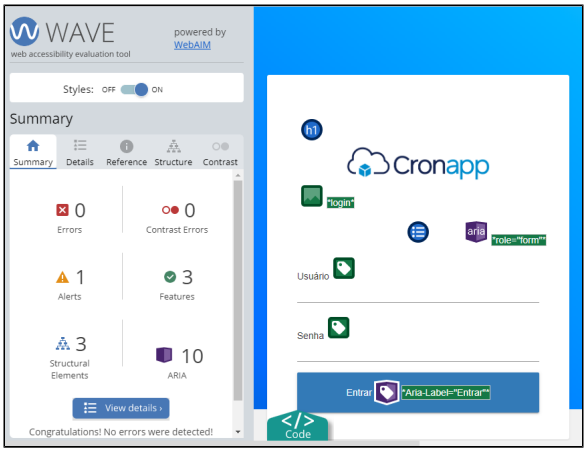

Figura 4 - *Plugin* Wave rodando numa aplicação Cronapp

As competências referentes à acessibilidade presentes no Cronapp são:

- **Estrutura do documento:** possui um HTML bem definido que respeita as regras de uso, como por exemplo, possuir um `DOCTYPE` apropriado, subordinação obrigatória entre as tags é respeitada e tags são corretamente aninhadas e fechadas;
- **Divisão de áreas de informação:** tem como objetivo organizar a estrutura da aplicação de forma a garantir a consistência entre as páginas definindo bem as divisões topo, conteúdo, menu, rodapé e possui como algumas regras de uso ter a seção do conteúdo principal marcada com a regra *main*, seção que tem um formulário de busca marcado com a regra *search* e rodapé contendo as informações versão, sobre, contato, etc. e marcada com a regra *contentinfo*;
- **Semântica das tags:** visa manter a lógica, semântica, ordem dos elementos e tem como algumas regras de uso blocos marcados com `p`, `table` não é usado para posicionar elementos na tela e cabeçalhos marcados com `h1`, `h2`, ... `h6`;
- **Leitura da tela:** permitindo aos usuários cegos ou com baixa visão compreensão e acesso total às funcionalidades do sistema, desde manipulação de arquivos e fotos, cadastros, criação e edição de documentos de texto rico, navegação, entre outras funcionalidades.
- **Fluxo de leitura e tabulação:** consiste em manter a ordem lógica das apresentações para facilitar leituras, navegação entre *links* e possui as regras de uso conteúdo apresentados na sequência em que se espera que o usuário leia, paradas de tabulação aparecem na sequência em que se espera que o usuário acesse;

- **Uso de cabeçalhos:** possui regras de uso, como `h1` usado apenas uma vez na página, níveis não são saltados da hierarquia, ou seja, depois de `h1` se usa `h2` e assim por diante, o `h1` não precisa ser necessário o primeiro cabeçalho na ordem de leitura;
- **Atalhos para conteúdo e navegação:** tem como algumas regras de uso *link* para ir direto ao conteúdo principal da página, link que leva ao conteúdo é o primeiro elemento a receber foco com o pressionamento da tecla TAB, link para ir direto à navegação da página;
- **Totalmente acessível por teclado:** obtém como regras de uso a possibilidade de realizar toda e qualquer ação em uma página usando somente o teclado, controle de foco e visualização deve permitir uma experiência usável quando se navega por teclado;
- **Visual:** visa garantir que o texto possua contraste suficiente para a leitura, e tem regras de uso, como por exemplo, texto regular com contraste mínimo de 4.5:1 entre cor de fonte e de fundo, indicação visual evidencia o elemento que possui foco e texto grande (a partir de 18 pontos, ou 14 pontos com negrito) com contraste mínimo de 3:1 entre cor de fonte e de fundo;
- **Links:** possui algumas regras de uso como texto do *link* indica com clareza seu destino, *links* que levam ao mesmo destino possuem o mesmo texto e *links* que levam a destinos diferentes possuem textos diferentes;
- **Imagens:** contém regras de uso como imagens possuem o atributo `alt` que possui conteúdo variável que depende de como a imagem é usada na página, imagem decorativa inserida por CSS (preferencialmente) não necessita do atributo `alt` e imagem que faz parte do conteúdo inserida por HTML e não por CSS;
- **Tabelas:** tem regras de uso, como por exemplo, título da tabela (quando houver) informado por meio da tag `caption`, cabeçalhos contidos em tags `th` e dados contidos em tags `td`;
- **Formulários:** obtém algumas regras de uso como ao navegar por teclado o controle deve receber o foco, controle operável por teclado e controle associado a seu rótulo (*label*) de ao menos uma dessas formas (atributo id do controle possui o mesmo valor do atributo for do rótulo correspondente, tag do controle posicionada dentro da tag `label` correspondente) e controles que guardam forte relação entre si agrupados com uso de `fieldset`;
- **Multimídias:** dispõem algumas regras de uso, conteúdo multimídia não inicia automaticamente sua reprodução, controle de execução operável por teclado e legenda em vídeo que contém áudio falado.

Modal

Modais são pequenas janelas que ficam sobrepostas sobre a tela principal, usadas para chamar a atenção do usuário, exibir um conteúdo ou adicionar uma funcionalidade extra na tela. A janela modal é constituída de três áreas: ***header***, ***body*** e ***footer***; e para adicionar componentes nessa área, basta selecionar os componentes e arrastá-los sobre a área do modal (Figura 5).

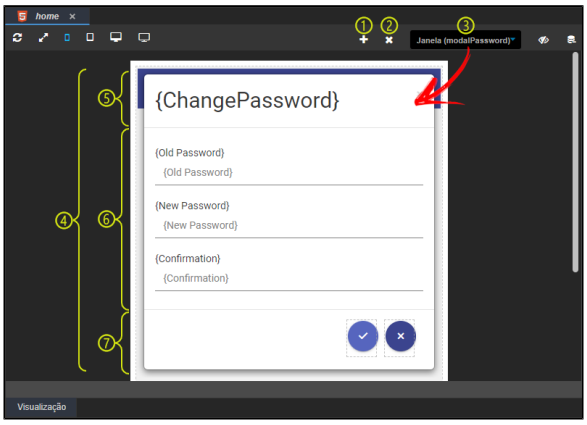

Figura 5 - Modal

1. **Adicionar janela:** cria uma modal;
2. **Remover janela:** remove a janela modal selecionada na caixa de seleção lateral;
3. **Caixa de seleção das janelas modais**;
4. **Janela do modal**: área de design do modal.
5. **Header**: cabeçalho do modal.
6. **Body**: conteúdo do modal.
7. **Footer**: rodapé do modal.

Seleção do modal

Para selecionar o modal e realizar algumas ações, como renomear, basta clicar no centro do modal e, na árvore DOM, clicar na `div` referente ao modal (destaque 1 da Figura 5.1), dessa forma o menu lateral irá exibir as propriedades do modal (2 da Figura 5.1).

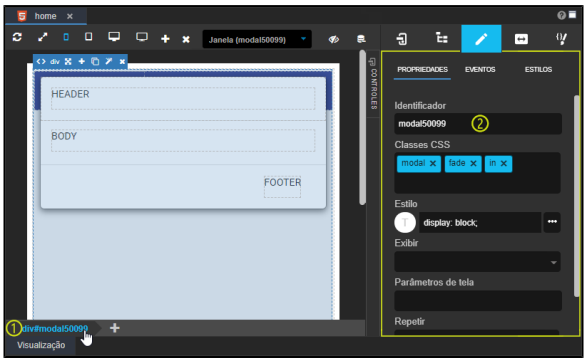

Figura 5.1 - Campo para renomear modal

Menu lateral

O menu lateral possui quatro abas que contém os componentes que são adicionados na área de design, onde mostra a configuração e a estilização desses componentes.

Figura 6 - Abas do menu lateral

1. Controles;
2. Hierarquia;
3. Propriedades;
4. Responsividade;
5. Estilos.

Controles

Contém os componentes visuais que são utilizados para o *design* das páginas e telas (Figura 6.1), esses componentes variam conforme o tipo de projeto - ou seja, se é uma *view web* ou *mobile*. O campo de busca permite pesquisar pelos componentes usando o termo em português ou inglês, independente do idioma configurado no Cronapp.

Os componentes visuais estão organizados em categorias, podendo um componente está vinculado a mais de uma categoria, caso possua relação. A última categoria exibirá todos os componentes disponíveis.

Acesse a página Componentes visuais para maiores informações.

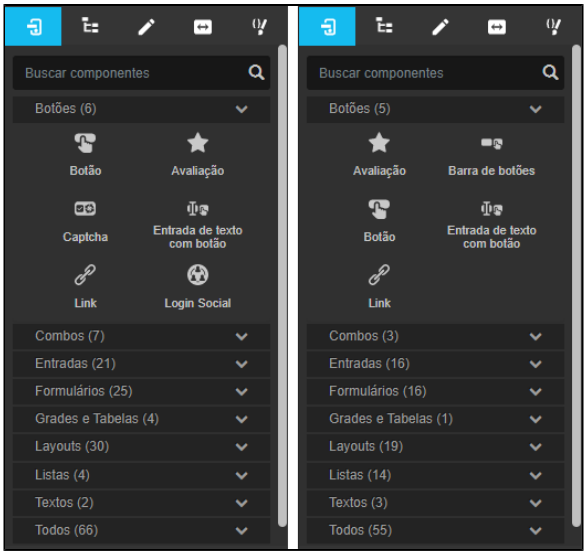

Figura 6.1 - Aba de controle no editor de *view web* e *mobile*

Hierarquia

Exibe uma árvore com a estrutura DOM (Modelo de Objeto de Documento) da página e destaca o elemento selecionado (destaques 1 e 3 da figura 6.2). Essa mesma navegação estrutural (*breadcrumb*) pode ser vista na parte debaixo da janela (destaque 2 da figura 6.2).

Figura 6.2 - Hierarquia da estrutura DOM

Árvore DOM e subcomponentes

O **Modelo de Objeto de Documento (*DOM*)** é uma interface de programação para documentos HTML, XML e SVG. Ele fornece uma representação estruturada do documento como em uma árvore. Imagine que o elemento `<html>` é a raiz dessa árvore e ele possui filhos, como o `<Head>` e o `<body>`, que por sua vez também possuem outros elementos filhos e assim sucessivamente. Cada elemento da árvore DOM pode possuir atributos e eventos (exemplo `<input type="text">`), no Cronapp chamamos respectivamente de Propriedades e Eventos. Clique aqui para mais detalhes sobre o conceito da árvore DOM.

No Cronapp, a estrutura DOM da página atual pode ser visualizada através da aba **Hierarquia** (destaque 1 da figura 6.2.1). Ao selecionar um componente visual na tela, o elemento que o representa na árvore DOM é destacado nessa hierarquia (2 da figura 6.2.1) e a sequência hierárquica, da raiz até o elemento, será exibida no formato de navegação estrutural (*breadcrumb*) na parte debaixo da janela (destaque 3 da figura 6.2.1). Na prática, quando selecionamentos um componente visual na área de edição, estamos selecionando todo o código do seu elemento HTML juntamente com todos os seus subelementos.

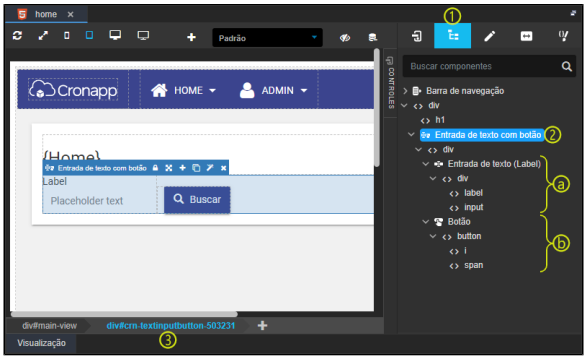

Figura 6.2.1 - Ferramentas que exibem a árvore DOM do componente selecionado: *breadcrumb* e aba Hierarquia

Alguns componentes visuais Cronapp são formados por outros componentes Cronapp (subcomponentes), como o Entrada de texto com botão, que contém os subcomponentes Entrada de texto e o Botão (destaques "a" e "b" da figura 6.2.1). Dessa forma, para selecionar, na área de *design*, um componente Cronapp dentro de outro componente, **não** é necessário desbloquear o componente principal. No exemplo da figura 6.2.2, mesmo com o componente **Entrada de texto com botão** bloqueado (destaque 1 da figura 6.2.2), o subcomponente **Botão** ainda é acessível ao passar o mouse por cima.

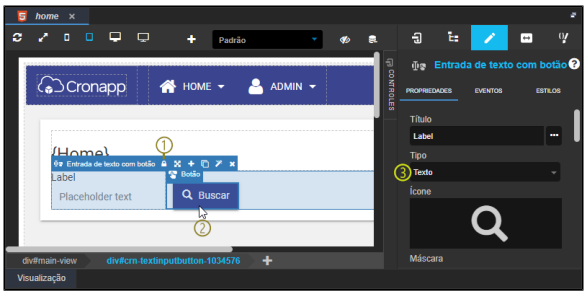

Figura 6.2.2 - Não é necessário desbloquear um componente para selecionar um componente Cronapp filho

Já para selecionar elementos HTML dentro de um componente, primeiro é necessário desbloquear o componente visual, clicando sobre o **ícone cadeado** na **barra de opções** (destaque 1 da figura 6.2.3). No exemplo abaixo, para tornar o elemento HTML `input` selecionável (destaque 2 da figura 6.2.3), primeiro foi necessário desbloquear o componente **Entrada de texto** (destaque 1).

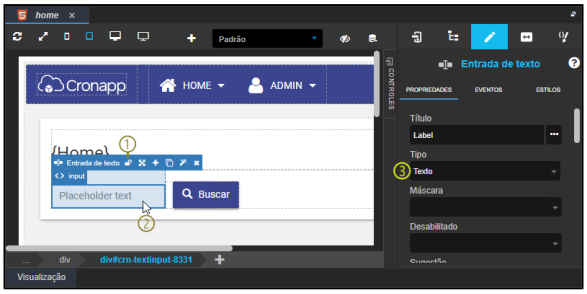

Figura 6.2.3 - É necessário desbloquear o componente principal antes de selecionar elementos HTML

Os campos das subabas **Propriedades** e **Eventos** exibidos no menu lateral (figura 6.3) fazem referência aos atributos e eventos do elemento HTML selecionado e, muitas vezes, também aos principais atributos e eventos de seus subelementos. O campo **Tipo** (destaque 3 da figura 6.2.2), exibido ao selecionar o componente **Entrada de texto com botão**, é uma propriedade do elemento HTML `input` do subcomponente **Entrada de texto**. Essa mesma propriedade também estará disponível ao selecionar

o componente **Entrada de texto** (destaque 3 da figura 6.2.3) e o elemento HTML que contém essa propriedade, o `input` (destaque 2 da figura 6.2.3).

Saber navegar entre os componentes e elementos é importante para permitir selecionar ou adicionar novas propriedades e eventos específicos. Por exemplo, para que o componente **Entrada de texto** funcione apenas como leitura, é necessário adicionar a propriedade **Apenas leitura** (ng-readonly) no elemento HTML `input`.

A navegação estrutural via *breadcrumb* (3 da figura 6.2.1) e o desbloqueio do ícone **cadeado** é ótima para acessar elementos próximos, como o elemento "pai" ou "filho", porém, para acessar elementos mais distantes recomendamos o uso da aba Hierarquia.

Acesse o tópico "Barra de opções" na documentação da Componentes visuais para informações sobre os demais ícones.

Propriedades dos componentes

As propriedades de um componente são exibidas sempre que o mesmo é selecionado na área de design e seu conteúdo pode variar a depender do item selecionado. Os campos de propriedades permitem alterar de forma *low-code* o conteúdo do elemento HTML, sejam atributos, eventos ou estilos (*inline*)

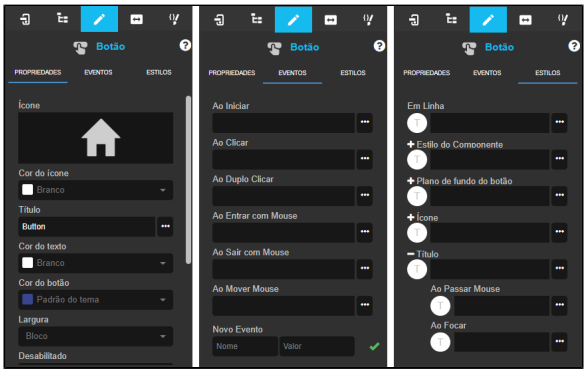

Figura 6.3 - Subabas Propriedades, Eventos e Estilos das propriedades do componente visual Botão

Subabas:

- Propriedades**:** permite adicionar ou editar as propriedades HTML, Angular, permissão de acesso e outros recursos.
- Eventos**:** permite adicionar eventos ao componente selecionado, como abrir um novo formulário, relatório, executar um bloco de programação ou ainda uma expressão.
- Estilos**:** permite editar ou adicionar CSS nos formatos em linha ou incorporado no componente selecionado.

Ao selecionar um componente visual da área de edição, apenas as principais propriedades e eventos são exibidas, para exibir mais, clique em **+ Exibir mais** (destaque 1 da figura 6.4) ou informe o nome da propriedade que deseja adicionar e seu valor no campo **Nova Propriedade** (2).

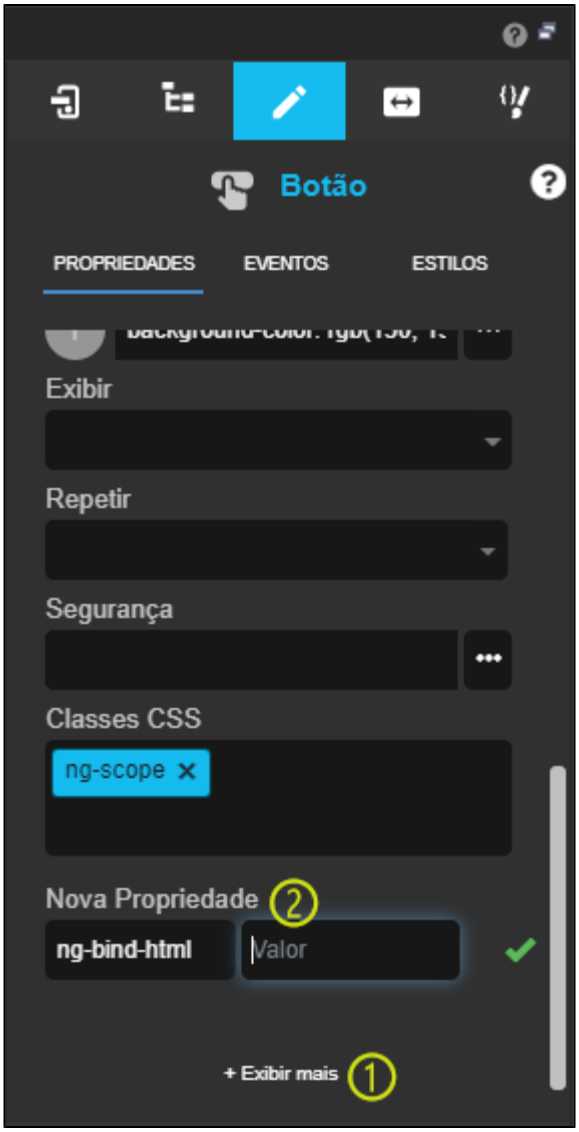

Figura 6.4 - Campo para adicionar novas propriedades e link para exibir as propriedades ocultas

Subaba Propriedade

Os campos dessa subaba alteram as características do componente selecionado através de propriedades HTML, Angular ou Cronapp. Seus títulos possuem nomes amigáveis, para saber o nome real da propriedade, posicione o cursor do mouse sobre o título e um *tooltip* exibirá o nome entre parênteses (tooltip da figura 6.5). Ao selecionar uma propriedade, alguns botões de configuração são exibidos (destaque 1 da figura 6.5):

- **Traduzir**: abre a janela de internacionalização e permite traduzir o conteúdo da propriedade. Acesse a documentação Internacionalização para mais detalhes.
- **Editar expressão**: abre uma janela para configurar o conteúdo da propriedade através de um Texto livre, Expressão, Campo da tela ou Bloco de programação.
- **Limpar**: exclui o valor da propriedade, deixando-a com o valor vazio.
- **Remover propriedade**: exclui a propriedade e seu valor do componente, na prática, remove o atributo HTML do componente.

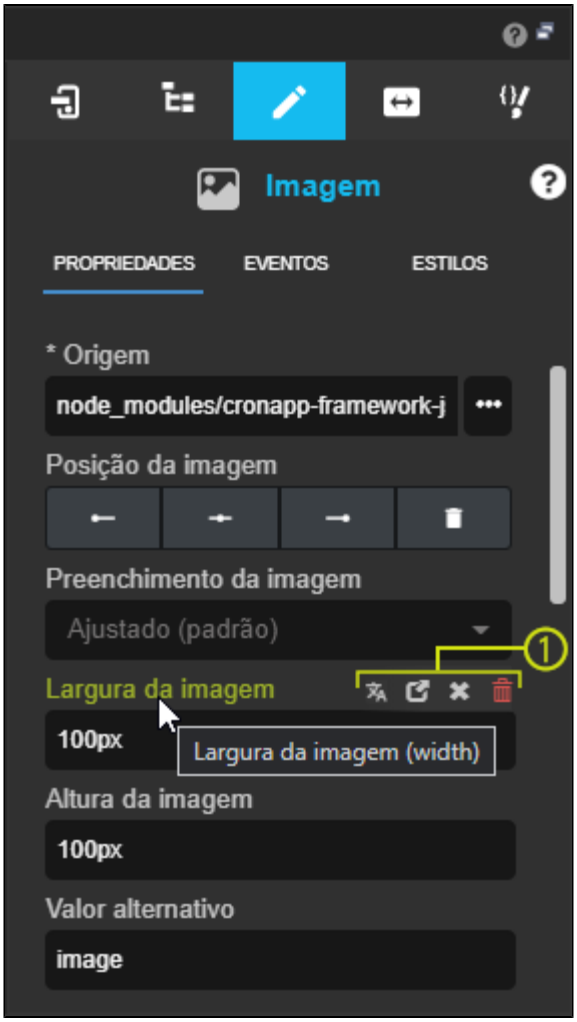

Figura 6.5 - Propriedades do componente selecionado

Acesse a documentação Propriedades dos componentes visuais para mais detalhes.

Subaba Eventos

Os campos dessa subaba alteram os eventos do componente selecionado através de propriedades HTML, Angular ou Cronapp. Seus títulos possuem nomes amigáveis, para saber o nome real do evento, posicione o cursor do mouse sobre o título e um *tooltip* exibirá o nome entre parênteses (tooltip da figura 6.6). Ao selecionar um evento, o botão **Limpar** (**x**) será exibido ao lado do título, ele irá excluir o valor do evento (destaque 1 da figura 6.6).

Todos os eventos possuem o botão "**...**" (2) ao lado do campo, esse botão abre uma janela para indicar qual a ação do evento.

Figura 6.6 - Eventos do componente selecionado

Acesse a documentação Eventos dos componentes visuais para mais detalhes.

Subaba Estilos

Os campos dessa subaba permitem etilizar o componente selecionado e seus subcomponentes. O primeiro campo (**Em Linha** - destaque 1 da figura 6.7) permite adicionar conteúdo CSS no formato *inline*, já os demais campos (2 da figura 6.7) são específicos para cada subcomponente contido no componente principal e o CSS é adicionado no formato incorporado. Os campos de estilo dos subcomponentes podem expandir para mudar o CSS dos subcomponentes em duas ações (destaque a da figura 6.7): Ao passar o mouse sobre o subcomponente e ao focar o subcomponente.

Todos os campos possuem o botão "**...**" ao lado, ao clicar, o editor CSS low-code será aberto.

Figura 6.7 - Edição do CSS do componente selecionado

Acesse a documentação do editor de Estilos - CSS para mais detalhes.

Responsividade

Essa aba permite configurar o tamanho do componente selecionado em relação ao tamanho da tela, podendo ser dividido em até 12 colunas horizontais. É possível configurar e ajustar a responsividade dos componentes para até 4 tamanhos de telas diferentes: **smartphones** (dispositivos pequenos), **tablets** (di spositivos médios), **monitores** e **televisores** (dispositivos grandes).

Clique em "+ Exibir todos dispositivos" ou "- Exibir apenas dispositivo selecionado" (destaque 1 da figura 6.8) para expandir ou ocultar a configuração dos demais dispositivos.

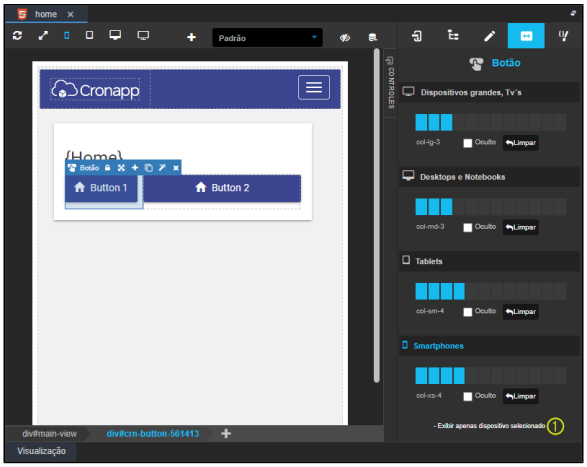

Figura 6.8 - Aba de Responsividade

Acesse a documentação sobre Responsividade para mais detalhes.

Estilos

Essa aba lista as classes CSS já criadas em seu projeto e permite criar novas (destaque 1 da figura 6.9), editar ou excluir (2). As classes criadas aqui serão armazenadas no arquivo `app.css` () e serão associadas aos componentes no formato CSS linkado ou externo.

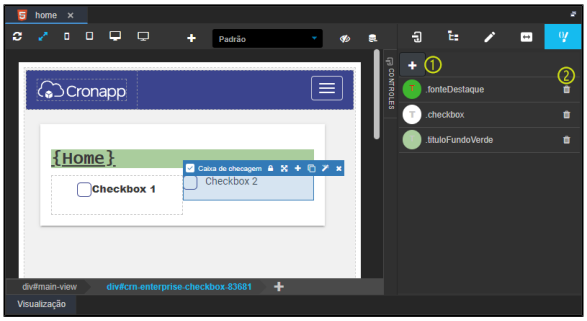

Figura 6.9 - Aba de Estilos

Após criar a classe, é possível vincular a um componente utilizando a propriedade **Classes CSS** do componente selecionado, basta clicar na última linha da caixa de classes para exibir uma caixa de seleção com as classes criadas.

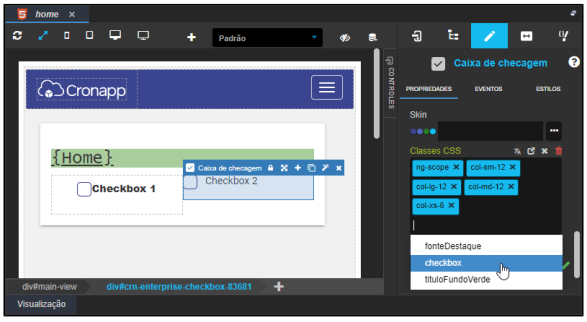

Figura 6.10 - Associando uma classe a um componente

Acesse a documentação sobre Estilos - CSS para mais detalhes.

Subabas visualização e código

O Editor de views possui duas subabas: **Visualização** e **Código** (destaque 2 da figura 7). A subaba **Visualização** exibe a tela no padrão WYSIWYG e permite editar seu conteúdo de forma *low-code*, já a subaba **Código** exibe o mesmo conteúdo no formato HTML e permite a edição em modo *high-code*. É possível exibir a subaba **Código** colocando a ide em Modo Avançado, ou habilitando a opção **Sempre mostrar código fonte**, disponível através do menu do sistema **Espaço de Trabalho** (1) > **Preferências**.

Para saber o trecho de código HTML de um componente visual, basta selecionar o componente na área de edição (3) e alterar da subaba **Visualização** para **Código** (2), o conteúdo estará selecionado (4).

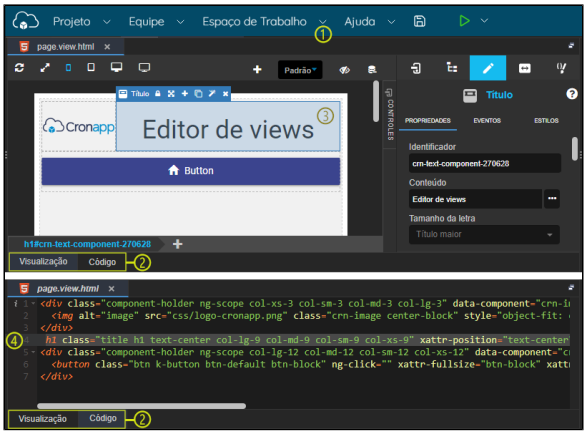

Figura 7 - Abas inferiores: Visualização e Código

A ação de um duplo clique sobre um componente visual pode variar a depender do componente selecionado. Em componentes que possuem configurações próprias, como Grade ou fonte de dados, a sua janela de configuração será exibida, já os demais componentes exibirão uma janela com o código HTML do componente selecionado (Figura 7.1), desde que a opção Modo Avançado esteja habilitada (destaque 1 da figura 7).

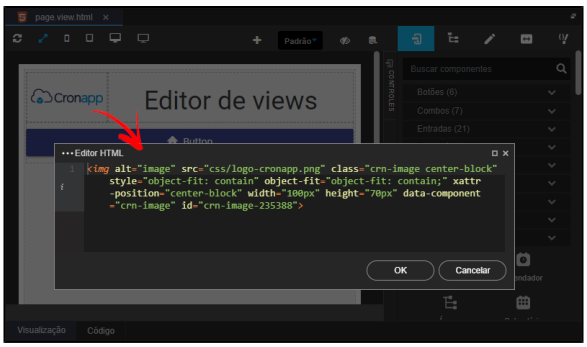

Figura 7.1 - Editor HTML do componente

Propriedades e Eventos da página

É possível associar propriedades e eventos diretamente a uma página. Isso pode ser feito acessando o menu de contexto dentro da área de edição da tela e selecionando a opção **Eventos da página** ou **Propriedades da página** (destaque 1 da figura 8). Por padrão, essas propriedades/eventos não existem quando criamos uma nova página, porém, ao acessar as propriedades ou eventos dessa página, um contêiner (`div#starter`) irá englobar todo o conteúdo da página. Além das propriedades e eventos contidos no contêiner, é possível inserir outros através da propriedade **Nova Propriedade** (2 da figura 8).

Para acessar a subaba **Código** é necessário acessar o editor de views no Modo Avançado, ou marcar a opção **Sempre mostrar código fonte**. acessível pelo menu do sistema **Espaço de Trabalho** > **Preferências**.

Para entender os demais elementos do menu de contexto (figura 8), acesse a documentação dos Componentes visuais.

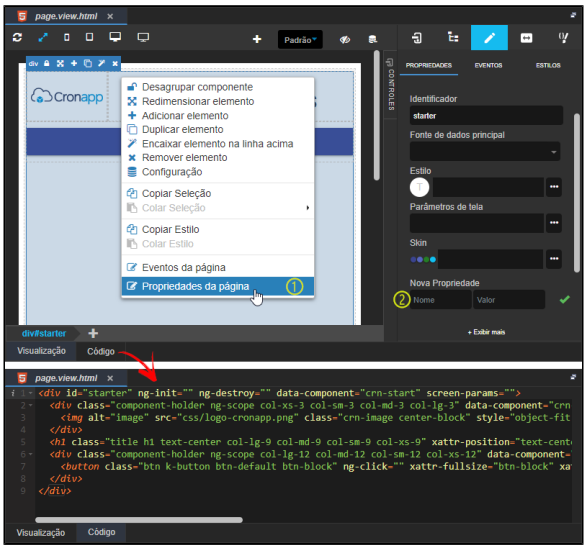

Figura 8 - Após acessar a propriedade da página, a div#starter será inserida para permitir configurações

Principais Propriedades

Nome	Propriedade	Função
Fonte de dados principal	primary-datasource	Define o comportamento geral da tela baseado nas ações da fonte de dados, dessa forma, em um formulário CRUD, se a fonte de dados mudar o status para edição, os componentes de exibição, como a grade, serão ocultados.
Exibir	ng-show	Diretiva Angular que exibe ou oculta a página.
Identificador	id	Identificador HTML da div principal, nesse caso sempre será `starter`.
Parâmetros da tela	screen-params	Definem parâmetros que serão passados a essa página via query string, dessa forma, ao configurarmos eventos que abram essa tela, serão exibidos esses parâmetros. Para obter um parâmetro, utilize a expressão: `params.<nome-da-queryString>`
Segurança	cronapp-security	Define quais grupos de acesso tem permissão para ver e editar.

Principais Eventos

Nome	Eventos	Função
Ao Iniciar	ng-init	Diretiva Angular que executa uma ação ao abrir a tela.
Ao Destruir	ng-destroy	Diretiva Angular que executa uma ação ao fechar a tela.

Acesse as documentações Propriedades dos componentes visuais e Eventos dos componentes visuais para mais detalhes.

Parâmetros da tela

Para exemplificar como ocorre a passagem de parâmetros entre as telas (query string), criamos uma página chamada **pessoa**, nela incluiremos dois parâmetros, sendo *nome* e *idade*. Para incluir os parâmetros, clique com o botão direito na view e acesse a opção **Propriedades da página**, como mostrado na figura 8, A seguir, acesse a propriedade **Parâmetros de tela**, clique no botão ao lado do campo (destaque 1 da figura 8.1) para exibir a janela de configurações dos parâmetros.

No campo **Nova propriedade** (2) informe o nome do parâmetro que deseja incluir na página, e em seguida clique em **Adicionar** (3), para este exemplo criamos os parâmetros *nome* e *idade*. A coluna **Pro priedade** (4) exibe o nome dos parâmetros inseridos na página e a coluna **Valor** (5) é dividido em duas caixas de seleção, a primeira informa o tipo do dado que será inserido e a segunda informa o conteúdo do parâmetro. Caso deseje excluir o parâmetro, basta clicar no ícone da lixeira (6). Para acessar os parâmetros da página, na própria página, é necessário utilizar `params.Nome_Do_Parâmetro`.

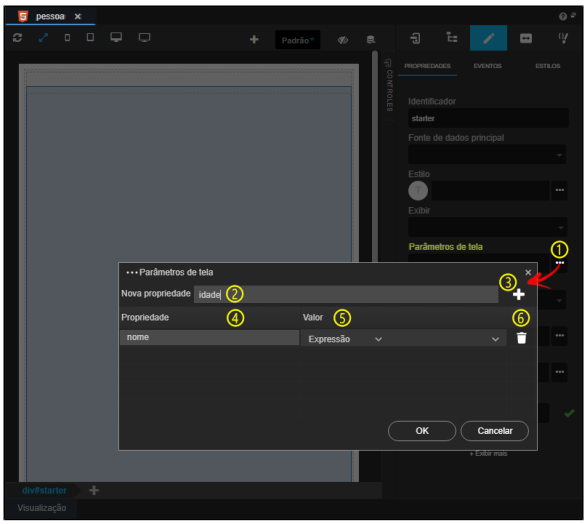

Figura 8.1 - Incluindo parâmetros de tela

Iremos exibir esses parâmetros no componente Título, inserimos dois componentes na view, um para cada parâmetro cadastrado. Acesse a propriedade **Conteúdo** (destaque 1 da figura 8.2), em seguida, a aba **Campos da Tela** (2) e clique na caixa de seleção **Campo** (3) para exibir e selecionar os parâmetros que inseridos anteriormente: *nome* e *idade.*

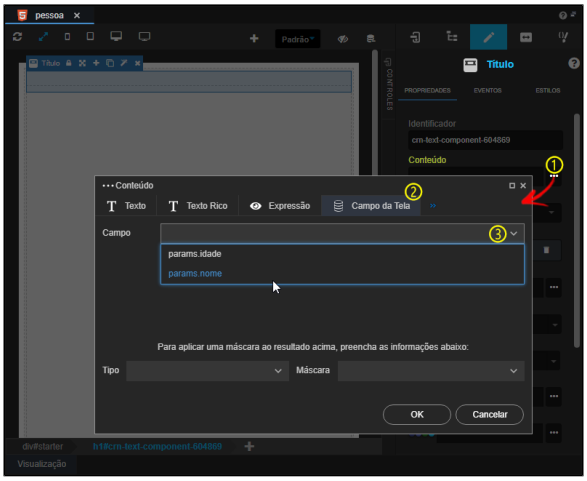

Figura 8.2 - Alimentando o componente Título a partir dos parâmetros de tela

Agora iremos enviar alguns dados da página **home** para preencher os parâmetros *nome* e *idade* da view **pessoa**. Para fazer isso nós incluímos um Botão, porém essa ação pode ser feita com qualquer outro componente ou Bloco de programação que chame outra página, o objetivo é somente transferir os dados de uma página para outra. Após abrir a janela do evento **Ao Clicar** (destaque 1 da figura 8.2),

acesse a aba **Formulário** (2) e selecione a view `Pessoa` no campo **Formulário** (3), ao final desta ação, os parâmetros da tela (*idade* e *pessoa*) serão exibidos, podemos agora incluir os valores de forma estática (exemplo da figura 8.3) ou dinâmico, selecionando algum campo da página home ou bloco de programação (Acesse o tópico "Aba Formulário" em Eventos dos componentes visuais para mais detalhes).

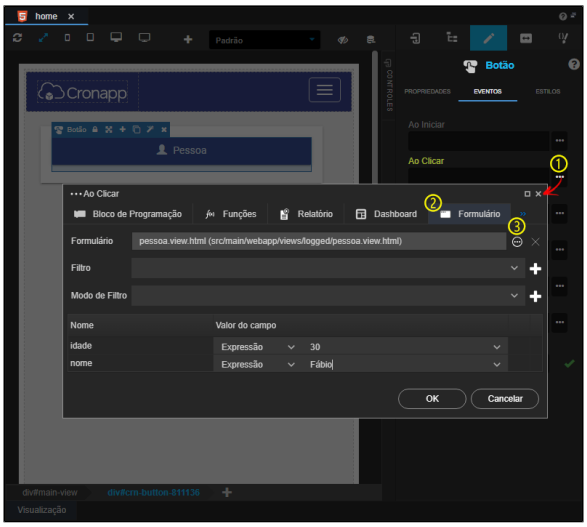

Figura 8.3 - Adicionando os valores para enviar para a view Pessoa

Na figura 8.4 é possível perceber a passagem de parâmetros da página **home** para a página **pessoa**

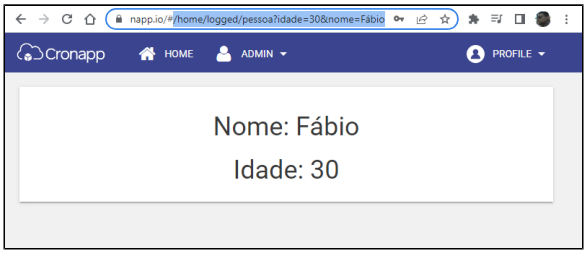

Figura 8.4 - Parâmetros idade e nome sendo preenchidos com o conteúdo da pessoa

Drag and Drop da árvore de arquivos

Para adicionar componentes em uma página, além da ação de arrastar e soltar a partir da aba Controles no menu lateral, também é possível obter componentes já configurados a partir da **árvore de recursos** ou **árvore de arquivos** (Modo Avançado desabilitado ou habilitado). Todo o conteúdo gerado a partir da árvore de recursos/arquivo também pode ser feito de forma *low-code*, a partir dos componentes visuais no Editor de views. Porém, o conteúdo gerado a partir da árvore permite agilizar ainda mais o processo de criação de páginas web e telas mobile.

Existem arquivos que não geram recursos no editor de views e arquivos que possuem várias opções. Por exemplo, após arrastar uma **página** (view) será exibido um menu (destaque 1 da figura 9) com opções para criar um Botão, com o evento **Ao clicar** (*ng-click*) já configurado, ou um componente Frame, que aponta para a página selecionada (figura 9.1). Já ao arrastar um arquivo `*.xml`, nenhuma opção será exibida, pois não existem ações configuradas para esse tipo de arquivo.

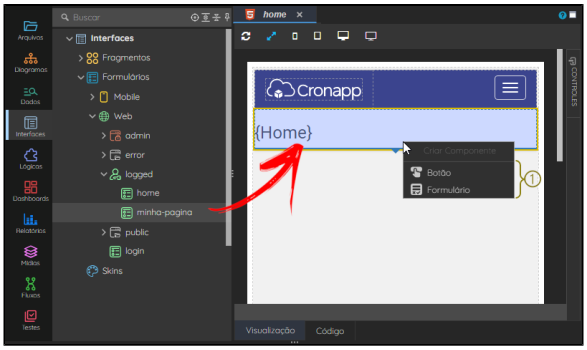

Figura 9 - Arrastando um item da árvore de recursos para o Editor de views

A opção selecionada no menu (destaque 1 da figura 9) pode abrir alguma janela de configuração do item ou exibir imediatamente seu resultado (figura 9.1).

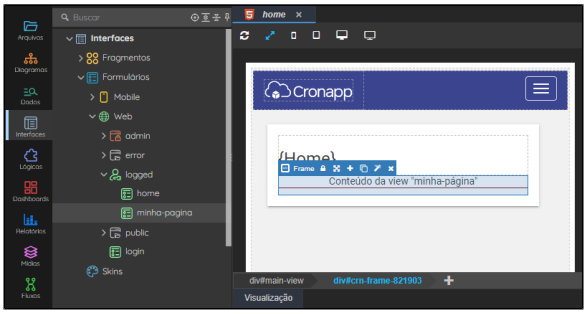

Figura 9.1 - Componente Frame gerado após selecionar a opção "Formulário" (figura 9)

Alguns componentes visuais na área de *design* podem exibir opções próprias ao arrastar um arquivo sobre ele. Por exemplo, ao arrastar uma **página** (view) até o componente Menu Dinâmico na tela, a opção "Formulário >" exibirá a estrutura configurada no **Menu Dinâmico**, permitindo adicionar o link da página selecionada em qualquer ponto na hierarquia do menu. Após selecionar a opção **(+)** (destaque 1 da figura 9.2), a janela de configurações do componente **Menu Dinâmico** será exibida com os principais campos já configurado. Esse mesmo procedimento ocorre no formulário mobile, porém, no mobile, o componente similar é o Barra de navegação (mobile).

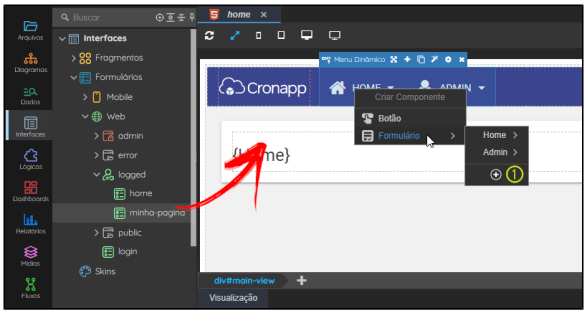

Figura 9.2 - Opção exibida ao arrastar uma página sobre o componente Menu Dinâmico

Os recursos arrastados durante a edição de uma página web devem pertencer ao projeto web. Assim como os recursos arrastados durante a edição de uma tela mobile, devem pertencer ao projeto mobile.

Não é esperado que recursos web funcionem em projetos mobile e vice-versa.

Arquivos aceitos no Editor web

A tabela abaixo apresenta a lista de arquivos que geram algum recurso após serem arrastado até o Editor de páginas web.

Arquivo	Onde soltar	Opções exibidas
Classe do Diagrama de dados	Corpo da página	• **Caixa de seleção dinâmica**: abre a janela de configuração do componente Caixa de seleção dinâmica já configurado com a Fonte de dados. • **CRUD**: abre a janela de configuração do Assistente de CRUD view. Após finalizar o assistente, um formulário com a entidade selecionada é gerado com Fonte de dados, Grade e campos da entidade. • **Grade**: abre a janela de configuração do componente Grade já configurado com a Fonte de dados.
Atributos das classes do Diagrama de dados	Corpo da página	• **Criar componente**: cria um componente visual baseado no tipo de dado do atributo. Esse recurso é útil para adicionar novos campos nas páginas CRUDs geradas a partir da classe do diagrama. Após arrastar até o formulário, o campo "Valor" (ng-model) é configurado esperando que exista um Componente visual fonte de dados com o mesmo nome da classe. ◦ **Texto**: cria o componente visual Entrada de texto com o tipo do campo configurado como Texto. ◦ **Texto Longo**: cria o componente visual Entrada de texto com o tipo do campo configurado como Texto. ◦ **UUID**: cria o componente visual Entrada de texto com o tipo do campo configurado como Texto. ◦ **Lógico**: cria o componente visual Caixa de checagem. ◦ **Caractere**: cria o componente visual Entrada de texto com o tipo do campo configurado como Texto. ◦ **Numérico**: cria o componente visual Entrada de texto com o tipo do campo configurado como Número. ◦ **Inteiro**: cria o componente visual Entrada de texto com o tipo de campo configurado como Inteiro. ◦ **Inteiro Longo**: cria o componente visual Entrada de texto com o tipo de campo configurado como Inteiro. ◦ **Decimal Grande**: cria o componente visual Entrada de texto com o tipo de campo configurado como Número. ◦ **Inteiro Grande**: cria o componente visual Entrada de texto com o tipo do campo configurado como Texto. ◦ **Inteiro Curto**: cria o componente visual Entrada de texto com o tipo do campo configurado como Texto. ◦ **Byte**: cria o componente visual Entrada de texto com o tipo do campo configurado como Número. ◦ **Data**: cria o componente visual Entrada de texto com o tipo do campo configurado como Data. ◦ **Data e Hora**: cria o componente visual Entrada de texto com o tipo do campo configurado como Data e Hora. ◦ **Carimbo de Data e Hora**: ria o componente visual Entrada de texto com o tipo do campo configurado como Data e Hora. ◦ **Hora**: cria o componente visual Entrada de texto com o tipo do campo configurado como Hora. ◦ **Binário**: cria o componente visual Entrada de texto com o tipo do campo configurado como Texto. ◦ **Arquivo no Banco**: cria o componente visual Envio de Arquivo Dinâmico. ◦ **Arquivo no Cloud**: cria o componente visual Envio de Arquivo Dinâmico. ◦ **Imagem no Banco**: cria o componente visual Imagem Dinâmica. ◦ **Imagem no Cloud**: cria o componente visual Imagem Dinâmica. ◦ **Versão**: cria o componente visual Entrada de texto com o tipo do campo configurado como Inteiro. ◦ **XML**: cria o componente visual Envio de Arquivo Dinâmico.
Fonte de dados	Corpo da página	• **Caixa de seleção dinâmica**: abre a janela de configuração do componente Caixa de seleção dinâmica já configurado com a Fonte de dados. • **CRUD**: abre a janela de configuração do Assistente de CRUD view. Após finalizar o assistente, um formulário com a entidade selecionada é gerado com Fonte de dados, Grade e campos da entidade.

		• **Grade**: abre a janela de configuração do componente Grade já configurado com a Fonte de dados.
Fragmento	Corpo da página	• **Fragmento**: adiciona o componente Fragmento e exibe o conteúdo selecionado.
View (formulário web)	Corpo da página	• **Botão**: abre a janela de configuração do evento **Ao Clicar** (ng-click). • **Formulário**: adiciona o componente Frame (web) e exibe o conteúdo selecionado.
View (formulário web)	Componente Menu Dinâmico	• **Formulário**: abre a janela de configuração do Menu Dinâmico com os principais campos já configurados.
Bloco de programação (cliente ou servidor)	Corpo da página	• **Botão**: abre a janela de configuração do evento **Ao Clicar** (ng-click) já com a primeira função selecionada.
Relatório	Corpo da página	• **Botão**: abre a janela de configuração do evento **Ao Clicar** (ng-click) já configurado para executar a função Abrir Relatório. • **Relatório**: abre a janela de configuração do componente Visualizador de relatório.
Relatório	Componente Menu Dinâmico	• **Formulário**: abre a janela de configuração do Menu Dinâmico com os principais campos já configurados.

Arquivos aceitos no Editor mobile

A tabela abaixo apresenta a lista de arquivos que geram algum recurso ser arrastado até o Editor de telas mobile.

Arquivo	Onde soltar	Opções exibidas
Classe do Diagrama de dados	Corpo da página	• **Caixa de seleção dinâmica**: abre a janela de configuração do componente Caixa de seleção dinâmica já configurado com a Fonte de dados. • **CRUD**: abre a janela de configuração do Assistente de CRUD view. Após finalizar o assistente, um formulário com a entidade selecionada é gerado com Fonte de dados, Lista Avançada e campos da entidade. • **Lista Avançada**: abre a janela de configuração do componente Lista Avançada já configurado com a Fonte de dados.

Atributos das classes do Diagrama de dados	Corpo da página	• **Criar componente**: cria um componente visual baseado no tipo de dado do atributo. Esse recurso é útil para adicionar novos campos nas páginas CRUDs geradas a partir da classe do diagrama. Após arrastar até o formulário, o campo "Valor" (ng-model) é configurado esperando que exista um Componente visual fonte de dados com o mesmo nome da classe.
		  ◦ **Texto**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Texto.
		  ◦ **Texto Longo**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Texto.
		  ◦ **UUID**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Texto.
		  ◦ **Lógico**: cria o componente visual Caixa de checagem.
		  ◦ **Caractere**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Texto

		◦ **Numérico**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Número. ◦ **Inteiro**: cria o componente visual Entrada de texto flutuante com o tipo de campo configurado como Inteiro. ◦ **Inteiro Longo**: cria o componente visual Entrada de texto flutuante com o tipo de campo configurado como Número. ◦ **Decimal Grande**: cria o componente visual Entrada de texto flutuante com o tipo de campo configurado como Número. ◦ **Inteiro Grande**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Texto. ◦ **Inteiro Curto**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Texto. ◦ **Byte**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Inteiro. ◦ **Data**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Data. ◦ **Data e Hora**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Data e Hora. ◦ **Carimbo de Data e Hora**: ria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Data e Hora. ◦ **Hora**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Hora. ◦ **Binário**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como configurado como Texto. ◦ **Arquivo no Banco**: cria o componente visual Envio de Arquivo Dinâmico. ◦ **Arquivo no Cloud**: cria o componente visual Envio de Arquivo Dinâmico. ◦ **Imagem no Banco**: cria o componente visual Imagem Dinâmica. ◦ **Imagem no Cloud**: cria o componente visual Imagem Dinâmica. ◦ **Versão**: cria o componente visual Entrada de texto flutuante com o tipo do campo configurado como Número. ◦ **XML**: cria o componente visual Envio de Arquivo Dinâmico.
Fonte de dados	Corpo da página	• **Caixa de seleção dinâmica**: abre a janela de configuração do componente Caixa de seleção dinâmica já configurado com a Fonte de dados. • **CRUD**: abre a janela de configuração do Assistente de CRUD view. Após finalizar o assistente, um formulário com a entidade selecionada é gerado com Fonte de dados, Lista Avançada e campos da entidade. • **Lista Avançada**: abre a janela de configuração do componente Lista Avançada já configurado com a Fonte de dados.
Fragmento	Corpo da página	• **Fragmento**: adiciona o componente Fragmento e exibe o conteúdo selecionado.
View (formulário mobile)	Corpo da página	• **Botão**: abre a janela de configuração do evento **Ao Clicar** (ng-click). • **Formulário**: adiciona o componente Frame (web) e exibe o conteúdo selecionado.
View (formulário mobile)	Componente Barra de navegação	• **Formulário**: abre a janela de configuração do Barra de navegação com os principais campos já configurados.
Bloco de programação (cliente ou servidor)	Corpo da página	• **Botão**: abre a janela de configuração do evento **Ao Clicar** (ng-click) já com a primeira função selecionada.