

E-Book Módulo 1 - Potencializando o Desenvolvimento de Telas com o Uso de Widgets

Desvendando o Front-end em Low-Code

Mentor Andrew Costa

#ecoapucrio

-  Introdução
-  Conceitos Fundamentais para Desenvolver Interfaces em Low-Code
-  Customizando Widgets
-  Desenvolvendo com DropDown
-  Radio Group
-  ButtonGroup
-  Gráficos (Charts)
-  Animações



Encerramento

Introdução

Introdução

Low-Code

O foco desse módulo é apresentar recursos que contribuam na criação de **telas ou páginas visuais** cada vez mais interessantes em softwares desenvolvidos usando a abordagem **Low-code**.

Essa abordagem permite a criação de soluções usando pouco ou nenhum código e tem sido adotada por diversas empresas no mundo.

Para entender de forma prática como criar telas em Low-Code usando diferentes recursos, será usada a plataforma Service Studio oferecida pela **OutSystems**, uma das líderes de mercado neste segmento.



O que são Widgets e como customizá-los

Entrando em detalhes, no módulo, você aprenderá **o que são widgets e como customizá-los.**

Você verá que widgets são componentes visuais que a plataforma da OutSystems oferece, e que a partir desses componentes, telas capazes de oferecer boas experiências aos usuários poderão ser construídas.

Alguns dos widgets vistos neste módulo são os seguintes: **Dropdown, Button Group e Radio Group.**

Criar gráficos e animações

Além desses widgets, você aprenderá a criar gráficos e incluir animações em suas soluções.

Dessa forma, experiências ainda mais impactantes poderão ser desenvolvidas e oferecidas aos seus usuários.

Ao final do módulo, você conseguirá usar e customizar diversos Widgets para desenvolver telas em suas **soluções Low-Code**.

Soluções Low-Code

Vamos então iniciar essa jornada de aprendizado!

Conceitos Fundamentais para Desenvolver Interfaces em Low-Code

Conceitos Fundamentais para Desenvolver Interfaces em Low-Code

Conheça orientações e explicações de conceitos fundamentais que irão lhe ajudar a desenvolver interfaces usando a abordagem Low-Code a partir da plataforma usada nas aulas.



Para desenvolver interfaces para algum sistema Low-Code, a trilha usa a plataforma **Service Studio** oferecida pela OutSystems, que é uma das líderes de mercado quando o assunto é Low-code.

Caso não tenha instalado o **Service Studio** em sua máquina, por favor, procure baixá-lo a partir do link: https://www.outsystems.com/Portal/Trial_Portal. Essa plataforma permite criar soluções Low-Code usando recursos que são ensinados nas diferentes aulas oferecidas na trilha.

Abaixo, seguem termos e recursos importantes usados no **Service Studio**. Vamos a eles.

- ***Project:***

Para criar alguma solução Low-Code, deve ser criado algum projeto na plataforma. Esse projeto poderá ser desenvolvido, por exemplo, (i) para Web, ou seja, poderá ser usado tanto em computadores, como smartphones e tablets, ou (ii) para algum tipo de dispositivo móvel, como smartphone e/ou tablet.

- ***Screen:***

Considerando que um projeto já foi criado, há um componente visual chamado **Screen** que permite criar uma nova tela ou página para a solução que estará sendo desenvolvida. Segunda a OutSystems,

Screen é um elemento de interface que contém outros elementos visuais e que poderão ser usados pelos usuários.

- **Widgets:**
Elementos visuais que ajudam o desenvolvedor a projetar e organizar a interface do usuário da sua solução Low-Code.
- **Variável Local (Local Variable):**
Existe somente no escopo do seu elemento pai, como por exemplo, uma **Screen** ou uma **Action**. Em uma Variável Local podem ser atribuídas informações que serão usadas localmente dentro nesse escopo. Exemplos de informações que podem ser atribuídas são números, textos etc.
- **Action:**
Permite definir um conjunto de passos que deverão ser executados em algum contexto considerado, como por exemplo, serem executados quando um botão for clicado por algum usuário, ou quem sabe quando uma nova tela for apresentada.

Com isso você conheceu os principais fundamentos para iniciar sua jornada de desenvolvimento de interfaces usando a abordagem Low-Code.

Customizando Widgets

Customizando Widgets

A plataforma do Service Studio permite que diversas customizações visuais possam ser realizadas em **widgets**. Para cada widget há duas abas que permitem que essas customizações aconteçam: **Properties e Styles**. De widget para widget as informações presentes em cada aba muda, devido o tipo de elemento visual considerado. A seguir, há uma breve descrição de cada aba.

Properties:

Nesta aba são apresentadas diversas propriedades que ao serem fornecidas pelo desenvolvedor impactam o widget. No componente visual **Button**, por exemplo, há propriedades como, **Enabled**, que indica que está habilitado ou não para ser clicado, assim como **Visible**, que indica se o botão deve estar ou não visível para o usuário.

Além dessas propriedades há outras, que permitem indicar o que deve acontecer quando o botão for clicado (p.ex **On Click**).

Na imagem são ilustradas quais propriedades compõe um widget **Button** na aba **Properties**.

OK **Button** **?**

Properties

Styles

Name

Confirmation Mes...

Enabled

Is Form Default

Visible

Style Classes

Attributes

Property

= Value

Events

On Click

Event

Handler

OK **Button** **?**

Properties

Styles

Style Classes

btn

Font

Style Size **100%** Color

Layout

Width **79.2891** Height **40px** Align Color

Margin

0px 0px
(Auto 16px Padding 16px 0px
0px
0px

Borders

Lines

1px
1px 1px
1px

Corners

4px 4px
4px 4px
4px 4px

Styles

Aba responsável por aplicar estilos visuais no Widget selecionado.

Estilos tem relação com cores, largura, altura do componente, tipografia, espaçamento em relação a outros elementos, além da borda a ser aplicada no widget.

Na imagem é apresentada a aba **Styles** de um widget **Button**.

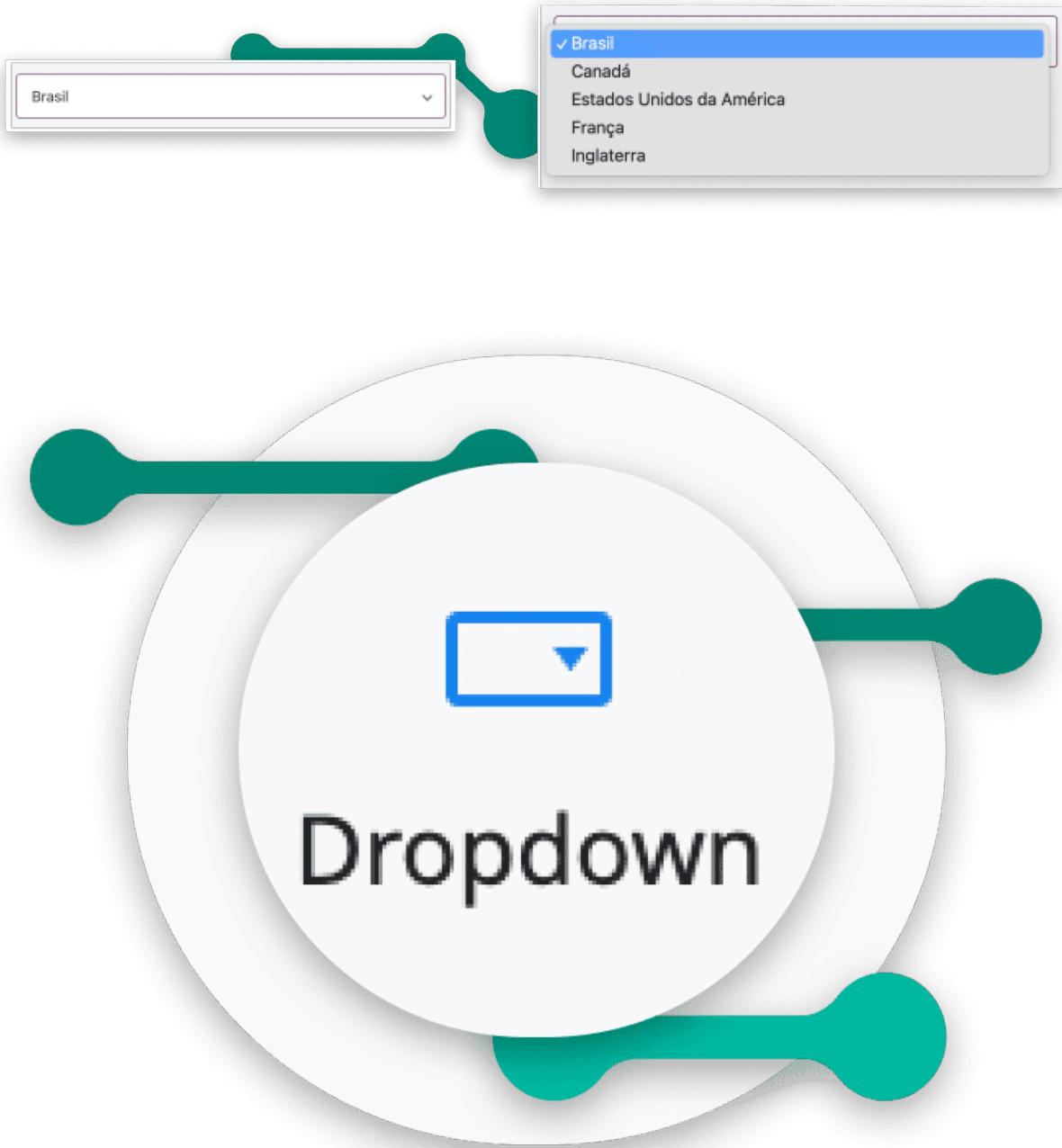
Com isso, você entendeu como certas customizações podem ser realizadas em Widgets oferecidos. **Use esse conhecimento para desenvolver boas experiências aos usuários.**

Desenvolvendo com DropDown

Conhecendo o Widget DropDown

Dropdown é um widget oferecido pela OutSystems que permite com que usuários façam a escolha de uma opção dentre várias oferecidas.

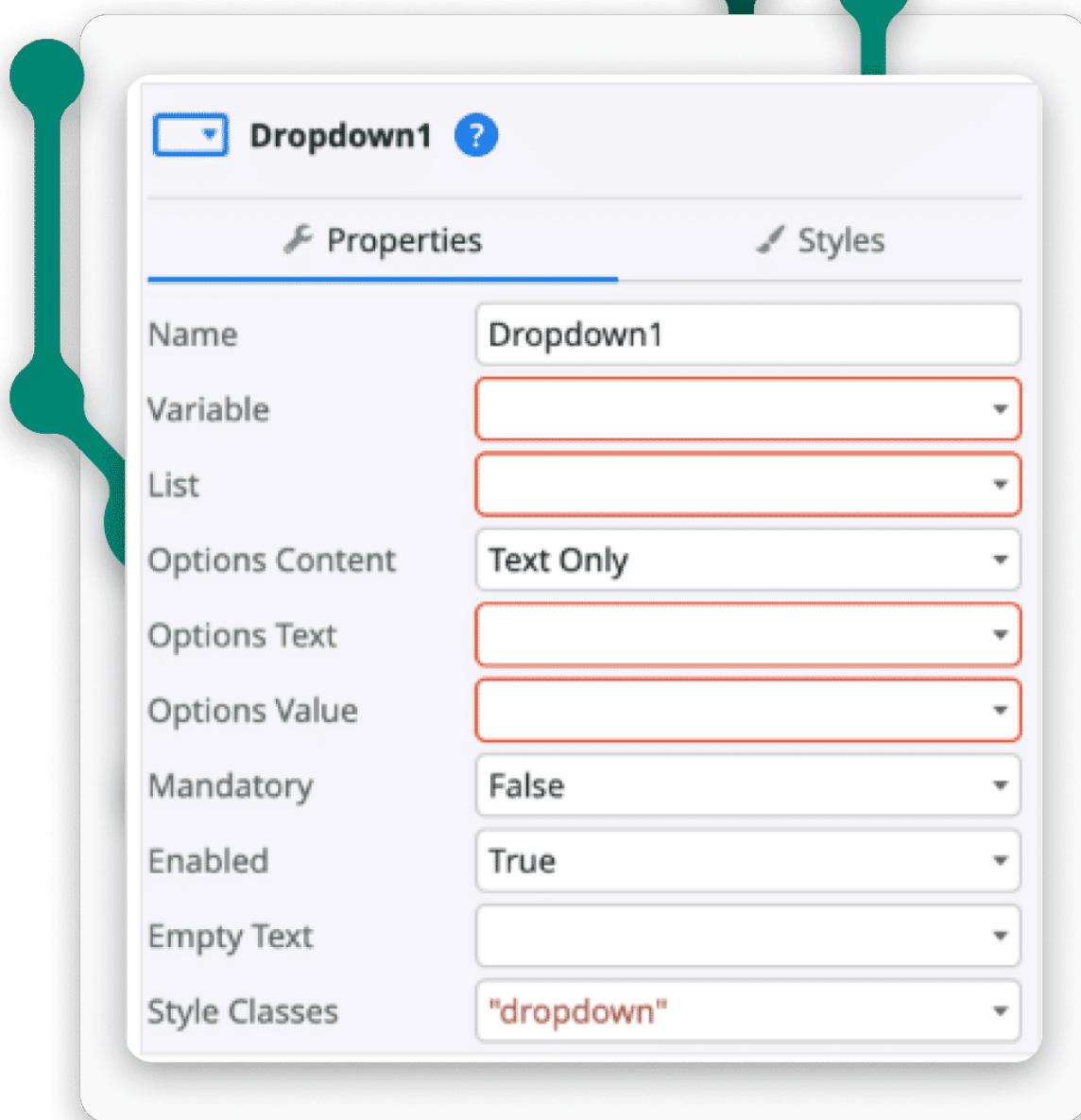
Perceba na imagem abaixo, um exemplo de uso do **Dropdown**. O usuário ao clicar no componente (lado esquerdo), opções são apresentadas (lado direito). No exemplo, são listados diferentes países.



A recomendação de uso é quando houver a necessidade de uma escolha dentre mais de quatro opções.

Essas opções podem ser de diferentes tipos, como por exemplo, textos ou imagens. O widget aparece disponível na plataforma do **Service Studio**, assim como ilustrado na imagem.

Ao arrastar e soltar em alguma **Screen**, o desenvolvedor deve preencher ao menos as quatro propriedades indicadas na imagem e explicadas em mais detalhe abaixo.



● **Variable:**

Variável que irá armazenar a informação da escolha feita no

Dropdown.

- ***List***

Fonte que fornece a lista de informações que será apresentada no widget. Para essa escolha, pode ser usada, por exemplo, uma entidade de banco de dados.

- ***OptionsText***

Textos que serão apresentados no componente visual, caso a propriedade ***Options Content*** esteja com ***Text Only***.

- ***OptionsValue***

Valor que será armazenado na variável indicada mais acima, quando uma opção for escolhida pelo usuário no widget.

Esse elemento visual pode ser usado em diversas situações, como por exemplo, formulários, questionários, ou qualquer outra situação que solicite a escolha de uma opção dentre várias oferecidas.

A partir de agora, **use Dropdown em suas soluções Low-Code conforme precisar.**

Radio Group



Radio Group é um widget que permite agrupar **Radio Buttons**, que representam opções que o usuário pode escolher, como uma questão de múltipla escolha.

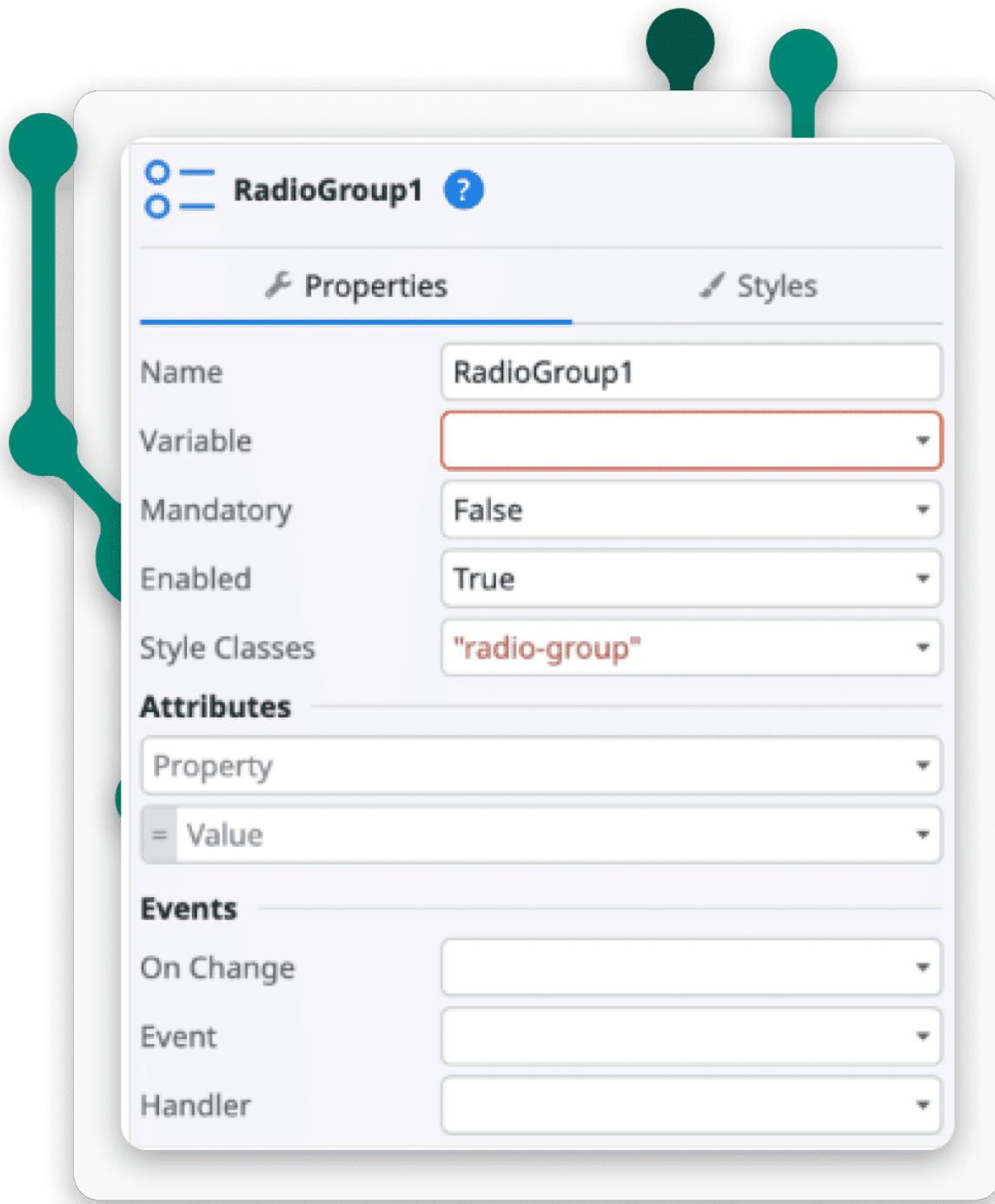
Veja na imagem a ideia desse elemento visual. Perceba que nela o usuário tem duas opções: **Sim ou**

Não

O widget aparece disponível na plataforma do **Service Studio** podendo ser adicionado em telas de qualquer solução Low-Code. Para usá-lo, vale destacar duas ações importantes que devem ser feitas.

Ação 1

Ao arrastar e soltar em alguma *Screen*, o desenvolvedor deve informar qual a variável (propriedade **Variable**) vinculada ao widget **RadioGroup** que irá guardar a resposta escolhida pelo usuário.



Ação 2

Para cada **RadioButton** que compõe o **RadioGroup** deve ser informado o valor (propriedade **Value**) que deve ser fornecido e armazenado na variável informada anteriormente do **RadioGroup**.



RadioButton1 ?

Properties

Styles

Name	RadioButton1
Value	
Enabled	True
Visible	True
Style Classes	"radio-button"

Attributes

Property
= Value

Events

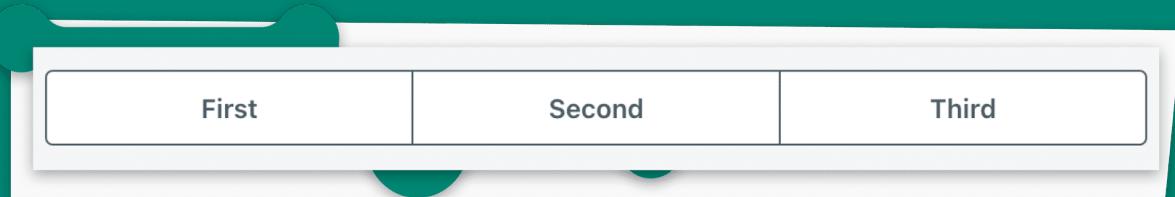
Event
Handler

ButtonGroup

ButtonGroup

ButtonGroup é um *widget* que permite oferecer opções de botões como escolha aos usuários.

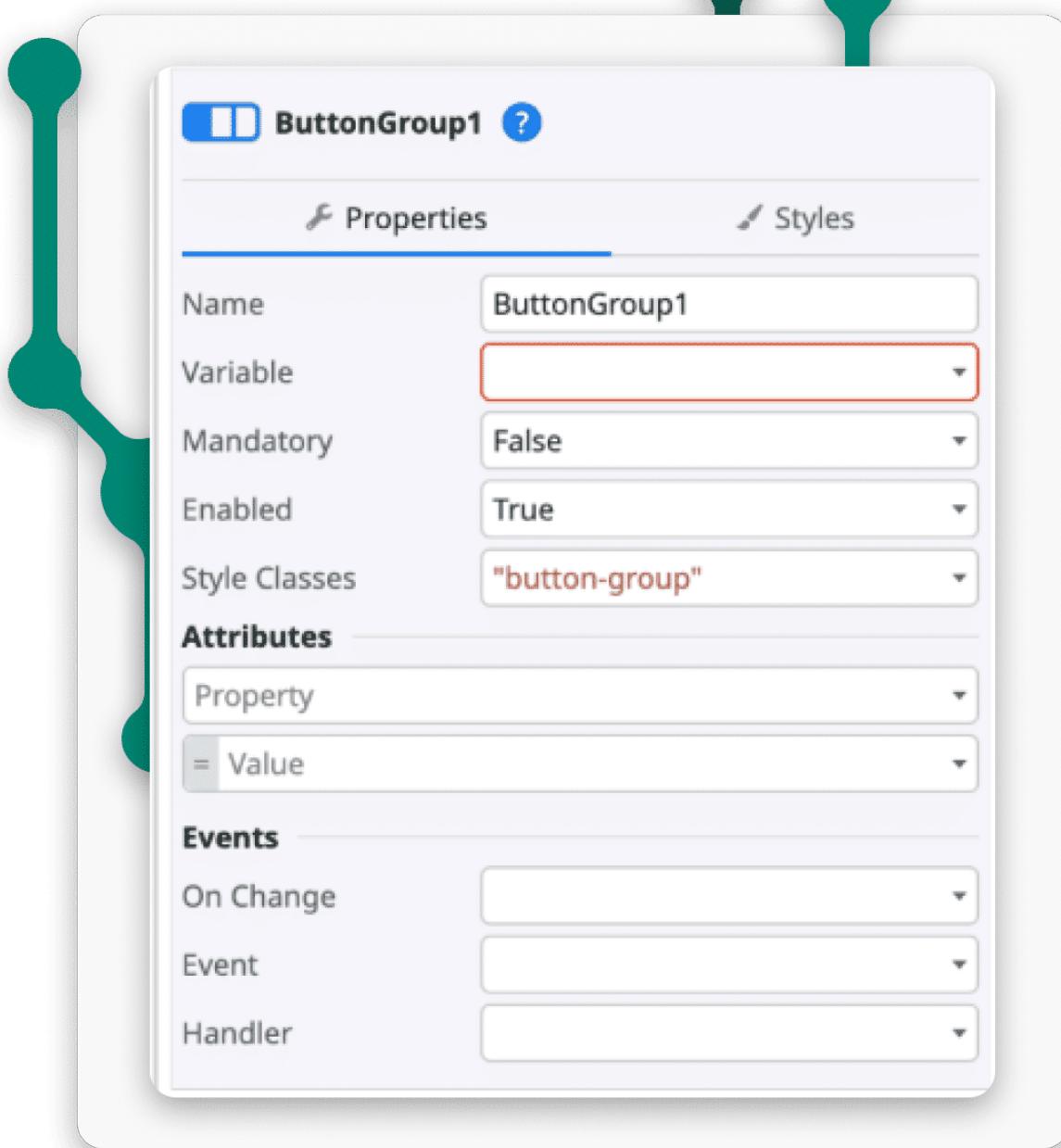
Veja na imagem como esse componente visual oferece botões agrupados. No exemplo, há três botões em que o usuário poderá escolher.



O widget aparece disponível na plataforma do **Service Studio** para ser adicionado em alguma tela, assim como apresentado na imagem.



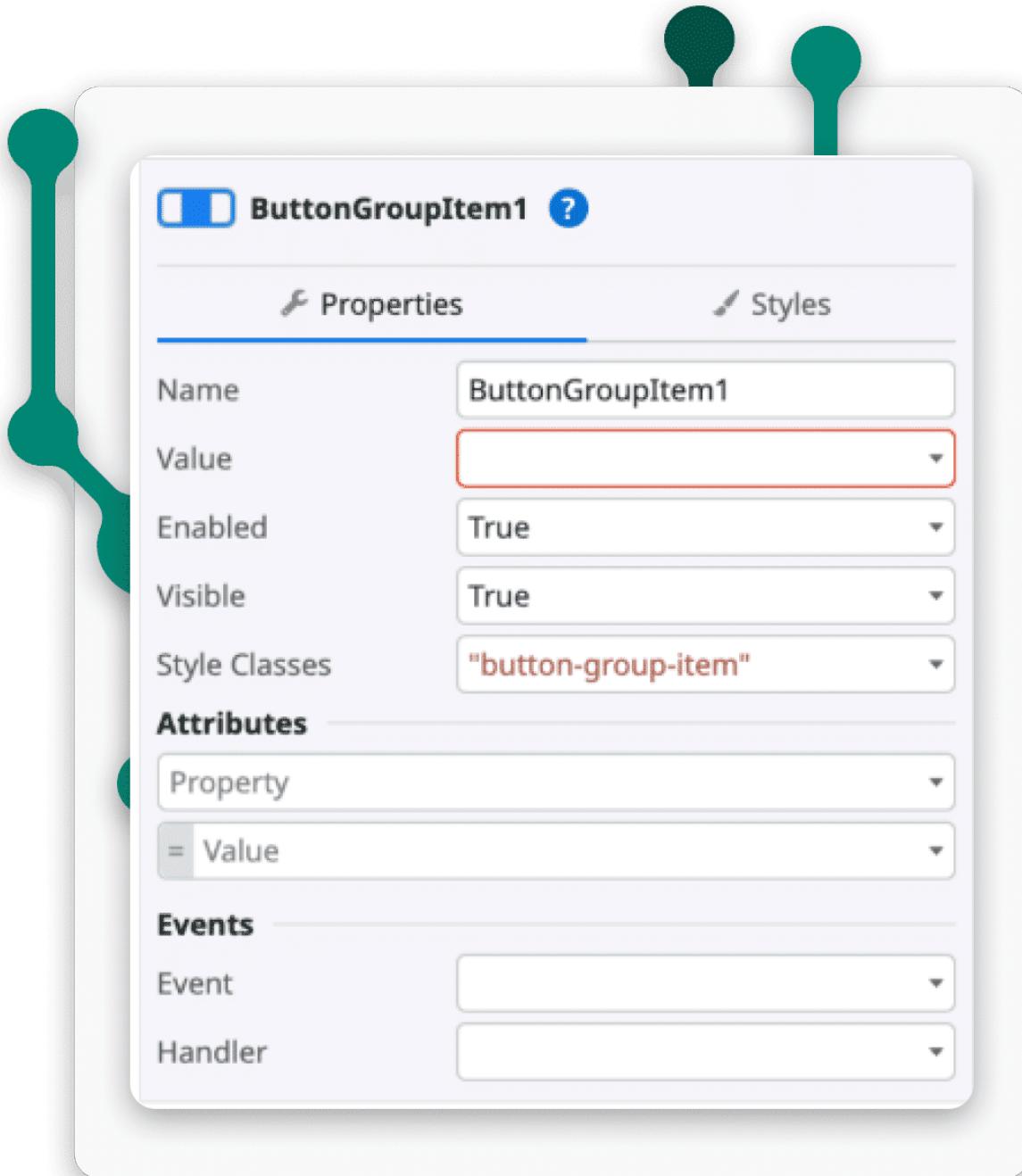
**Button
Group**



Ao arrastar e soltar em alguma **Screen**, o desenvolvedor deve informar qual a variável (propriedade **Variable**) vinculada ao widget **ButtonGroup**, que irá guardar a escolha feita pelo usuário dentre os botões oferecidos.

Além disso, para cada **ButtonGroupItem** que compõe o **ButtonGroup**, deve ser informado o valor (propriedade **Value**) que deve ser fornecido e armazenado na variável informada anteriormente do **ButtonGroup**.

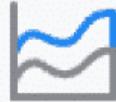
Perceba que cada botão que compõe esse grupo de botões é um **ButtonGroupItem**.



Gráficos (Charts)

Gráficos (Charts)

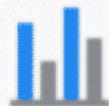
Um tipo de recurso comum que está presente em diversos aplicativos, assim como sistemas web é o uso de gráficos, que ajudam a oferecer visões para análise de informações.



Area Chart



Bar Chart



Column
Chart



Donut
Chart



Line Chart



Pie Chart

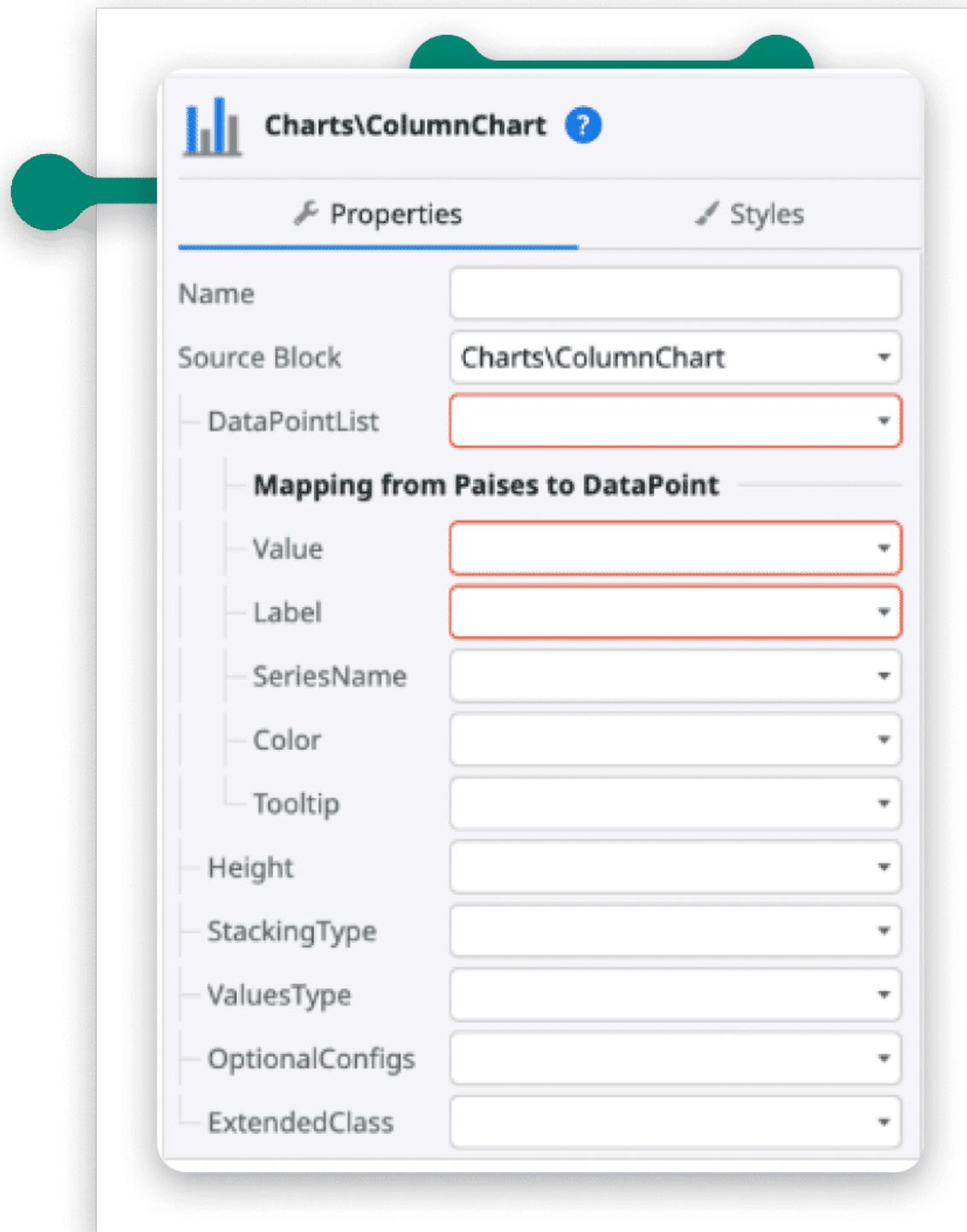


Radar
Chart

A fim de facilitar o desenvolvimento de soluções que permitam a inclusão de diferentes tipos de gráficos, o **Service Studio** oferece um conjunto de widgets para esse tipo de situação. Na plataforma, há as seguintes opções: **Area Chart**, **Bar Chart**, **Column Chart**, **Donut Chart**, **Line Chart**, **Pie Chart** e **Radar Chart**.

Vale destacar que cada opção irá permitir a criação de diferentes tipos de gráficos com diferenças em suas respectivas configurações.

Ao lado, há a indicação visual de quais informações devem ser fornecidas pelo desenvolvedor para configurar minimamente um gráfico do tipo Coluna (**Column Chart**).



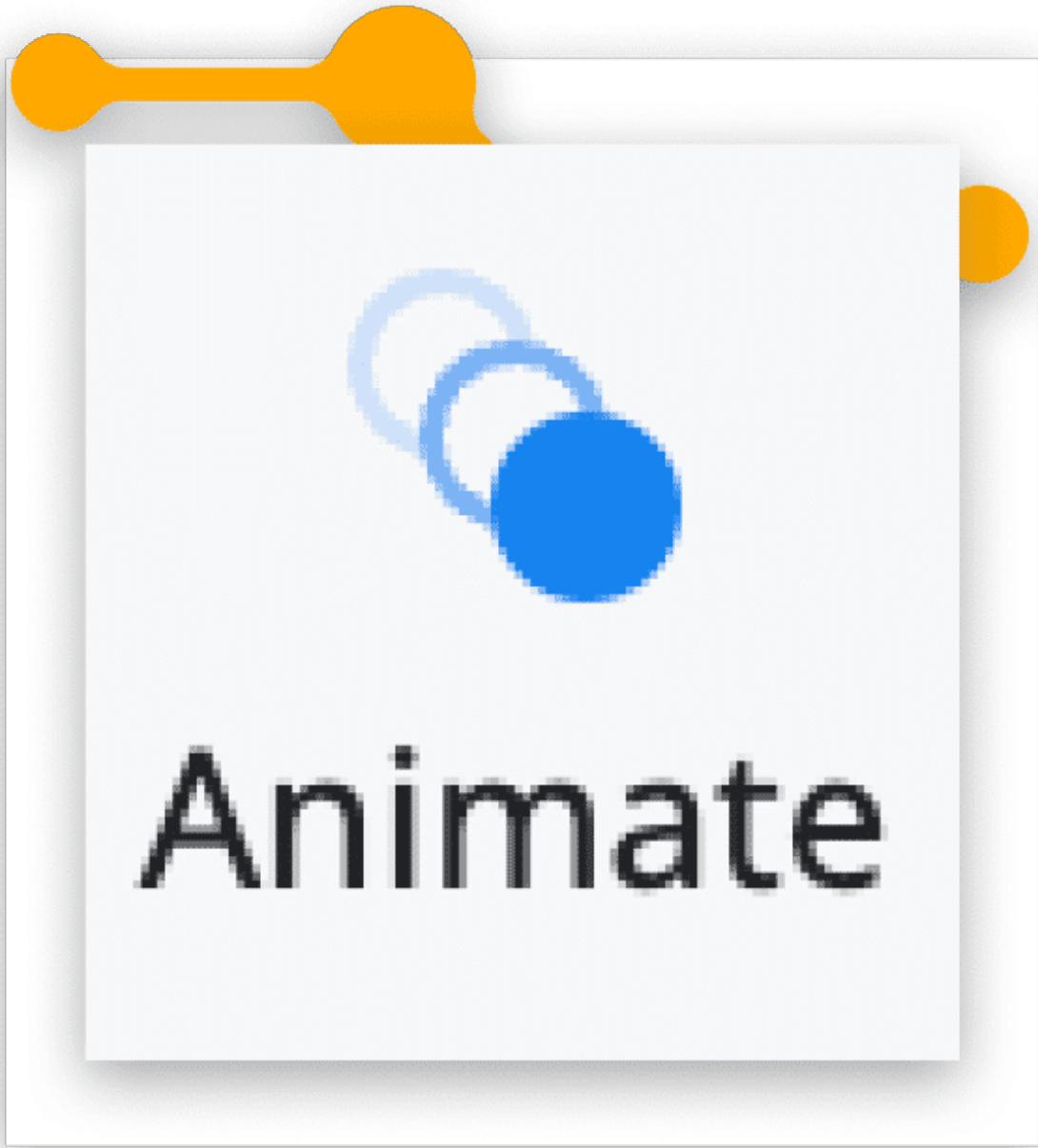
Abaixo, há a explicação de cada propriedade indicada em vermelho na imagem acima.

- ***DataPointList***
Lista com as informações que serão consideradas para montar o gráfico. Uma possibilidade é usar um **Aggregate**. Se seu **Aggregate** tiver o nome `Itens`, bastaria colocar nesse campo ***Itens.List***.
- ***Value***
Valor que deve ser considerado para montar cada coluna do gráfico. Espera-se que sejam dados do tipo numérico.
- ***Label***
Informação que será usada para identificar o que é cada coluna.

Animações

Animações

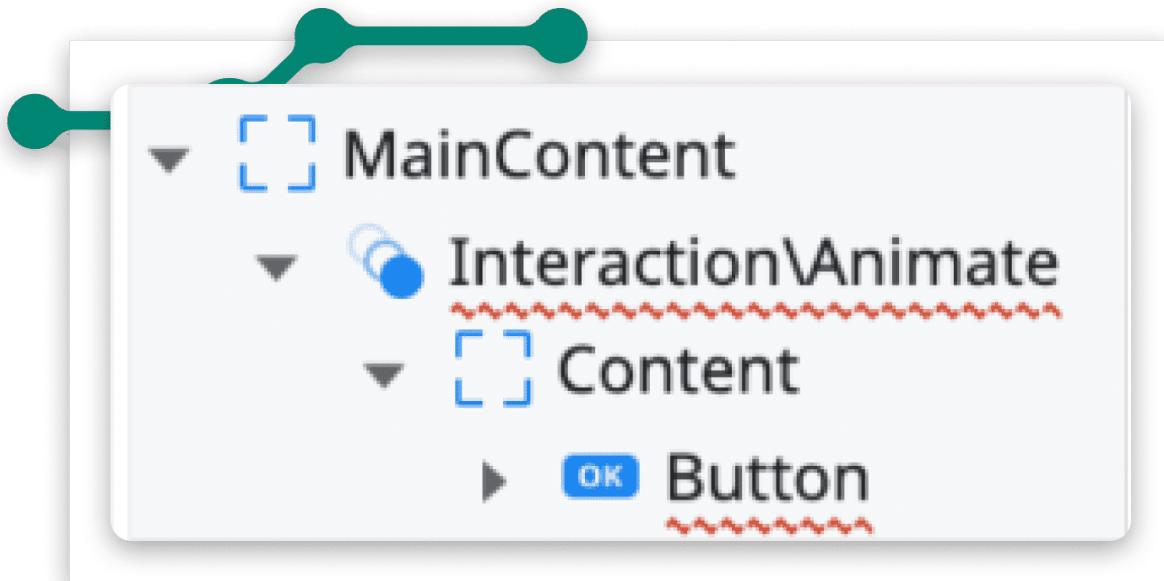
A fim de oferecer experiências ainda mais impactantes para os usuários, a plataforma da OutSystems permite que diversas animações possam ser inseridas. Para isso, são oferecidos dois recursos: **Animate** e **Animated Label**.



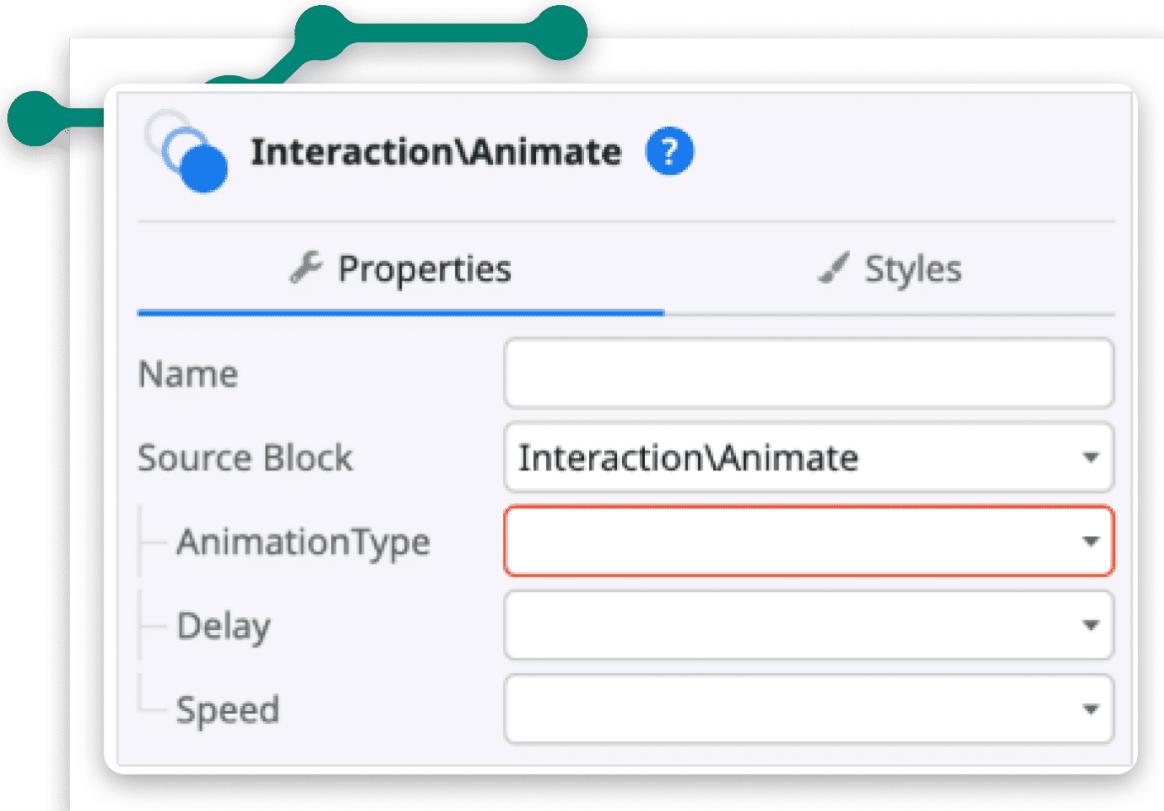
Animate oferece diferentes animações que podem ser aplicadas em elementos de interface.

Alguns exemplos de animações são os seguintes: aparecer, quicar, mover da esquerda para direita, mover da direita para esquerda, dentre outros tipos.

Caso deseje, por exemplo, aplicar uma animação em um botão, tal elemento deve estar dentro do recurso **Animate** incluído na tela, assim como apresentado ao lado.



Na sequência, devem ser preenchidos os campos indicados abaixo no Animate incluído.



- ***AnimationType***

Tipo de animação que irá acontecer no elemento da interface desejado.

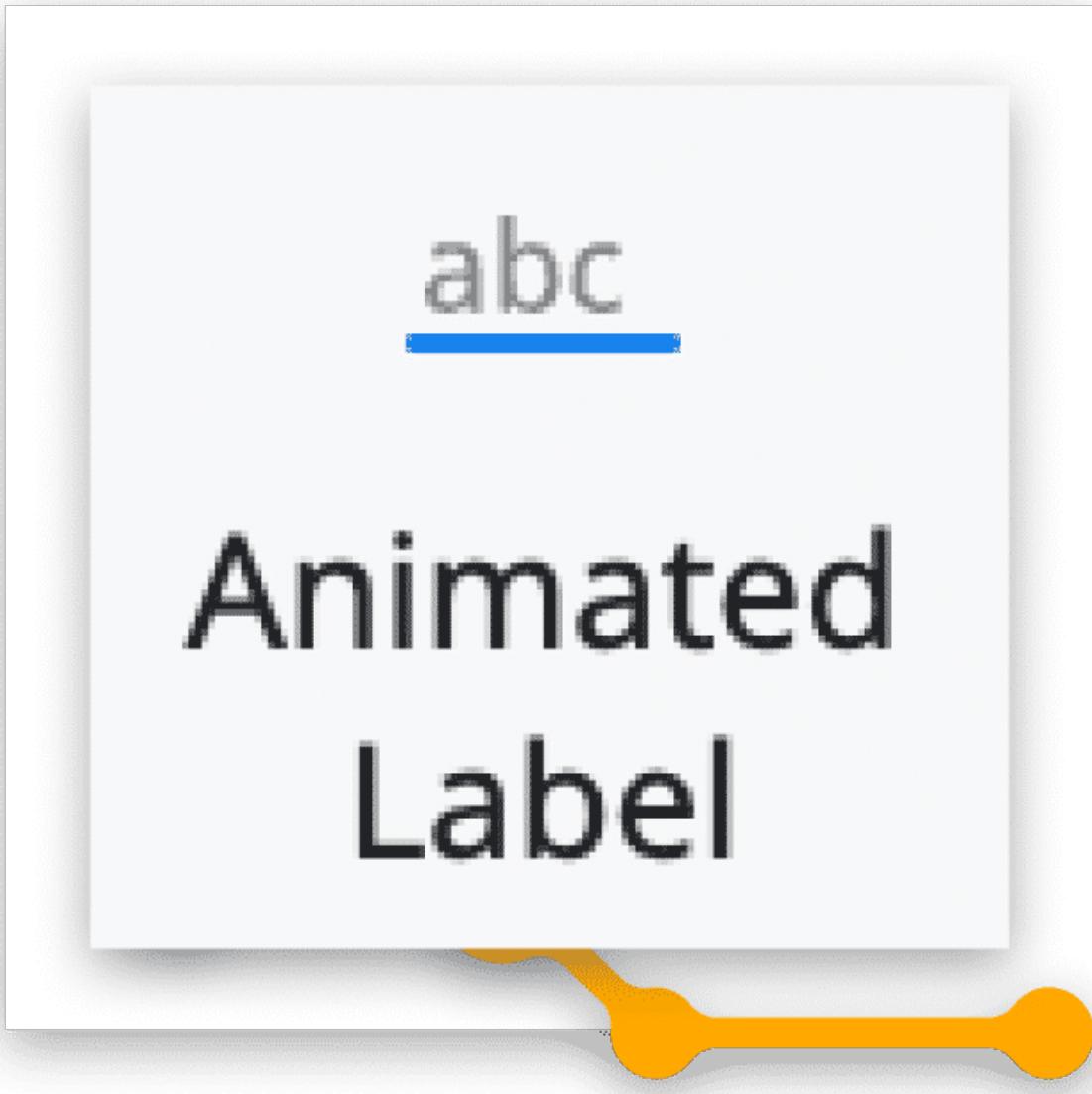
- ***Delay***

Tempo em milissegundos para iniciar a animação.

- ***Speed***

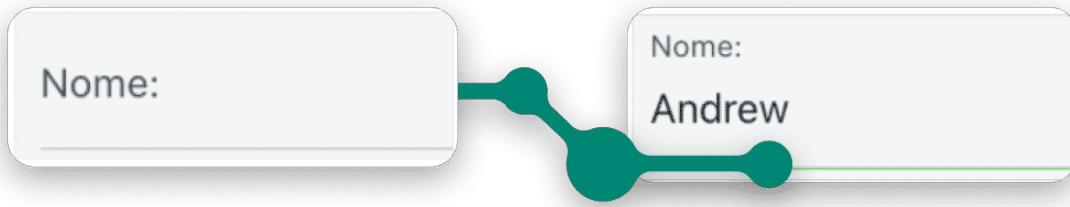
Velocidade de duração da animação. A plataforma oferecer três opções: Fast (Rápida), Normal e Slow (Devagar).

Vamos agora conhecer o Animated Label?



Animated Label permite que seja oferecida uma animação considerando um label e uma caixa de texto oferecida. Conforme o usuário selecione o campo e inicie a escrever, a caixa de texto se destaca visualmente, oferecendo uma boa experiência de uso.

Veja a seguir como o recurso animado funciona em um campo **Nome** sem ter qualquer texto fornecido, e na sequência tendo um nome escrito.

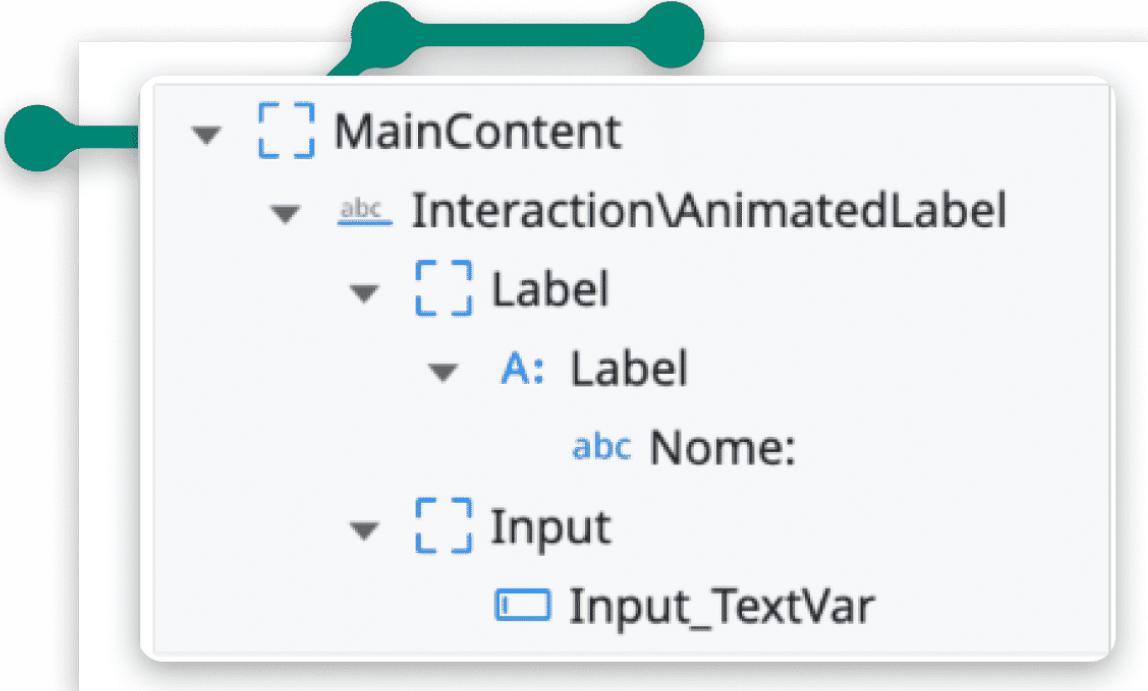


Sem texto fornecido

Com texto fornecido

Portanto, quando um **Animate Label** é adicionado em uma tela, ele é composto por dois conteúdos:

Label e Input. Lembre-se que para a caixa de texto (**Input**), uma variável deve ser indicada para armazenar o texto fornecido pelo usuário.



Encerramento

Clique na seta abaixo para fazer o download deste E-book.



File Attachment Block

No file added





PARABÉNS

Você completou este Módulo