



**Aprenda  
App Inventor**

# APP AGENDA

Aplicativo que permite que o usuário armazene nome e telefone de vários amigos, exibindo na tela os dados já cadastrados

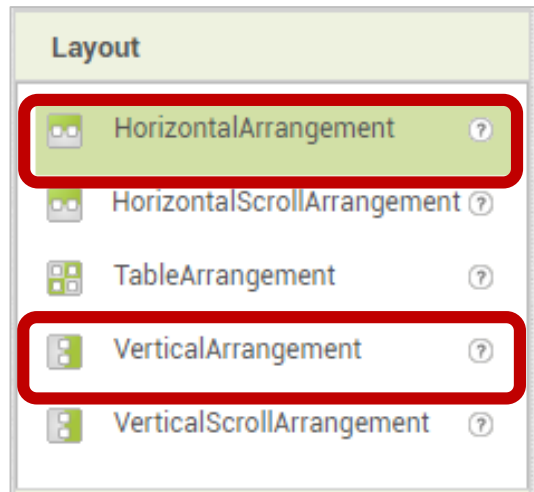
## ➤ O APP AGENDA IRÁ CONTER:

- ✓ 1 componente vertical
- ✓ 4 componentes horizontais
- ✓ 2 caixas de texto para receber os dados
- ✓ 2 labels para indicar o que deve ser digitado
- ✓ 1 botão para cadastrar dados
- ✓ 1 componente ListView para exibir os dados
- ✓ 1 componente TinyDB para armazenamento

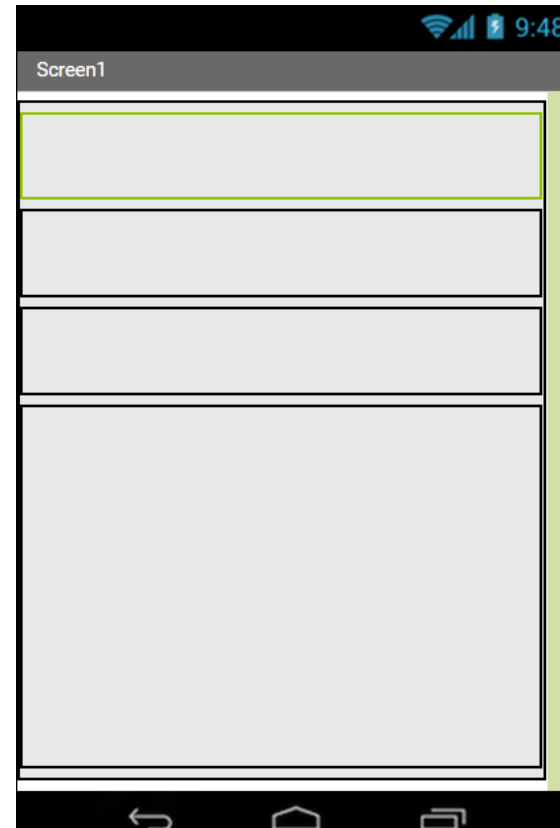


# MONTANDO A INTERFACE

- É necessário usar os componentes horizontais e verticais ([Aba Layout](#)) para exibir os itens de forma organizada



*Adicione 1 layout vertical, e dentro dele 4 layouts horizontais.*



# AJUSTANDO LAYOUTS

- É preciso ajustar as propriedades de cada layout para obter a aparência desejada
  - Para o layout vertical, atribua Height (altura) e Width (largura) como Fill Parent (preencher a tela).
  - Para os três primeiros layouts horizontais, defina Height como 12% e Width como Fill Parent.
  - Para o último layout horizontal, defina Height como Fill Parent e Width como 70%.
  - Para todos, defina o alinhamento horizontal e o alinhamento vertical como CENTER.

Properties

HorizontalArrangement3

AlignHorizontal  
Center : 3 ▾

AlignVertical  
Center : 2 ▾

BackgroundColor  
■ Default

Height  
12 percent...

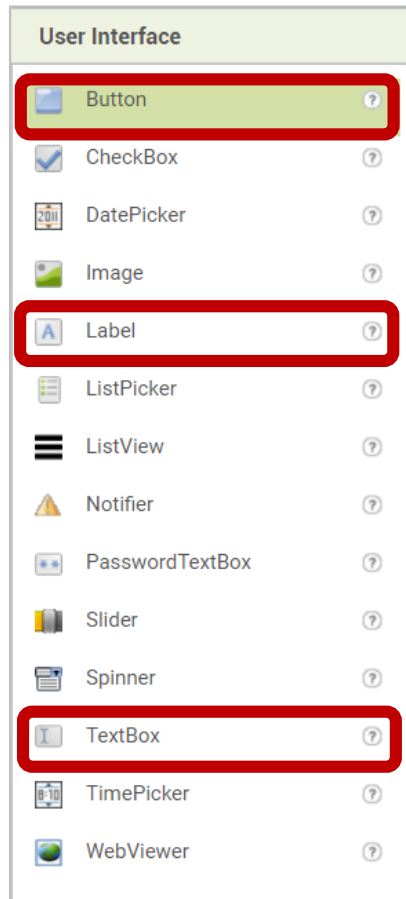
Width  
Fill parent...

Image  
None...

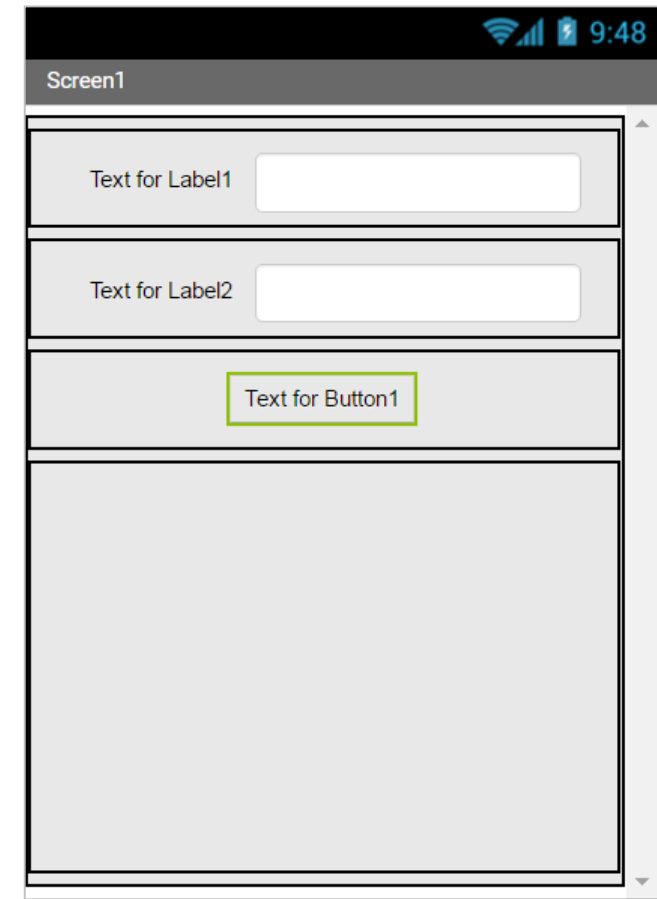
Visible  
☒

# ADICIONANDO COMPONENTES

- Para que o usuário possa cadastrar os dados da agenda, são necessários Labels (rótulos), TextBoxs (caixas de texto) e um Button (botão), todos da aba User Interface

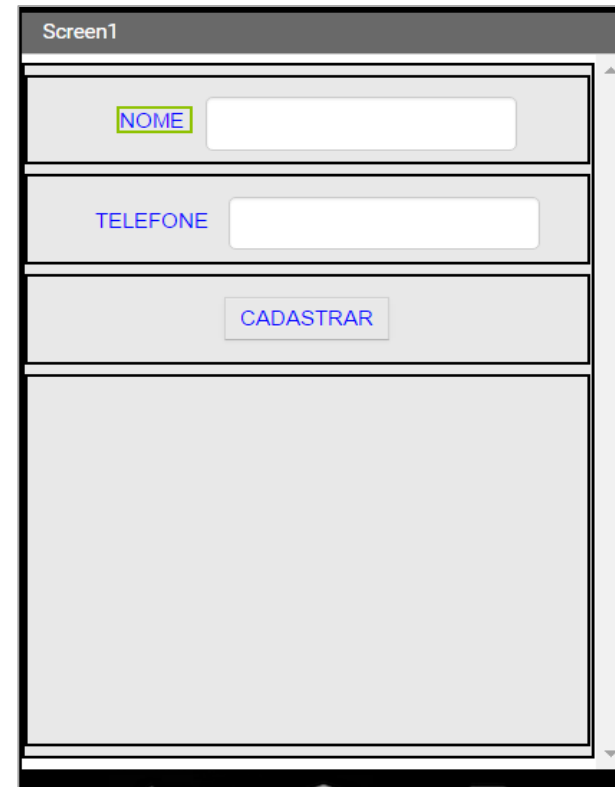
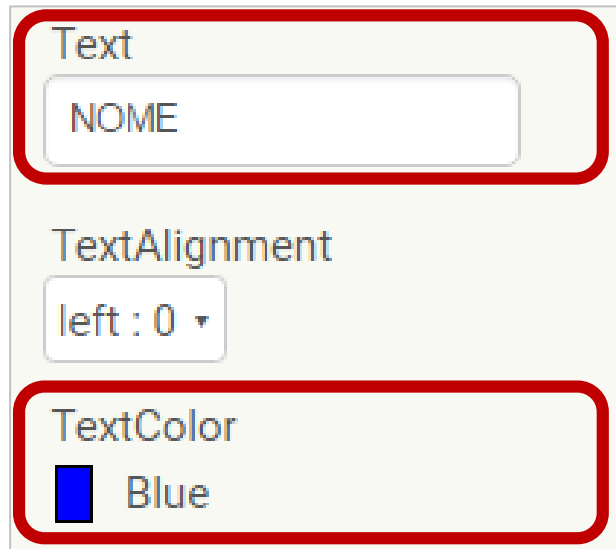


- No primeiro layout horizontal, acrescente um Label e um TextBox.
- No segundo layout horizontal, acrescente outro Label e outro TextBox.
- No terceiro layout horizontal, acrescente um Button



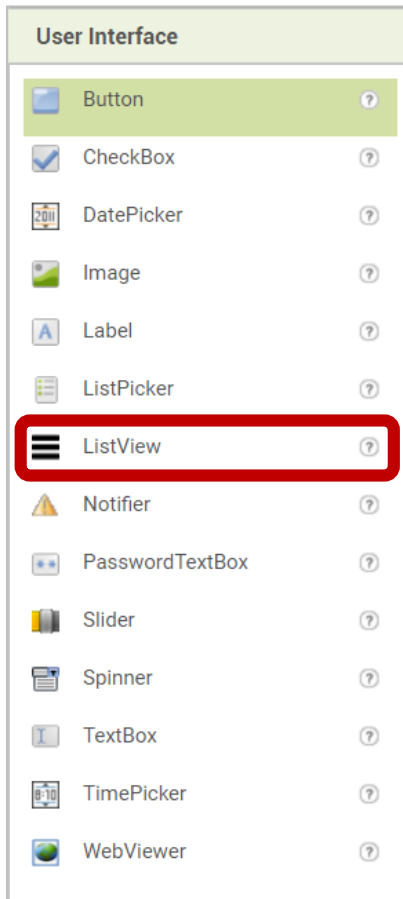
# CONFIGURANDO COMPONENTES

- As propriedades Text (texto) e TextColor (cor do texto) dos Labels (rótulos) e do Button (botão) podem ser modificadas para que o usuário entenda como deve interagir, e para que o aplicativo fique mais colorido (Dica: apague as hints das caixas de texto)



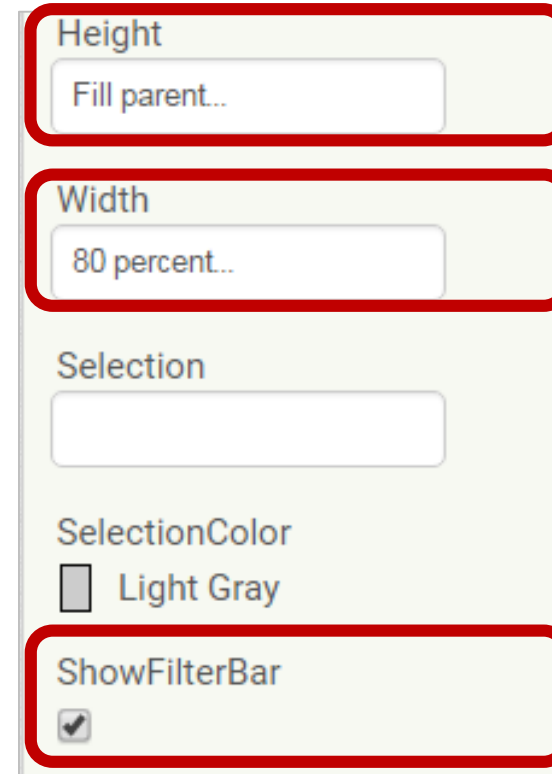
# ADICIONANDO LISTVIEW

- Para que o usuário possa visualizar os amigos já cadastrados em sua agenda, será usado um componente ListView (visualizador de lista) da aba User Interface



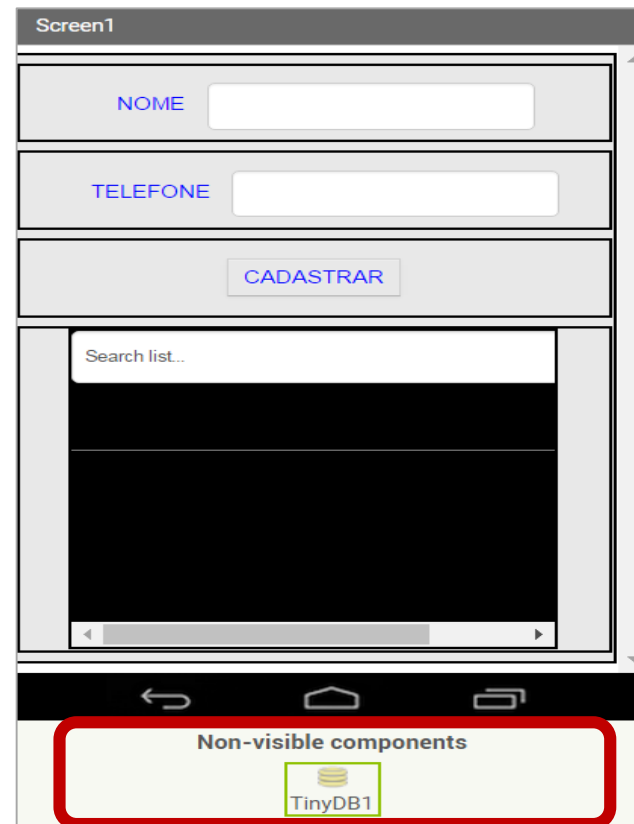
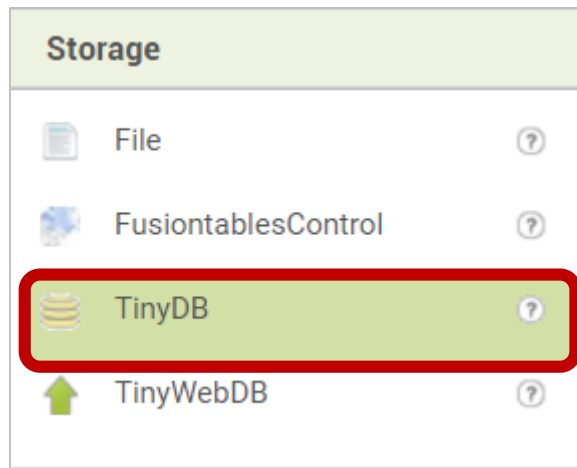
- Atribua Height (altura) como Fill Parent (preencher a tela) e Width (largura) como 80%.

- Marque a opção ShowFilterBar para permitir que sejam feitas buscas na agenda.



# ADICIONANDO TINYDB

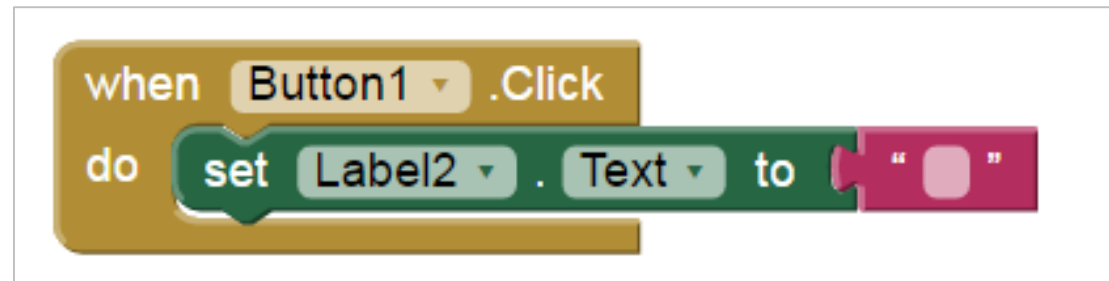
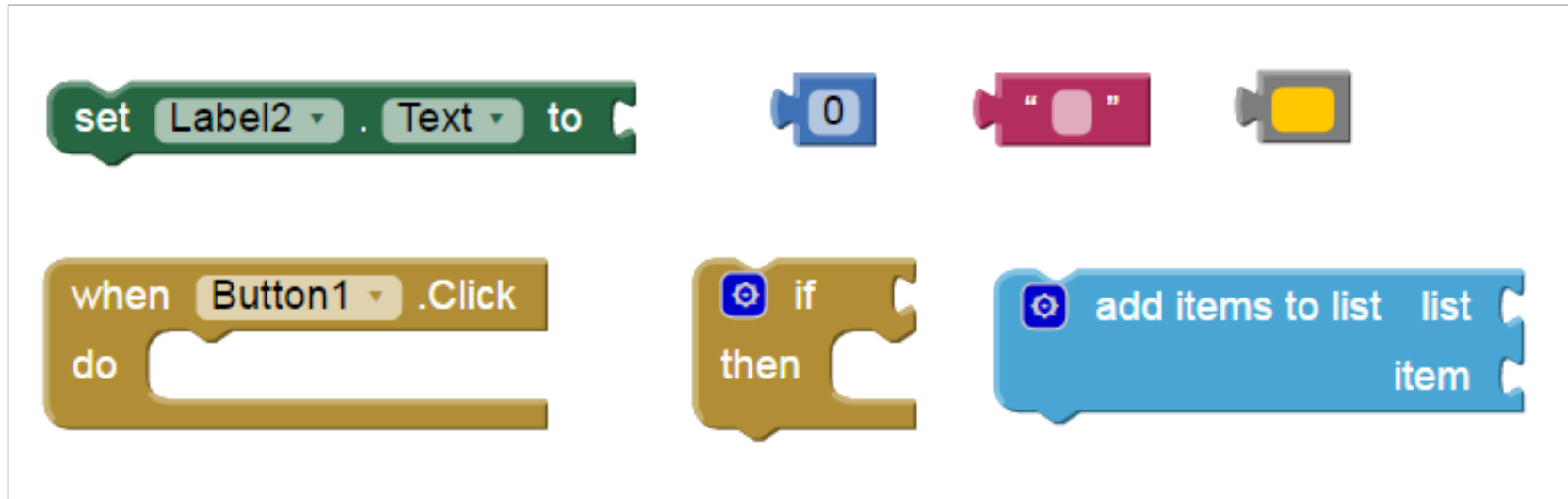
- TinyDB (aba Storage) é um componente de banco de dados que permite que os dados de uma aplicação sejam armazenados no aparelho. Ele não é um componente visível, e por isso fica abaixo da tela quando adicionado.





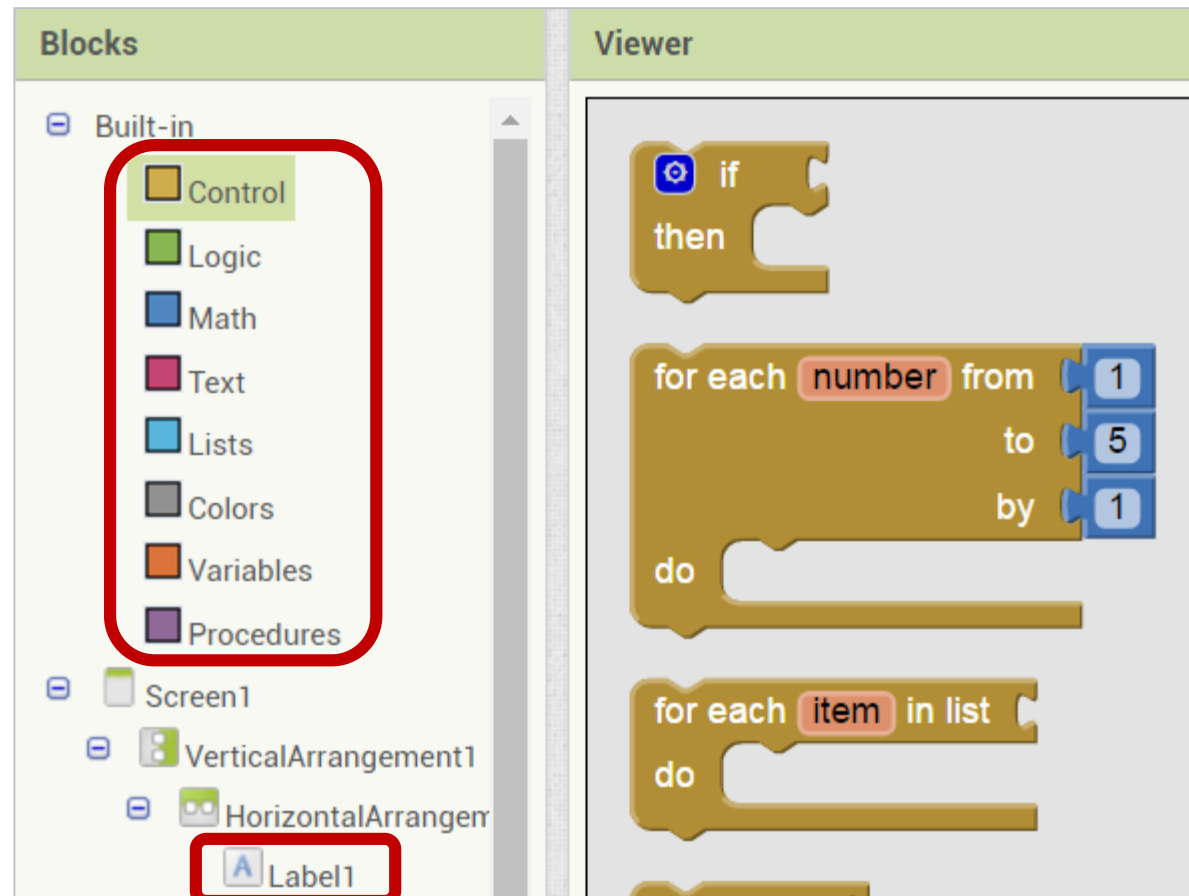
# PROGRAMANDO NO APP INVENTOR

- A programação no App Inventor é toda baseada em encaixe de blocos, como se fosse um quebra-cabeças. É preciso prestar atenção nos formatos para saber quais instruções podem ser usadas em cada caso.



# PROGRAMANDO NO APP INVENTOR

- Para saber as opções disponíveis, basta clicar em um dos tipos de bloco ou em um dos componentes adicionados (lado esquerdo do editor de blocos)

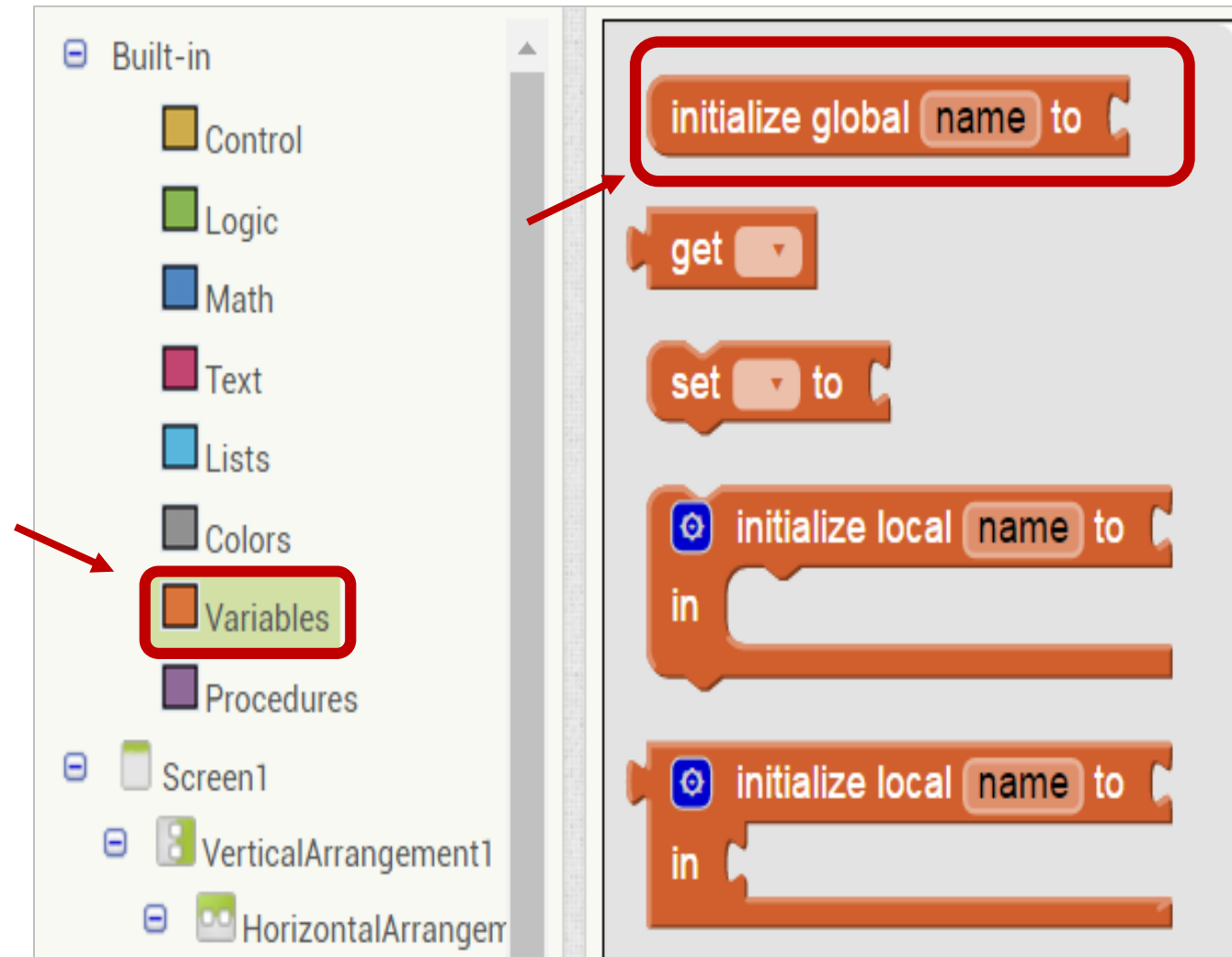


# COMEÇANDO A PROGRAMAR

- O funcionamento da agenda é bem simples:
  - O usuário vai digitar o nome e o telefone de um amigo nas duas caixas de texto da tela
  - Em seguida, vai clicar no botão Cadastrar
  - Os dados serão exibidos no componente ListView no formato Amigo – Telefone (com um hífen separando os dados)
- Como é necessário guardar dados de vários amigos, uma lista será usada
- Para que os dados não se percam depois que o aplicativo for fechado, será usado um banco de dados local temporário chamado TinyDB

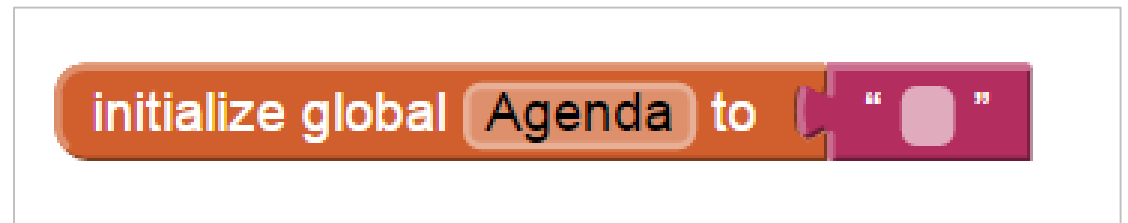
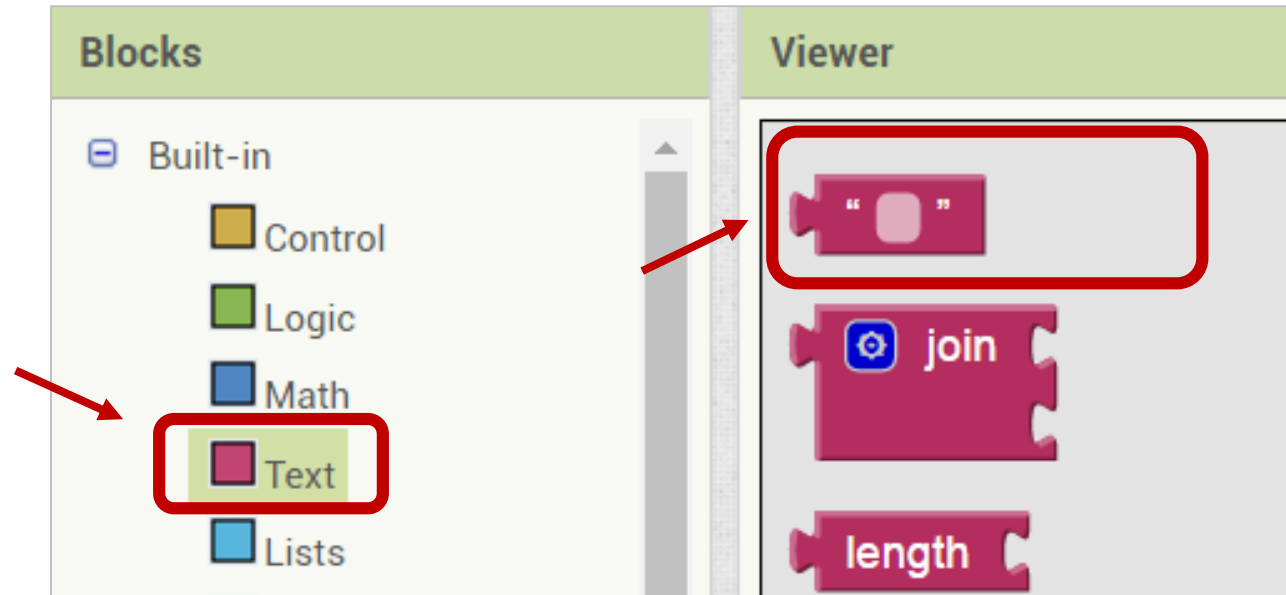
# PASSO 1 – CRIAR VARIÁVEL GLOBAL

- Para que a agenda guardada no banco de dados possa ser manipulada (recuperada, alterada e armazenada), o primeiro passo é criar uma variável global que irá contê-la
- No menu Variables, escolha a primeira opção (initialize global name to)
- Substitua name por Agenda para renomear a variável



# PASSO 1 – CRIAR VARIÁVEL GLOBAL

- Essa variável deverá ter valor inicial "" (String em branco)
- Para atribuir, no menu Text, escolha a primeira opção e encaixe a peça na anterior
- Essa será a primeira instrução do programa

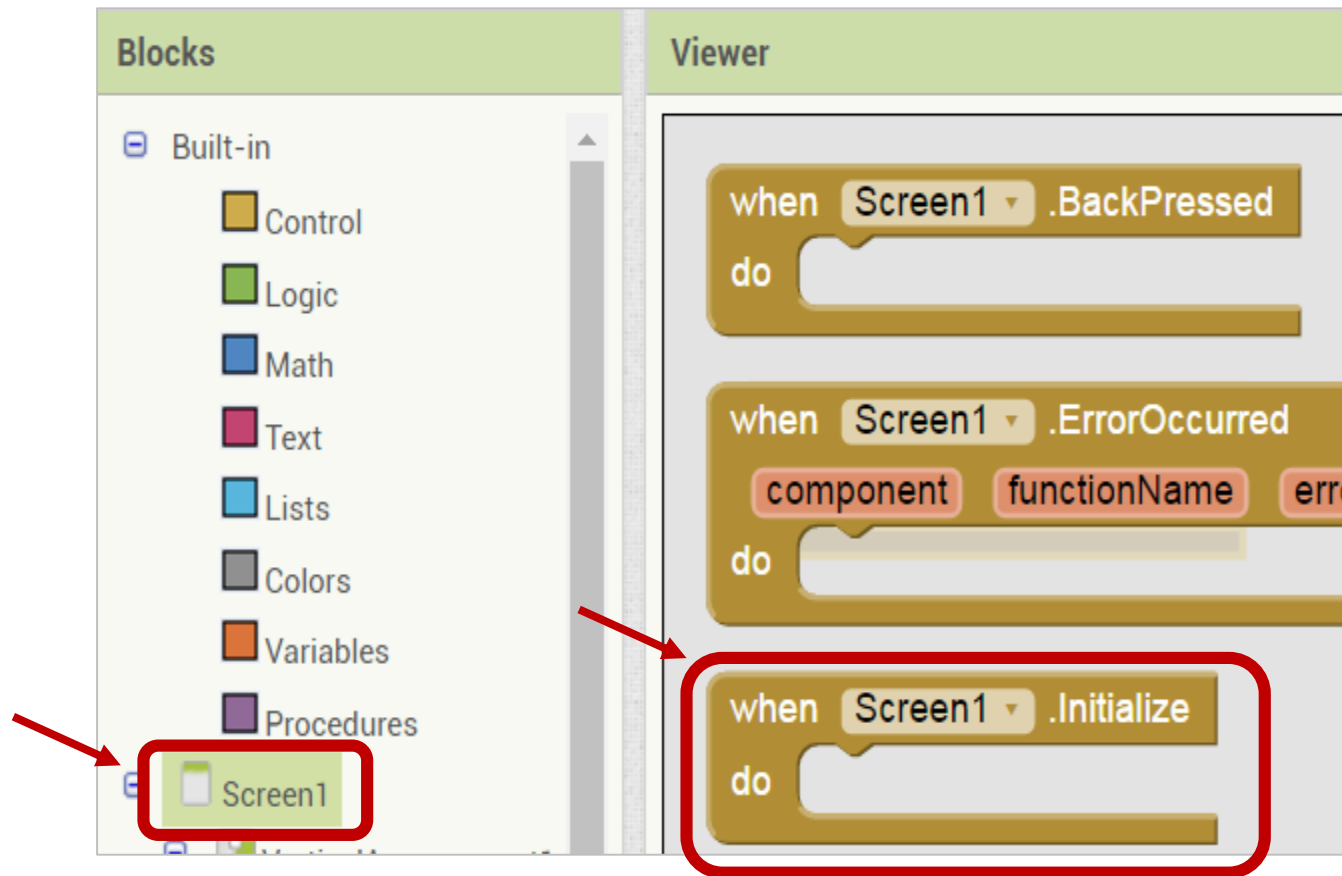


# PASSO 2 – RECUPERAR O BANCO

- Sempre que a aplicação for iniciada, será preciso primeiro recuperar o banco de dados existente, para que ele não seja sobreposto.
- Esse banco de dados será guardado na variável global criada previamente
- Caso a aplicação esteja sendo usada pela primeira vez, o banco estará vazio, e a variável global poderá assumir o valor de um String, um número ou uma lista.
- Para que os dados fiquem organizados no banco de dados, é possível criar várias tags (o equivalente a colunas).
  - Obs: Neste exemplo, em vez de usar duas tags (uma para Nome e outra para Telefone) para guardar os dados de cada amigo no banco de dados, será criada uma única tag para armazenar Strings no formato Nome - Telefone.

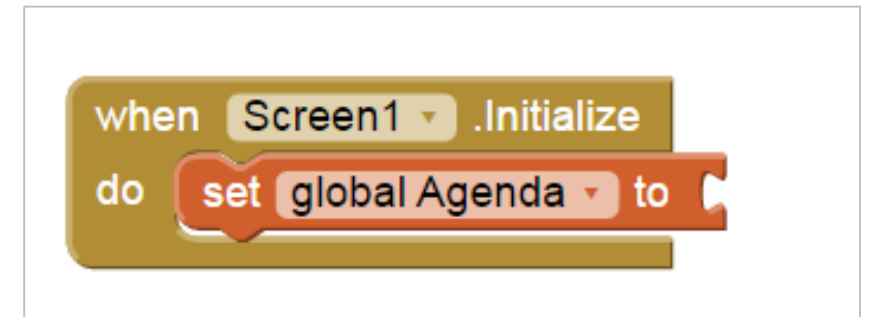
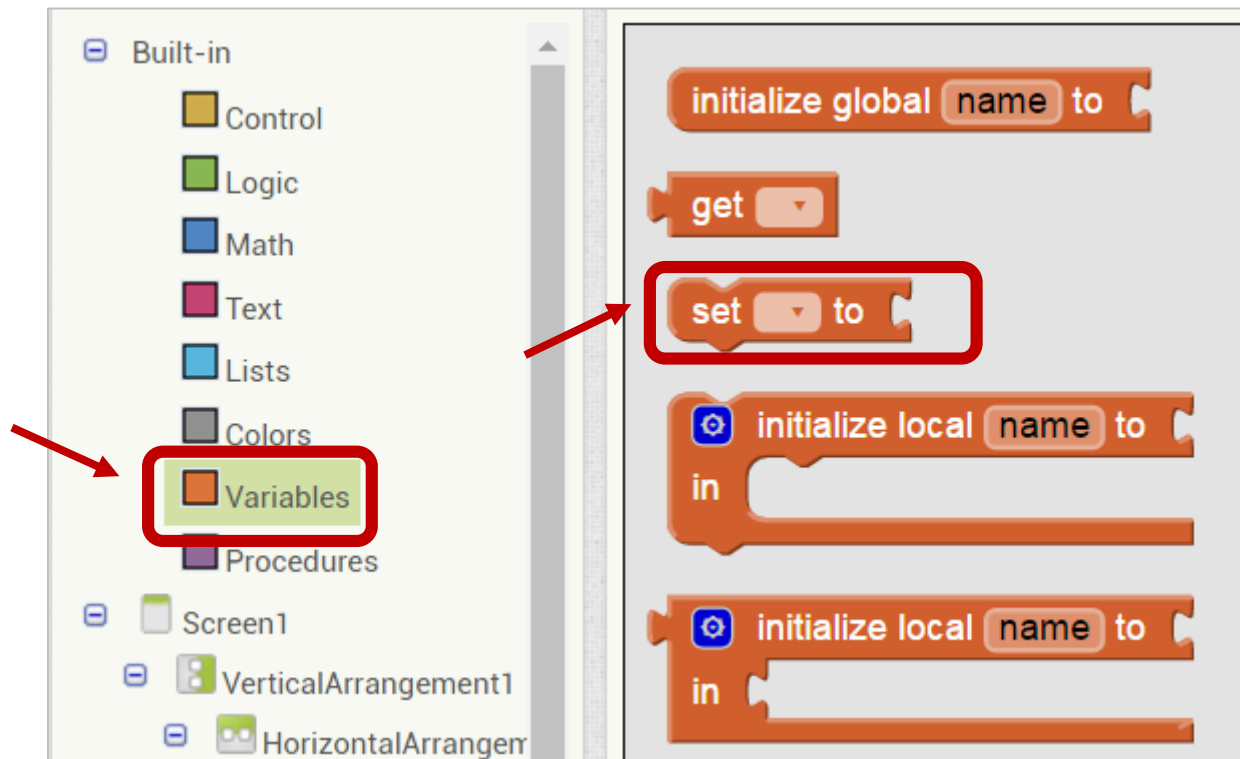
# PASSO 2 – RECUPERAR O BANCO

- Para definir o que deve acontecer toda vez que a aplicação for iniciada, escolha o componente Screen1 e o evento correspondente



# PASSO 2 – RECUPERAR O BANCO

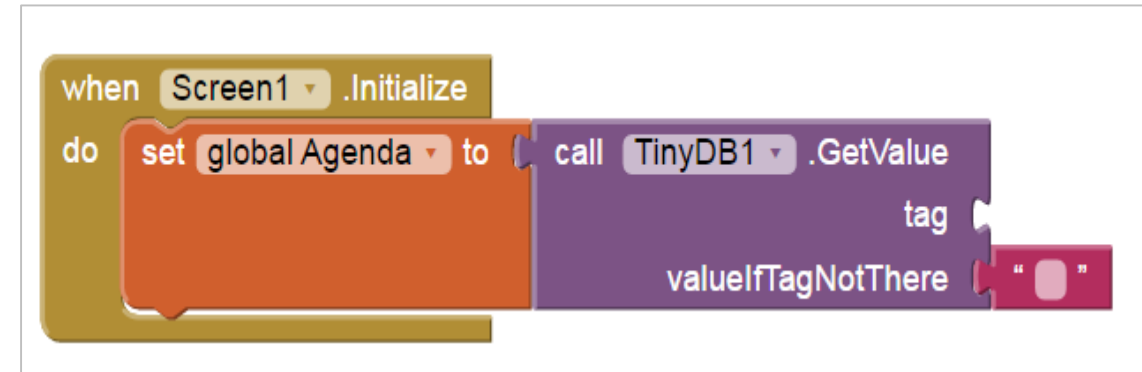
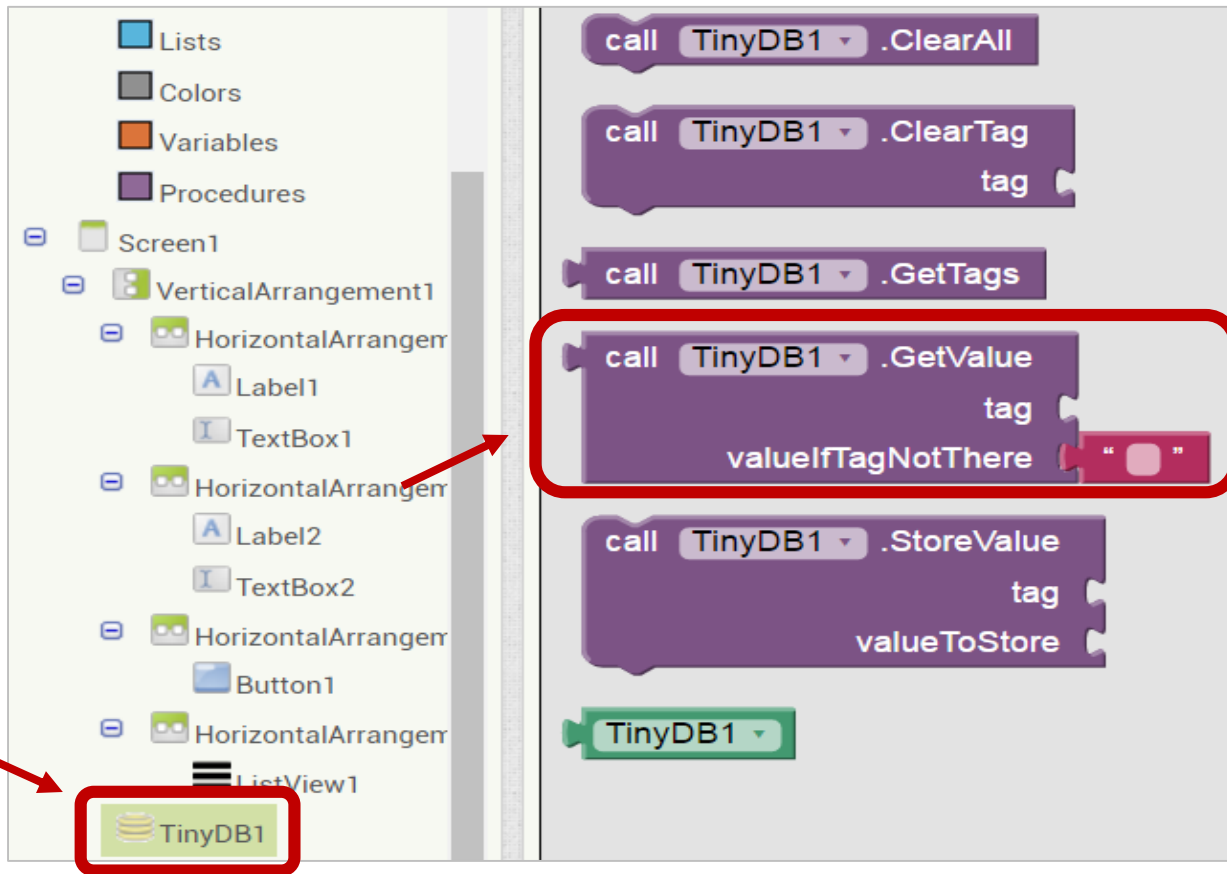
- Para alterar o valor da variável global criada inicialmente, escolha o tipo de bloco Variables, e a opção set. Escolha a variável global Agenda na lista de opções do próprio bloco, e encaixe-o no bloco criado no passo anterior.





# PASSO 2 – RECUPERAR O BANCO

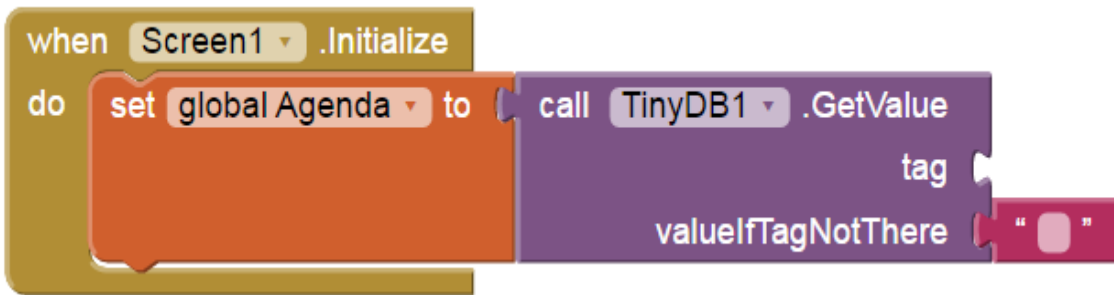
- Em seguida, escolha o componente TinyDB e a opção GetValue. Encaixe esse bloco na estrutura já montada.



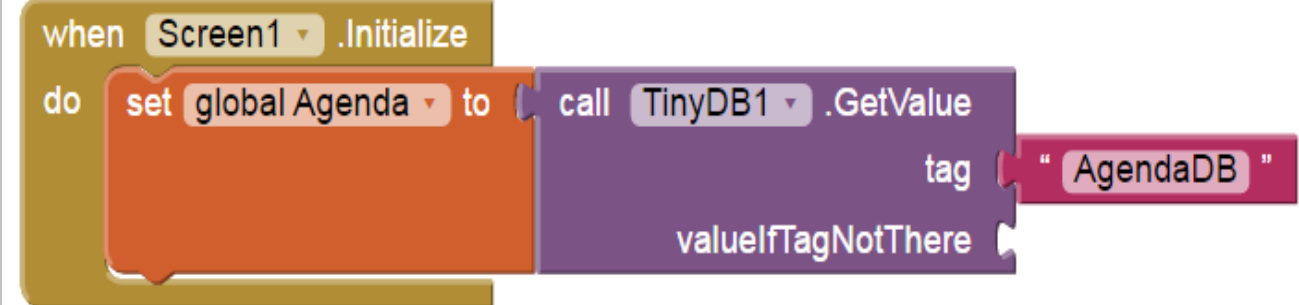
# PASSO 2 – RECUPERAR O BANCO

- Para concluir essa instrução, é preciso definir o nome da tag que deve ser buscada no banco de dados (para esse exemplo, será AgendaDB), e o valor a ser assumido pela variável caso o banco esteja vazio (será uma lista vazia).
- Troque a posição do bloco com String vazio, conectando-o no encaixe de cima do bloco roxo, e altere seu valor para AgendaDB

**Antes**

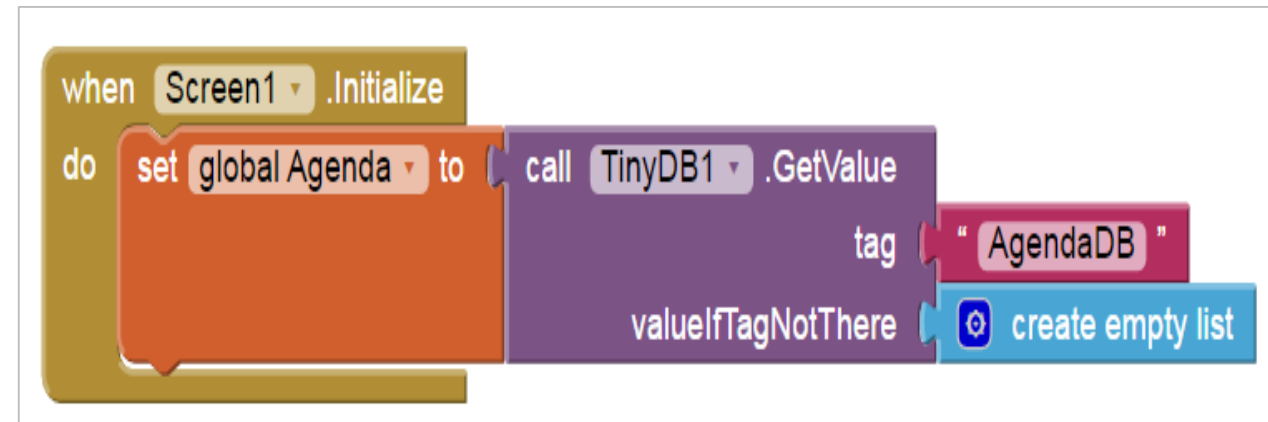
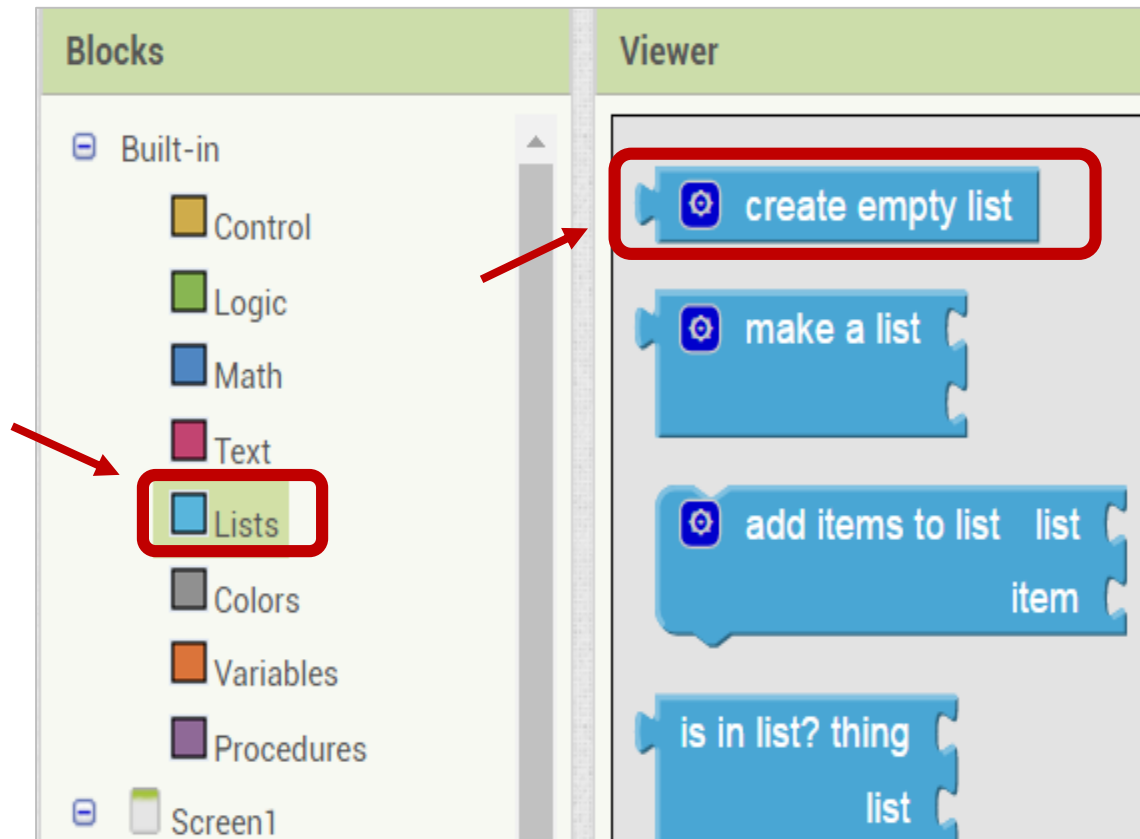


**Depois**



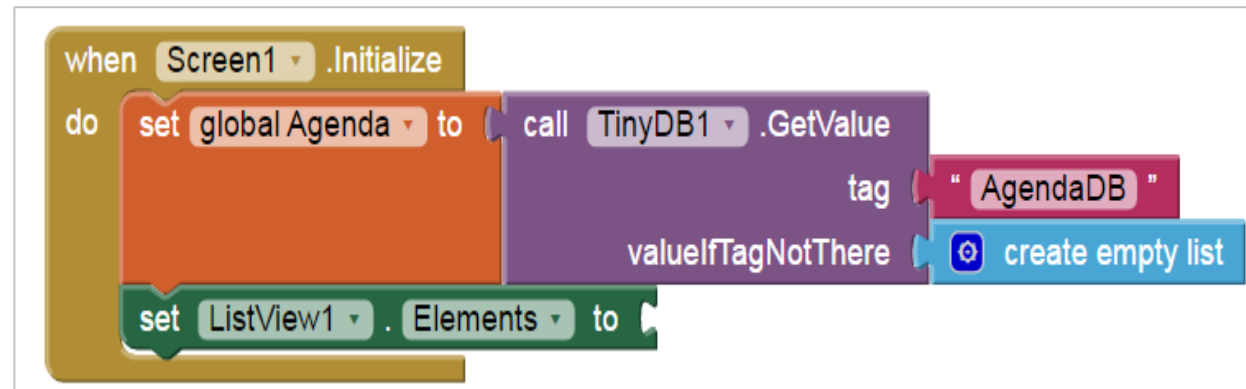
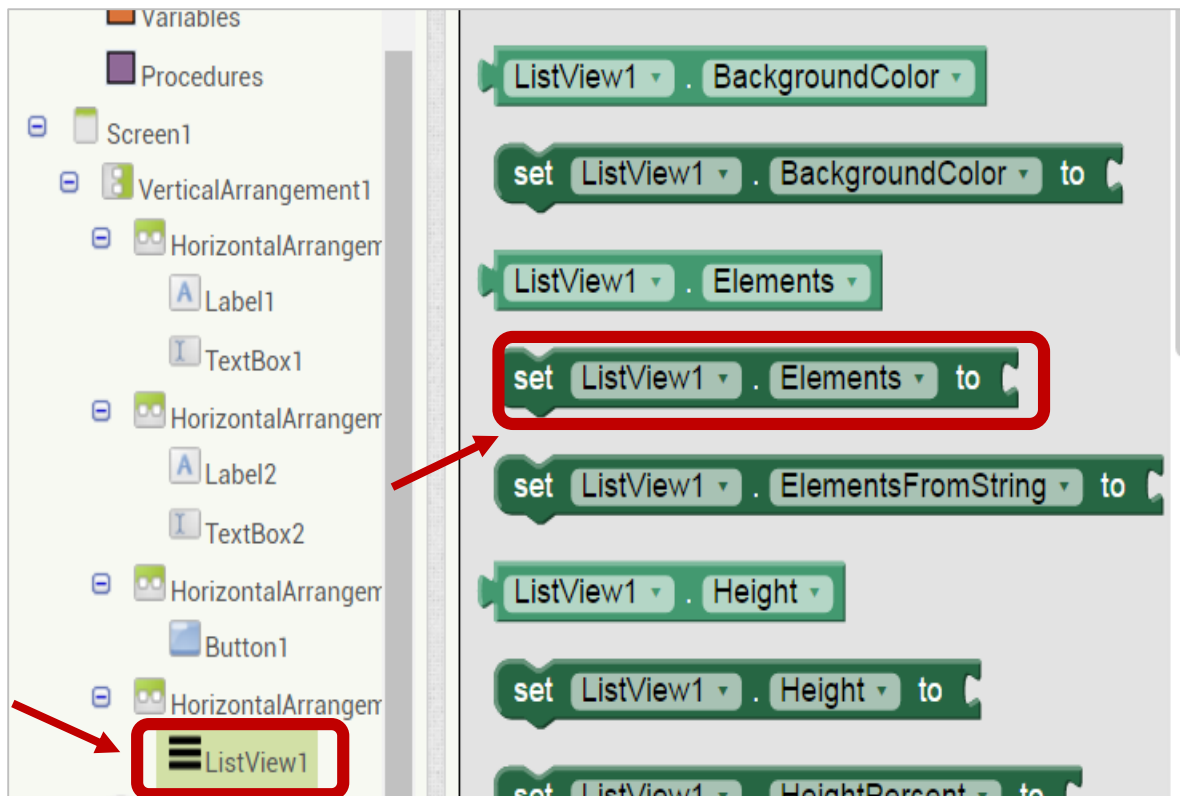
# PASSO 2 – RECUPERAR O BANCO

- Basta agora definir que uma lista vazia seja atribuída à variável caso o banco esteja vazio. Escolha o tipo de bloco Lists e a opção create empty list.



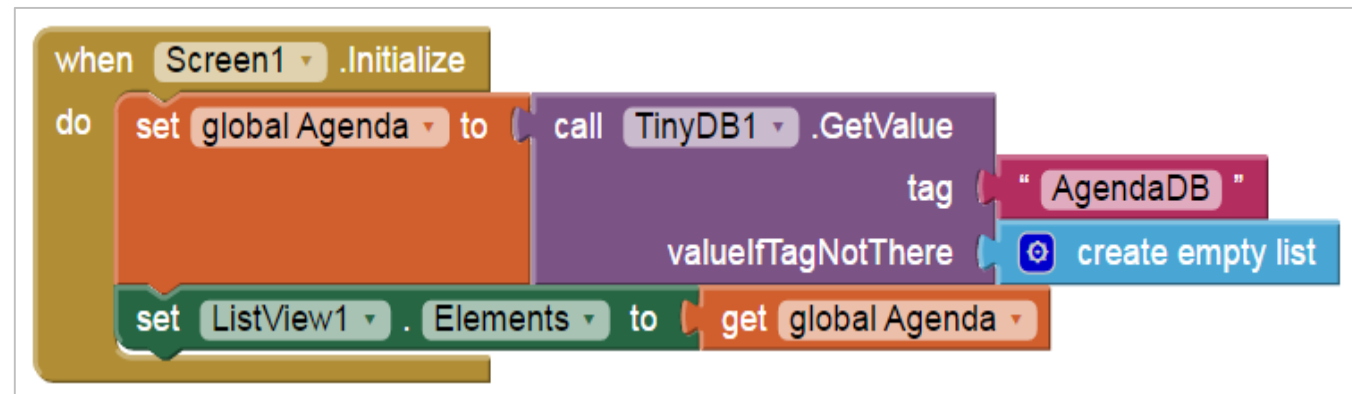
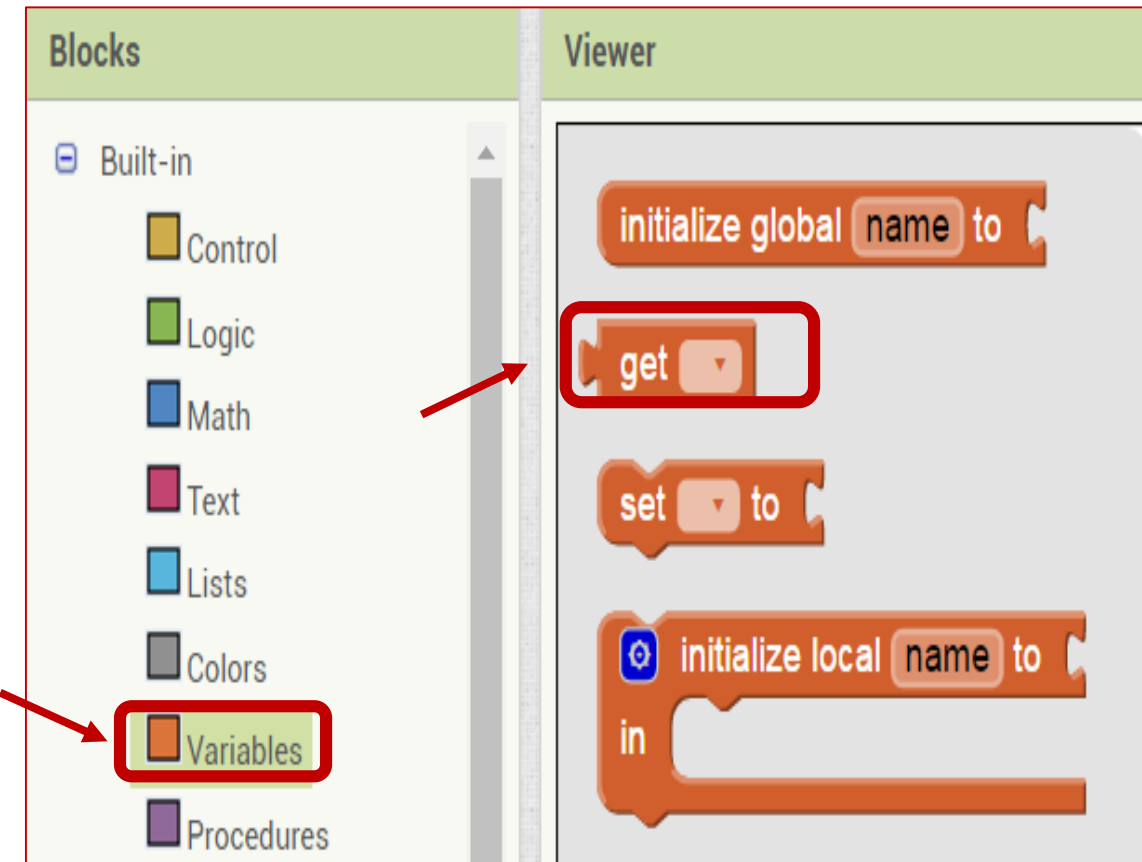
# PASSO 2 – RECUPERAR O BANCO

- Para que o conteúdo dessa variável seja sempre exibido na parte de baixo da tela, é preciso escolher o componente ListView, e a opção set ListView elements. Ele deve ser encaixado na parte de baixo do bloco anterior.



# PASSO 2 – RECUPERAR O BANCO

- Agora basta obter o valor da variável global Agenda. Escolha o tipo de bloco Variables, e a opção get. Na lista de opções, escolha global Agenda e encaixe.

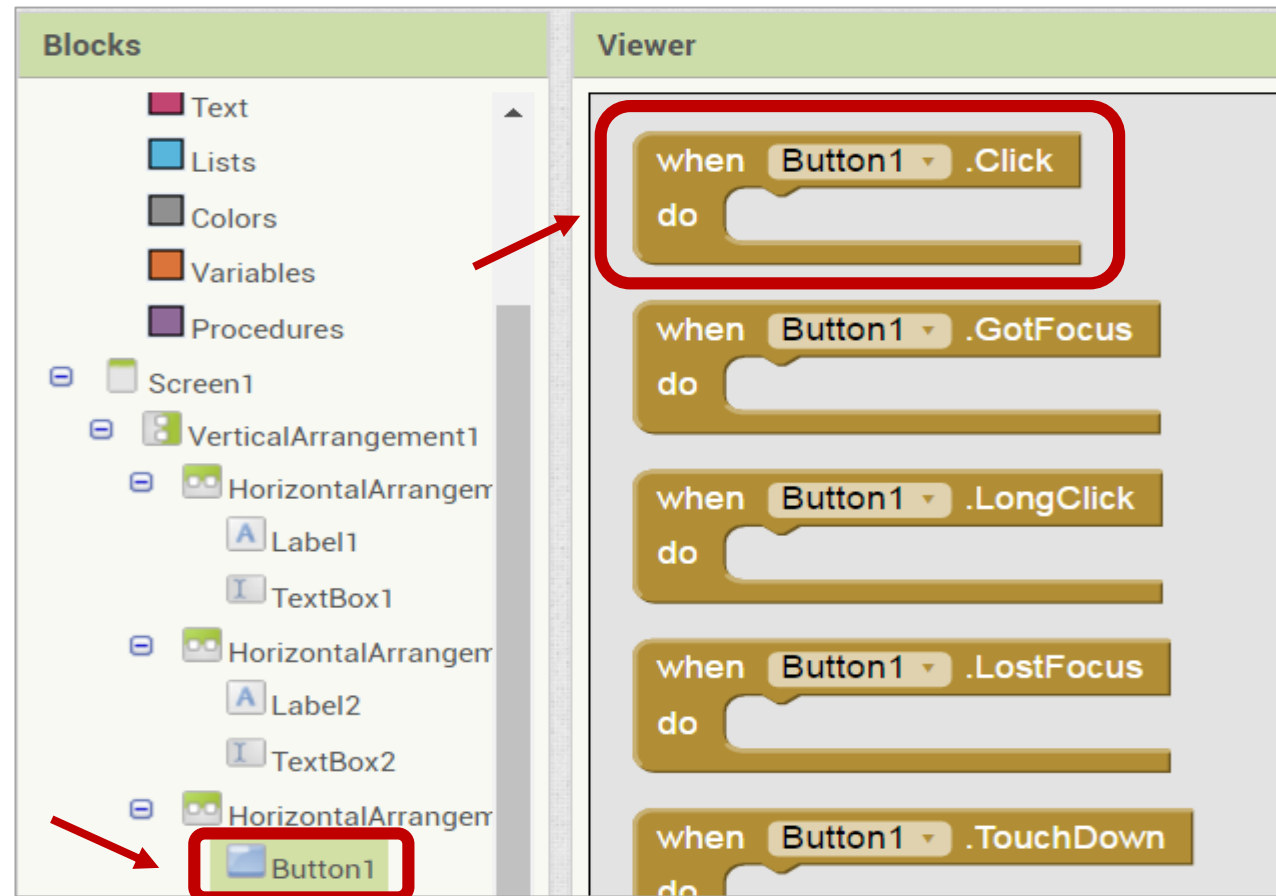


# PASSO 3 – GRAVAR O NOVO DADO

- Quando o botão Cadastrar for clicado, será preciso:
  - Recuperar os textos informados nas caixas de texto para Nome e Telefone
  - Transformá-los em um único String, unindo-os por hífen (Ex: Nome – Telefone)
  - Adicionar esse String à lista já existente e armazenada na variável global criada
  - Armazenar a variável global no banco de dados
  - Exibir o conteúdo dessa lista no componente ListView
  - Limpar o conteúdo das caixas de texto para que novos dados possam ser acrescentados

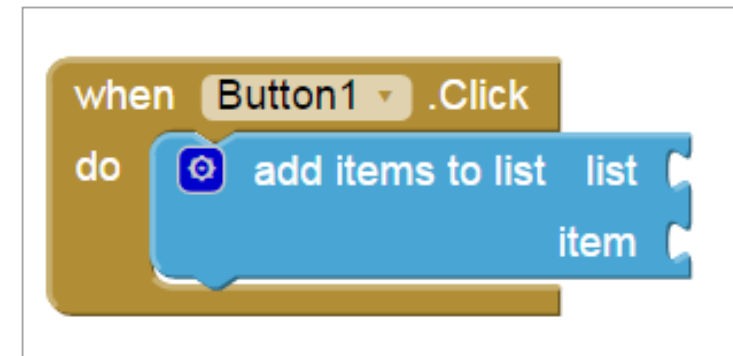
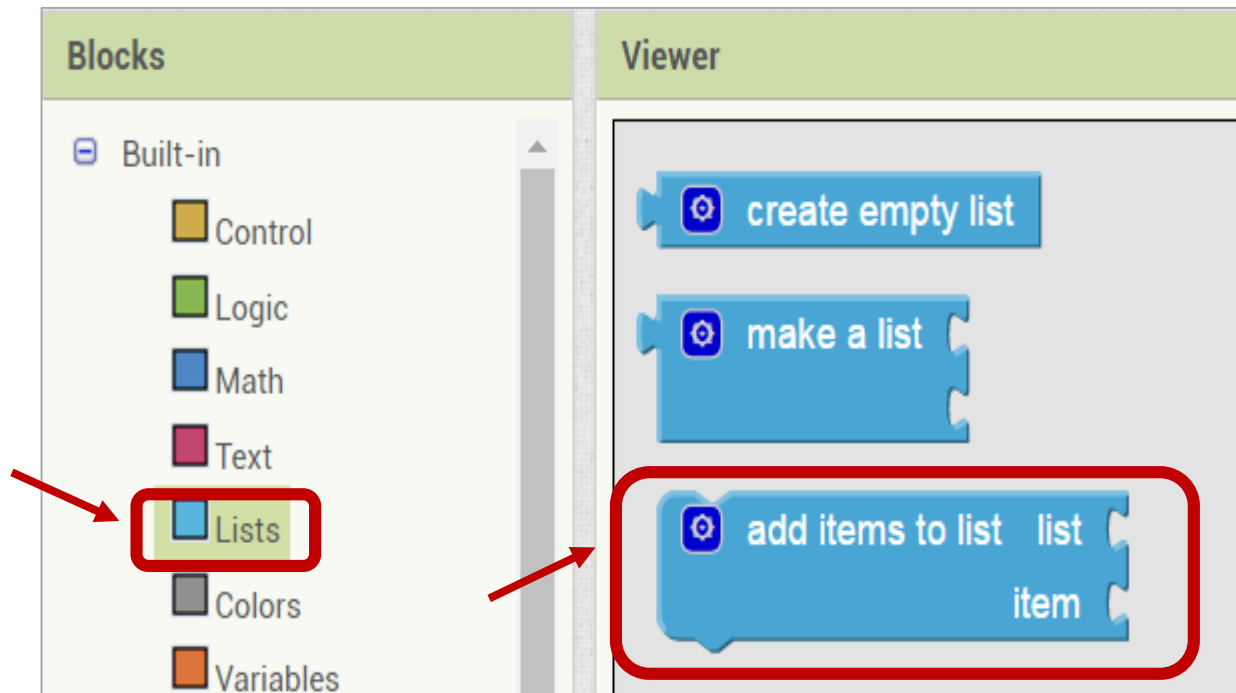
# PASSO 3 – GRAVAR O NOVO DADO

- Inicialmente, é necessário escolher o componente botão adicionado e selecionar a opção when Button1 Click.



# PASSO 3 – GRAVAR O NOVO DADO

- Agora é preciso indicar que se deseja acrescentar um item a uma lista. Selecione o tipo de bloco Lists e a opção add items to list, encaixando essa peça na criada anteriormente.





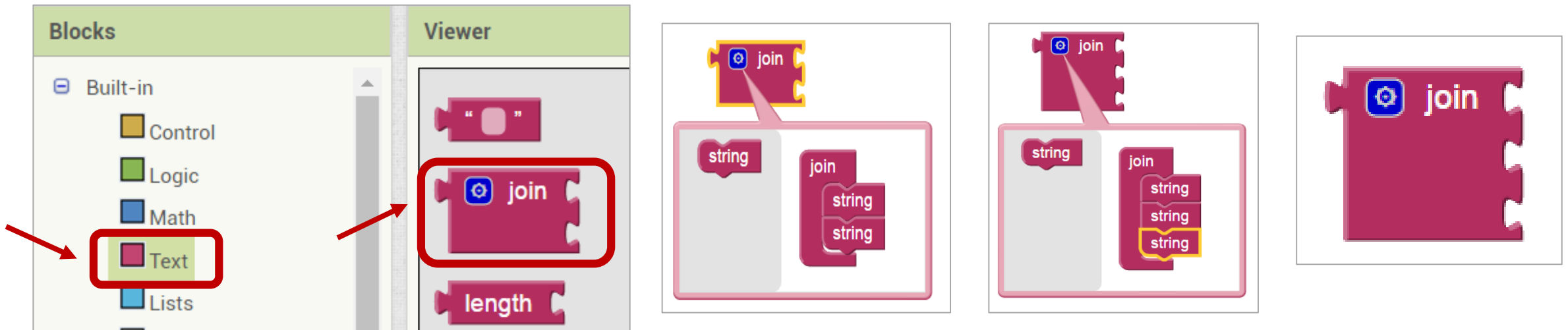
# PASSO 3 – GRAVAR O NOVO DADO

- A lista que deve receber um novo elemento é **global Agenda**. Escolha o tipo de bloco Variables, e a opção get. Na lista de opções, escolha a variável e encaixe.



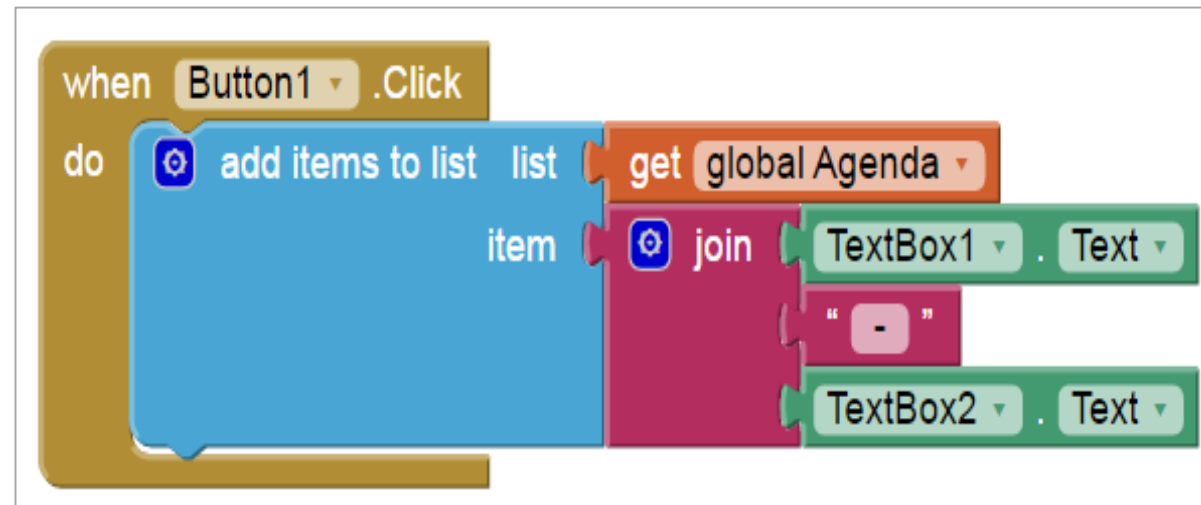
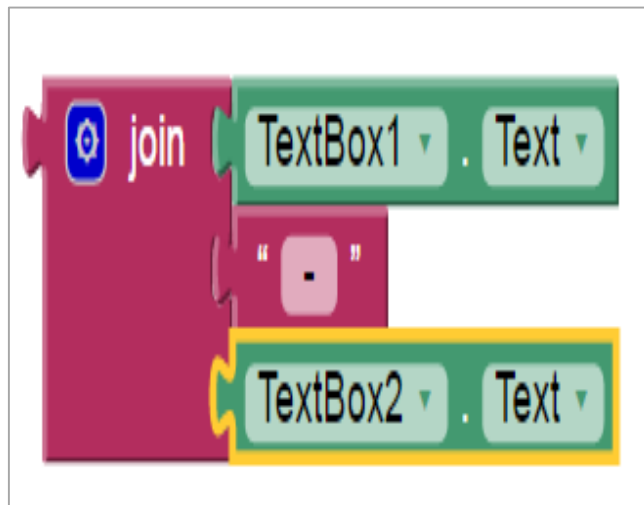
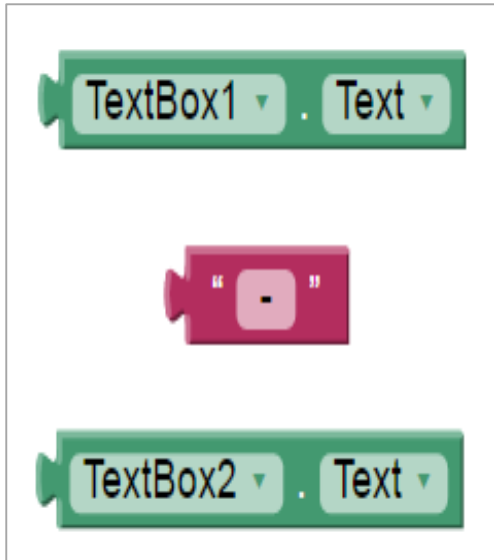
# PASSO 3 – GRAVAR O NOVO DADO

- O item que se deseja adicionar na lista será uma junção do nome do amigo, com um hífen, e com seu telefone. Para isso, basta escolher o tipo de blocos Text e a opção join. Como ela apresenta apenas duas possibilidades de encaixe, clique na engrenagem azul e arraste a peça String para incluí-la abaixo das outras duas, obtendo um bloco com três encaixes.



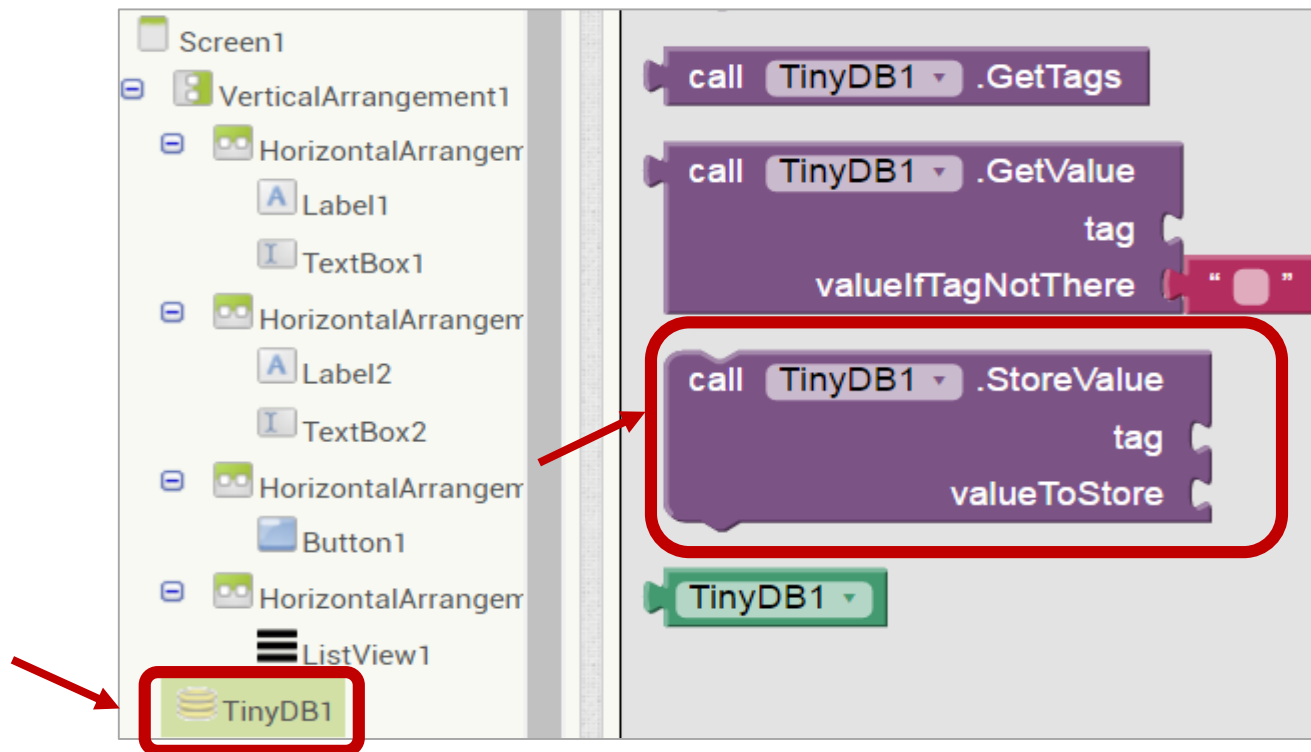
# PASSO 3 – GRAVAR O NOVO DADO

- Agora serão definidos os Strings a serem unidos: o conteúdo da primeira caixa de texto (componente TextBox1, opção TextBox1.text), um hífen (tipo de bloco Text, opção " - "), preenchida com – ) e o conteúdo da segunda caixa de texto (componente TextBox2, opção TextBox2.text). Encaixe todas as peças no bloco join já criado, e em seguida no bloco maior.



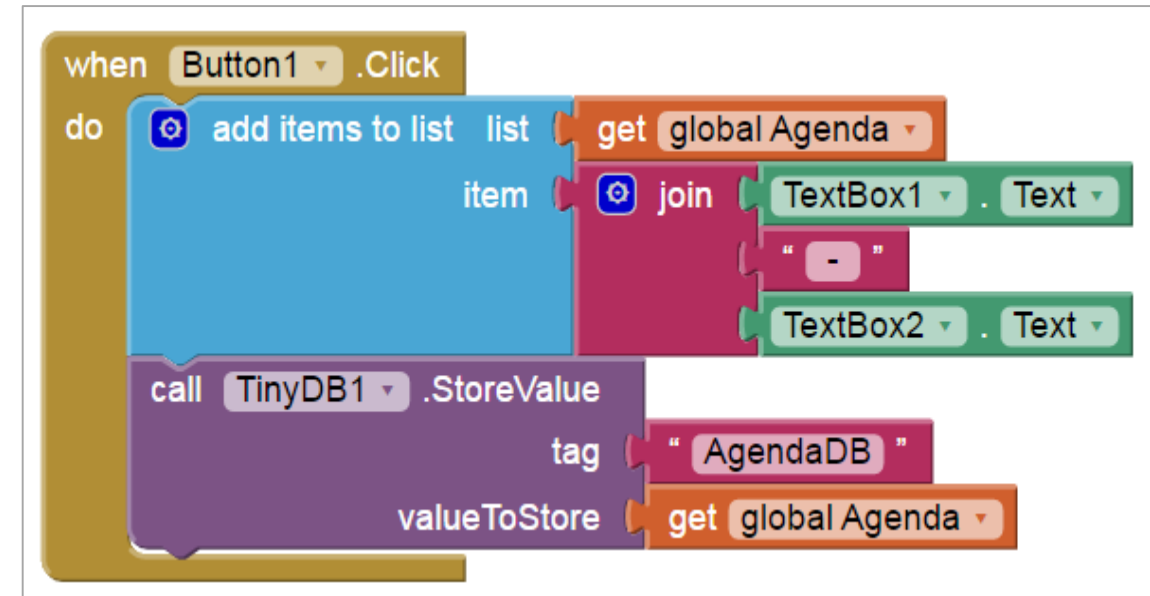
# PASSO 3 – GRAVAR O NOVO DADO

- Como a variável global Agenda foi modificada, é preciso atualizar o banco de dados com o novo valor. Mais uma vez, será necessário escolher o componente TinyDB1, selecionando agora a opção StoreValue.



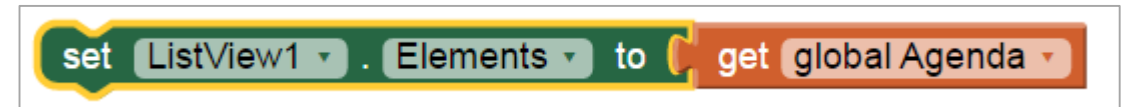
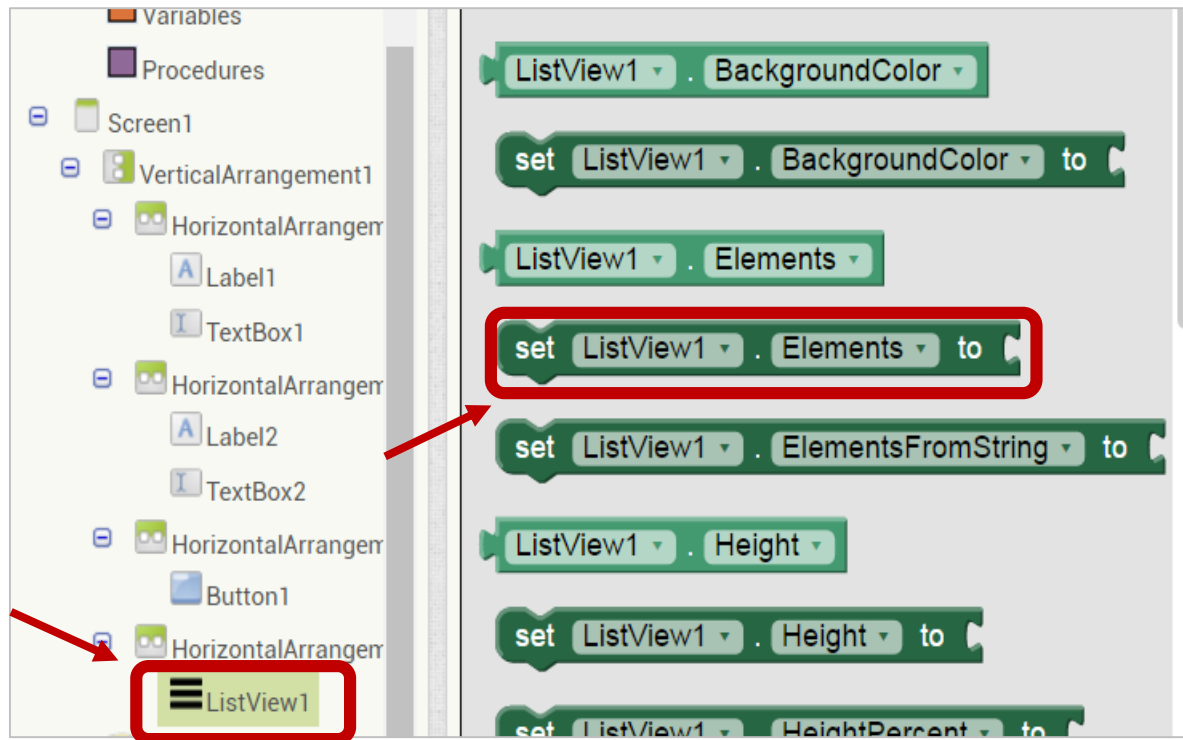
# PASSO 3 – GRAVAR O NOVO DADO

- A tag usada no banco de dados é AgendaDB, e esse String deve ser informado em um elemento " " do tipo de blocos Text. Já o valor a ser armazenado é o conteúdo da variável global Agenda, obtida na opção get do tipo de blocos Variable. Essas novas peças devem ser encaixadas no bloco maior que está sendo construído.



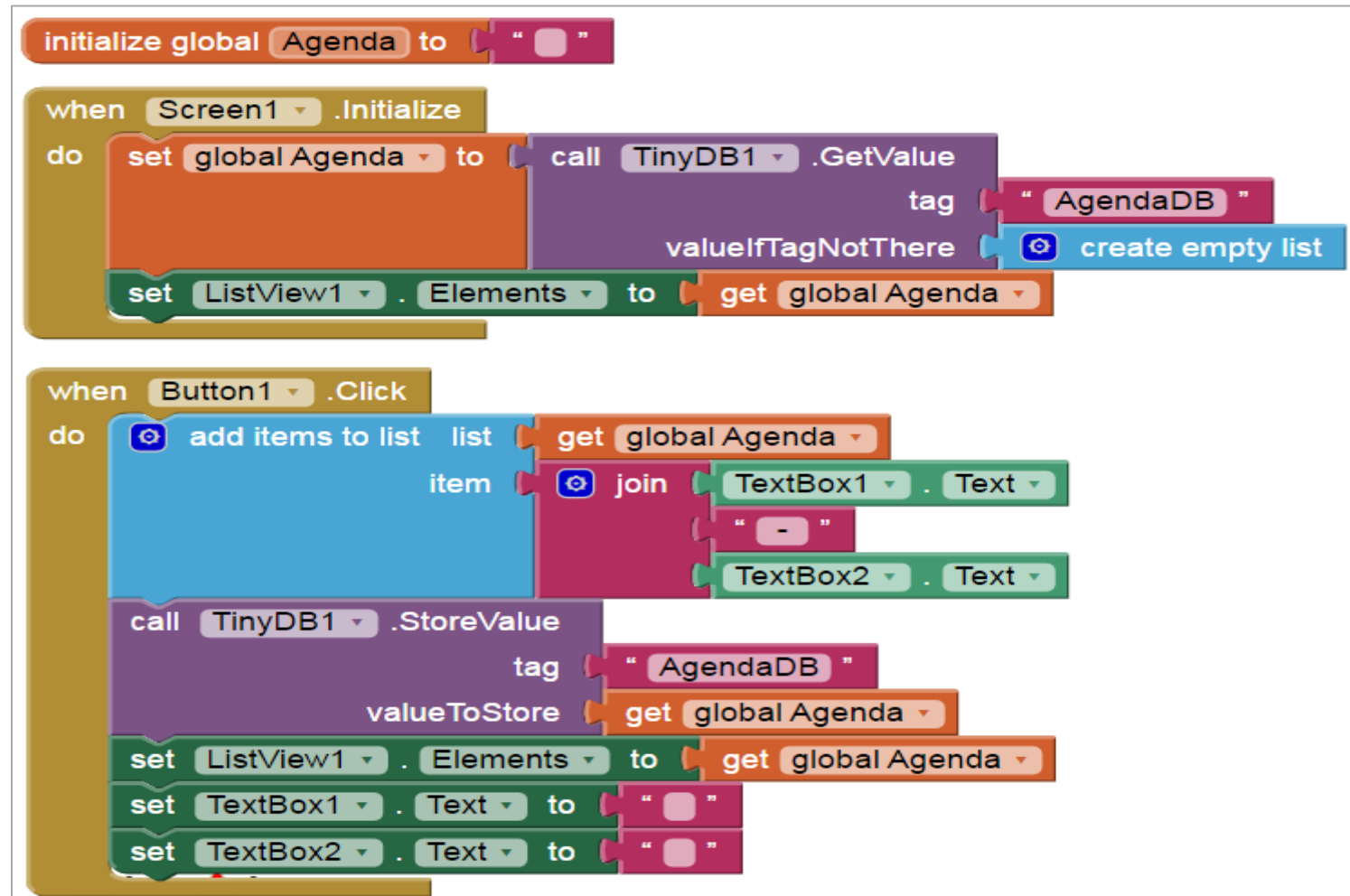
# PASSO 3 – GRAVAR O NOVO DADO

- Os dados estão armazenados, mas ainda não é possível vê-los na tela, pois é preciso atribuir a variável global Agenda aos elementos do ListView a cada atualização. Para isso, basta selecionar o componente ListView e a opção set ListView Elements.



# PASSO 3 – GRAVAR O NOVO DADO

➤ Sua agenda está pronta! Teste o aplicativo e faça as melhorias que desejar.



# AGENDA – VERSÃO 2.0

- Uma melhoria bastante interessante seria fazer uma aplicação com mais de uma tela.

The mockup shows a mobile application interface for registration. At the top, there's a status bar with signal, battery, and time (9:48). Below it is a yellow header with the text "Minha Agenda". The main content area is yellow and contains two input fields: "Amigo:" and "Telefone:". Below these fields are two buttons: "Cadastrar" and "Ver lista". The bottom of the screen features a black navigation bar with standard Android icons (back, home, recent apps).

**Tela 1 - Cadastro**

The mockup shows a mobile application interface for search and visualization. At the top, there's a status bar with signal, battery, and time (9:48). Below it is a yellow header with the text "Minha Agenda". The main content area is yellow and contains a search bar with the placeholder text "Search list..". Below the search bar is a horizontal line. At the bottom of the main content area is a scroll indicator (a grey bar with arrows). Below this is a yellow footer with a button labeled "Voltar". The bottom of the screen features a black navigation bar with standard Android icons (back, home, recent apps).

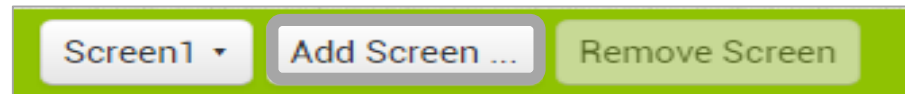
**Tela 2 – Busca e Visualização**



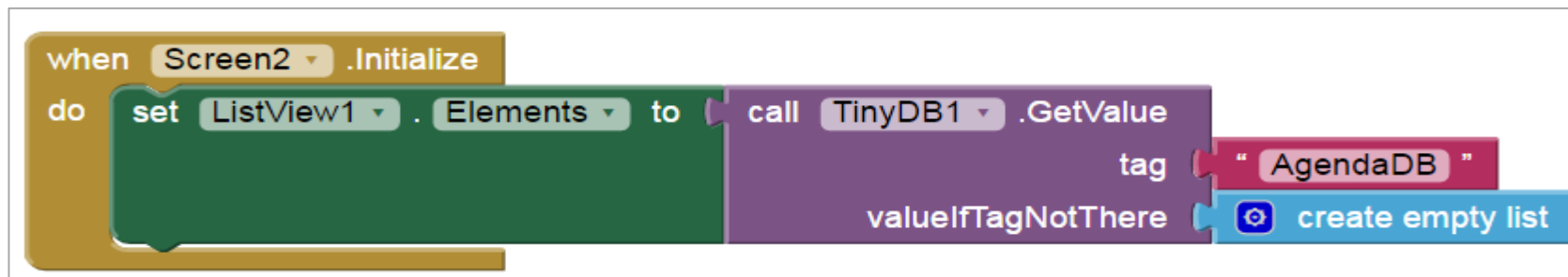
# AGENDA – VERSÃO 2.0

## ➤ Para isso, seria preciso:

- Acrescentar uma nova tela (opção disponível logo acima do editor de interface)

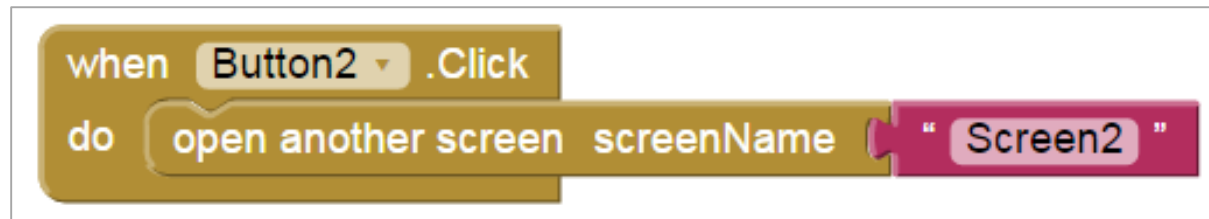


- Montar o layout da nova tela com um painel vertical e outros horizontais.
- Adicionar na segunda tela um componente TinyDB com o mesmo nome do que está na primeira tela. As tags usadas também devem ser as mesmas.
- Retirar o componente ListView da primeira tela e adicioná-lo na segunda tela.
- Adicionar o código para definir os elementos do ListView no evento de inicialização da tela 2.

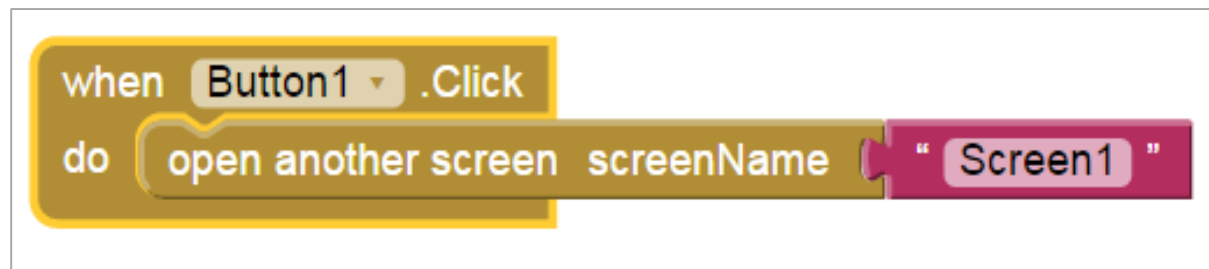


# AGENDA – VERSÃO 2.0

- Acrescentar um novo botão na tela 1
- Programar a opção Click desse botão da tela 1 para que ele abra uma nova tela (tipo de bloco Control , opção open another screen)

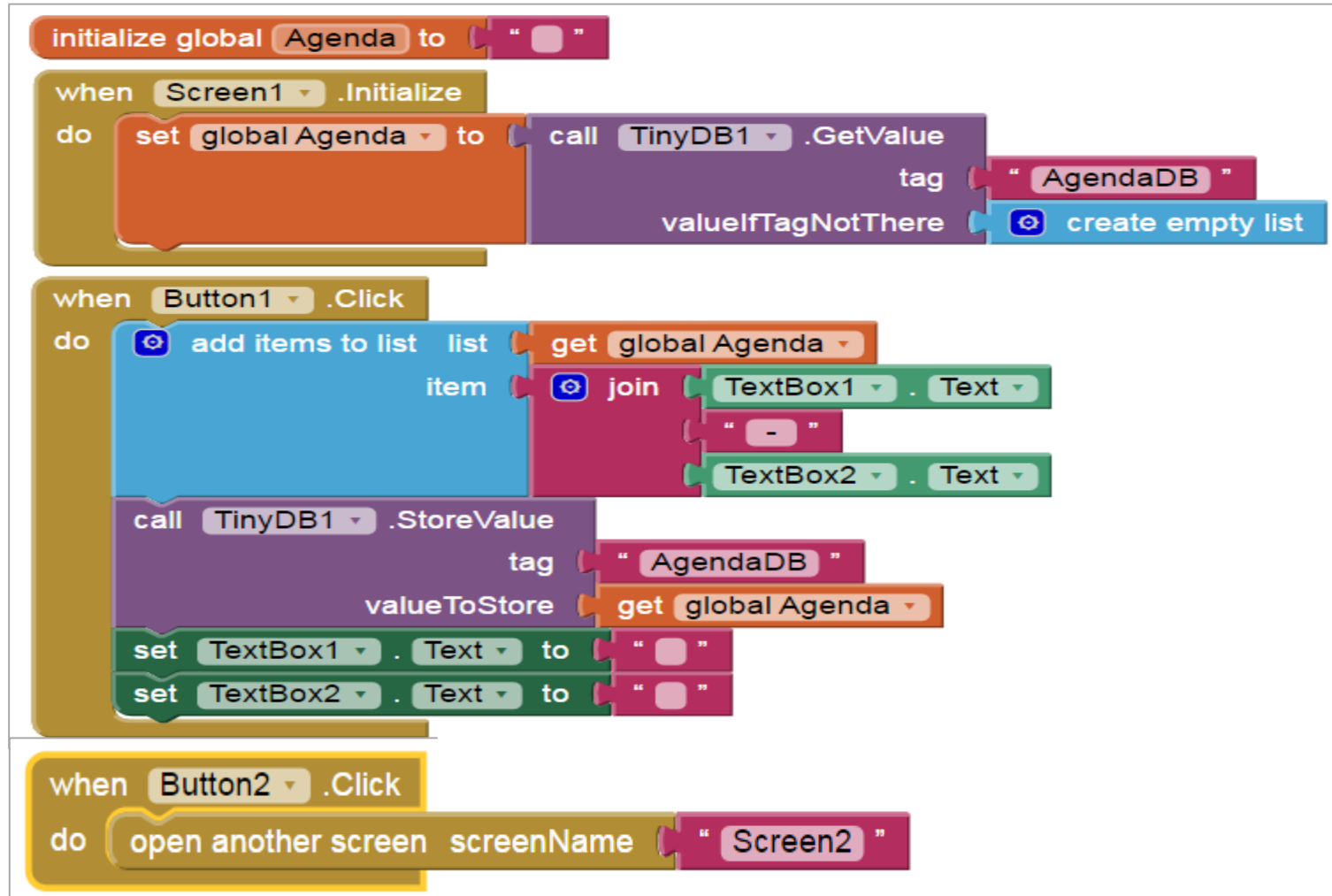


- Acrescentar um botão na tela 2 para poder voltar à tela 1, programando sua opção de forma semelhante ao que foi feito na tela 1



# AGENDA – VERSÃO 2.0

## ➤ Código completo da Tela 1



# AGENDA – VERSÃO 2.0

## ➤ Código completo da Tela 2

