

Tema	DADOS ESPACIAIS		
Descrição da Aula	As crianças aprenderão a usar a Interface de Programação de Aplicativos ou API para obter dados em tempo real sobre o número de humanos no espaço de um banco de dados de código aberto.		
Aula	ADV-C73		
Tempo de aula	55 minutos		
Meta	 <ul style="list-style-type: none"> Desenvolver um Aplicativo de Dados Espaciais usando APIs. 		
Recursos necessários	<ul style="list-style-type: none"> Recursos da professora: <ul style="list-style-type: none"> Laptop / PC com WebCam Use as credenciais de login do Gmail. Fone de ouvido com microfone Bloco de notas e caneta Recursos do aluno: <ul style="list-style-type: none"> Laptop / PC com WebCam Use as credenciais de login do Gmail. Fone de ouvido com microfone (opcional) Bloco de notas e caneta 		
Estrutura de Aula	Introdução Atividade Dirigida pela Professora Atividade Dirigida pelo Aluno Fechamento Dicas e Sugestões do Projeto	5 minutos 10 minutos 30 minutos 5 minutos 5 minutos	
<p>Observação: O design parcial do aplicativo está disponível para os alunos em Atividade do Aluno 6, para os alunos com mais dificuldade, basta dar uma visão geral do design e começar a explicar o conceito.</p>			
SESSÃO DE INTRODUÇÃO - 5 min			



A professora inicia a apresentação de slides  **dos slides 1 a 19.**
Consulte as anotações da oradora e siga as instruções de cada slide.



ATIVIDADE DA PROFESSORA

- 10 min

Professora Começa a Compartilhar a Tela

Dizer	Fazer
<p>Já tem um tempo que os humanos começaram a viajar para o espaço e eles até permanecem lá quando necessário. Não seria fascinante saber quem são todos eles e de qual país eles são?</p> <p>Então, hoje, iremos criar um aplicativo que vai obter as informações dos humanos que estão atualmente presentes no espaço e que buscará informações deles na NASA ou na Wikipedia.</p> <p>Faremos isso usando duas APIs diferentes e alguns outros componentes, como Listview (visualizadores de lista), Web Viewer (visualizadores da web), botões, variáveis, organizações verticais e organizações horizontais.</p>	
	<p>Abra a Atividade da Professora 1.</p>

Em cada etapa, teste o aplicativo em tempo real clicando em **Conectar** e selecionando **Assistente AI**



no topo do site. Em seguida, um código de barras irá aparecer. Digitalize o código de barras do MIT no telefone e mostre para o aluno, aproximando o telefone da câmera do laptop.

A GUI ou o design do aplicativo possui os seguintes recursos:

- **Screen 1:**
 - 3 botões:
 - Lista de Pessoas no Espaço (btnLista)
 - Informações das Pessoas no Espaço (btnInfo)
 - CLIQUE PARA SABER SOBRE AS PESSOAS NO ESPAÇO (bntClique)
 - Organizações Horizontal e Verticais
 - 1 Navegador Web



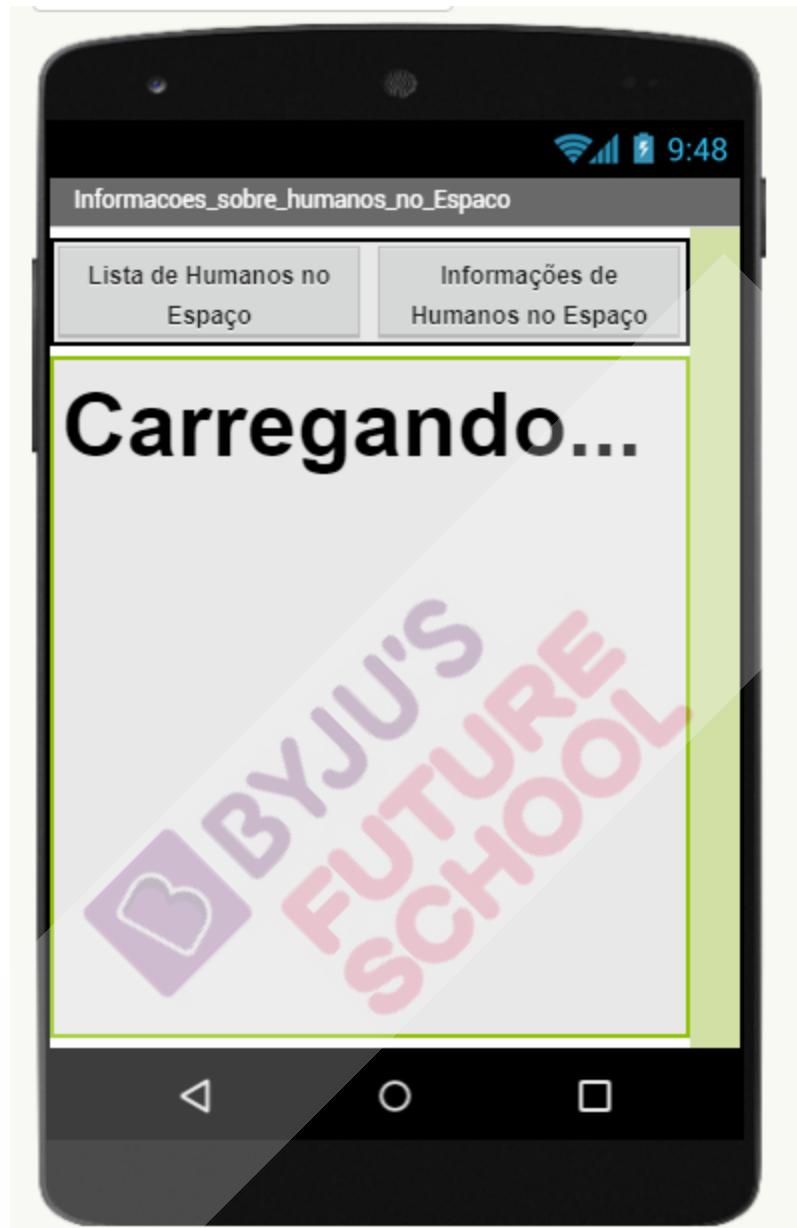
○ **Tela 2:**

→ 2 botões:

- Lista de Pessoas no Espaço
- Informações das Pessoas no Espaço

→ 1 Rótulo

→ 1 WebViewer



Hoje, seguiremos o seguinte roteiro para projetar e programar o aplicativo:

1. Adicione **Screen1** e personalize seus componentes. Este deve ter o nome dos humanos no espaço.

- **Adicione 2 botões** na parte superior que irão atuar como uma Barra de Navegação e personalize eles. A barra de navegação é usada para ir de uma tela para outra, alternando entre as telas. Deve haver uma Tela para a Navegação. Cada botão deve representar uma tela. Uma tela deve ter o nome dos humanos no espaço, e a segunda tela deve mostrar a página da Wikipedia de acordo com a pessoa selecionada na lista de os humanos no espaço. E adicione outro botão no centro da tela que dará a informação sobre o humano no espaço.
2. **Código** para Screen1.
 3. Adicione **Screen2** e personalize seus componentes. Esta tela deve aparecer quando um usuário clica em qualquer nome na Screen1 e deve conter as informações de um humano no espaço.
 4. **Programe** para a Screen2.

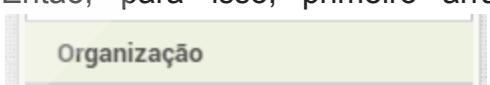
OBSERVAÇÃO: **Na Atividade do Aluno 5** estão as imagens que serão necessárias para essa aula. Os alunos devem abrir o MIT e começar um projeto novo. Para os alunos que tiverem mais dificuldade, na **Atividade do Aluno 6** tem o design parcial.

Vamos começar.

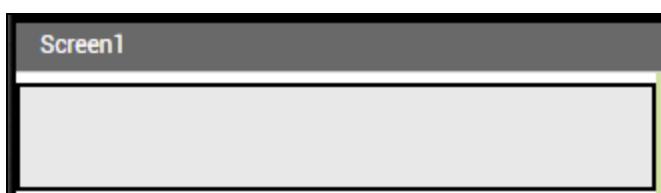
1. Personalizando os componentes da Screen1:

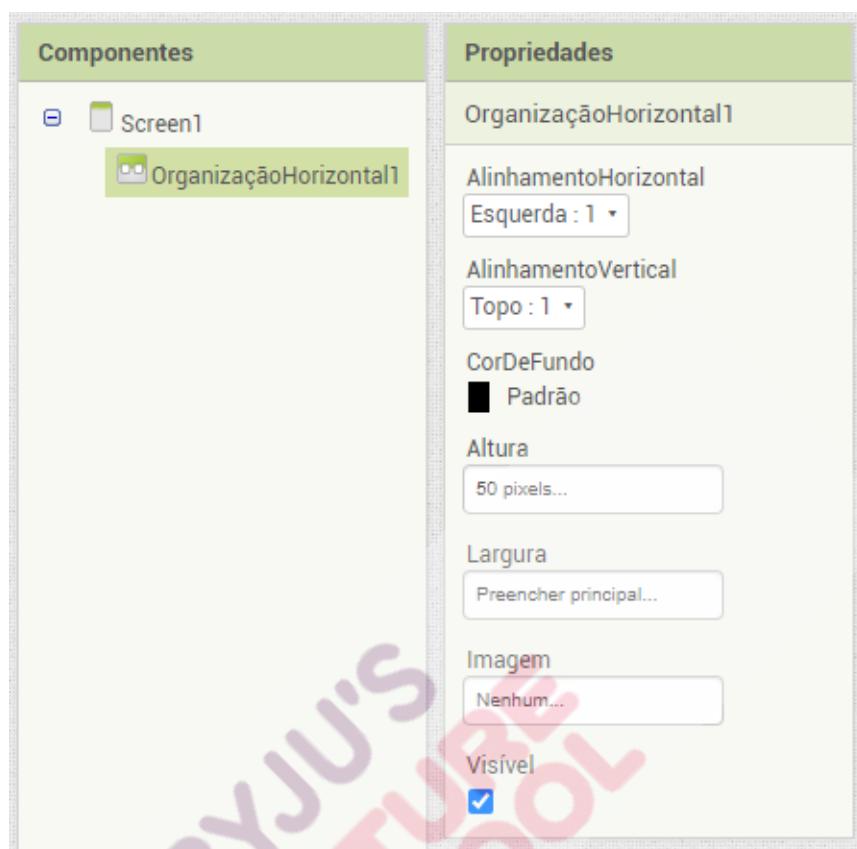
→ **Adicione uma organização horizontal e 2 botões:** Adicione 2 botões dentro de uma organização horizontal na parte superior que funcionarão como uma barra de navegação.

→ Então, para isso, primeiro arraste uma organização horizontal

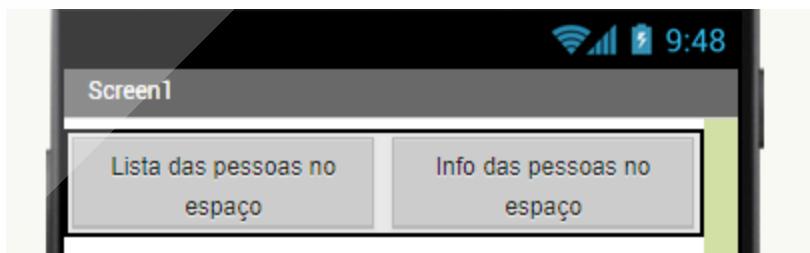


da seção Paleta e defina sua altura como 50px e largura como preencher principal.



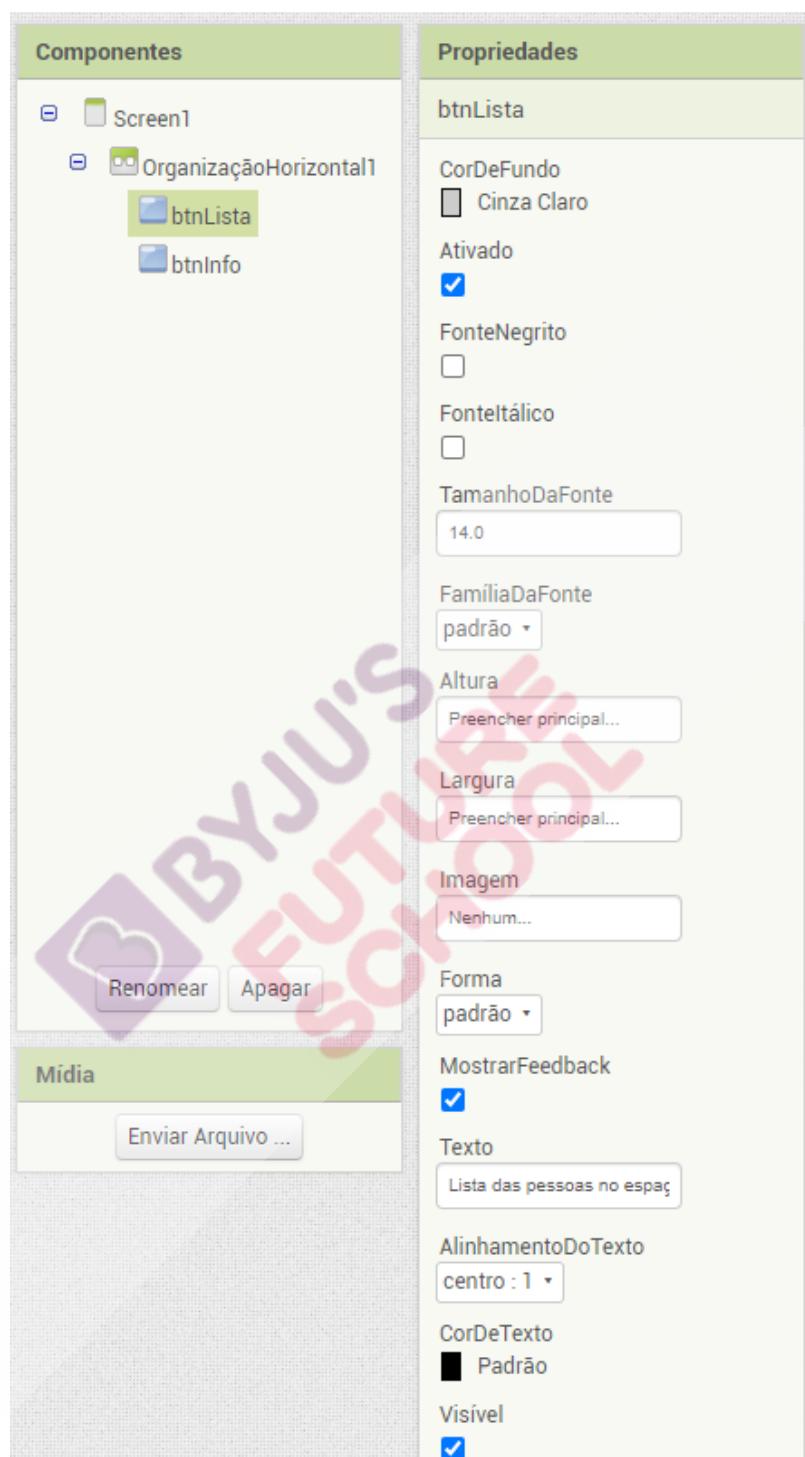


- Em seguida, arraste 2 botões de **Interface de Usuário** e solte-os dentro de **OrganizaçãoHorizontal1** e defina a altura e a largura dos botões de forma que ambos possam preencher toda a tela horizontalmente e defina uma cor de fundo para eles.

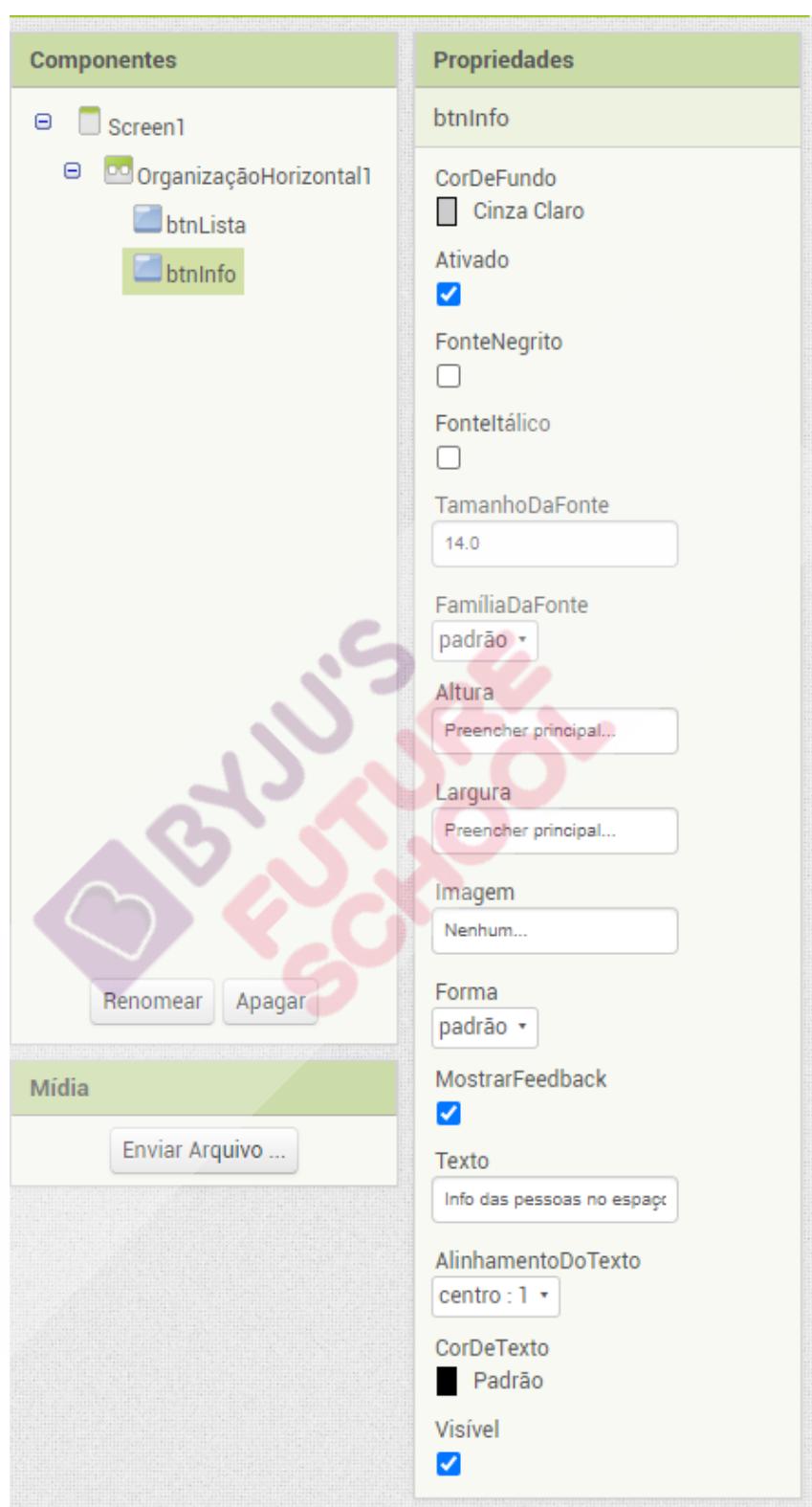


- Personalize os botões: Ambos os botões terão as mesmas propriedades de estilo, conforme o exemplo:

Botão 1: o que levará para a lista das pessoas (btnLista)



Botão 2: o que levará para as informações das pessoas (btnInfo)

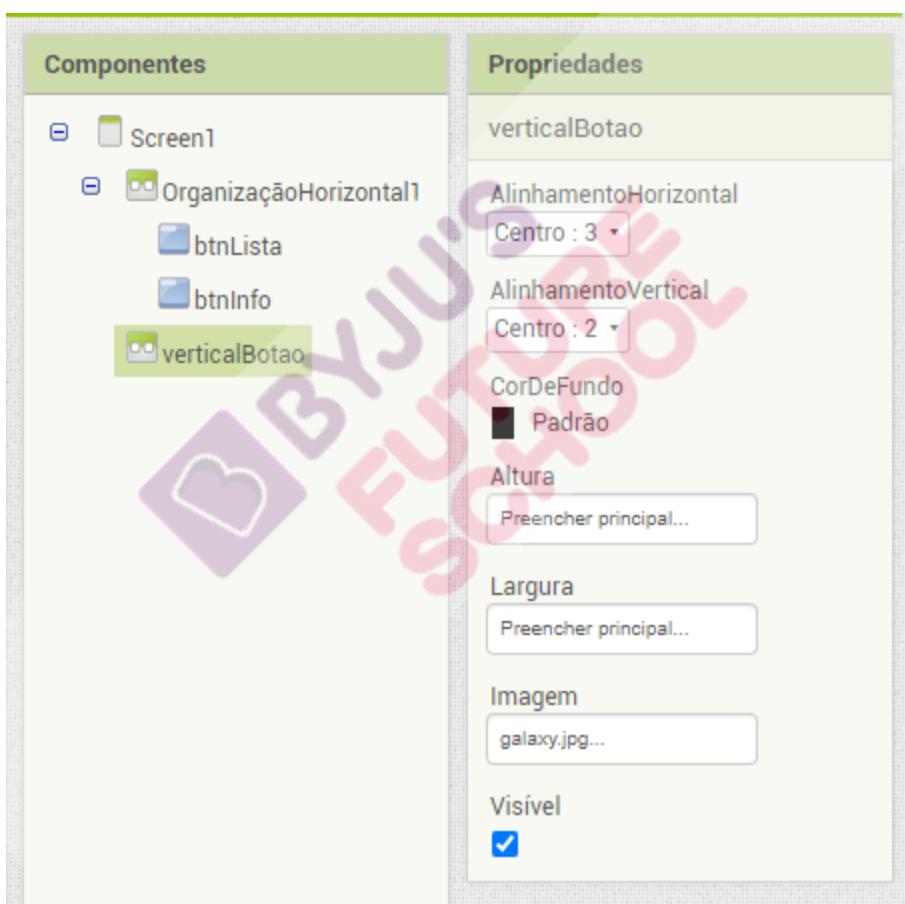


Observação: Para o aluno que está usando o design completo, as imagens já foram

enviadas para o aluno. Peça ao aluno para selecionar uma imagem para este componente.

- Em seguida, adicione uma Organização Vertical. Para isso, arraste um Organização Vertical de da Paleta.
- Vamos renomear a **OrganizaçãoVertical** para **verticalBotao** porque essa organização vertical vamos colocar o botão coletará o nome das pessoas no espaço.

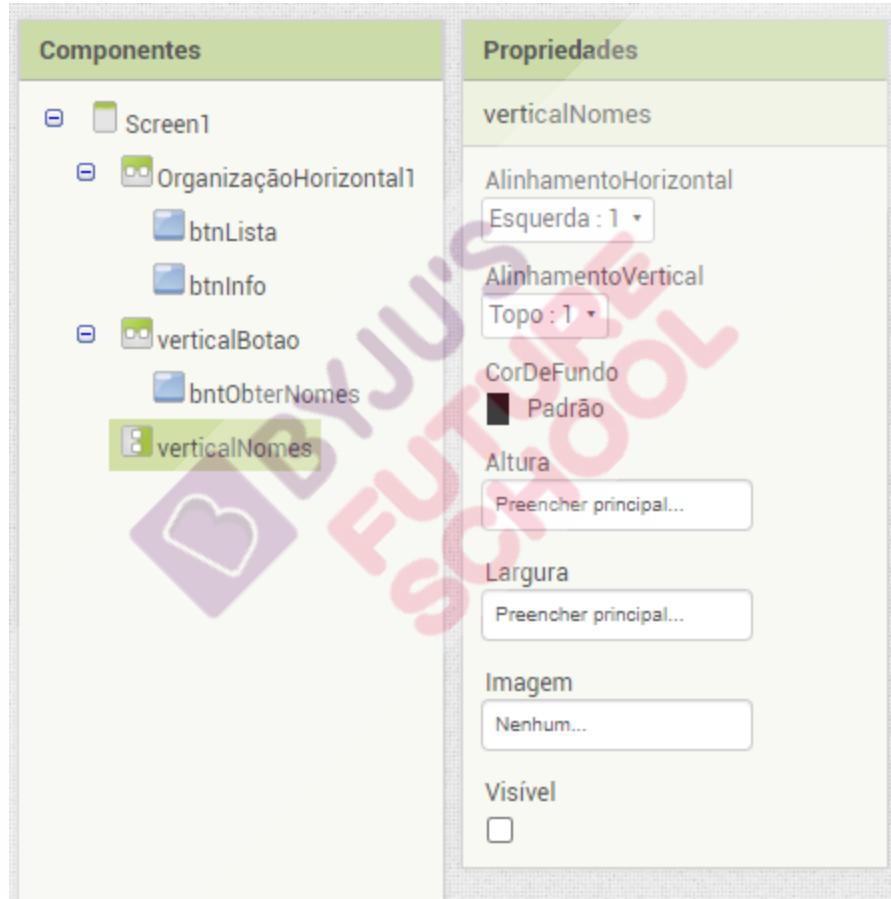
- Em seguida, defina o estilo das propriedades para:



- Agora, arraste um componente de botão, da seção de Interface do usuário e solte ele dentro da organização verticalBotao. Personalize e renomeie ele.
Agora, defina o estilo das propriedades deste botão:

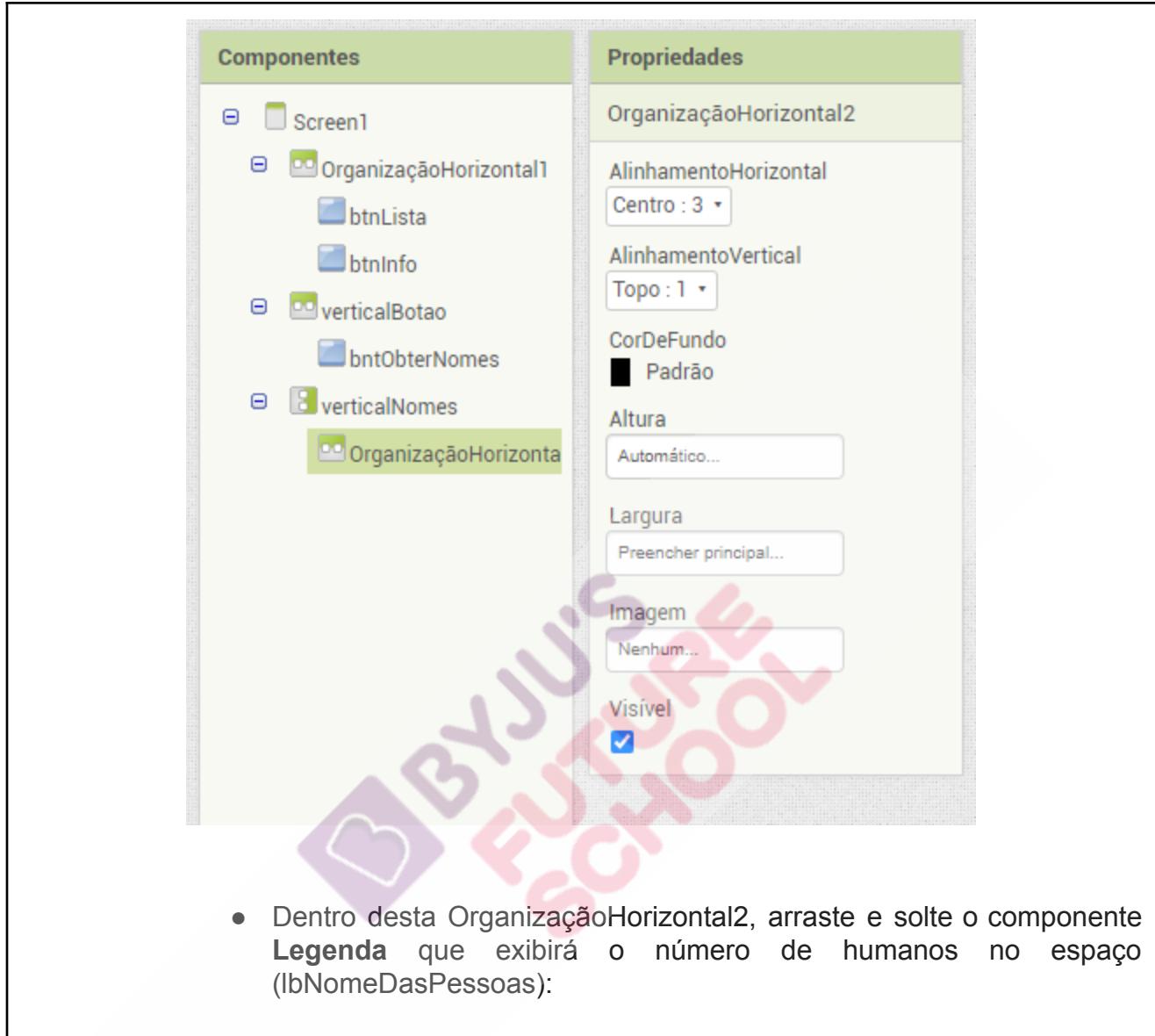
Componentes	Propriedades
<ul style="list-style-type: none"> Screen1 Organiza�oHorizontal1 <ul style="list-style-type: none"> btnLista btnInfo verticalBotao <ul style="list-style-type: none"> bntObterNomes 	<p>bntObterNomes</p> <p>CorDeFundo Nenhum</p> <p>Ativado <input checked="" type="checkbox"/></p> <p>FonteNegrito <input checked="" type="checkbox"/></p> <p>Fontelt�lico <input checked="" type="checkbox"/></p> <p>TamanhoDaFonte 26</p> <p>Fam�liaDaFonte mono-espaciada</p> <p>Altura Autom�tico...</p> <p>Largura Preencher principal...</p> <p>Imagen Nenhum...</p> <p>Forma padr�o</p> <p>MostrarFeedback <input checked="" type="checkbox"/></p> <p>Texto CLIQUE PARA SABER SOI</p> <p>AlinhamentoDoTexto centro : 1</p> <p>CorDeTexto Branco</p> <p>V�isivel <input checked="" type="checkbox"/></p>
<p>Midia</p> <p>galaxy.jpg</p> <p>Enviar Arquivo ...</p>	

- Agora vamos adicionar uma nova organização vertical, para listar o nome das pessoas no espaço
 - Primeiro, adicione uma Organização Vertical e renomeie para **verticalNomes**, porque essa organização vertical conterá os nomes dos humanos que estiveram no espaço.
 - Em seguida, defina as propriedades. Vamos definir sua visibilidade como 'falsa'. Iremos definir a visibilidade deste componente como 'verdadeira' quando o botão  bntObterNomes for clicado.



- Dentro da verticalNomes, adicione um Organização Horizontal para conter um componente de Legenda. (para conseguir adicionar a nova organização , deixe visível a verticalNomes e adicione a organização, depois que finalizar as adições só deixar como invisível novamente).





Componentes

- Screen1
- OrganizaçãoHorizontal1
 - btnLista
 - btnInfo
- verticalBotao
 - bntObterNomes
- verticalNomes
 - OrganizaçãoHorizontal2

Propriedades

OrganizaçãoHorizontal2

AlinhamentoHorizontal
Centro : 3

AlinhamentoVertical
Topo : 1

CorDeFundo
Padrão

Altura
Automático...

Largura
Preencher principal...

Imagen
Nenhum...

Visível

- Dentro desta OrganizaçãoHorizontal2, arraste e solte o componente **Legenda** que exibirá o número de humanos no espaço (lbNomeDasPessoas):

The screenshot shows a software interface for managing components and properties. On the left, a tree view titled "Componentes" lists the following structure:

- Screen1
 - OrganizaçãoHorizontal1
 - btnLista
 - btnInfo
 - verticalBotao
 - bntObterNomes
 - verticalNomes
 - OrganizaçãoHorizontal2
 - IbNomeDasPessoas

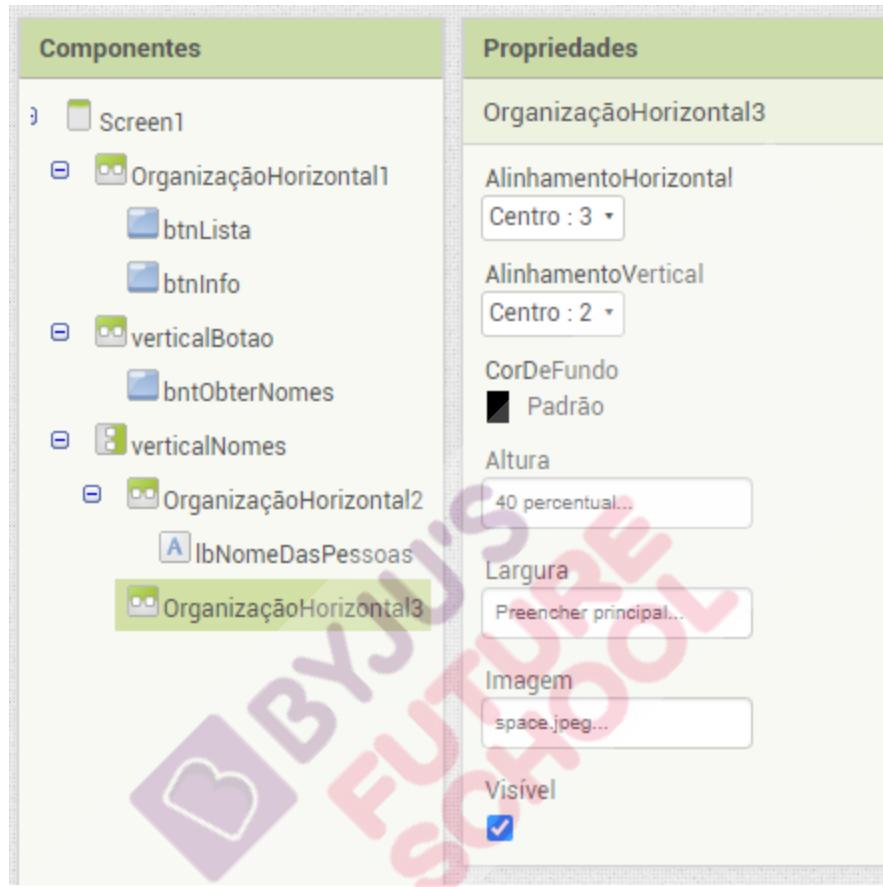
At the bottom of the component tree, there are "Renomear" and "Apagar" buttons.

On the right, a "Propriedades" (Properties) panel displays settings for the selected component, "IbNomeDasPessoas".

Propriedade	Valor
CorDeFundo	<input type="checkbox"/> Nenhum
FonteNegrito	<input checked="" type="checkbox"/>
FonteItálico	<input type="checkbox"/>
TamanhoDaFonte	14.0
FamíliaDaFonte	padrão ▾
HTMLFormat	<input type="checkbox"/>
TemMargens	<input checked="" type="checkbox"/>
Altura	Automático...
Largura	Automático...
Texto	Quantidade de Pessoas atualmente no Espaço
AlinhamentoDoTexto	esquerda : 0 ▾
CorDeTexto	<input type="checkbox"/> Padrão
Visível	<input checked="" type="checkbox"/>

Below the properties panel, there is a "Mídia" (Media) section showing an image file "galaxy.jpg" and a "Enviar Arquivo ..." button.

→ Agora adicione outra OrganizaçãoHorizontal dentro da organização **VerticalNomes** e personalize ela. Esta organização horizontal conterá o número total de pessoas atualmente presentes no espaço.



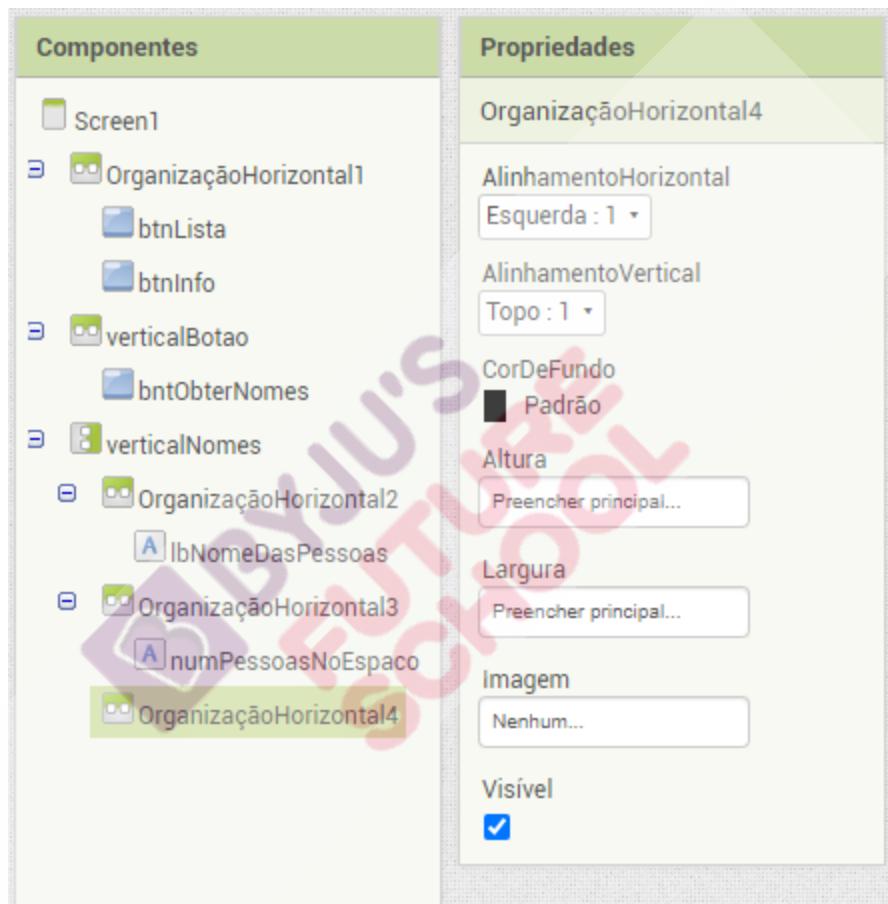
- Dentro dessa organização horizontal, arrastamos e soltamos o componente Legenda. Esta Legenda conterá o número total de pessoas atualmente presentes no espaço. Renomeie a Legenda para **numPessoasNoEspaco** e também personalize.

Defina as propriedades desta Legenda:

Vamos manter o campo Legenda vazio porque iremos atualizar a legenda com o número total de humanos atualmente presentes no espaço.

Componentes	Propriedades
Screen1	numPessoasNoEspaco
OrganizaçãoHorizontal1	CorDeFundo Nenhum
btnLista	FonteNegrito <input checked="" type="checkbox"/>
btnInfo	FonteItálico <input type="checkbox"/>
verticalBotao	TamanhoDaFonte 150
bntObterNomes	FamíliaDaFonte padrão
verticalNomes	HTMLFormat <input type="checkbox"/>
OrganizaçãoHorizontal2	TemMargens <input checked="" type="checkbox"/>
lbNomeDasPessoas	Altura Automático...
OrganizaçãoHorizontal3	Largura Preencher principal...
numPessoasNoEspaco	Texto
	AlinhamentoDoTexto centro : 1
	CorDeTexto Branco
	Visível <input checked="" type="checkbox"/>

→ Agora adicione outra OrganizaçãoHorizontal dentro da organização **VerticalNomes** e estilize ela. Esta organização horizontal manterá a lista de todos os humanos atualmente presentes no espaço em um visualizador de lista.



Deixando a organização **verticalBotao** invisível, essa será a aparência da organização **verticalNomes**



- Agora, vamos adicionar um Visualizador de Listas. Este Visualizador de Listas conterá os nomes das pessoas.

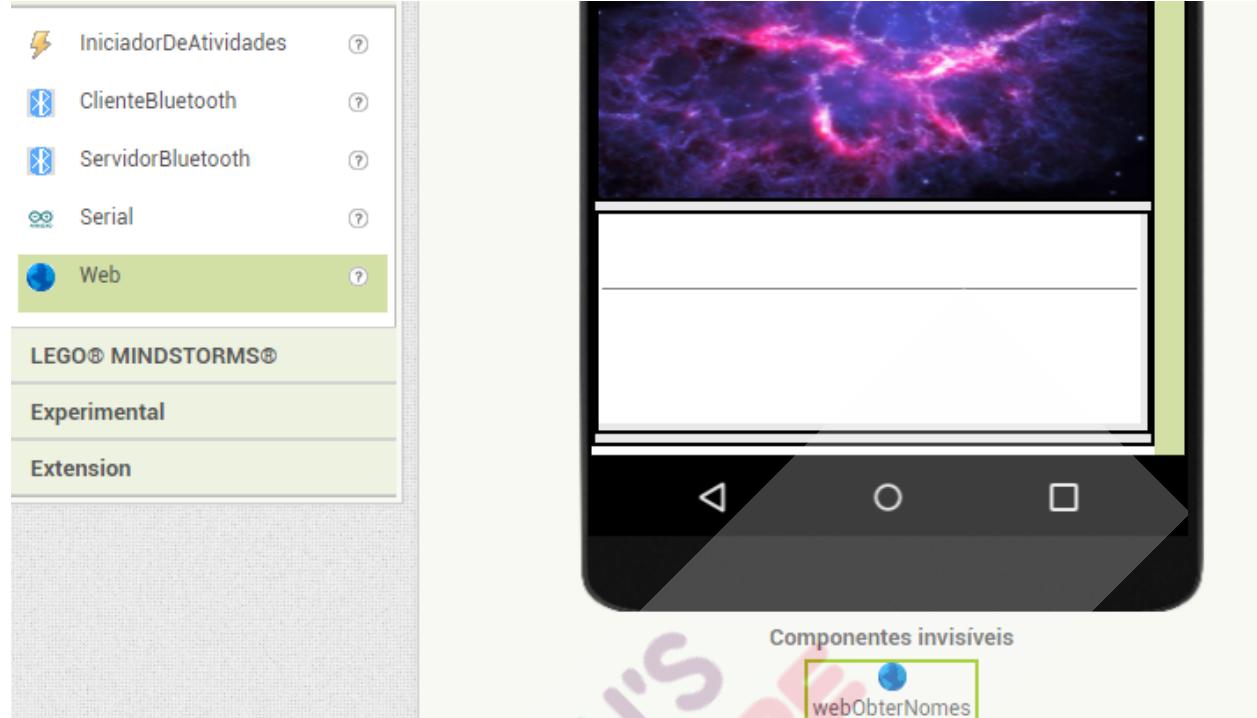
Componentes
Screen1 OrganizaçãoHorizontal1 btnLista btnInfo verticalBotao btntObterNomes verticalNomes OrganizaçãoHorizontal2 lbNomeDasPessoas OrganizaçãoHorizontal3 numPessoasNoEspaco OrganizaçãoHorizontal4 VisualizadorDeListas1

Propriedades
VisualizadorDeListas1 CorDeFundo <input type="checkbox"/> Branco CadeiaDeElementos FontSizeDetail 14.0 FamíliaDaFonte padrão ▾ FontTypefaceDetail padrão ▾ Altura Preencher principal... Largura Preencher principal... ImageHeight 200 ImageWidth 200

ListData
Click to Add/Delete Data ListViewLayout >MainText ▾ Orientation vertical ▾ Seleção CorDeSeleção <input checked="" type="checkbox"/> Cinza MostrarBarraDeFiltragem <input type="checkbox"/> CorDeTexto <input checked="" type="checkbox"/> Preto TextColorDetail <input type="checkbox"/> Padrão TamanhoDoTexto 22 Visível <input checked="" type="checkbox"/>

BYJU'S
FUTURE
SCHOOL

- Agora, adicionaremos uma API da Web arrastando o componente Web da seção Conectividade e soltando em qualquer parte do celular
- Vamos dar o nome de **webObterNomes**, pois essa API nos dará todos os nomes de humanos no espaço.



The image shows the Scratch interface. On the left, the script editor lists the following scripts:

- IniciadorDeAtividades
- ClienteBluetooth
- ServidorBluetooth
- Serial
- Web

The "Web" script is highlighted with a green background. Below the script editor, there are three categories: "LEGO® MINDSTORMS®", "Experimental", and "Extension".

The stage background is a purple nebula. At the bottom, there are three control buttons: a triangle pointing left, a circle, and a square.

Text overlay on the right side of the stage area:

Componentes invisíveis

webObterNomes

Excelente! Então fechamos a Screen1.

OBS: não esqueça de tornar visível a organização **verticalBotao** e invisível a **verticalNomes**



2. Agora vamos começar a programar a Screen1:

→ Primeiro, initialize 3 variáveis:

inicializar global nomes para criar lista vazia

inicializar global listaDeObjetos para criar lista vazia

inicializar global nomeSelecionado para " "

1. **Em primeiro lugar**, vamos iniciar uma variável **nomes** que conterá todos os **nomes das pessoas atualmente presentes no espaço**. Estamos fazendo com que essa variável seja uma lista vazia. Como sabemos, a lista é um tipo de dado que pode conter muitos dados dentro dela. São nas listas que podemos armazenar muitos valores dentro de uma variável, em vez de ter várias variáveis para vários valores.
2. **Em segundo lugar**, vamos iniciar uma variável **listaDeObjetos** que conterá todos os **objetos que recebemos da API**. Obteremos os nomes das pessoas presentes no espaço da API na forma de objetos. A API usa objetos para enviar dados. Dentro dos objetos, temos os valores (nome das pessoas).

Por exemplo:

name: john - aqui o objeto é *name* e seu valor é *john*.

Outro exemplo pode ser:

```
pessoa
{
    "firstname": john
    sobrenome: cena
    idade: 37
}
```

Aqui, o objeto é 'pessoa'.

Estamos transformando essa variável em uma lista vazia.

Porque da API, obteremos muitos objetos. Para armazenar muitos objetos dentro de uma variável, estamos usando List.

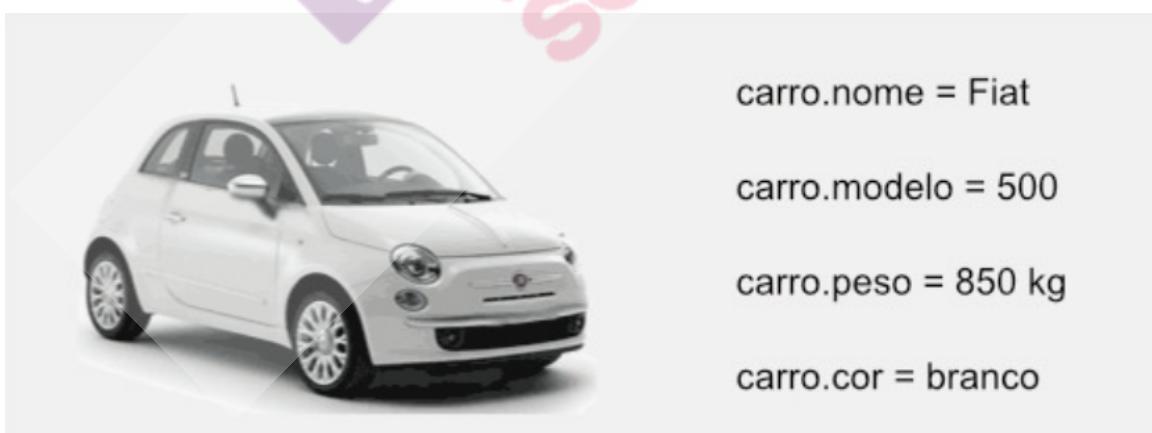
Você obterá o bloco de código de lista vazia em:



3. Em terceiro lugar, faremos outra variável com uma string vazia. A variável **nomeSelecionado** será usada para armazenar o item de lista selecionado do Visualizador de Listas.

A Lista é apenas uma coleção de valores, enquanto Object é uma coleção de Propriedades e Valores:

**Lista {valor1, valor2, valor3, valor4} igual a Matrizes
Objeto {Propriedade1: Valor, Propriedade2: Valor,
Propriedade3: Valor}**



Objetos e seus valores são como:

```
carro{  
nome: fiat,  
modelo: 500,
```

```

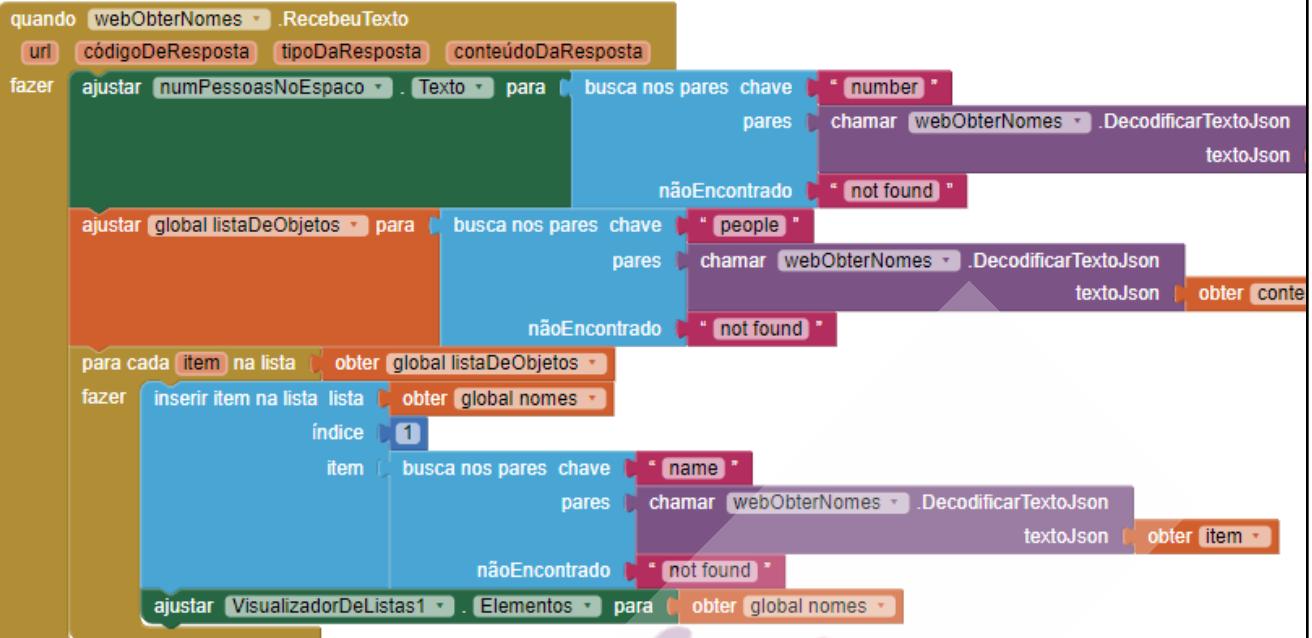
    peso: 850 kg,
    cor branca
}
  
```

O carro é o objeto principal aqui e tem:

- nome que tem o valor *fiat*
- modelo que tem o valor *500*
- peso que tem o valor *850kg*
- cor que tem o valor *branco*

→ Agora, teremos uma função de botão de clicar (**bntobterNomes**). Isso irá ocultar a organização **verticalBotao** porque esta organização vertical não é mais necessária, e mostra a organização **verticalNomes**. Esta organização vertical *mostrará todas as pessoas que estão presentes no espaço* e atualizará o número na Legenda.





- Primeiro, estamos ocultando a organização **verticalBotao**, porque uma vez que o botão é clicado, não precisamos mais dessa organização vertical.

quando bntObterNomes Clique

fazer ajustar verticalBotao Visível para falso

- Então, estamos mostrando a organização **verticalNomes** porque esta é a principal organização vertical que conterá todos os nomes de humanos no espaço da API.

quando bntObterNomes Clique

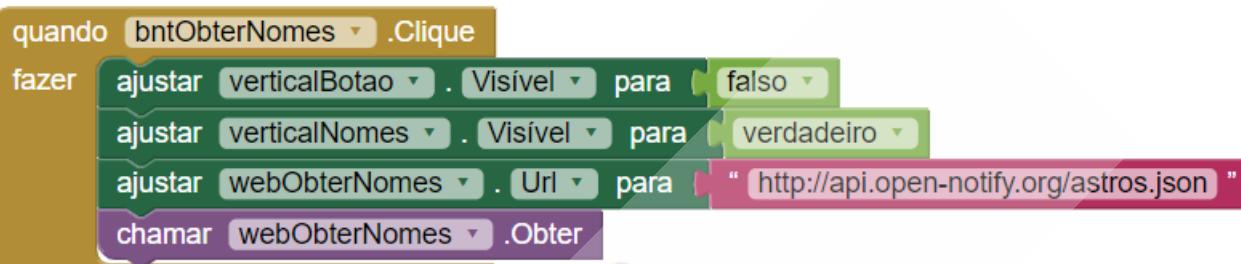
fazer ajustar verticalBotao Visível para falso

ajustar verticalNomes Visível para verdadeiro

- Em seguida, definimos a URL da API.



- Usamos a função OBTER para obter os dados da API.



Você pode encontrar o bloco de código

chamar webObterNomes .Obter em 

Quando **chamar webObterNomes .Obter** é chamado, ele aciona o evento



- Portanto, dentro de **webObterNome.RecebeuTexto**, escreva o código para obter os dados da API.

Os dados da API estão no formato JSON. Portanto, usamos o seguinte bloco de código para obter os dados que estão em JSON e decodificá-los.

chamar webObterNomes .DecodificarTextoJson
textoJson

que está em  webObterNomes

O formato que API está enviando dados é:



Portanto, temos 3 objetos - *people*, *number* e *message*.

1. **People** contém uma lista de objetos.

Dentro de cada objeto, existem dois objetos com valores. Um objeto é '**name**'. Este tem o nome das pessoas que estão presentes no espaço. O segundo objeto fornece os nomes das naves espaciais que não são necessárias para este aplicativo.

2. **Number** contém o número total de pessoas que estiveram no espaço.
3. **Message** contém o status da API.

Por exemplo: Se quisermos acessar o primeiro nome:

```
{  
  - people: [  
    - {  
      name: "Christina Koch",  
      craft: "ISS"  
    },  
    - {  
      name: "Alexander Skvortsov"  
      craft: "ISS"  
    },  
    - {  
      name: "Luca Parmitano",  
      craft: "ISS"  
    },  
    - {  
      name: "Andrew Morgan",  
      craft: "ISS"  
    },  
    - {  
      name: "Oleg Skripochka",  
      craft: "ISS"  
    },  
    - {  
      name: "Jessica Meir",  
      craft: "ISS"  
    }  
  ],  
  number: 6,  
  message: "success"  
}
```

people[0].name

Código para obter o bloco JSON **JsonTextDecode**:

Como os dados da API estão em um par de valores-chave, obteremos o bloco que fornece o valor de uma chave específica na lista:

Controle

Lógica

Matemática

Texto

Listas

Dictionaries

Cores

Variáveis

Procedimentos

busca nos pares chave
pares
nãoEncontrado "not found"

- Primeiro, atualizamos o Legenda (**numPessoasNoEspaco**) com o número total de pessoas presentes no espaço, que obtemos da API.



Isso significa que ele irá se referir ao **número** do objeto e obterá o valor presente no objeto **number** da API.

Resultado da Legenda **numPessoasNoEspaco**:



- Agora, colocaremos todos os dados do objeto **people** na variável **listaDeObjetos**.



Portanto, agora a variável **listaDeObjetos** terá todos os objetos que estão dentro do objeto **people** (que são os nomes das pessoas no espaço).

- Agora, faremos um loop por esta variável **listaDeObjetos** porque há muitos nomes dentro da variável **listaDeObjetos**.

Conforme estamos em loop, colocamos o valor do nome do objeto um por um dentro da variável **names** na **listaDeObjetos**.



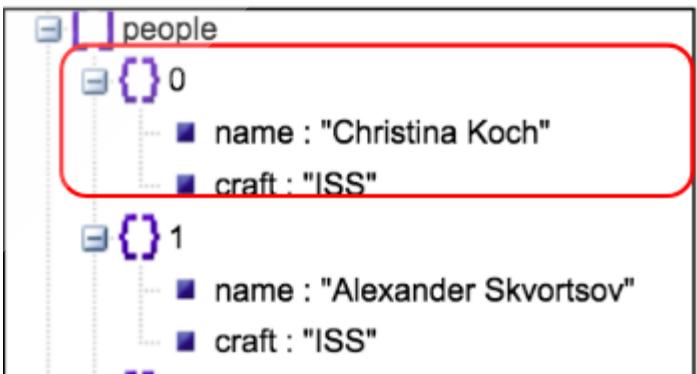
Usaremos um loop **for each** que executa o código dado para cada elemento da lista. E o elemento da lista em loop é armazenado na variável **item**.

Dessa forma, podemos ler cada elemento da variável **listaDeObjetos** e armazená-lo na variável **item**. E uma vez que o loop atinge o final da lista, o loop **for each** irá parar de funcionar.

O bloco de código a seguir fornecerá o nome de cada pessoa do objeto de **people** que é o nome da pessoa em 0º objeto, 1º objeto, 2º objeto e assim por diante... que serão os valores armazenados na variável **item**.



Por exemplo:

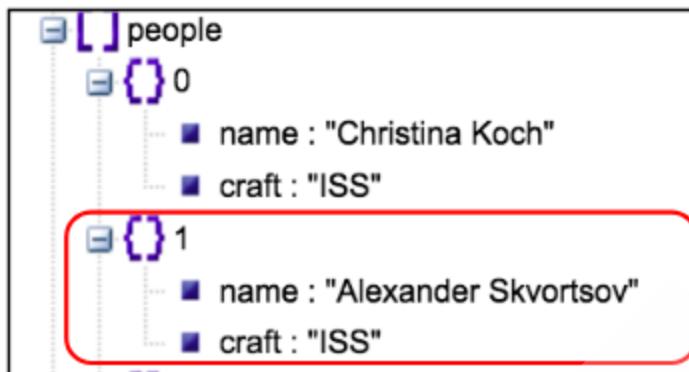


```

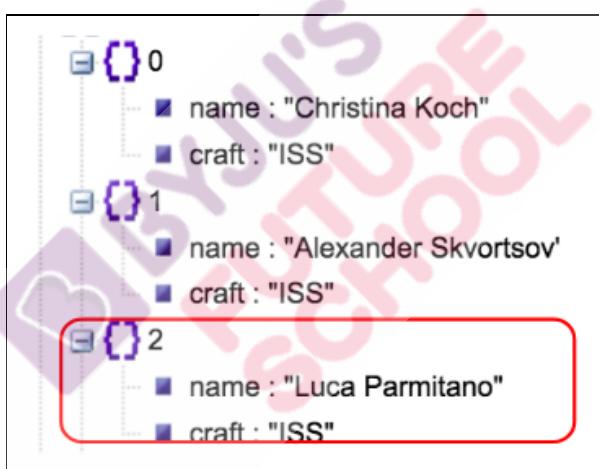
people
[0]
  name : "Christina Koch"
  craft : "ISS"
[1]
  name : "Alexander Skvortsov"
  craft : "ISS"

```

- Primeiro, ele lerá o primeiro objeto e obterá o valor que é o nome do objeto e o armazenará na variável de **nomes** que é uma lista. Portanto, a lista armazenada na variável **name** tem 'Christina Koch'.



- Agora, ele lerá o segundo objeto e obterá o valor que é o **name** do objeto e o armazenará na lista de variáveis **nomes**. Portanto, a lista armazenada na variável **nome** tem 'Christina Koch' e 'Alexander Skvortsov'.



- Agora, ele lerá o terceiro objeto e obterá o valor que é o nome do objeto e o armazenará nas variáveis **nomes**. Portanto, a lista armazenada nas variáveis **nomes** agora contém 'Christina Koch', 'Alexander Skvortsov' e 'Luca Parmitano'.

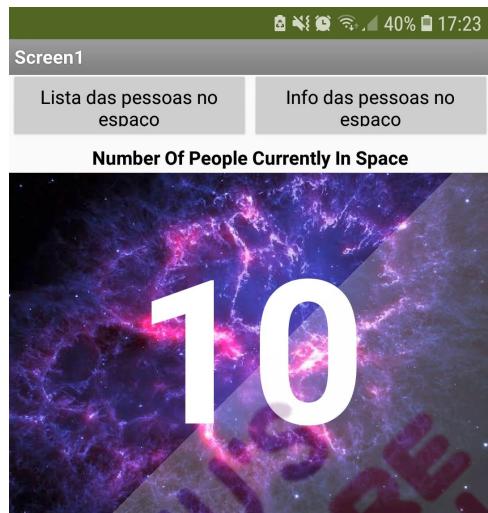
E o loop termina assim que todos os dados da variável **lista_de_objetos** forem lidos.

Agora, a variável **nomes** contém todos os nomes da API.

- Agora, colocaremos as variáveis **nomes** dentro do Visualizador de Listas. Portanto, todos os nomes de humanos presentes no espaço são exibidos em formato de lista.

ajustar VisualizadorDeListas1 para obter global nomes

Resultado do Visualizador de Lista:



Matthias Maurer

Kayla Barron

Tom Marshburn

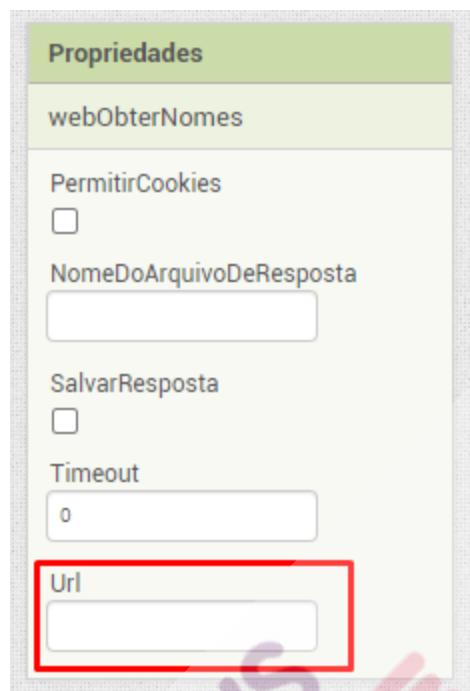
Raja Chari

Ye Guangfu

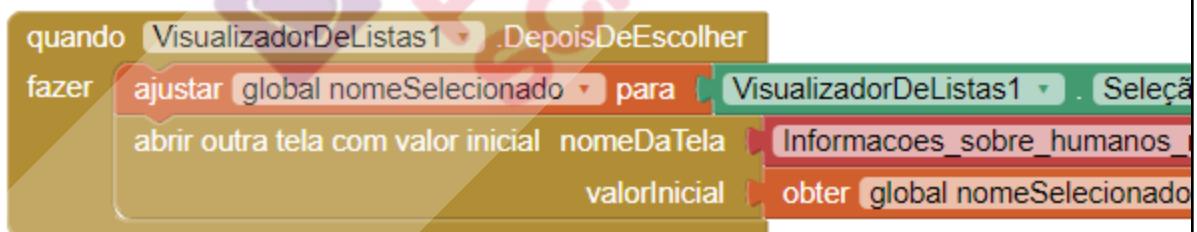
Wang Yaping

Excelente! Conseguimos buscar e exibir os nomes dos humanos presentes no espaço, bem como o número total de pessoas.

Observação: Se, por algum motivo, a saída não for exibida, coloque o link da API na caixa de texto da URL da visualização da web na seção de design:



- Agora adicione uma funcionalidade: Quando esta visualização de lista é clicada, ela deve abrir a respectiva página de informações daquela pessoa. Para isso, adicione uma função onclick para o item da lista que quer dizer: o que deve acontecer quando um usuário clica em qualquer nome da lista?

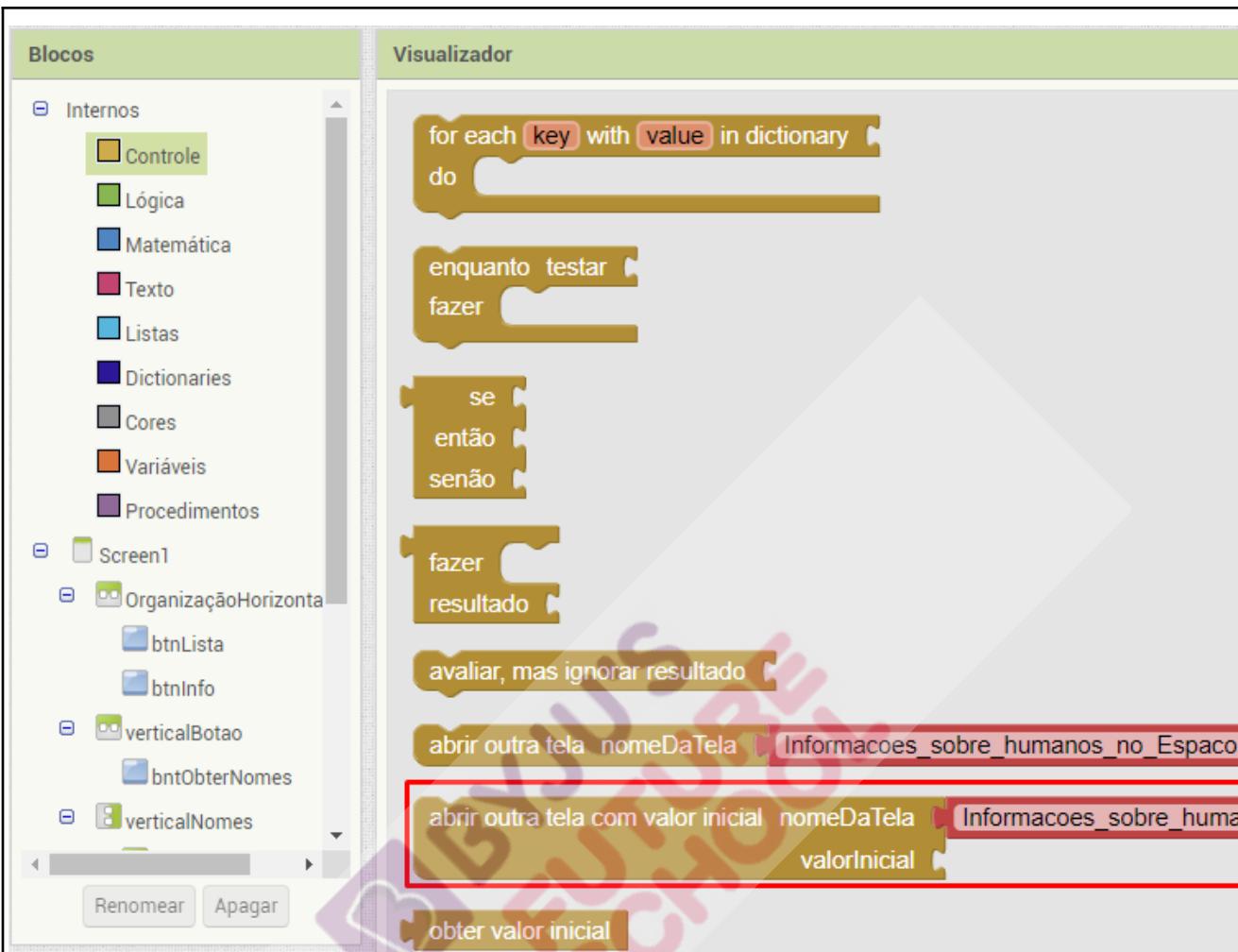


Assim, quando o item na lista é clicado, o nome clicado da pessoa é armazenado na variável **nomeSelecionado** e ele navega para outra tela (**Informacoes_sobre_humanos_no_Espaco**) que iremos criar agora.

Então, basicamente, quando clicamos no nome da pessoa, ela vai navegar para outra tela e mostrar suas informações.

E enquanto navegamos para outra tela, queremos usar a variável **nomeSelecionado** em outra tela também.

Portanto, vamos usar o bloco de código "Abrir outra tela com valor inicial" a partir de:



E obter a variável nomeSelecionado.

Em seguida, ao clicar no **btnInfo**, abrir a tela **Informacoes_sobre_humanos_no_Espaco**



3. Adicionar uma nova tela:

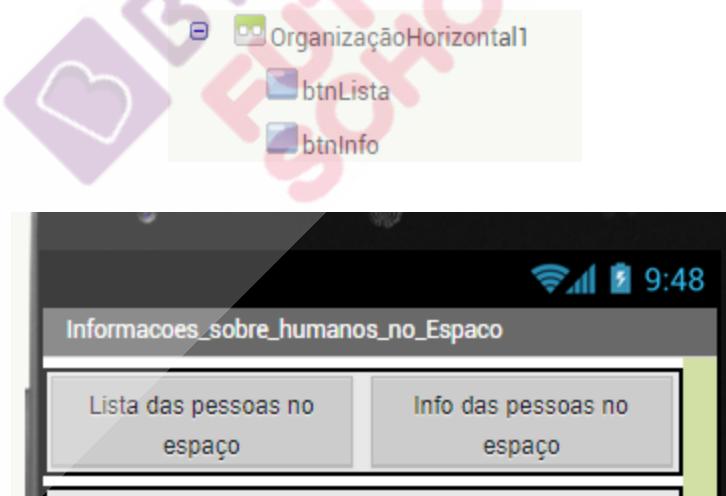
Vamos adicionar uma nova tela.

Renomeie a tela e para **Informacoes_sobre_humanos_no_Espaco**

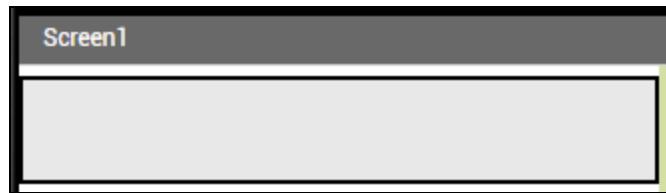


Observação: Se você quiser dar um nome de tela, só poderá dar um nome de tela ao adicionar uma nova tela. Uma vez que a tela foi criada, você não pode renomear a tela. Você terá que criar uma nova tela com um novo nome então.

4. Assim como na Screen1 vamos adicionar 2 botões dentro de um organização horizontal na parte superior que funcionará como uma barra de navegação:



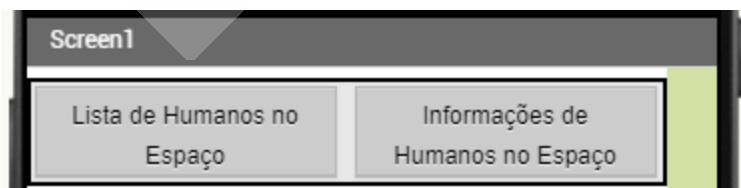
Portanto, para isso, primeiro, adicione o organização horizontal de 50 px de altura e largura que preenche toda a tela:



Defina o estilo de OrganizaçãoHorizontal1, da seguinte maneira:

Componentes	Propriedades
Informações_sobre_humanos_n	OrganizaçãoHorizontal1
OrganizaçãoHorizontal1	AlinhamentoHorizontal Esquerda : 1 ▾
btnLista	AlinhamentoVertical Topo : 1 ▾
btnInfo	CorDeFundo Padrão
OrganizaçãoVertical1	Altura 50 pixels...
Legenda1	Largura Preencher principal...
NavegadorWeb1	Imagen Nenhum...
	Visível <input checked="" type="checkbox"/>

Em seguida, adicione 2 botões, defina a altura e a largura do botão de forma que ambos os botões preencham a tela inteira e forneçam a cor de fundo ao botão.



Personalize os botões: da mesma forma que foi feita na Screen 1

Componentes

- Informacoes_sobre_humanos_n
- OrganizaçãoHorizontal1
 - btnLista
 - btnInfo
- OrganizaçãoVertical1
 - Legenda1
 - NavegadorWeb1



Mídia

- galaxy.jpg
- space.jpeg

[Enviar Arquivo ...](#)

Renomear **Apagar**

Propriedades

btnLista

CorDeFundo
 Cinza Claro

Ativado

FonteNegrito

FonteItálico

TamanhoDaFonte
14.0

FamíliaDaFonte
padrão ▾

Altura
Preencher principal...

Largura
Preencher principal...

Imagen
Nenhum...

Forma
padrão ▾

MostrarFeedback

Texto
Lista das pessoas no espaço

AlinhamentoDoTexto
centro : 1 ▾

CorDeTexto
 Padrão

Visível

Componentes

- Informacoes_sobre_humanos_n
- OrganizaçãoHorizontal1
 - btnLista
 - btnInfo
- OrganizaçãoVertical1
 - Legenda1
 - NavegadorWeb1

Mídia

- galaxy.jpg
- space.jpeg

Propriedades

btnInfo

CorDeFundo
Cinza Claro

Ativado

FonteNegrito

FonteItálico

TamanhoDaFonte
14.0

FamíliaDaFonte
padrão

Altura
Preencher principal...

Largura
Preencher principal...

Imagem
Nenhum...

Forma
padrão

MostrarFeedback

Texto
Info das pessoas no espaço

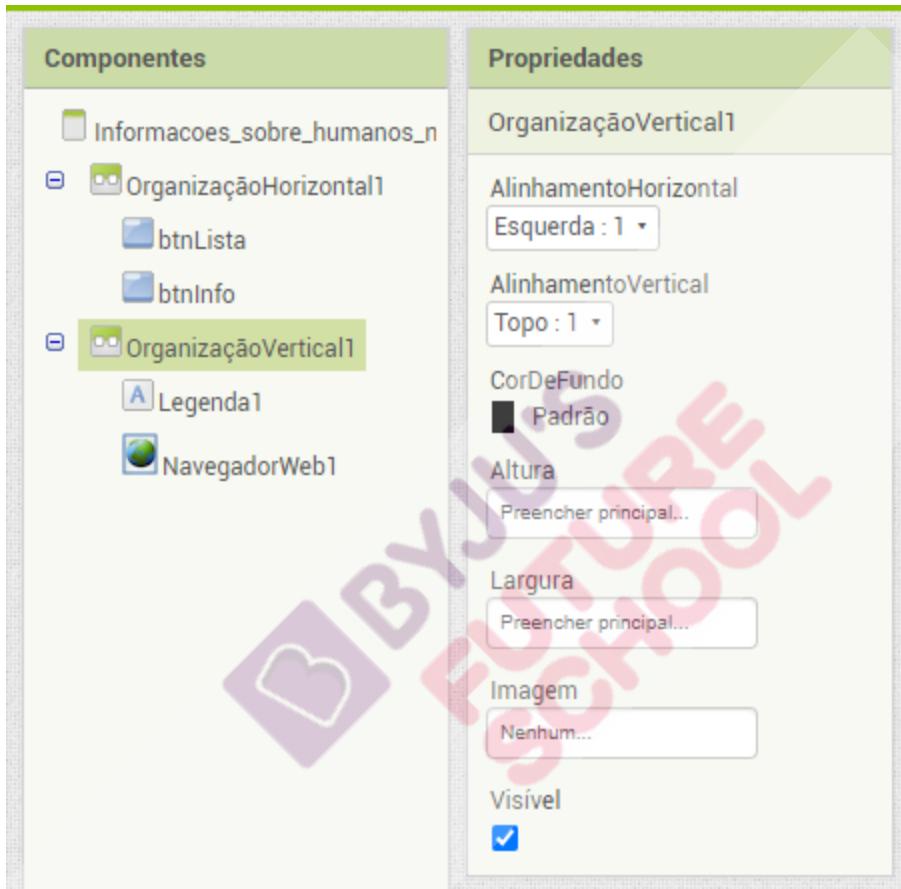
AlinhamentoDoTexto
centro : 1

CorDeTexto
Padrão

Visível

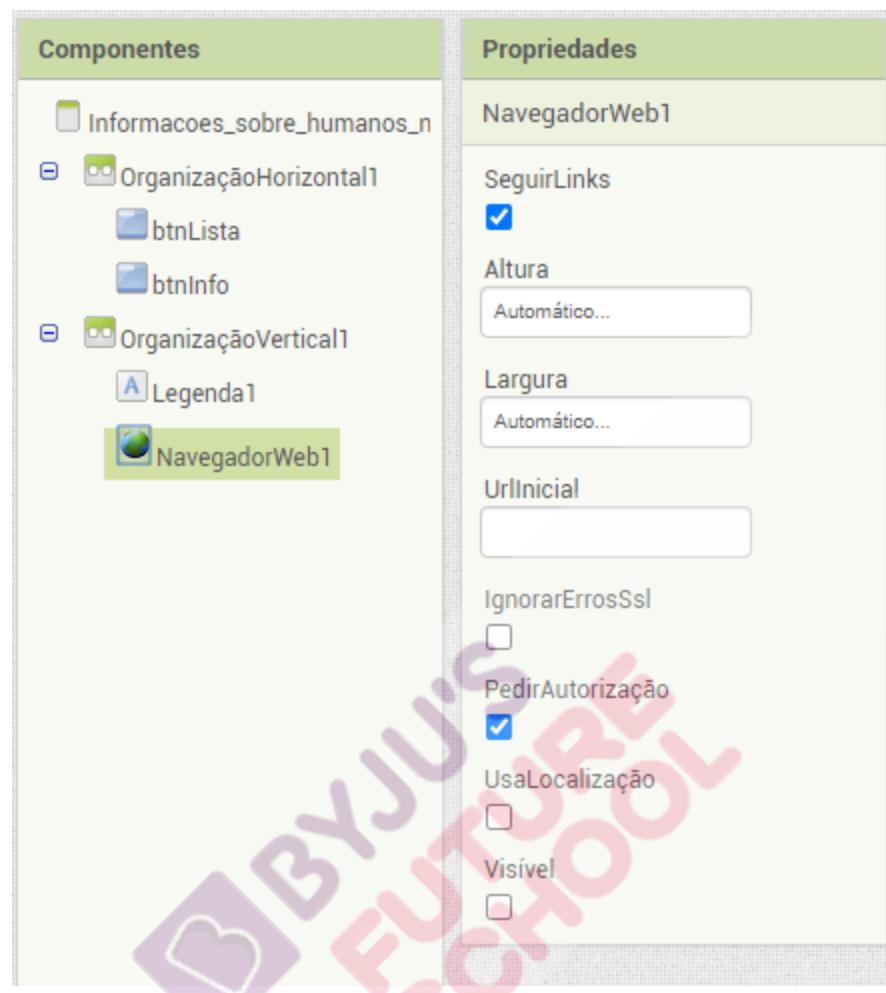
Agora, adicione uma organização vertical dentro dela coloque uma etiqueta que mostrará “Carregando” e um Navegador Web.

Personalize a organização vertical e a etiqueta, da seguinte forma:



Personalize a Legenda e o Navegador Web, da seguinte forma:

Componentes	Propriedades
<ul style="list-style-type: none">☐ Informacoes_sobre_humanos_n⊖  OrganizaçãoHorizontal1<ul style="list-style-type: none">☐ btnLista☐ btnInfo⊖  OrganizaçãoVertical1<ul style="list-style-type: none"> Legenda1 NavegadorWeb1	<p>Legenda1</p> <p>CorDeFundo <input type="checkbox"/> Nenhum</p> <p>FonteNegrito <input checked="" type="checkbox"/></p> <p>FonteItálico <input type="checkbox"/></p> <p>TamanhoDaFonte 50</p> <p>FamíliaDaFonte padrão ▾</p> <p>HTMLFormat <input type="checkbox"/></p> <p>TemMargens <input checked="" type="checkbox"/></p> <p>Altura Automático...</p> <p>Largura Automático...</p> <p>Texto Carregando... ▾</p> <p>AlinhamentoDoTexto centro : 1 ▾</p> <p>CorDeTexto  Padrão</p> <p>Visível <input checked="" type="checkbox"/></p>



4. Código para Screen2:

- Primeiro, vamos configurar que ao clicar nos botões para navegar entre as telas
- `btnLista` navegue até “Screen1” e ao clicar no `btnInfo` navegue até “`Informacoes_sobre_humanos_no_Espaco`”.

quando **btnLista** .Clique
 fazer **abrir outra tela nomeDaTela** **Screen1**

quando **btnInfo** .Clique
 fazer **abrir outra tela nomeDaTela** **Informacoes_sobre_humanos_no_Espaco**

- Em seguida, defina uma URL variável:

inicializar global url para " "

- Então, quando a tela é carregada, estamos ocultando o Navegador Web, porque mostraremos o Navegador Web quando ele for carregado.

quando **Info_Of_Human_In_Space** .Inicializar
 fazer **ajustar NavegadorWeb1 Visivel para falso**

- Primeiro defina a variável da URL para 'juntar'. Temos a API seguida pela variável **nomeSelecionado** (esta variável é definida em **screen1** quando clicamos no nome do item da lista).

ajustar global url para juntar " https://mahdihat791.github.io/v2/Space_data/index.html?name= " obter valor inicial

Por exemplo: Na **Screen1**, se você clicar em: **Christina Koch**, então a variável **nomeSelecionado** será **Christina Koch** e irá rolar para esta tela, e definir a URL da variável para https://mahdihat791.github.io/v2/Space_data/index.html?name=Christina_Koch

OU

Na Screen1, se você clicar em: Oleg Skripchka, a variável **nomeSelecionado** será **Oleg_Skripochka** e ela irá rolar para esta tela, e

definir a URL variável para

https://mahdihat791.github.io/v2/Space_data/index.html?name=Oleg%20Skrípochka

O link https://mahdihat791.github.io/v2/Space_data/index.html?name= está disponível em **Atividade do Aluno 3**

3. Em seguida, passamos este URL para o NavegadorWeb:



4. Ocultar etiqueta com texto *Carregando ...*



5. Mostre a visualização da web.



A saída será a respectiva página da pessoa clicada na visualização de lista.

Mas haverá alguns casos em que você não obterá a página real da Wikipedia dessa pessoa. A razão para isso pode ser:

- Existem poucas pessoas com esse nome.
OU
- A API da Wikipedia não suporta a forma como o nome é passado / escrito.

Nesses casos, você obterá a saída na segunda tela assim:

Selecione um nome

Mas tudo bem. Clique no nome de outra pessoa e verifique.

A professora interrompe o Compartilhamento de Tela

 ATIVIDADE DO ALUNO - 30 min.

- Agora é a sua vez de compartilhar sua tela.

- Peça ao aluno que pressione a tecla ESC para voltar ao painel.
- Oriente o aluno a iniciar o Compartilhamento de Tela.
- A professora entra em Tela Cheia.

O aluno inicia o Compartilhamento de Tela

Agora você terá que aplicar os aprendizados e criar o aplicativo.

[Atividade do Aluno 1: MIT APP INVENTOR](#)

Observação: O design completo do aplicativo está disponível para os alunos em [Atividade do Aluno 6](#), para os alunos com mais dificuldade, basta dar uma visão geral do design e começar a explicar o

[Atividade do Aluno 2: API HUMANOS NO ESPAÇO](#)

[Atividade do Aluno 3: API DE](#)

conceito.

Você tem que codificar para Screen1 e a tela **Informacoes_sobre_humanos_no_Espaco**

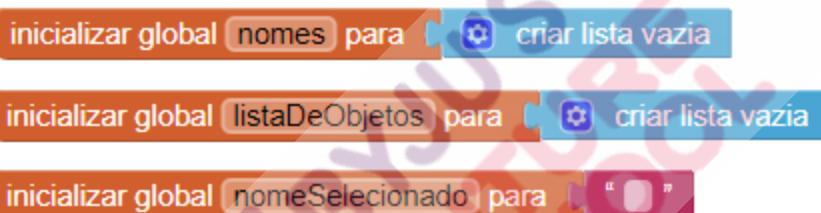
INFORMAÇÕES SOBRE HUMANOS NO ESPAÇO

[Atividade do Aluno 4: JSON VIEW](#)

[Atividade do Aluno 5: PASTA DE ATIVOS COMPLETOS](#)

[Atividade do Aluno 6: PROJETO COMPLETO](#)

1) Inicializar variável global:

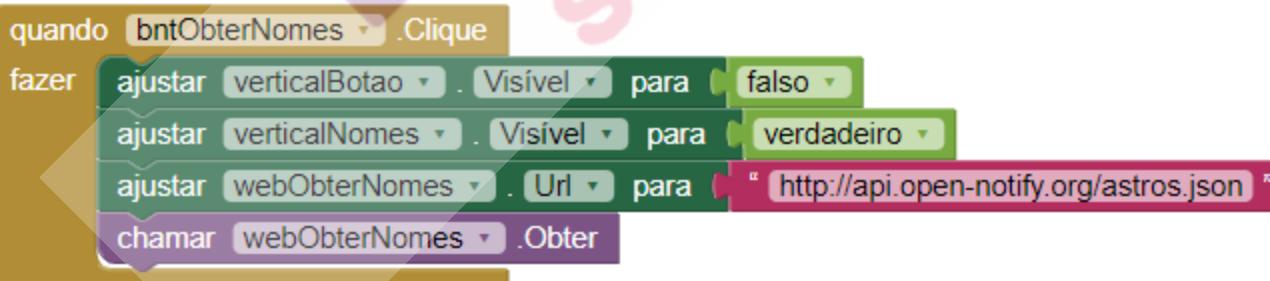


```

inicializar global [nomes] para [criar lista vazia]
inicializar global [listaDeObjetos] para [criar lista vazia]
inicializar global [nomeSelecionado] para [0]

```

2) Adicionar funcionalidade no botão Obter_Lista_de_Humanos:

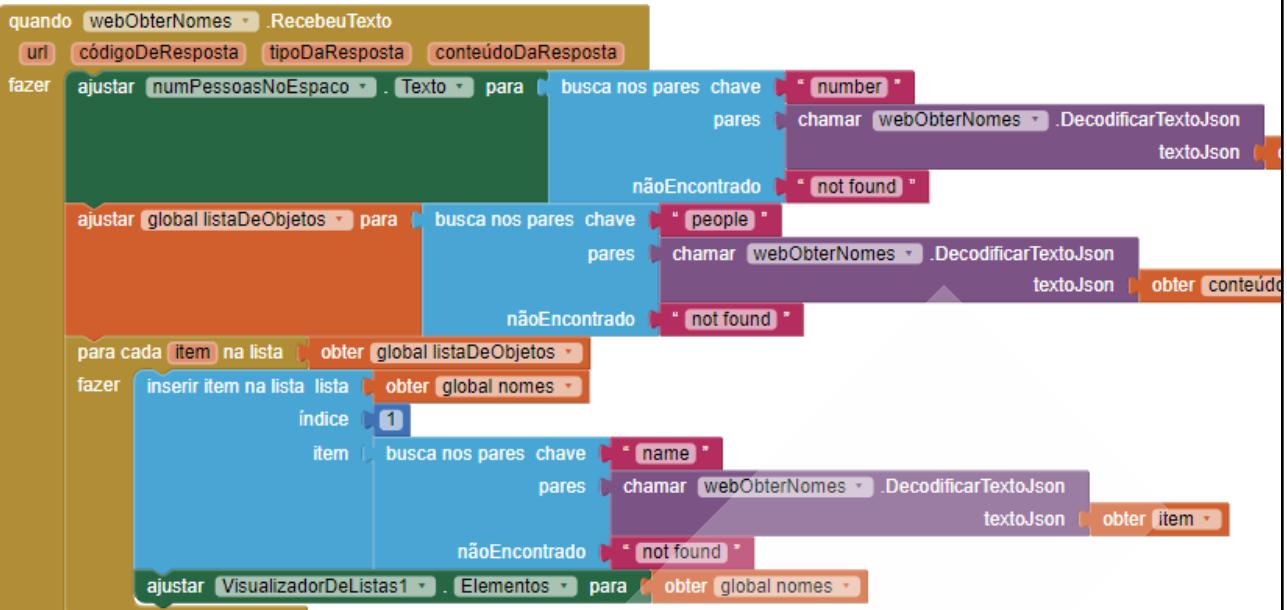


```

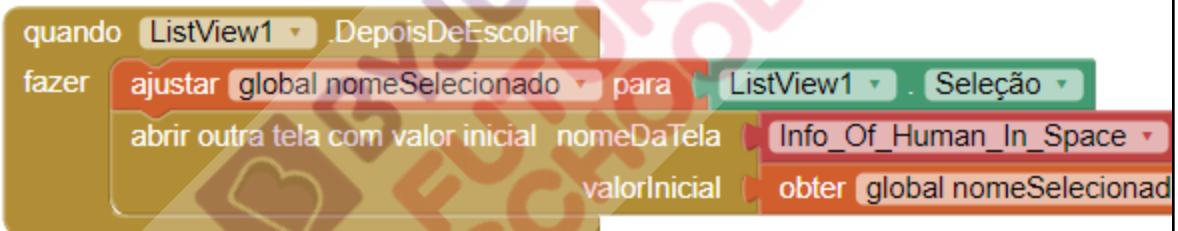
quando [bntObterNomes clique]
fazer [ajustar verticalBotao para falso, ajustar verticalNomes para verdadeiro, ajustar webObterNomes para URL "http://api.open-notify.org/astros.json", chamar webObterNomes para Obter]

```

3) Adicione funcionalidade para obter o nome dos humanos do espaço:



4) Adicione o código para Visualizador de Listas:



5) Adicione o código para navegar até Informações_sobre_humanos_no_Espaço:



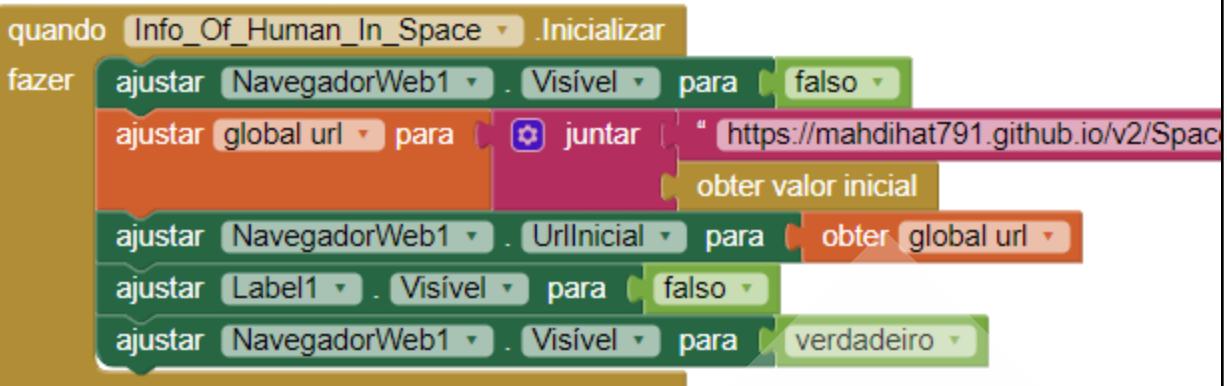
Código para Informações_sobre_humanos_no_Espaço:

1) Initialize a variável global:

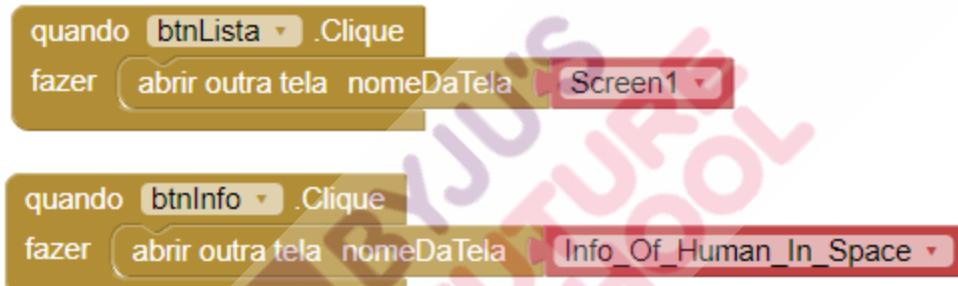
inicializar global url para " "

2) Exibir as informações do ser humano no espaço quando a tela for

inicializada:



3) Código para navegar para Screen1 e Informações_sobre_humanos_no_Espaço:



A professora orienta o aluno a interromper o Compartilhamento de Tela

A professora inicia o Compartilhamento de Tela

SESSÃO DE FECHAMENTO - 5 mins

A professora inicia a apresentação de slides  do slide 20 a 23.
Consulte as notas da oradora e siga as instruções em cada slide.

DICAS E SUGESTÕES DO PROJETO - 5 min.

A professora inicia a apresentação de slides  no slide 24.
Consulte as notas da oradora e siga as instruções em cada slide.

Dizer	Fazer
<p>LOCALIZANDO A ESTAÇÃO ESPACIAL</p> <p>Objetivo do Projeto:</p> <p>Hoje, você aprendeu como obter o número de pessoas no Espaço usando APIs. Você também aprendeu como obter seus nomes na API e exibi-los no aplicativo.</p> <p>Neste projeto, você vai localizar a ISS (Estação Espacial Internacional) no mapa e obter o número de passes. Você usará as APIs que o ajudarão a obter a localização atual da ISS.</p> <p>História:</p> <p>Você sabe o que é a ISS?</p> <p>A Estação Espacial Internacional ou ISS é o maior objeto artificial flutuando no espaço. Ela viaja ao redor da Terra a uma velocidade média de 27.700 km / h. À noite, ela pode ser facilmente vista da Terra, pois voa 320 quilômetros acima de nós. 16 países, incluindo os EUA, Rússia, Japão, Canadá e muitos estados membros da ISS trabalharam juntos para construir a Estação.</p> <p>É o maior satélite em órbita baixa da Terra, regularmente visível a olho nu da superfície da Terra.</p> <p>Neste projeto, vamos localizar a posição atual da ISS e marcar a localização no mapa com um marcador. Além disso, você também exibirá a altitude e o número de passagens - significa o número de vezes que a estação espacial passou por cima - da API.</p> <p>Estou muito animada para ver o resultado do seu projeto e sei que você se sairá muito bem.</p>	

Boa sorte!
Tchauzim!



A professora inicia a apresentação de slides no slide 25.
Consulte as notas da oradora e siga as instruções em cada slide.

Professora Para de Compartilhar a Tela

× End Class

Professora clica em:

Atividades Adicionais		Ajude o aluno com a programação.
------------------------------	--	----------------------------------

Atividade	Nome da Atividade	Links
Atividade da Professora 1	APP COMPLETO	https://drive.google.com/file/d/1P1VOjjqzRs27cf9sq0W8yvpXqTPqpIMT/view?usp=sharing
Atividade do Aluno 1	MIT APP INVENTOR	https://appinventor.mit.edu/
Atividade do Aluno 2	API DE HUMANOS NO ESPAÇO	http://api.open-notify.org/astros.json
Atividade do Aluno 3	API DE INFORMAÇÕES SOBRE HUMANOS NO ESPAÇO	https://mahdihat791.github.io/v2/Space_data/index.html?name=
Atividade do Aluno 4	JSONView	https://chrome.google.com/webstore/detail/jsonview/chklaanhfefnbpoihckbnefhakgolnmc

Atividade do Aluno 5	ARQUIVO DAS IMAGENS	https://drive.google.com/drive/folders/1Oer_rEJw2Xhys09cgpKdv9aO0lzQhBdR?usp=sharing
Atividade do Aluno 6	APP PARCIAL	https://drive.google.com/file/d/1QmimMtXniSfAq6kpYJNF6GjbtuvUylf1/view?usp=sharing
Solução de Projeto da Professora	LOCALIZANDO A ESTAÇÃO ESPACIAL	(vídeo) (código)
Recursos Visuais		https://s3-whjr-curriculum-uploads.whjr.online/27dbaa0a-0111-4af0-94c9-c3e252a89994.html COM NOTAS
Questionário	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/4501a9be-69c3-4f0b-a3c1-50b9c8b03742.pdf

Referência da Professora - NÃO SE DEVE ADICIONAR AO PAINEL DE LINKS DE ATIVIDADES

Referência para a professora Link de recurso visual (SEM NOTAS)	Link de recurso visual	https://s3-whjr-curriculum-uploads.whjr.online/7f299807-0327-459a-8064-822eedc7faee.html
--	------------------------	---