

## APP NATIVO DE DELIVERY



### Qual foi nosso OBJETIVO neste MÓDULO?

As crianças aprenderão a construir seu próprio app de delivery, que pode ser utilizado por donos de restaurantes, para tornar o seu negócio digital, enquanto usuários poderão utilizar o mesmo app para fazer pedidos. As crianças aprenderão a esboçar os caminhos do usuário enquanto constroem a front-end e a back-end do banco de dados.

### O que APRENDEMOS na aula de HOJE?

- Projetar e programar a tela 'Obrigado' (App Usuário).
- Projetar e programar a tela 'Conversa' (App Usuário).

### Que CONCEITOS/BLOCOS DE CÓDIGO cobrimos hoje?

- Adicionamos a funcionalidade para exibir a mensagem Obrigado após fazer um pedido.
- Adicionamos a funcionalidade para iniciar a conversa com o chefe.
- Adicionamos a funcionalidade para salvar a conversa no firebase realtime database.

Como FIZEMOS as atividades?

## Design do App

### DESIGN PARA A Tela “Obrigado” do App Usuário

Na tela Obrigado, o usuário foi capaz de ver a mensagem Obrigado na tela, e o botão Conversar com o Chefe levou o usuário à tela Conversa.

Resultado da Tela Obrigado:

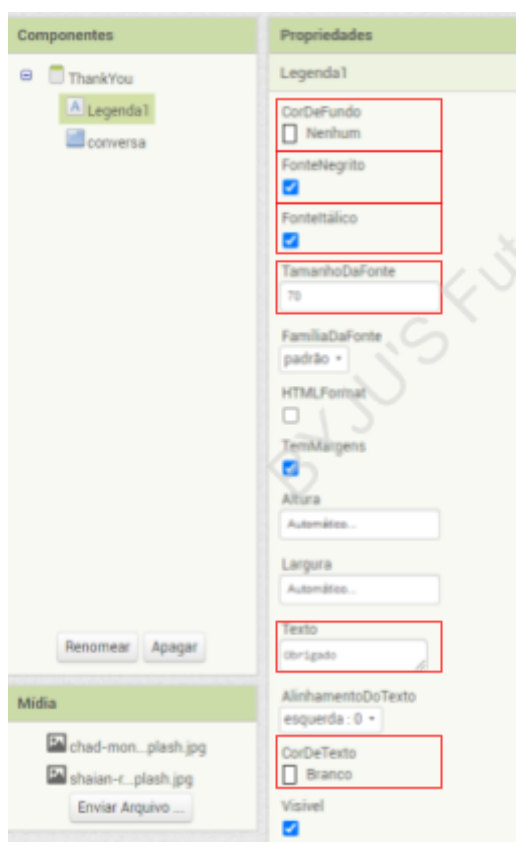


### Design da Tela Obrigado:

A tela 'Obrigado' compreendeu dois componentes de design:

- Uma **Legenda** para exibir a mensagem “Obrigado”
- Um **botão** para inicializar a conversa entre o Admin/Chefe e o Usuário.

1. **Exibimos a mensagem “Obrigado”:** Dentro da Tela Obrigado, colocamos a mensagem “Obrigado” com a ajuda de um componente legenda.

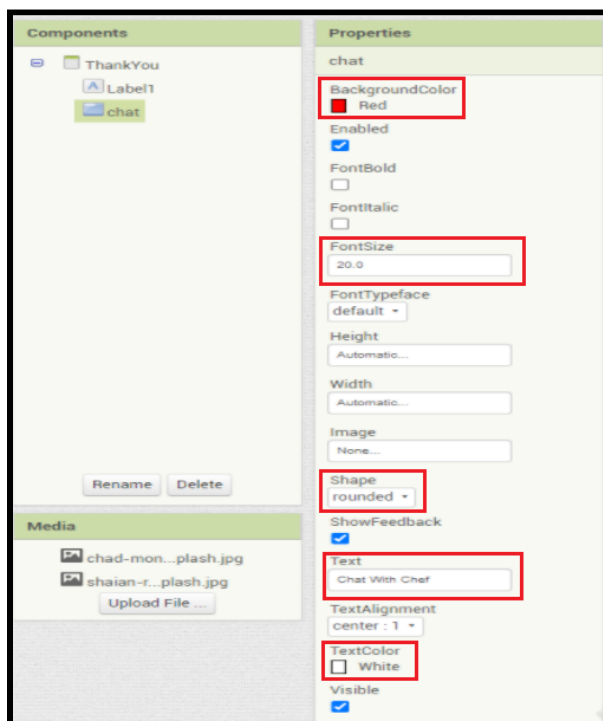


Propriedades da Legenda 'Obrigado'

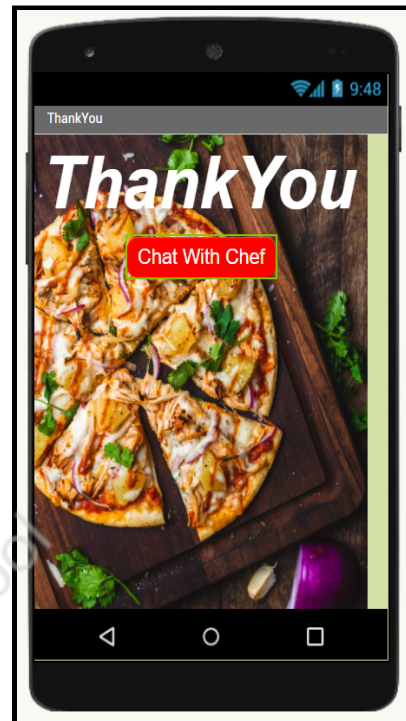


Resultado

2. **Adicionamos um botão - “Conversar com o Chefe”:** Criamos um botão para iniciar a conversa entre o Admin e o Usuário. O nome do botão é “Conversar com o Chefe”.



Propriedades do Botão 'Conversar com o Chefe'



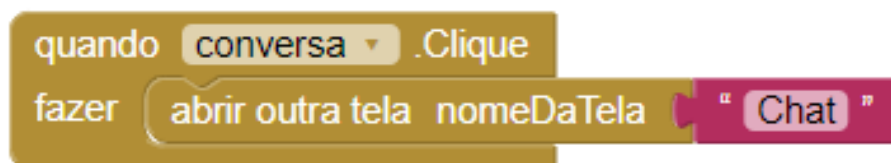
Resultado

### Código & Explicação

#### CÓDIGO PARA A TELA "Obrigado" do App Usuário

O objetivo aqui foi para que, uma vez que o usuário clicasse no Botão 'Conversar com o Chefe', o app abrisse a tela Conversa.

**Código para a Tela Conversa:** Concluimos o código para o botão "Conversar com o Chefe" - Quando um usuário clica no botão "Conversar com o Chefe", ele deve redirecionar para outra tela, ou seja, a tela "Conversa".



**Adicionamos uma nova Tela:** Clique na opção **Adicionar Tela...** para adicionar uma nova

tela e renomeá-la para Conversa.

---

## Design do App

---

### DESIGN PARA A Tela “Conversa” do App Usuário

Na tela Conversa, o usuário pode conversar com o admin do app ou com o chefe do item prato.

Resultado da tela Conversa:



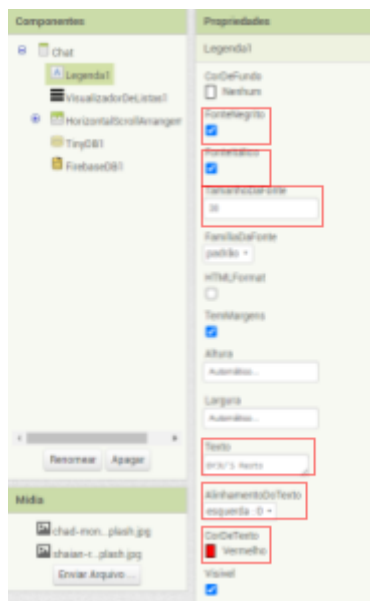
### Dados Salvos no Firebase Realtime database:



### Design da Tela Conversa:

Na tela 'Conversa', temos vários componentes de design:

1. **Adicionamos uma Legenda para exibir o nome do app:** Dentro da tela **Conversa**, colocamos o nome do app "BYJU'S Resto" com a ajuda de um componente legenda. Arrastamos e soltamos o componente Legenda no topo da tela Conversa.

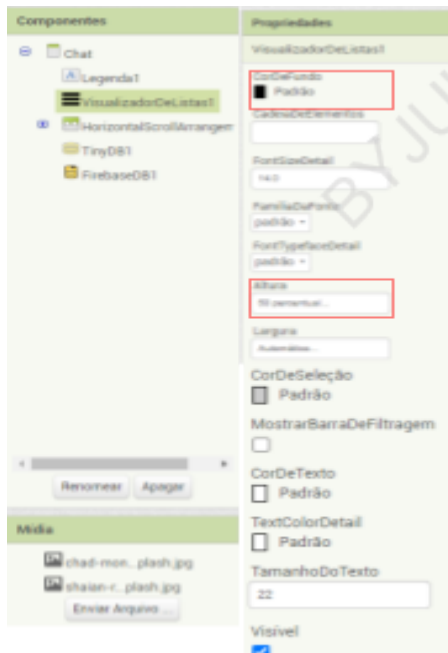


Propriedades do Componente 'Legenda1'



Resultado

2. **Adicionamos um VisualizadorDeListas1:** Arrastamos e soltamos o componente **VisualizadorDeListas1** abaixo da legenda1.



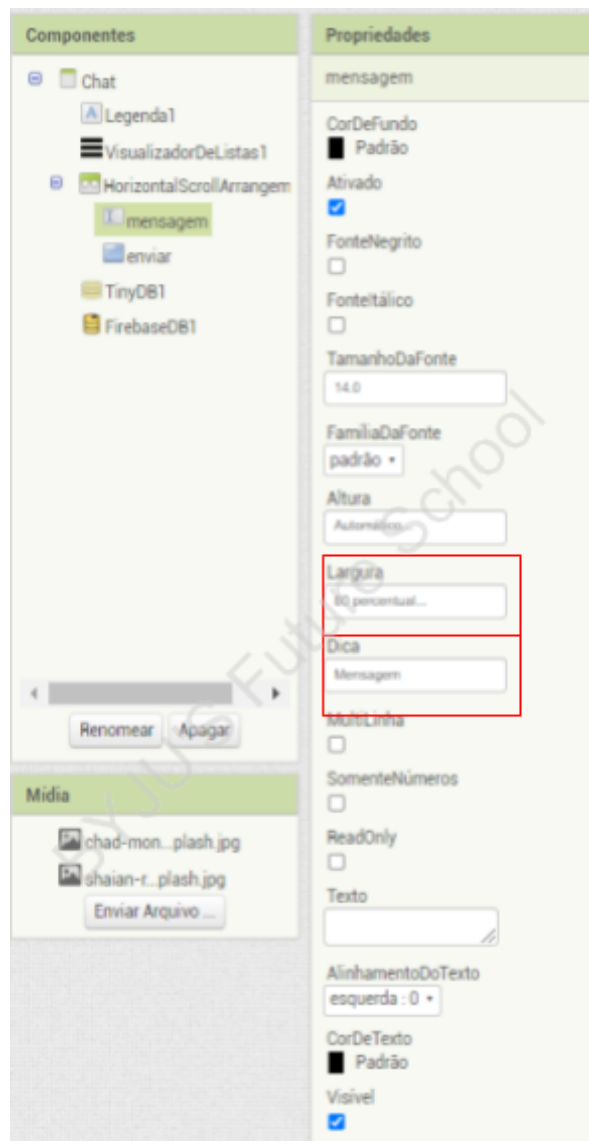
Propriedades do componente 'VisualizadorDeListas1'



Resultado

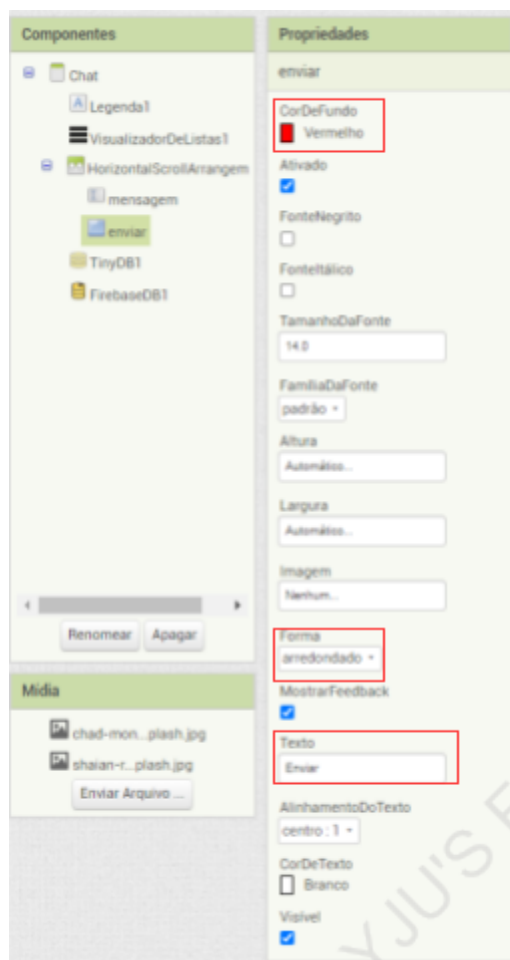
3. **Colocamos a CaixaDeTexto "mensagem" e o botão "Enviar":** Arrastamos e soltamos o componente **HorizontalScrollArrangement** abaixo do **VisualizadorDeListas1** e, dentro

desse componentes, arrastamos e soltamos mais dois componentes - Um é um componente CaixaDeTexto com o nome **“mensagem”** e outro é um componente Botão com o nome **“Enviar”**.



Propriedades do componente CaixaDeTexto mensagem



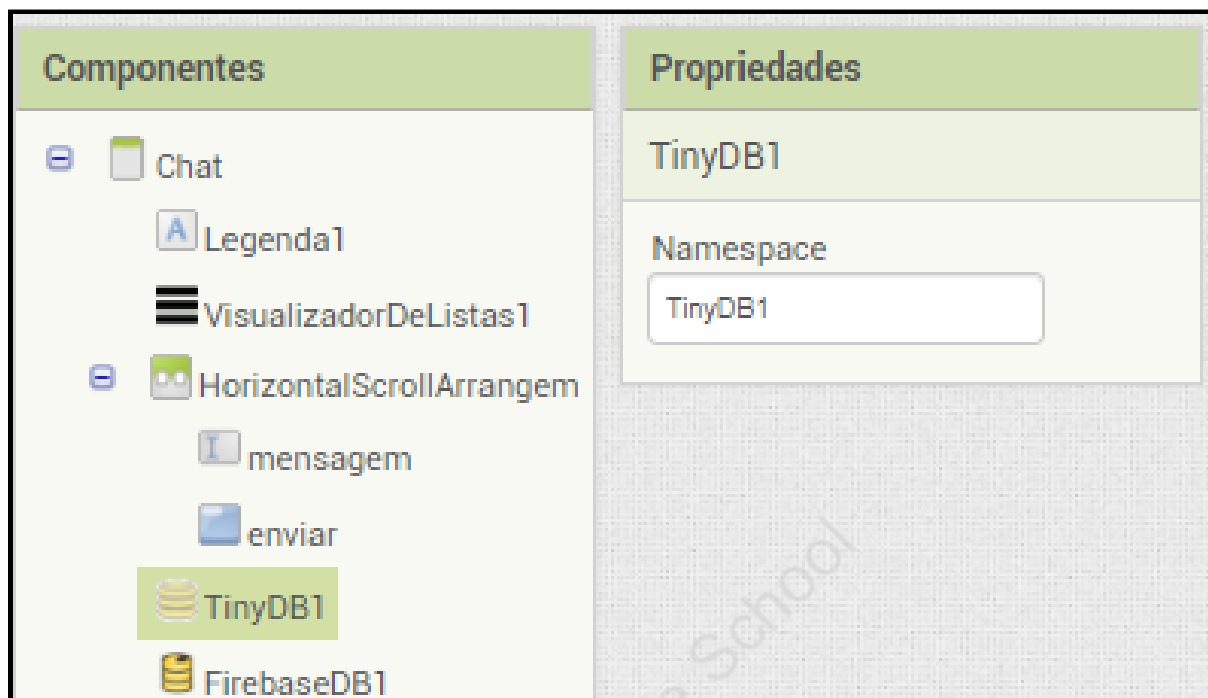


Propriedades do Botão “Enviar”

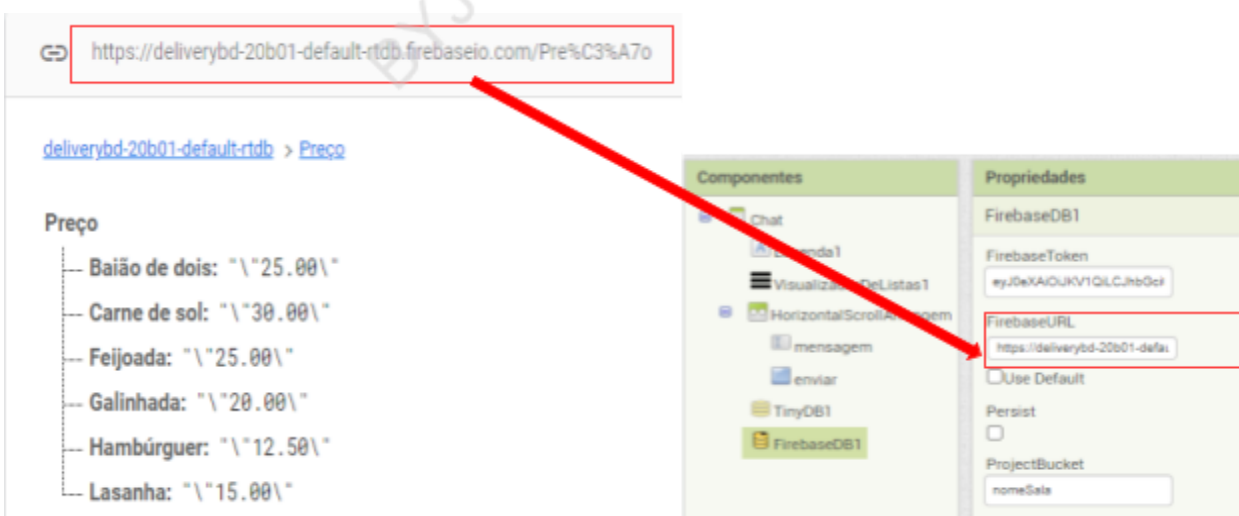


Resultado

4. **Adicionamos um componente TinyDB:** Arrastamos e soltamos o componente **TinyDB** do bloco **TinyDB** ao final. Precisamos de algum banco de dados para que possamos salvar a conversa entre o usuário e o chefe. É por isso que utilizamos o **ccomponente TinyDB**, ele salva os dados localmente para nosso app para que possamos utilizá-los em outras telas.



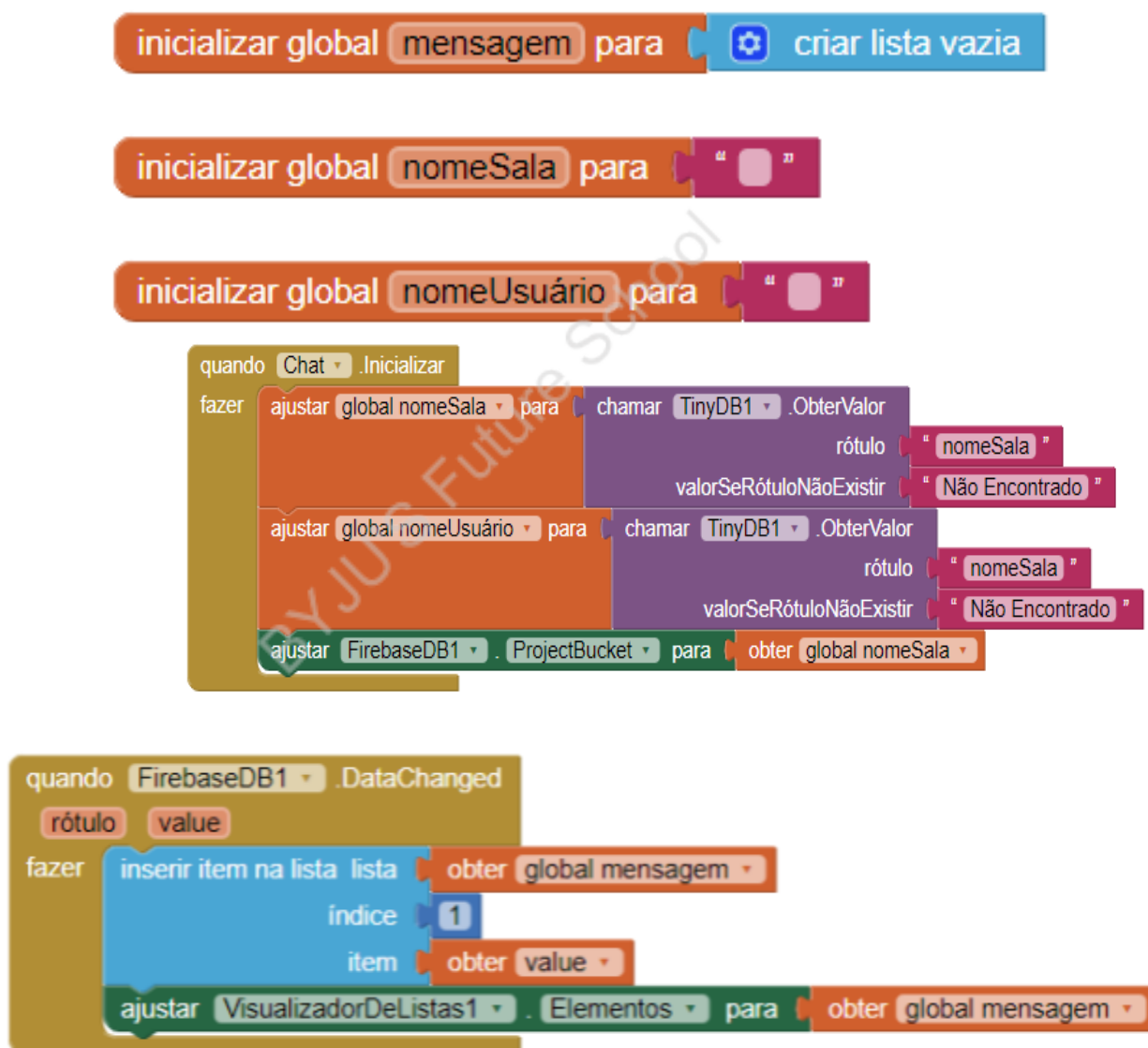
5. **Adicionamos um componente FirebaseDB:** Arrastamos e soltamos o componente FirebaseDB depois do componente TinyDb. Precisamos modificar a FirebaseURL de acordo com o projeto que criamos no Firebase Realtime Database para conectarmos o app ao nosso banco de dados.



## Código &amp; Explicação

## CÓDIGO PARA A TELA “Conversa” do App Usuário

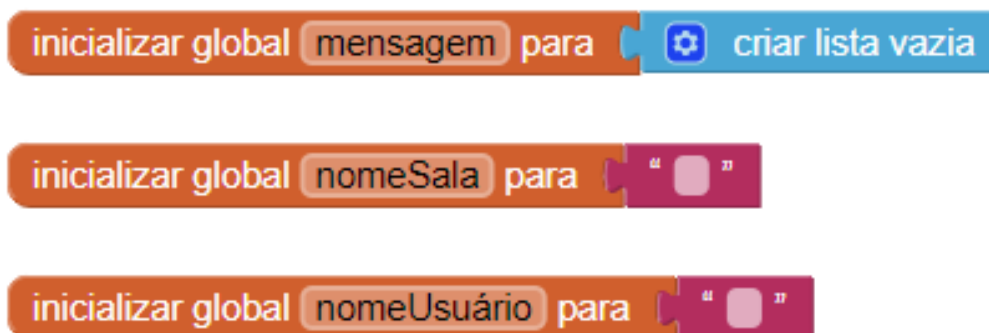
Abaixo, o código para a Tela Conversa:



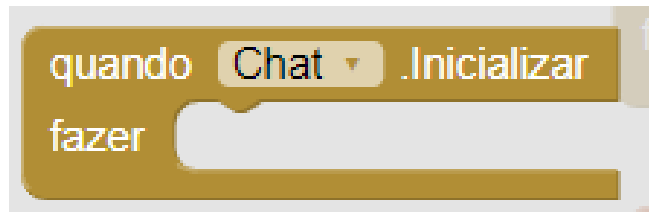


1. **Código para Inicializar a Variável:** Inicializamos todas as variáveis globais necessárias para a tela **Conversa** do App Usuário da mesma maneira que fizemos antes para as outras telas. Aqui, precisamos de três variáveis.

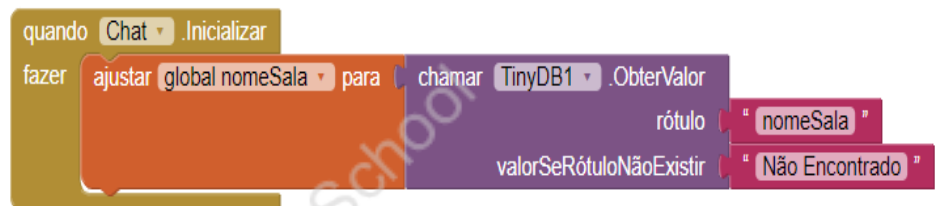
- a. **mensagem:** Aqui, tornamos o nome da variável para **mensagem** e a declaramos como uma *lista vazia*, para salvar todas as mensagens na lista quando um usuário iniciar a conversa.
- b. **nomeSala:** Aqui, tomamos o nome da variável para **nomeSala** e a declaramos como um **texto vazio** "". Como o nome sugere, essa variável foi utilizada para salvar o nome da sala dentro da qual mostramos a conversa.
- c. **nomeUsuário:** Aqui, tomamos o nome da variável para **nomeUsuário** e a declaramos como um **texto vazio** "". Como o nome sugere, essa variável foi utilizada para salvar o nome do usuário para quando ele iniciar a conversa.



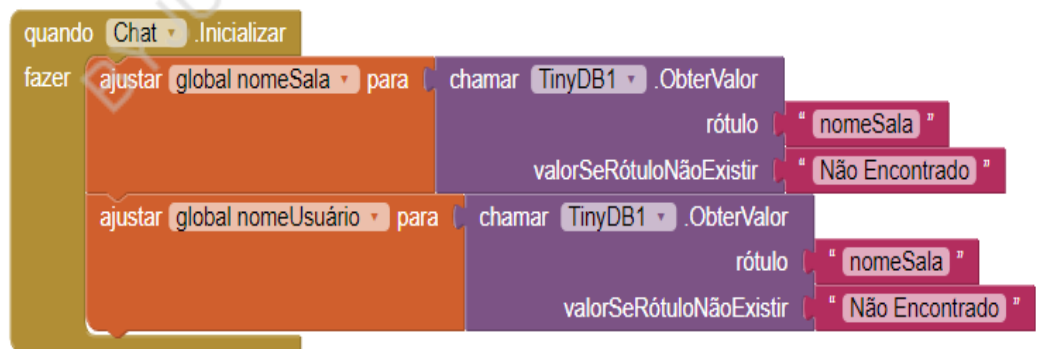
2. **Código para Inicializar a Tela Conversa:** Depois de declarar todas as variáveis, escrevemos o código para a inicialização da 'Conversa'.



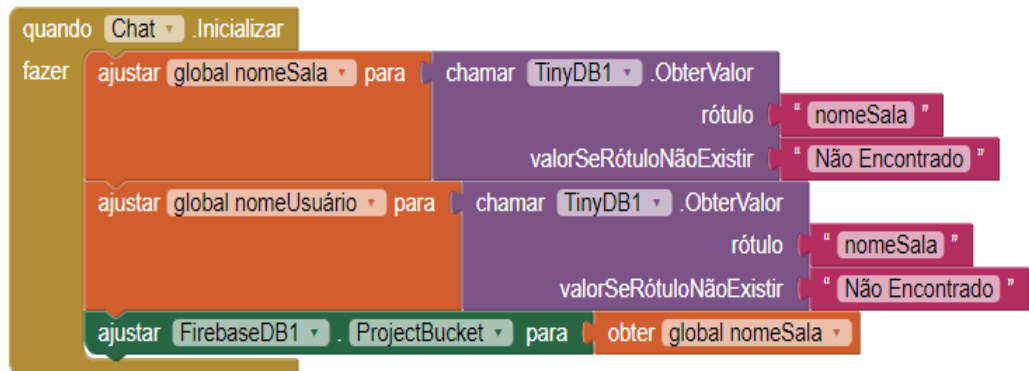
- Quando a conversa for inicializada, ajustamos todos valores das variáveis para seus respectivos valores :
  - **global nomeSala** recebe o valor do **TinyDB1** com o rótulo “nomeSala” que já salvamos, durante a programação da tela **PedidoFeito** na C45. Para isso, chamamos o **TinyDB1** e pegamos seu valor.



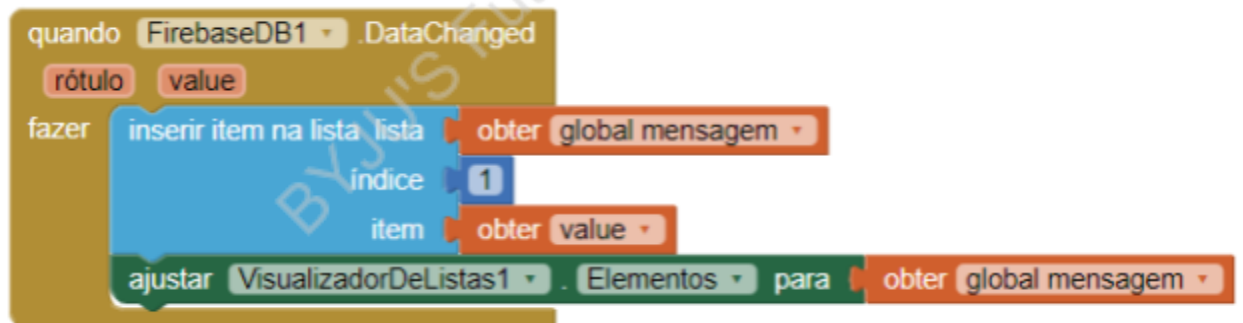
- Da mesma maneira, ajustamos o valor para a **global nomeSala**, agora, ajustamos o valor para a **global nomeUsuário**. Ela, também, pega o valor do **TinyDB1** com o rótulo “nomeSala”. Da mesma maneira, colocamos o código para a **global nomeSala**. Aqui, escolhemos os mesmos valores para **nomeUsuário** e **nomeSala**. Criamos a **nomeSala** com o mesmo nome de **nomeUsuário**.



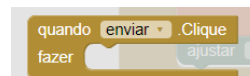
- Ajustamos, também, o projectBucket do Firebase com o nome que salvamos na **global nomeSala**, pois, criamos um novo projectBucket no banco de dados Firebase com o nome do usuário presente na **nomeSala** para que toda a conversa entre o chefe e um usuário fosse salva no banco de dados.



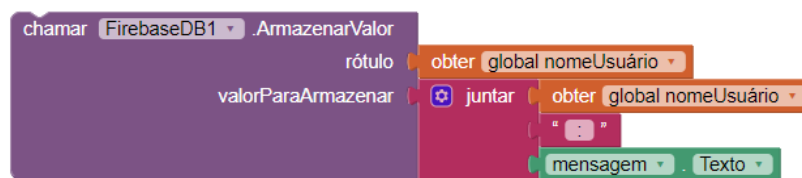
3. **Código para FirebaseDb1.DataChanged:** Agora, havia um problema - se o mesmo usuário enviasse, novamente, uma mensagem, então, no banco de dados Firebase, a nova mensagem seria atualizada em frente ao nome do usuário. Pois, no Firebase o rótulo que utilizamos é **nomeUsuário** para armazenar os dados no banco de dados. Portanto, **nomeUsuário** sempre foi o mesmo, e no banco de dados, se o rótulo com o mesmo nome já estivesse presente, então, o valor seria atualizado com um novo valor. Portanto, utilizamos uma lista para armazenar todas as mensagens - quando o banco de dados é modificado, ele continua adicionando todas as mensagens à lista **global mensagens**.



4. **Código para o botão “Enviar”:** Agora, precisávamos programar o botão **Enviar**. Quando um usuário clica no botão **Enviar**, ele deve armazenar a mensagem no FirebaseDB com o nome do usuário.



- Primeiro, escolhemos o comando **ajustar** do bloco **Enviar**.
- A partir do bloco **FirebaseDb**, precisamos do comando que foi utilizado para armazenar os valores no firebase.
  - Juntamos o rótulo com o comando **obter variável nome** do bloco **Variáveis** e escolhemos o valor **global nomeUsuário** da lista suspensa.
  - Em **valorParaArmazenar**, queremos salvar o valor da sintaxe abaixo:  
Nome usuário: mensagem



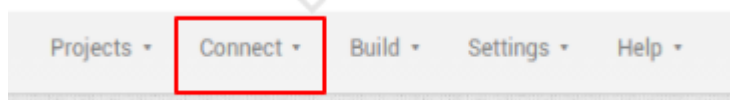
- Ajustamos o texto da variável mensagem para texto vazio.



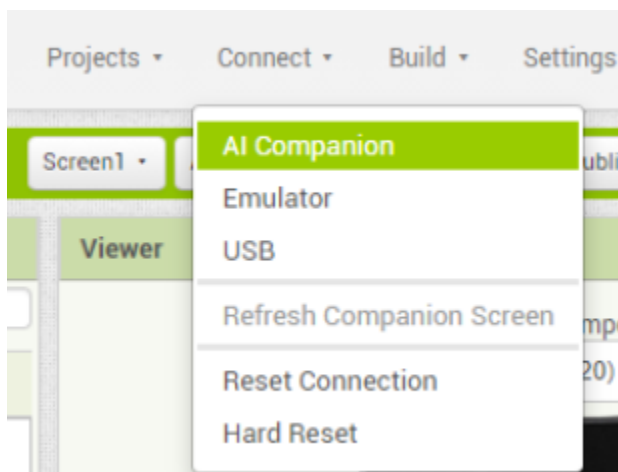
### Testando a Aplicação:

Conectamos a aplicação ao nosso smartphone para testá-la:

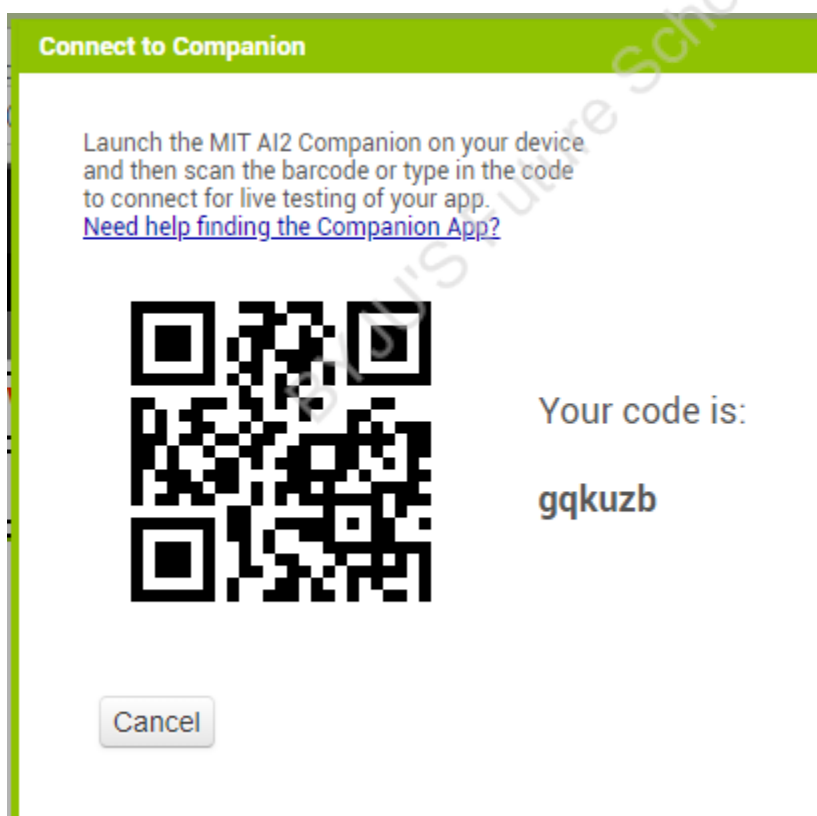
- 1) Clicamos em **Conectar**:



- 2) Selecionamos **Assistente AI**:

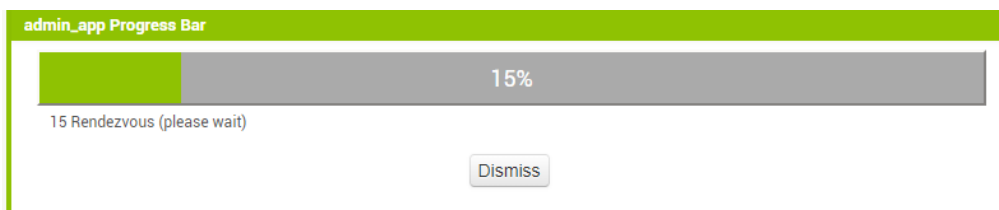


3) Escaneamos o QR code:



A conexão ao smartphone começa:





- 4) Uma vez que o app foi conectado ao smartphone, você verificou o resultado das telas Obrigado e Conversa. Além disso, verificou o resultado no firebase realtime database.

### Qual o PRÓXIMO PASSO?

Na próxima aula, vamos começar a projetar e a programar a tela 'VerificarPedidos' do App Admin, para que o admin possa verificar os pedidos feitos.