

EASYTALK



Integrantes do Grupo:

- Cláudia Mikaelly da Silva;
- Diego Ribeiro de Sousa;
- Edson dos Santos Teixeira Junior;
- Letícia dos Santos Castro;
- Luís Tiago Ferreira Melo dos Santos;
- Nicole Freitas de Oliveira;
- Rafaella de Oliveira Barros;

Docente:

Maurício Neves Asenjo

2023

SUMÁRIO

INTRODUÇÃO

INSTALAÇÃO

CÓDIGOS

LOGIN

- BASE DE DADOS FIREBASE
- TELA DE LOGIN

CADASTRO

FUNCIONALIDADES

- OPÇÕES
- TRADUÇÃO
- AUDIODESCRIÇÃO
- TRANSCRIÇÃO

2ª VERSÃO

- LOGIN
- CADASTRO
- CHAT

AGRADECIMENTOS



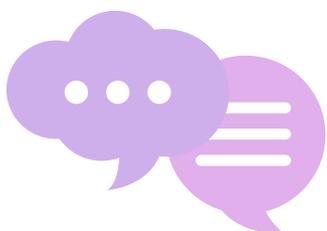
INTRODUÇÃO

A criação desse App teve como objetivo explorar nossos aprendizados obtidos no curso técnico em informática e contribuir com um software inclusivo que visa atender necessidades na comunicação entre pessoas que possam ter alguma deficiência que dificulte esse processo.

Criamos um aplicativo de mensagens que além de enviar e receber mensagens de texto, também inclui outras funcionalidades, como:

- Conversão de texto para áudio;
- Transcrição de áudio para texto;
- Tradução de mensagens;
- Detecção de texto em imagem e sua conversão para áudio;

Para isso, resolvemos iniciar desenvolvendo as funcionalidades listadas acima e integrá-las a uma interface gráfica. Em seguida, foi criada a tela de chat, onde ocorre o envio e recebimento de mensagens entre dispositivos.



INTRODUÇÃO

Para criar nosso aplicativo, optamos por utilizar a linguagem de programação Python, uma escolha cada vez mais popular no mercado corporativo. Isso se deve à sua capacidade de proporcionar maior agilidade, produtividade e qualidade nas entregas.

O Python oferece várias vantagens, tais como:

- Facilidade de aprendizado
- Sintaxe intuitiva
- Diversidade de bibliotecas abertas
- Multiplataforma, entre outras.

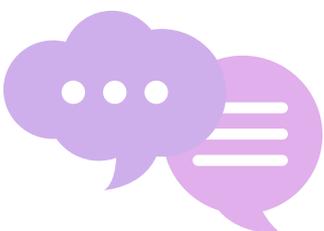
Nas próximas páginas haverá o passo a passo de cada etapa desenvolvida neste software.



INSTALAÇÃO

Para iniciar o desenvolvimento do nosso App, tivemos que instalar:

- **Visual Studio Code:** editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS;
- **Python:** linguagem de programação que para ser executada pela sua máquina, precisa que haja um interpretador que deve ser instalado;
- **Flet:** framework que permite utilizar a interface do Flutter programando em Python;



INSTALAÇÃO

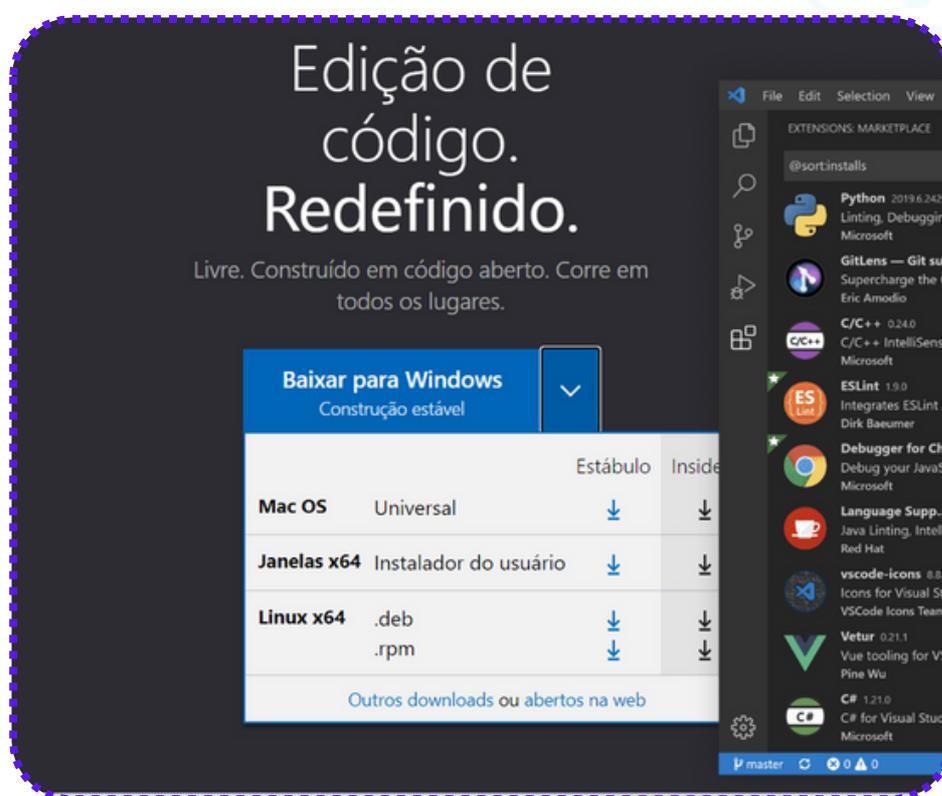
1º PASSO:

• Visual Studio Code

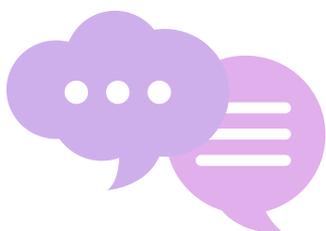
Para instalá-lo, basta acessar o site:

<https://code.visualstudio.com/>

e baixar a versão compatível com seu sistema operacional, como mostrado abaixo, e em seguida instalá-lo. Caso tenha dúvidas na instalação, basta consultar a documentação do VS Code.



OBS: o VS Code não é obrigatório para o desenvolvimento deste App, ele pode ser substituído por outro editor de código.

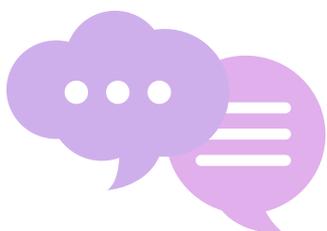
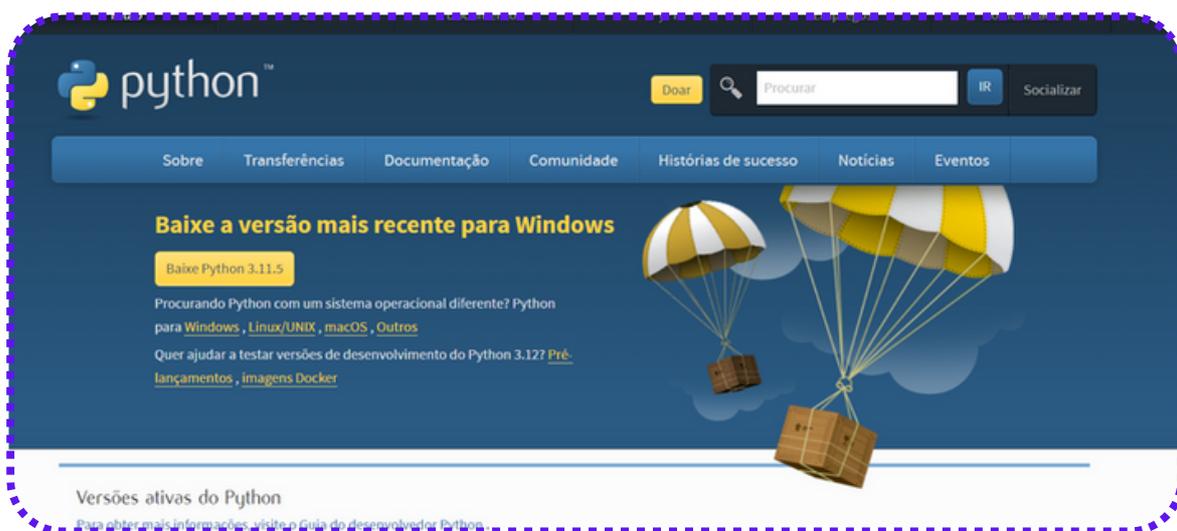


INSTALAÇÃO

2º PASSO:

- **Python**

Para instalá-lo, basta acessar o site: <https://www.python.org/downloads/> e seguir os mesmos critérios usados na instalação anterior.

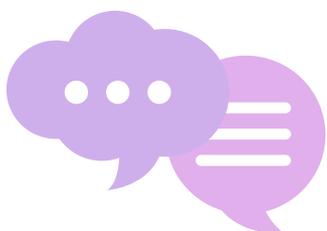
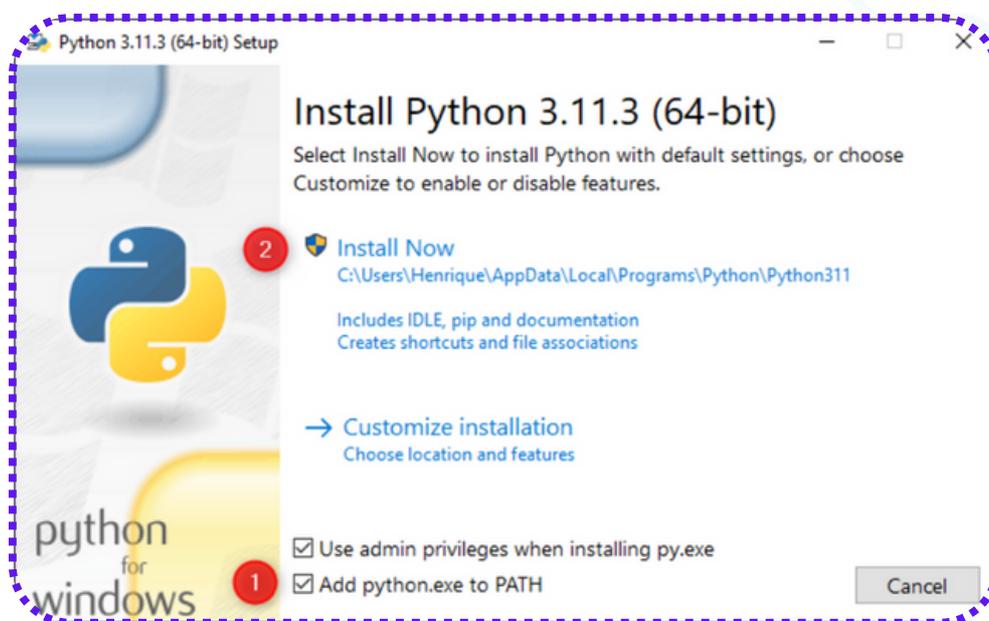


INSTALAÇÃO

2º PASSO:

• Python

É importante que no processo de instalação seja selecionado as duas opções abaixo. Caso a opção “Add python.exe to PATH” não seja marcada, após instalar o Python deve-se inserir o caminho onde ele está armazenado, nas Variáveis de Ambiente do seu dispositivo, para que os comandos usados por esse interpretador sejam reconhecidos pelo seu aparelho.



INSTALAÇÃO

3º PASSO:

- **Flet**

Para a instalação deste é necessário que você já tenha um editor de código e um interpretador do Python instalados, pois você deve usar o PIP que é um gerenciador de pacotes escritos em Python, para instalar o Flet que é um desses pacotes.

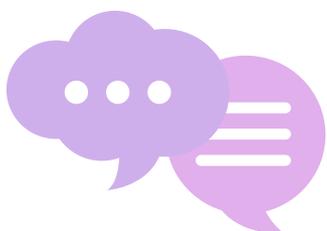
Para instalar pacotes usando o PIP, basta digitar no Prompt de Comando:

```
pip install nome_do_pacote
```

Nesse caso, você irá digitar:

```
pip install flet
```

Após a instalação do Flet você já pode começar criar os arquivos em Python e usar este framework.



CÓDIGOS

Os códigos de cada arquivo criado durante o desenvolvimento deste App, estão disponíveis nos links abaixo:

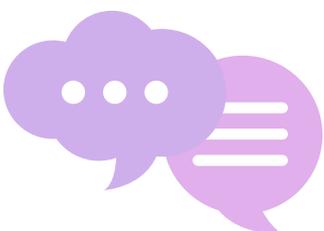
<https://github.com/LeticiaSanttos/appUsingFlet>

<https://github.com/LeticiaSanttos/EasyTalk---Flet>

- O **1º link** mostra a primeira versão do aplicativo, onde o enfoque estava nas funcionalidades.
- O **2º link** mostra a segunda versão do aplicativo, onde o enfoque estava na criação de um aplicativo de mensagens e a aplicação destas funcionalidades nele.

OBS.: sempre que nos arquivos houver **import nome_do_pacote**, você deve no Prompt de Comando digitar: **pip install nome_do_pacote**

Para que assim sejam instaladas todas as bibliotecas do Python necessárias para as funcionalidades deste App.



LOGIN

1º PASSO:

- **Criar base de dados no Firebase**

Neste App criamos uma tela de login e cadastro que para ser validada necessita de uma base de dados, para isso usamos o Firebase. Para utilizá-lo, basta acessar o link abaixo, colocar sua conta do Google e criar um projeto.

Link: [FIREBASE](#)

X Criar um projeto(Passo 1 de 3)

Vamos começar nomeando o projeto[®]

Nome do projeto

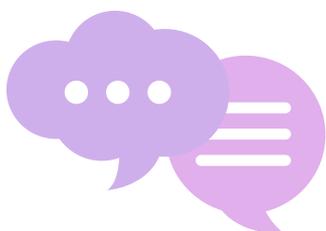
cadastro

cadastro-6f71b

Aceito os [Termos do Firebase](#)

Confirmo que vou usar o Firebase exclusivamente para fins relacionados ao meu comércio, empresa, ofício ou profissão.

Continuar

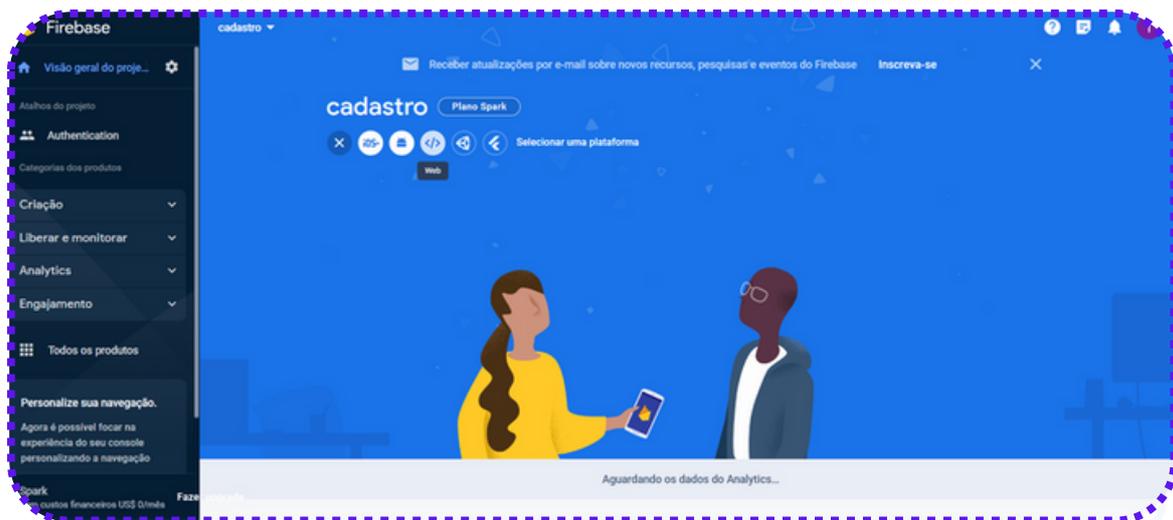


LOGIN

1º PASSO:

- **Criar base de dados no Firebase**

Em seguida clique na opção **Web** mostrada abaixo para gerar o código de autenticação que será importante para a conexão do aplicativo com o Firebase.



LOGIN

1º PASSO:

- **Criar base de dados no Firebase**

Você deverá colocar o nome do aplicativo, em seguida será gerado o código de autenticação que você deverá copiar e colar no arquivo Python que mostraremos na próxima página.



2 Adicionar o SDK do Firebase

Usar o npm Usar a tag <script>

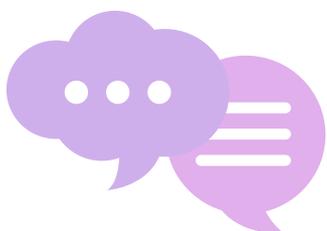
Se você já estiver usando o [npm](#) e um bundler de módulos, como [webpack](#) ou [Rollup](#), execute o seguinte comando para instalar o SDK mais recente ([saiba mais](#)):

```
$ npm install firebase
```

Depois inicialize o Firebase e comece a usar os SDKs dos produtos.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyAomzV8pGkyMBUfWhgpDUYhmbn2vzBqkaM",
  authDomain: "cadastro-6f71b.firebaseio.com",
  projectId: "cadastro-6f71b",
  storageBucket: "cadastro-6f71b.appspot.com",
  messagingSenderId: "187658957157",
  appId: "1:187658957157:web:42eb4b9ae82878b1e4b8c",
  measurementId: "G-RQ1EQ19TGK"
};
```



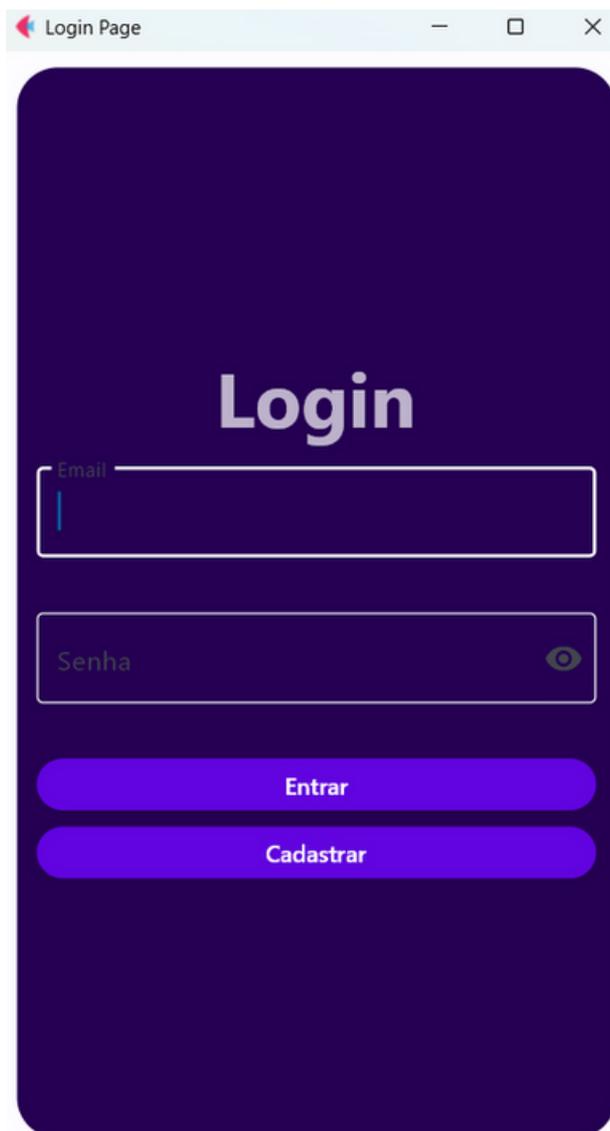
LOGIN

2º PASSO:

- **Criar tela de login usando o Flet**

Para isso, você deve criar o arquivo `page_login.py` e colocar o código disponível no link da página Códigos.

[Ir para pág. Códigos](#)



The image shows a mobile application window titled "Login Page". The interface is dark-themed with a purple gradient. At the top, the word "Login" is displayed in a large, white, sans-serif font. Below the title, there are two input fields: the first is labeled "Email" and the second is labeled "Senha" (Password), which includes a small eye icon for toggling visibility. At the bottom of the form, there are two prominent, rounded rectangular buttons: "Entrar" (Login) and "Cadastrar" (Register), both in a bright purple color with white text.



CADASTRO

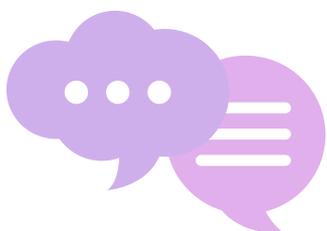
- **Criar tela de cadastro usando o Flet**

Para isso, você deve criar o arquivo `page_cadastro.py` e colocar o código disponível no link da página Códigos.

[Ir para pág. Códigos](#)



The image shows a screenshot of a mobile application window titled "Login Page". The main content is a dark purple screen with the word "Cadastro" in large white text at the top. Below the title are two white input fields: the first is labeled "Email" and the second is labeled "Senha". At the bottom of the screen, there are two white buttons: the top one is labeled "Cadastrar" and the bottom one is labeled "Já possui uma conta? Entre Aqui". The window has standard OS window controls (back, minimize, maximize, close) at the top.

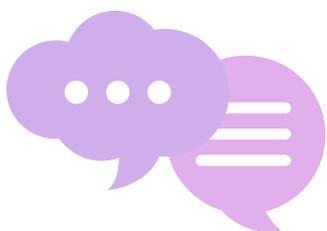
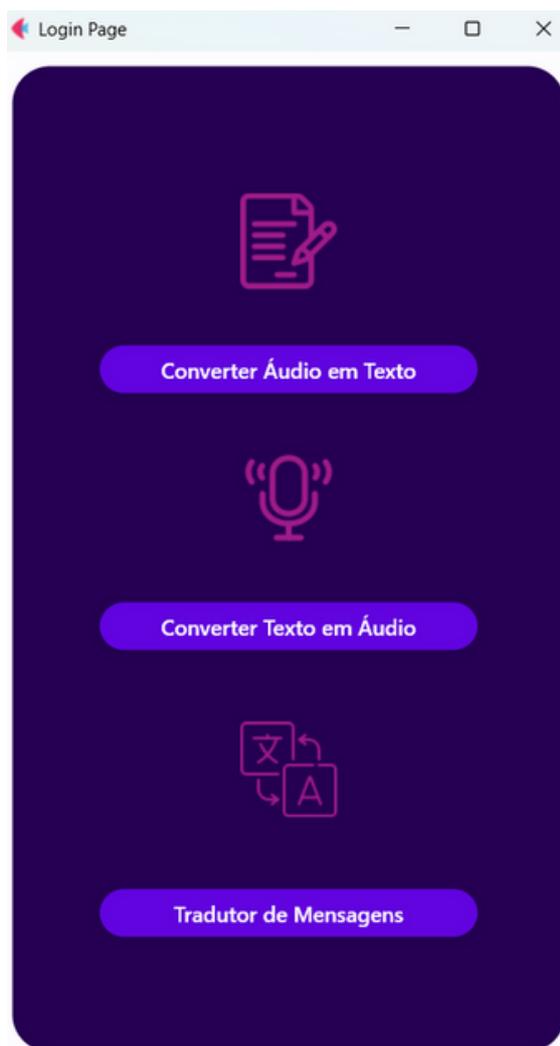


FUNCIONALIDADES

- **Criar tela de opções usando o Flet**

A tela de opções apresentará as funcionalidades disponíveis para teste ao usuário por meio de ícones que representam cada uma delas e um botão que redirecionará o usuário para uma tela específica contendo a funcionalidade selecionada.

[Ir para pág. Códigos](#)

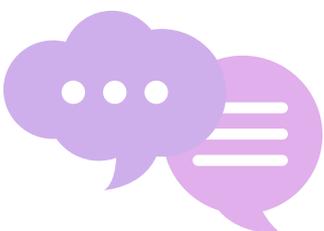
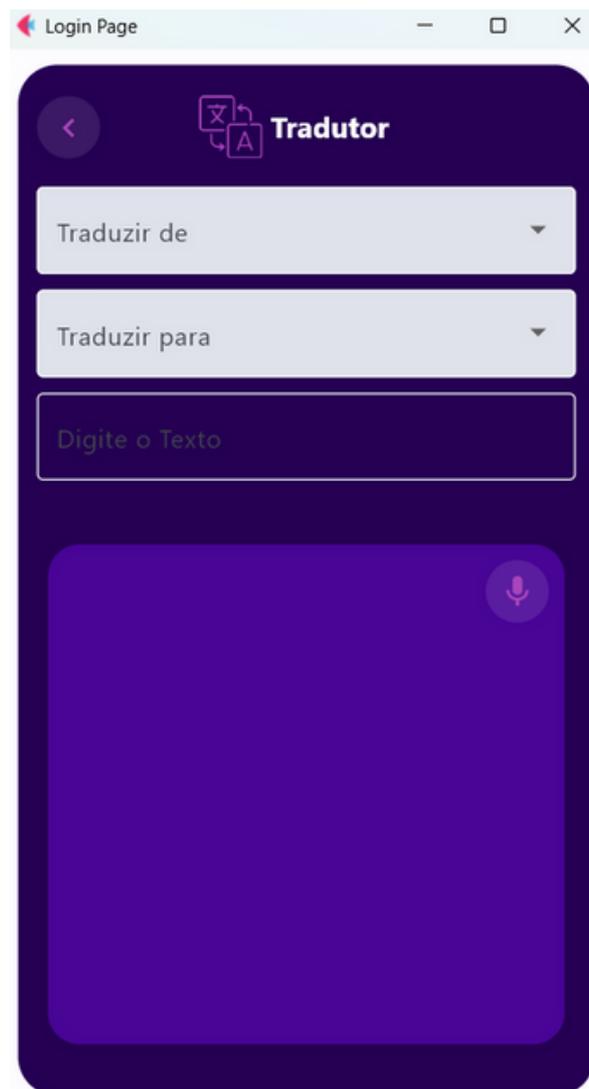


TRADUÇÃO

- **Criar tela de tradução usando o Flet**

A tela de tradução contém dois campos de seleção que correspondem respectivamente, ao idioma do qual a mensagem será traduzida e para qual idioma se quer traduzir. Há também um botão que acionará o evento de tradução e outro para ouvir a mensagem traduzida.

[Ir para pág. Códigos](#)

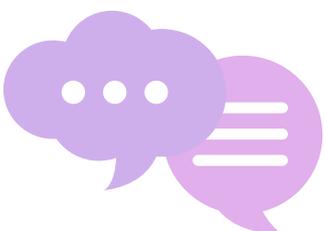
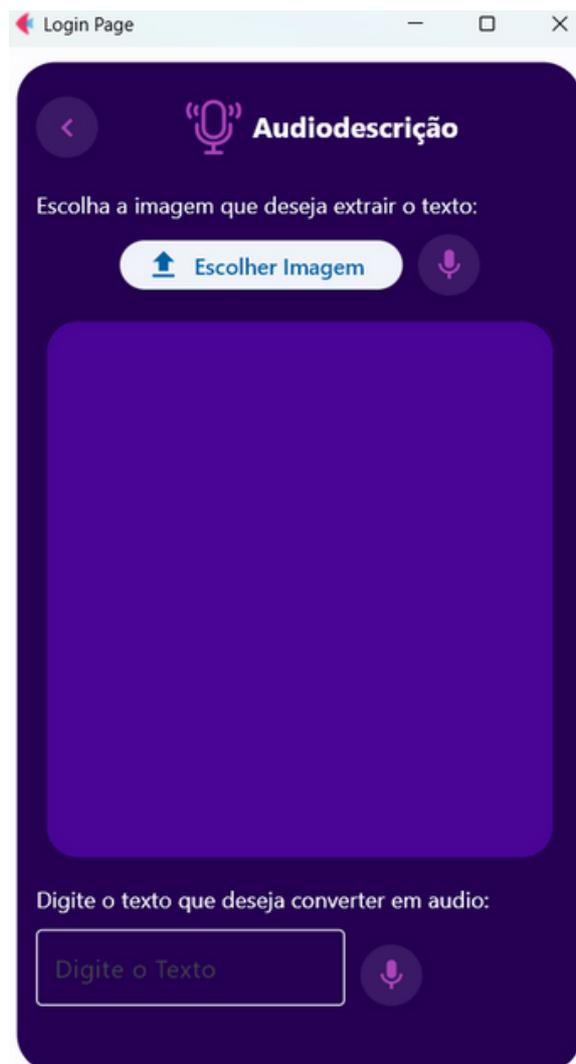


AUDIODESCRIÇÃO

- **Criar tela de audiodescrição usando o Flet**

A tela de audiodescrição contém um botão para extrair o texto de uma imagem e outro botão para ouvir o texto extraído. Também possui um campo para que o usuário digite o texto que quiser e um botão para ouvir o texto.

[Ir para pág. Códigos](#)

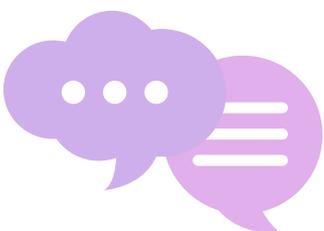


TRANSCRIÇÃO

- **Criar tela de transcrição de áudio usando o Flet**

A tela de transcrição de áudio contém um botão para o usuário gravar um áudio que em seguida será transcrito.

[Ir para pág. Códigos](#)

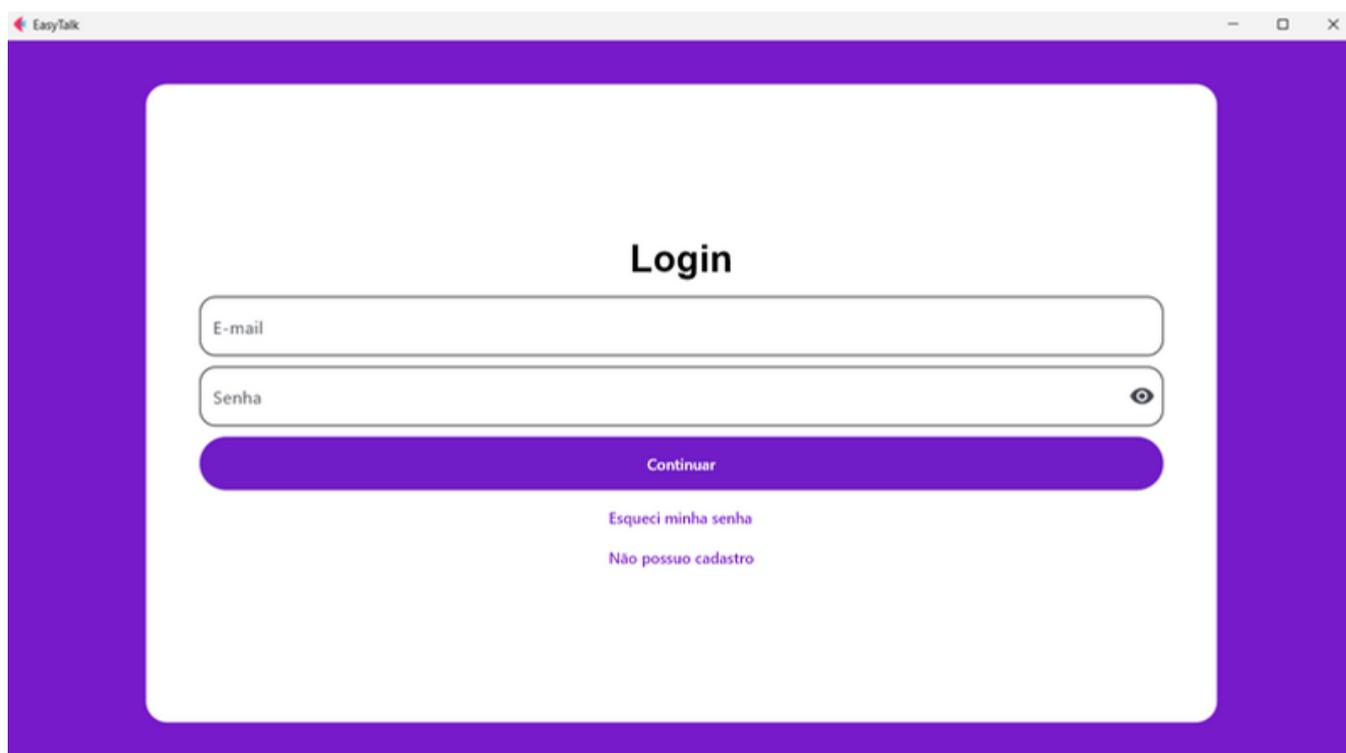


2ª VERSÃO

- **Tela de login**

Esta é a nova tela de login, possui dois campos necessários para identificação do usuário, além de uma opção para redefinição de senha e outra para a criação de um cadastro.

[Ir para pág. Códigos](#)



EasyTalk

Login

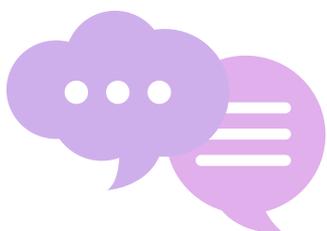
E-mail

Senha

Continuar

[Esqueci minha senha](#)

[Não possuo cadastro](#)



2ª VERSÃO

- **Tela de cadastro**

Esta é a nova tela de cadastro, possui quatro campos necessários para o cadastro do usuário e uma opção para ser direcionado para a tela de login.

[Ir para pág. Códigos](#)

EasyTalk

Cadastro

Nome

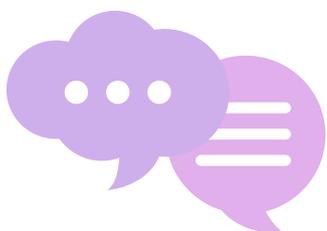
E-mail

Senha

Confirme sua senha

Cadastrar

Já tenho uma conta. Fazer login.

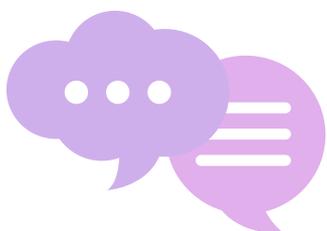
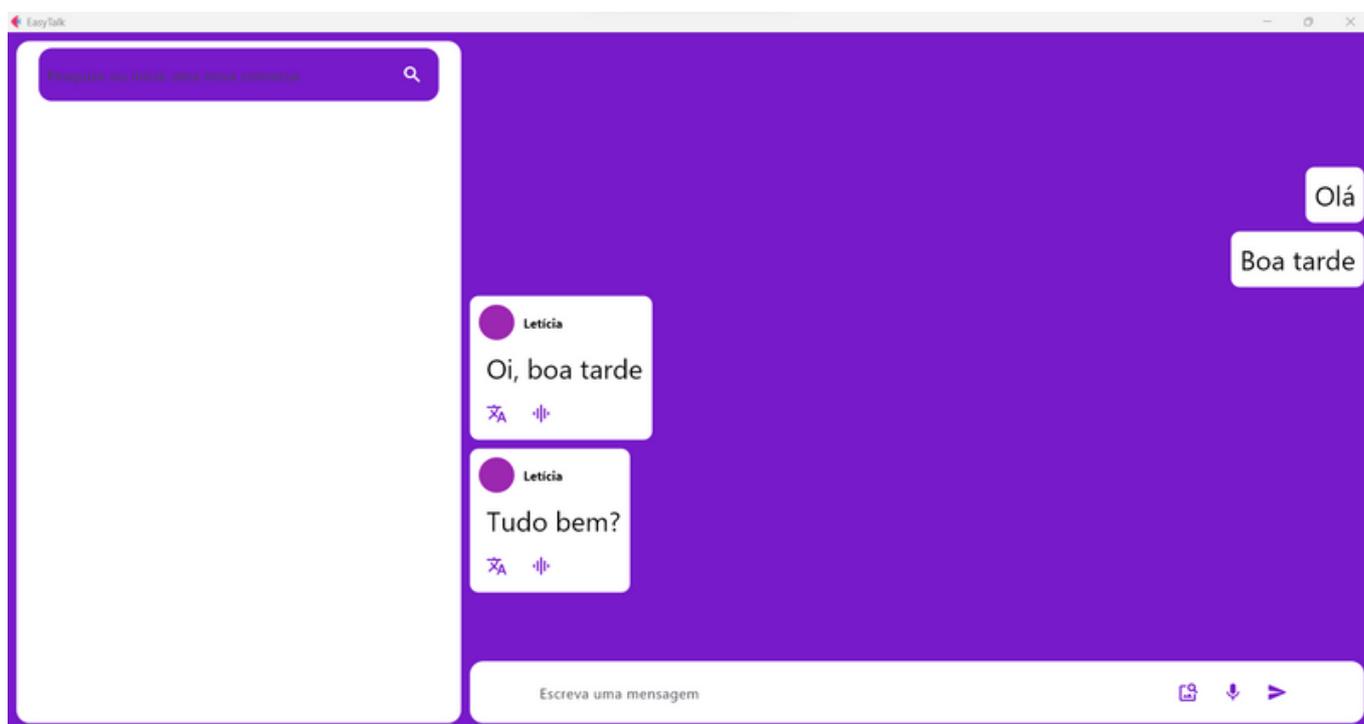


2ª VERSÃO

- **Tela do chat com as funcionalidades**

Esta é a visão geral do aplicativo de mensagens, onde pode-se observar uma conversa entre dois usuários.

[Ir para pág. Códigos](#)

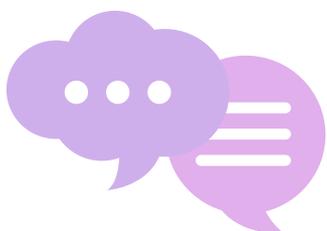


2ª VERSÃO

- **Tela do chat com as funcionalidades**

Nas mensagens recebidas há o nome do usuário com quem se está conversando e dois ícones abaixo da mensagem deste, que representam respectivamente as funcionalidades de tradução da mensagem e de audiodescrição.

[Ir para pág. Códigos](#)

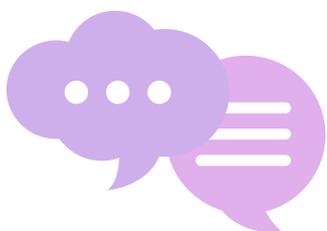


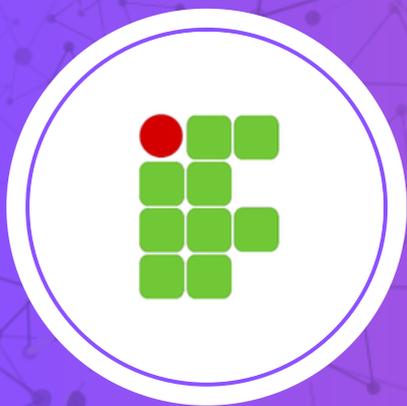
2ª VERSÃO

- **Tela do chat com as funcionalidades**

No campo direcionado para o envio das mensagens, há a possibilidade de além de escrever mensagens de texto, importar uma imagem que possua algum texto, ou gravar um áudio para em seguida transcrevê-los.

[Ir para pág. Códigos](#)





EASYTALK 

AGRADECIMENTOS

Primeiramente, agradecemos a Deus por nos permitir alcançar nossos objetivos ao longo dos anos de estudo. Também gostaríamos de expressar nossa sincera gratidão aos orientadores, Maurício, Marcelo e Thiago, e a todos os nossos familiares, amigos, IFSP e outros apoiadores que tornaram possível o encerramento deste Trabalho de Conclusão de Curso.

2023