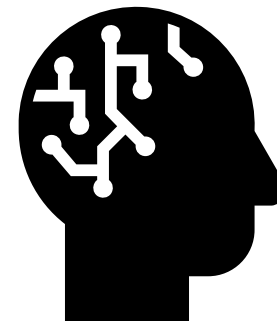


ClerkBot

Bot de envio de mensagens para WhatsApp

O que é um bot?

Bot é um software criado para executar determinadas tarefas de modo automático, ou seja, atuam por conta própria sem precisar de um humano fazendo aquela tarefa.



Os bots normalmente costumam imitar ou substituir um comportamento humano.

Exemplos de bot

ChatBot

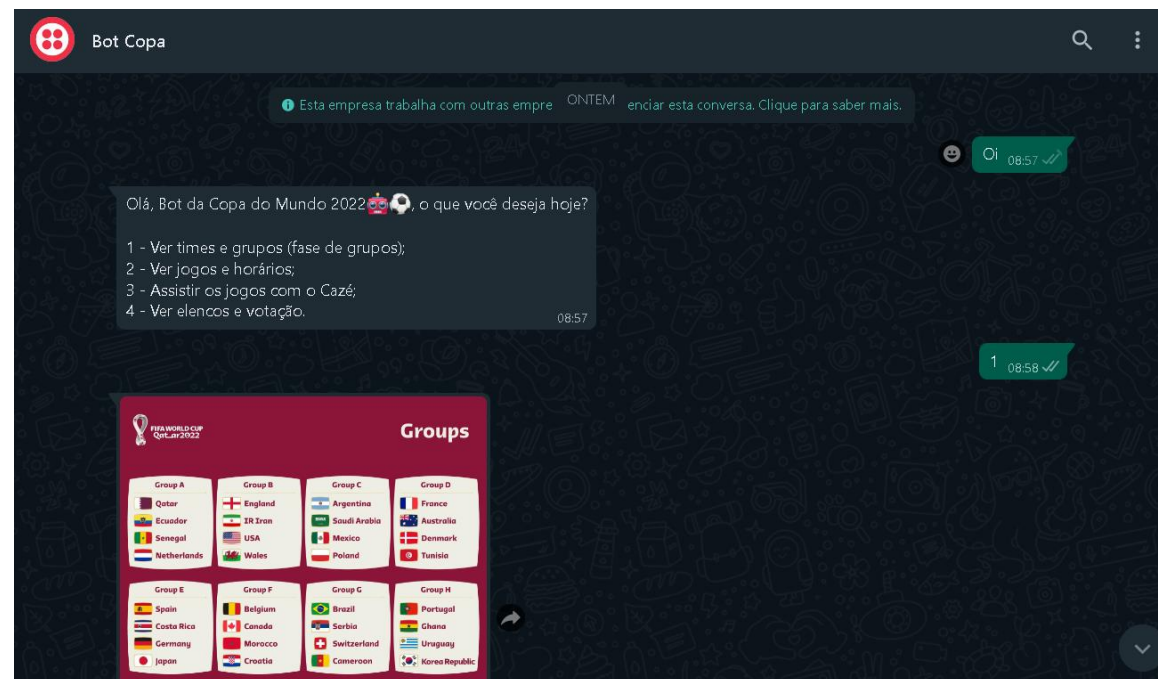


ChatBot

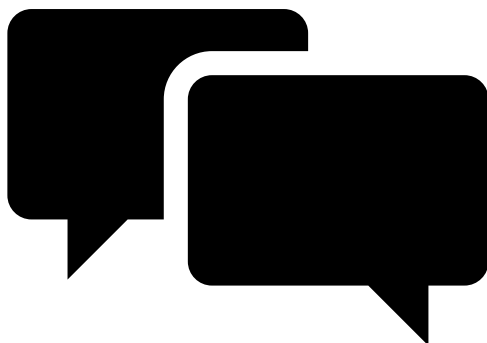


O ChatBot como o próprio nome já diz, robôs que conversam. Eles são desenvolvidos para se comunicar com os usuários e responder as perguntas dos clientes, grande parte das vezes através de uma Inteligência Artificial(IA).

ChatBot da copa

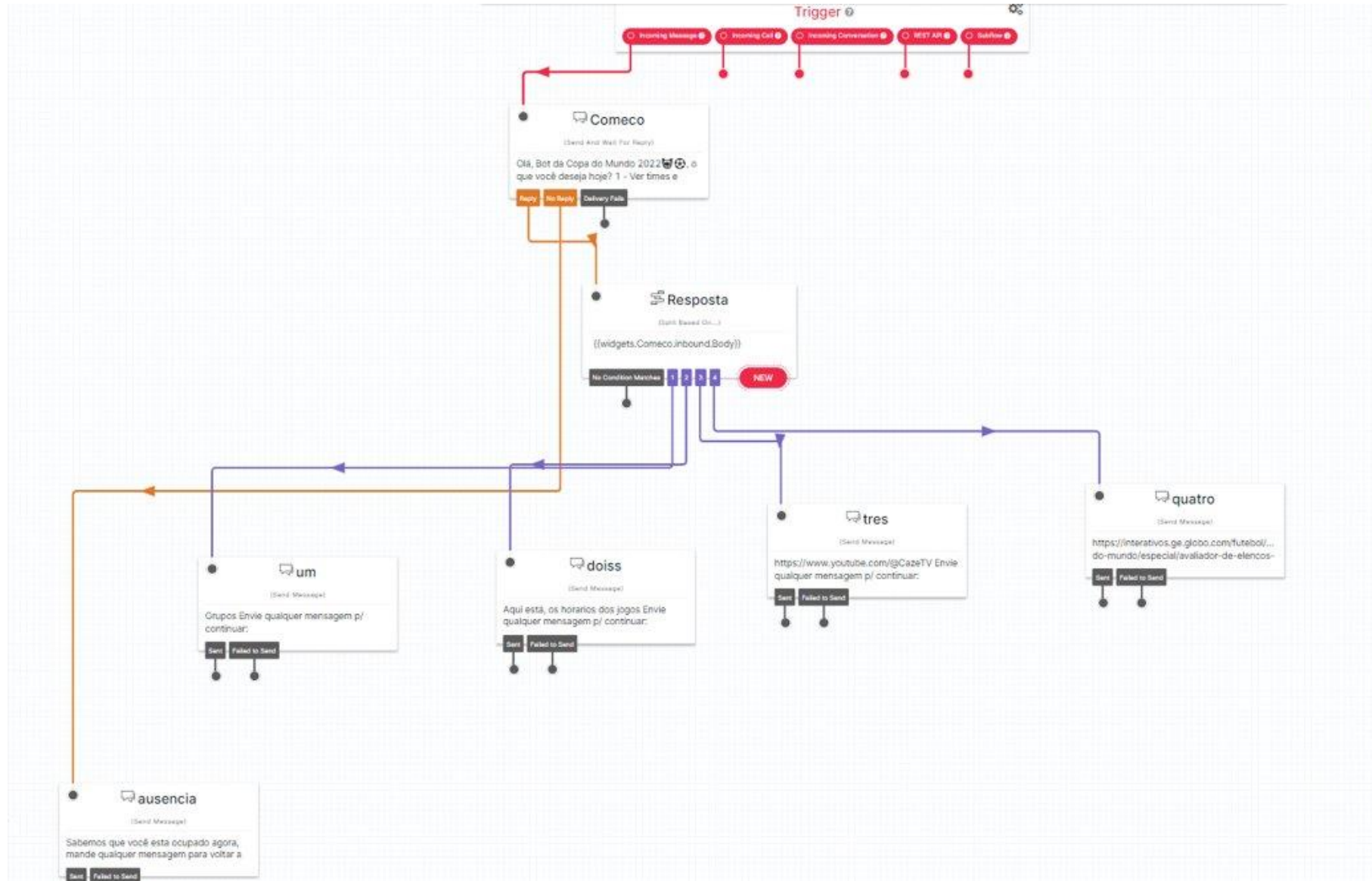


Como fizemos o bot na twilio?



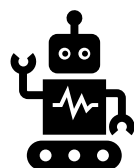
- Funciona basicamente como uma pipeline, com caixas que estruturam as conversas.
- Criamos uma caixa para iniciar a conversa, após isso separamos a conversa pela opção escolhida, criamos também uma mensagem de ausência caso o usuário demore a responder

ChatBot da Twilio



Bot mal-intencionado

Bot mal-intencionado



Qualquer bot que viole as intenções do proprietário do site ou os termos de serviço, são considerados bots maliciosos. Embora algumas dessas atividades sejam ilegais, os bots não precisam infringir nenhuma lei para serem considerados maliciosos.

Além disso, o tráfego excessivo de bots pode sobrecarregar os recursos de um servidor web, reduzindo ou interrompendo o serviço para os usuários humanos legítimos que estejam tentando usar um site ou um aplicativo. Às vezes, isso é intencional e assume a forma de um ataque DoS ou DDoS

Criação do bot

Passo a Passo

1. Crie uma conta no site da Twilio
2. Entre em *Studio*, entre em *Overview* e crie um novo *Flow*
3. Crie um conjunto de pipeline usando as ferramentas oferecidas
4. Entre em *Messaging*, entre em *Settings*, logo entre em *WhatsApp SandBox* , por fim envie a mensagem obrigatória para o numero disponibilizado
5. Teste o seu Bot

Passo a Passo

1. Foi pensado nas funcionalidades do bot e como ele seria;
2. Escolheu-se a linguagem de programação a ser usada (Python);
3. Foi utilizado duas bibliotecas durante o Desenvolvimento do projeto:
 - Selenium: Para o desenvolvimento geral do Back-End do projeto;
 - PySimpleGUI: Para o desenvolvimento das janelas executáveis do projeto.
4. Foi criado uma intro para a janela do bot;
5. Por fim criou-se o executável do projeto para a instalação/utilização do bot em outras máquinas;

Ambiente de desenvolvimento



O que é uma IDE?

IDE vem do inglês Integrated Development Environment, ou seja, um ambiente de desenvolvimento integrado é um software para criar aplicações que combina ferramentas comuns de desenvolvimento em uma única interface gráfica.

Para criar o nosso bot nós usamos a IDE “Visual Studio Code”.

Essa IDE possibilita desenvolver em qualquer linguagem de programação e baixar extensões que facilitam o processo de criação do código.



Visual Studio Code

Ambiente de desenvolvimento



O que é a Twilio

Twilio é uma empresa americana com sede em San Francisco, Califórnia, que fornece ferramentas de comunicação programáveis para fazer e receber chamadas telefônicas, enviar e receber mensagens de texto e realizar outras funções de comunicação usando suas APIs de serviço da web.

Esse serviço cognitivo nos permite criar um ChatBot através de um pipeline



Tecnologias utilizadas

Python

Utilizamos a linguagem de programação Python por ser conhecida pela facilidade na sintaxe e na criação de bots.

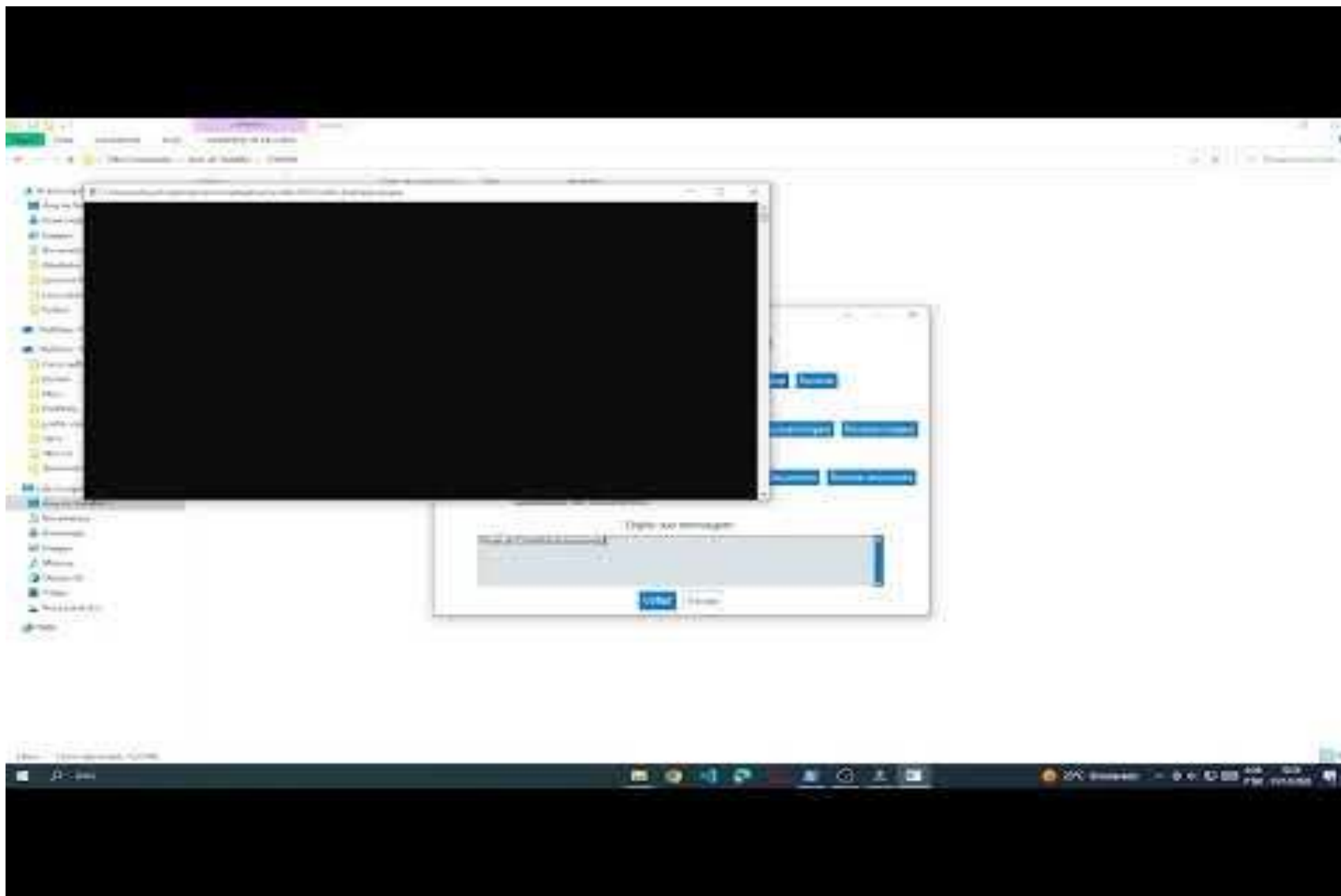
GitHub

Utilizamos o GitHub para todos da squad terem acesso ao código e para termos o controle de versão de cada alteração que fizemos.

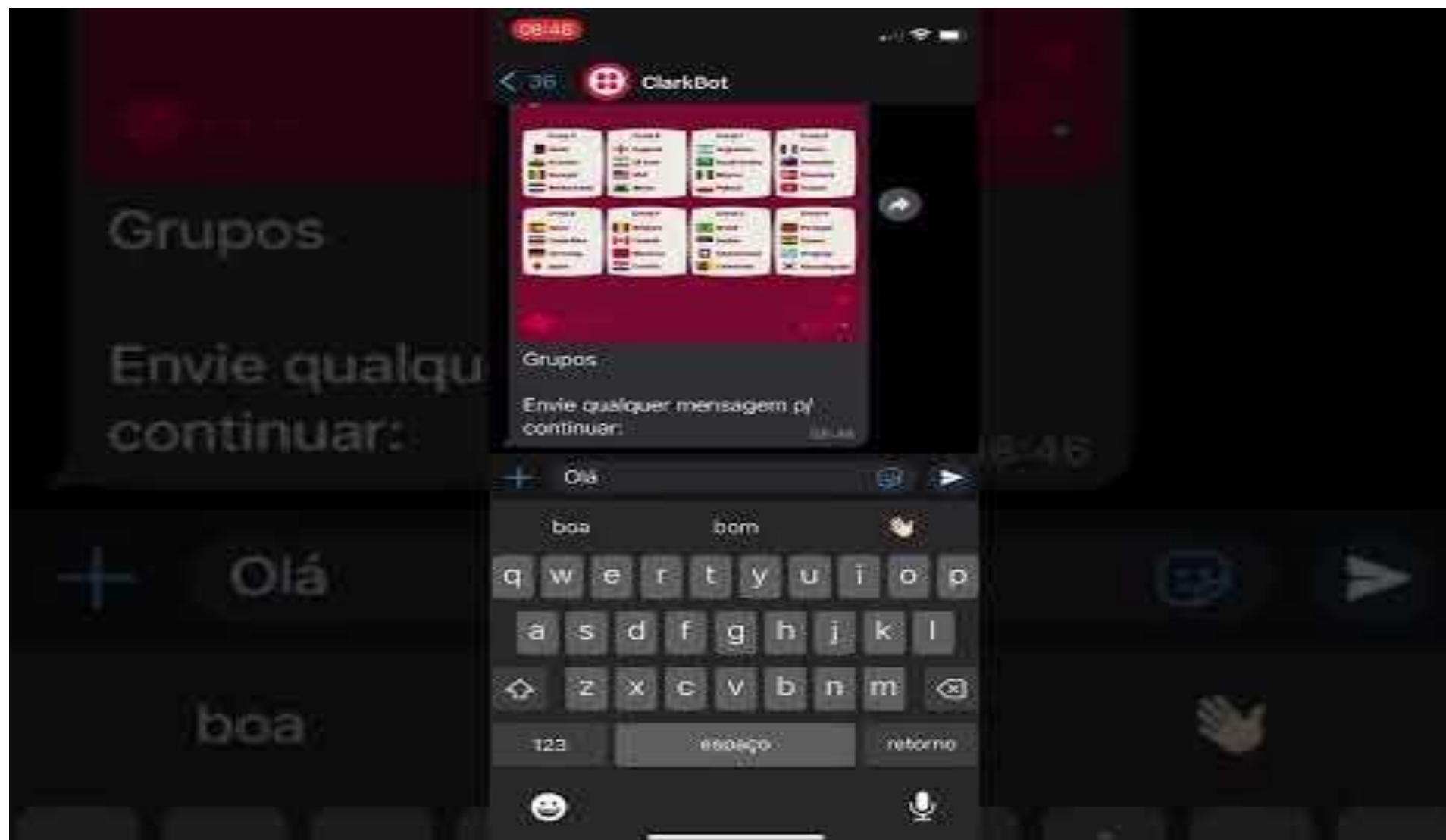
Git

Utilizamos o Git para versionar nosso código no GitHub e criar ramificações para teste.

Bot do WhatsApp



Bot Conversacional



Funcionalidades

ClerkBot

- Armazena dados de texto, vídeo, fotos e documentos;
- Abre o WhatsApp;
- Seleciona os contatos definidos;
- Por fim, envia as mensagens desejadas.


Bot Twilio

- Recebe mensagem de saudação para iniciar a conversa;
- Envia mensagem oferecendo quatro opções de respostas;
- Respondendo com a opção desejada.

Código fonte

Front-End

Fizemos o Front e o Back em páginas separadas para conseguir ter melhor legibilidade e separação do código



```
1  import PySimpleGUI as sg
2  from backChatBot import sendMessage
3  from time import sleep
4  from tkinter import *
5  from tkinter.ttk import *
6  import tkVideoPlayer
```

Janela inicial

```
13 # Criar janela e layouts
14 def janelaInicial():
15     sg.theme('Reddit')
16     layout = [
17         [sg.Image(source="assets/logo_def.png")],
18         [sg.Button("Bot", size=(15,20), font="arial 16 ", button_color='#5271FF'), sg.Button("Como funciona?", font="arial 16 ", :
19     ]
20     return sg.Window('ClerkBot', layout=layout, element_justification="center", finalize=True, size=(720, 500), background_color=
21
```

Janela do bot



```
22 ~ def janelaEnviar():
23     sg.theme('Reddit')
24     layout = [
25         [sg.Text("Enviar mensagens", font="arial 24", justification="center")],
26         [sg.Text("Digite o nome do contato e/ou grupo", font="arial 10")],
27         [sg.Text("Contato:", font="arial 12"), sg.Input(key="contato", size=(25,2), font="arial 12"), sg.Button("Adicionar"),
28         [sg.Text("Contatos:", font="arial 12"), sg.Text(f"{' ' if len(listContato) < 0 else listContato}", key="contatos", for
29         [sg.Text("Imagem ou Video:", font="arial 12"), sg.Input(key="img", size=(25,2), font="arial 12"), sg.FileBrowse(), sg.
30         [sg.Text("Quantidade de imagem:", font="arial 12"), sg.Text(f"{len(listImages)}", font="arial 12", key="imgs", size=(30,
31         [sg.Text("Documento:", font="arial 12"), sg.Input(key="doc", size=(25,2), font="arial 12"), sg.FileBrowse(), sg.Button
32         [sg.Text("Quantidade de documentos:", font="arial 12"), sg.Text(f"{len(listDocs)}", font="arial 12", key="documentos",
33         [sg.Text("Digite sua mensagem:", font="arial 12")],
34         [sg.Multiline(key="newMessage", size=(80,5))],
35         [sg.Button("Voltar", font="arial 13"), sg.Button("Enviar", font="arial 13")],
36     ]
37
38     return sg.Window('ClerkBot', Layout=layout, element_justification="center", finalize=True, icon='assets/logo_robot.ico')
```

Janelas de como funciona

```
50 def help1():
51     sg.theme('Reddit')
52     layout = [
53         [sg.Image(source="assets/ajuda1.png")],
54         [sg.Button("Voltar", size=(10), font="arial 16 ", button_color='#5271FF'),sg.Button("Continuar", size=(10), font="arial 16 "
55     ]
56     return sg.Window('ClerkBot', Layout=layout, element_justification="center", finalize=True, size=(720, 500), background_color='#0
57 def help2():
58     sg.theme('Reddit')
59     layout = [
60         [sg.Image(source="assets/ajuda2.png")],
61         [sg.Button("Voltar", size=(10), font="arial 16 ", button_color='#5271FF'),sg.Button("Continuar", size=(10), font="arial 16 "
62     ]
63     return sg.Window('ClerkBot', Layout=layout, element_justification="center", finalize=True, size=(720, 500), background_color='#0
64 def help3():
65     sg.theme('Reddit')
66     layout = [
67         [sg.Image(source="assets/ajuda3.png")],
68         [sg.Button("Voltar", size=(10), font="arial 16 ", button_color='#5271FF'),sg.Button("Tela Inicial", size=(10), font="arial 1
69     ]
70     return sg.Window('ClerkBot', Layout=layout, element_justification="center", finalize=True, size=(720, 500), background_color='#0
```

Back-End

```
1 from time import sleep
2 import pyautogui
3 from selenium.webdriver.chrome.service import Service
4 from selenium.webdriver.support import expected_conditions as EC
5 from selenium.webdriver.support.wait import WebDriverWait
6 from selenium.webdriver.common.by import By
7 from webdriver_manager.chrome import ChromeDriverManager
8 from selenium import webdriver
```

Como biblioteca principal, utilizamos o Selenium, mas utilizamos o PyAutoGui para alguns detalhes.

Docs String



```
16 def sendMessage(message, contacts, images, documents):
17     """
18     Abre o navegador no WhatsApp Web, espera a autenticação do usuário (QR code)
19     e envia as mensagens, imagens e arquivos para todos os usuários informados.
20
21     Parâmetros
22     -----
23     ``message`` : string
24         Mensagem a ser enviada.
25     ``contacts``: array
26         Vetor com todos os contatos.
27     ``images``: array
28         Vetor com todas as imagens.
29     ``documents``: array
30         Vetor com todos os documentos.
31
32     Retorno
33     -----
34     | None
35
36     """
37
```


Docs String

```
16 def sendMessage(message, contacts, images, documents):  
17     Abre o navegador no WhatsApp Web, espera a autenticação do usuário (QR code) e  
18     envia as mensagens, imagens e arquivos para todos os usuários informados.  
19
```

Parâmetros

```
22     message : string  
23         Mensagem a ser enviada.  
24     contacts : array  
25         Vetor com todos os contatos.  
26     images : array  
27         Vetor com todas as imagens.  
28
```

Abrir o navegador e instalar o driver

```
38 # Download e definição do driver
39 servico = Service(ChromeDriverManager().install())
40 navegador = webdriver.Chrome(service=servico)
41 sleep(10)
42 navegador.get("https://web.whatsapp.com/")
43 WebDriverWait(navegador, timeout=3600).until(EC.presence_of_element_located((By.ID, 'pane-side')))
44 sleep(3)
```

Henrique de Moraes Botelho da Silva, ontem • Video inicial

Execução do código

Referências

Referências usadas



Video twilio: <https://youtu.be/EE6ecyi9zSU>

PySimpleGUI: <https://www.pysimplegui.org/en/latest/>

Selenium : <https://selenium-python.readthedocs.io/>



Guilherme
Henrique Botelho
Lucas Pereira
Luis
Nicolas Gomes

Escola SENAI Suíço-Brasileira Paulo Ernesto Tolle
R. Bento Branco de Andrade Filho, 379 - Santo
Amaro, São Paulo - SP, 04757-000