

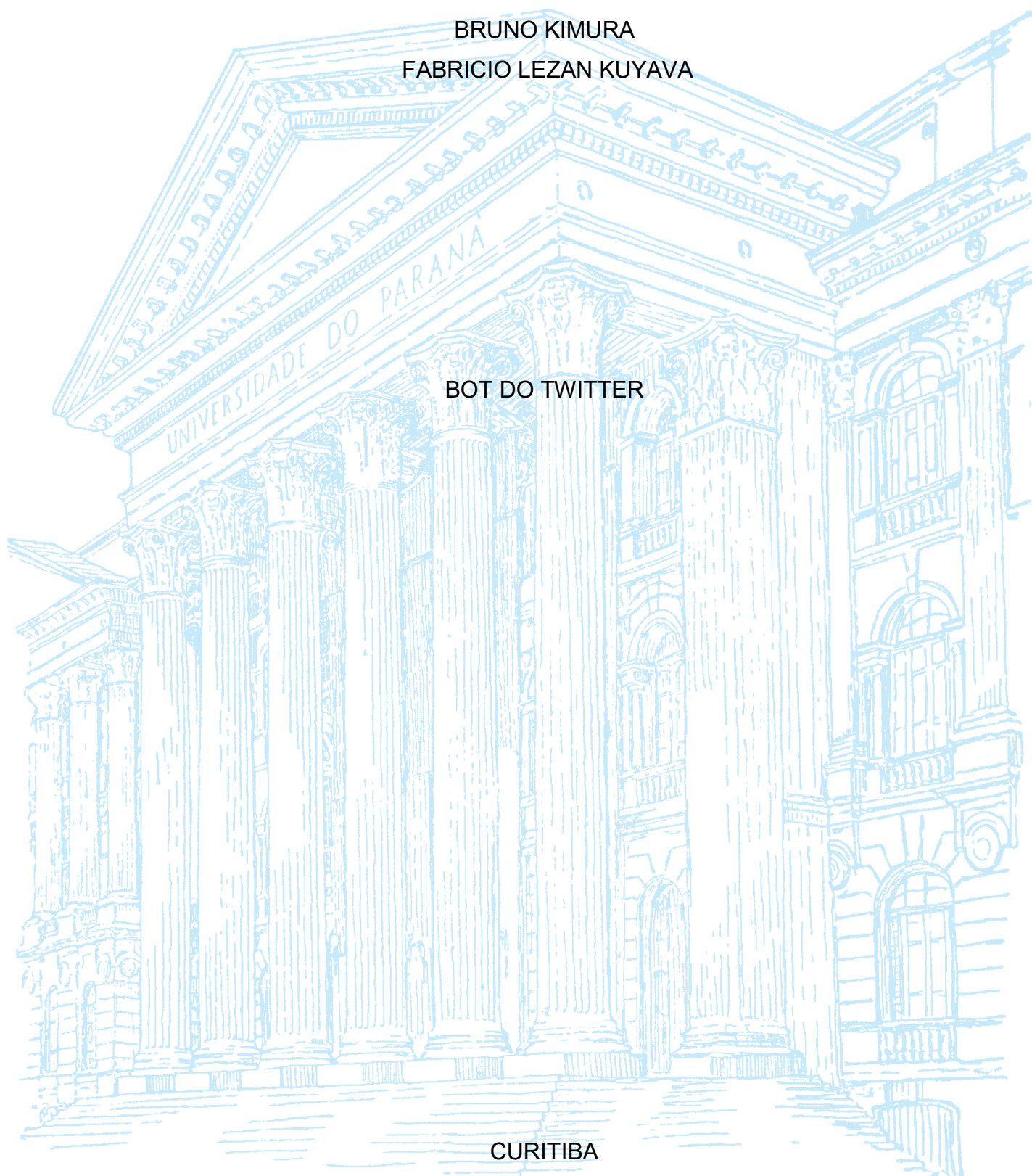
UNIVERSIDADE FEDERAL DO PARANÁ

BRUNO KIMURA
FABRICIO LEZAN KUYAVA

BOT DO TWITTER

CURITIBA

2019



BRUNO KIMURA (GRR20194159)
FABRICIO LEZAN KUYAVA (GRR20194212)

BOT DO TWITTER

Trabalho apresentado como requisito parcial à obtenção de nota, Curso de Programação, Setor de Ciências Agrárias, Universidade Federal do Paraná.

Orientador: Prof. Jackson de Lima

CURITIBA

2019

RESUMO

Esse trabalho foi criado a partir das ideias dos alunos Fabricio Lezan Kuyava e Bruno Kimura, com o intuito de facilitar a vida de usuários do Twitter que desejam obter respostas rápidas sobre algum conteúdo específico. O @Botkimura1 foi o bot criado para o usuário acessar essa matriz de dados e funções, a fim auxiliar e responder o usuário quando requisitado. Nosso trabalho criou forma graças aos conhecimentos adquiridos nas aulas de Programação, as aulas de Funções, Matrizes, Entrada de dados, enfim, todas as aulas e explicações do professor tiveram um ponto importante na montagem desse trabalho. As monitorias feitas também auxiliaram bastante nosso grupo, tanto nas dúvidas recorrentes as provas quanto na montagem do trabalho. Nosso programa tem 69 linhas de programação e se apresenta em perfeito estado de funcionamento, sendo específico e direto. O relatório foi montado no formato do TCC da UFPR, e menciona todas as coisas feitas durante os momentos da programação. Os slides da apresentação foram montados com embasamento de todo o processo de programação e montagem do relatório, a fim de mostrar todo o processo desde seu início até o final.

LISTA DE FIGURAS

FIGURA 1 – DADOS DO BOT	Erro! Indicador não definido.
FIGURA 2 – RESPONDER AS MENÇÕES	Erro! Indicador não definido.
FIGURA 3 – ARMAZENAR A ID	Erro! Indicador não definido.
FIGURA 4 – MOSTRAR A ID.....	9
FIGURA 5 – IMPORTAÇÃO DE BIBLIOTECAS.....	10
FIGURA 6.1 – INSERÇÃO DE DADOS.....	10
FIGURA 6.2 – DADOS DA CONTA DO TWITTER.....	11
FIGURA 7 – FUNÇÕES.....	11
FIGURA 8 – CHARADAS E MÚSICAS.....	12
FIGURA 9 – DECLARAÇÃO DE VARIÁVEIS.....	12
FIGURA 10 – RESPONDER OS TWEETS.....	13
FIGURA 11 – LOOP.....	14
FIGURA 12 – RESULTADO 1.....	15
FIGURA 13 – RESULTADO 2.....	15

SUMÁRIO

1	INTRODUÇÃO	6
1.1	JUSTIFICATIVA.....	6
1.2	OBJETIVOS.....	6
1.2.1	Objetivo geral.....	Erro! Indicador não definido.
1.2.2	Objetivos específicos.....	Erro! Indicador não definido.
2	FUNDAMENTAÇÃO TEÓRICA.....	7
3	METODOLOGIA.....	8
3.1	BIBLIOTECAS UTILIZADAS.....	8
3.2	FUNÇÕES CRIADAS.....	8
3.3	EXPLICAÇÕES ADICIONAIS.....	10
4	APRESENTAÇÃO DE RESULTADOS.....	15
5	CONSIDERAÇÕES FINAIS.....	16
5.1	RECOMENDAÇÕES PARA TRABALHOS FUTUROS.....	16
	REFERÊNCIAS.....	17

1 INTRODUÇÃO

Atualmente os maiores meios de comunicação e entretenimento são as redes sociais. Em cima disso, o Bot Kimura foi criado para entreter os usuários do Twitter, uma das redes mais utilizadas no momento. A intenção deste programa é desenvolver uma personalidade robótica para interagir com os usuários quando requisitado. Este programa consegue utilizar todas as funções da rede social, como curtir, retweetar, mandar mensagens, etc.

1.1 JUSTIFICATIVA

A finalidade deste trabalho é meramente voltada ao entretenimento, fornecendo aos usuários do Twitter uma personalidade robótica que interage com seus seguidores.

1.2 OBJETIVOS

O objetivo do Bot é a instantânea interação entre o usuário e a inteligência artificial. Dando parâmetros para que a inteligência possa reconhecer uma palavra-chave, e assim, fornecer uma resposta conforme o pedido do usuário.

1.2.1 OBJETIVO GERAL

Dar aos usuários a oportunidade de receber uma mensagem aleatória, conforme o conteúdo pedido, sem que o usuário precise sair da rede social.

2 FUNDAMENTAÇÃO TEÓRICA

O Twitter é uma rede social que permite aos usuários o rápido compartilhamento de mensagens, mídia e conteúdo em geral, usualmente encarada como um meio de entretenimento. O aplicativo, utilizado tanto para o meio profissional quanto pessoal, atualmente conta com 316 milhões de usuários ativos todos os meses.

Por sua praticidade e agilidade no compartilhamento de informações, o Twitter é considerado o “SMS da internet”. Por conta desta popularidade e da rápida expansão de informações dentro da rede social normalmente não formal, é um ótimo meio para a atuação de um bot voltado para o entretenimento.

3 METODOLOGIA

Para a realização deste trabalho, foram utilizados diversos tutoriais para a utilização das bibliotecas.

Para o melhor aprendizado das diferentes funções que a biblioteca tweepy pode fornecer, foi feita a leitura da documentação do programa. Dessa forma, pôde se extrair o máximo potencial fornecido pelos desenvolvedores.

3.1 BIBLIOTECAS UTILIZADAS

Para esse programa foram utilizadas duas bibliotecas: Tweepy e Random.

O Tweepy é uma biblioteca desenvolvida pelo próprio Twitter para a linguagem python. A partir dessa biblioteca é possível automatizar funções que o usuário pode fazer, como curtir, compartilhar e escrever postagens.

A biblioteca Random permite que a inteligência artificial selecione um elemento aleatório de uma lista.

3.2 FUNÇÕES CRIADAS

- Dados do bot

```

4  #DADOS DA CONTA DO BOT NO TWITTER
5  CONSUMER_KEY = 'ur9ZK4X5R6oEVjRZtbRCBvNXY'
6  CONSUMER_SECRET = 'nekk9Th72aA9dsM2LZ153vcUaA5ncmIbX1ynu9lJ9ri9b2RiWM'
7  ACCESS_KEY = '114080557921825996;8-ps7Utf5cATAUGeIXSosJXvkzZ28VHC'
8  ACCESS_SECRET = 'bBfQ1oFrpcXoNX0SYpDX6XUfEfCEaiL2OwRfTlBCJB8nK'
9
10 #INSERIR AS INFORMAÇÕES DA CONTA DO TWITTER NO PROGRAMA
11 auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
12 auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
13 api = tweepy.API(auth)

```

Figura 1

- Responder as menções

```

47 for mention in mentions:
48
49     print(str(mention.id) + ' - ' + mention.text)
50     id_repetido = mention.id
51
52     armazenar_id(id_repetido, file_name)
53
54     if 'piada' in mention.text.lower():
55         print("Contando uma piada!")
56         api.update_status('@' + mention.user.screen_name + ' ' + random.choice(charadas), mention.id)
57
58     elif 'amo esse bot' in mention.text.lower():
59         print(":o")
60         api.retweet(mention.id)
61
62     elif 'testando' in mention.text.lower():
63         print("Opa!")
64         api.update_status('@' + mention.user.screen_name + ' Olá!', mention.id)
65
66     elif 'musica' in mention.text.lower():
67         print("Vou cantar uma música pra você.")
68         api.update_status('@' + mention.user.screen_name + ' ' + random.choice(musicas), mention.id)
69

```

Figura 2

- Armazenar a ID do usuário já respondido

```

24 def armazenar_id(arquivo):
25     escrever_id = open(arquivo, 'w')
26     escrever_id.write(str(id_repetido))
27     escrever_id.close()
28     return

```

Figura 3

- Mostrar a ID do usuário já respondido

```

18 def devolver_id(arquivo):
19     ler_id = open(file_name, 'r')
20     id_repetido = ler_id.read().strip()
21     ler_id.close()
22     return id_repetido

```

Figura 4

3.3 EXPLICAÇÕES ADICIONAIS

1. Importação das bibliotecas

```
1 import tweepy
2 import random
```

Figura 5

2. Inserção de dados da conta do Bot dentro do programa Python

```
4 #DADOS DA CONTA DO BOT NO TWITTER
5 CONSUMER_KEY = 'ur9ZK4X5R6oEVjRZtbRCBvNXY'
6 CONSUMER_SECRET = 'nekk9Th72aA9dsM2LZ153vcUaA5ncmIbX1ynu9lJ9ri9b2RiWM'
7 ACCESS_KEY = '114080557921825996;8-ps7Utf5cATAUGeIXSosJXvkzZ28VHC'
8 ACCESS_SECRET = 'bBfQ1oFrpcXoNX0SYpDX6XUfEfCEaiL2OwRfTlBCJB8nK'
9
10 #INSERIR AS INFORMAÇÕES DA CONTA DO TWITTER NO PROGRAMA
11 auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
12 auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
13 api = tweepy.API(auth)
```

Figura 6.1

Para inserir a conta do Bot dentro do programa, é necessário coletar os dados da conta fornecidos pelo próprio Twitter:

Consumer API keys

AhpIWmebOsOtBCZsjgSSuZxVA (API key)

97D9jpoR6M4bhwkYecAh6mlwVda0LbwU7AJaGeSyBe3508nuOk (API secret key)

Regenerate

Access token & access token secret

1138937981409083392-Uvp3UNxBMVIGeFT1XRuAemCC1qpDby (Access token)

ouaPIkgMXcgkIw3WZVIPz6P1YxynoVS5IzUtUzsvTxAe4 (Access token secret)

Read and write (Access level)

Figura 6.2

Cada dado foi colocado dentro de uma variável, que em seguida é processado pelos comandos da biblioteca tweepy.

Feitos esses passos, o Bot estará devidamente conectado ao programa.

3. Funções para armazenamento de ID's de tweets já respondidos

Para evitar que o Bot responda continuamente o mesmo tweet várias vezes, é necessário armazenar o ID de cada tweet dentro de um arquivo: `ultimos_id.txt`.

```
16 arquivo = 'armazenamento_de_id.txt'
17
18 def armazenar_id(arquivo):
19     escrever_id = open(arquivo, 'w')
20     escrever_id.write(str(id_repetido))
21     escrever_id.close()
22     return
23
24 def devolver_id(arquivo):
25     ler_id = open(arquivo, 'r')
26     id_repetido = ler_id.read().strip()
27     ler_id.close()
28     return id_repetido
```

Figura 7

Na linha 19, o programa irá abrir o arquivo `armazenamento_de_id.txt`.

Em seguida, na linha 20, o programa irá inserir o ID do tweet que está sendo respondido.

Na linha 21, o programa fecha o arquivo `txt`.

Na linha 25, o arquivo é aberto novamente, para que o ID inserido anteriormente possa ser lido.

Se já houver o ID dentro do arquivo, quer dizer que o programa já respondeu este tweet, portanto não é necessário responde-lo novamente.

4. Listas de charadas e músicas

```

30 charadas = ['O que é um ponto verde na parede?' + 60*' ' + 'Uma ervilha caindo de paraquedas',
31             'O que é uma minhoca com sono?' + 55*' ' + 'Uma dorminhoca',
32             'O que o pato disse pra pata durante o jogo?' + 50*' ' + 'Estamos empatados',
33             'Qual o vinho que não tem álcool?' + 50*' ' + 'O-vinho de codorna',
34             'Qual a comida que liga e desliga?' + 60*' ' + 'Strog(on)off']
35
36 musicas = ['Portal - Still alive https://www.youtube.com/watch?v=Y6ljFaKRTTrI&t=28s',
37            'Portal 2 - Want you gone https://www.youtube.com/watch?v=dVVZaZ8y06o',
38            'Poppy - Computer boy https://www.youtube.com/watch?v=5Ao5mg11xIk',
39            'hello world https://www.youtube.com/watch?v=Yw6u6YkTgQ4',
40            'Computer error song https://www.youtube.com/watch?v=mKkLjJHwRec',
41            'Elektronik supersonik https://www.youtube.com/watch?v=MNYG-xu-75Q']

```

Figura 8

Como exemplo, foi inserido dentro de listas algumas piadas e músicas que podem ser mandadas ao usuário aleatoriamente.

5. Declaração de variáveis

```

45 ultimo_id = devolver_id(arquivo)
46 mentions = api.mentions_timeline(ultimo_id)

```

Figura 9

A variável 'ultimo_id' chama a função devolver_id, ou seja, irá fazer a leitura do arquivo armazenamento_de_id.txt.

A variável 'mentions' utiliza um comando do tweepy, que irá mostrar todos os tweets em que o bot foi mencionado, exceto aqueles já respondidos.

6. Responder os tweets

```

49 for mention in mentions:
50
51     print(str(mention.id) + ' - ' + mention.text)
52     id_repetido = mention.id
53
54     armazenar_id(id_repetido, arquivo)
55
56     if 'piada' in mention.text.lower():
57         print("Contando uma piada!")
58         api.update_status('@' + mention.user.screen_name + ' ' + random.choice(charadas), mention.id)
59
60     elif 'amo esse bot' in mention.text.lower():
61         print(":o")
62         api.retweet(mention.id)
63
64     elif 'testando' in mention.text.lower():
65         print("Opa!")
66         api.update_status('@' + mention.user.screen_name + ' Olá!', mention.id)
67
68     elif 'musica' in mention.text.lower():
69         print("Vou cantar uma música pra você.")
70         api.update_status('@' + mention.user.screen_name + ' ' + random.choice(musicas), mention.id)

```

Figura 10

Finalmente, a seção onde o usuário irá receber as respostas.

Na linha 51, será imprimido na tela de comando (cmd) a ID do tweet em que o bot foi mencionado e a mensagem dentro dela.

Na linha 52, esse tweet que irá sendo respondido vai ser guardado na variável 'id_repetido', que será inserido dentro do arquivo 'armazenamento_de_id.txt', na linha 54.

Depois, estão as respostas:

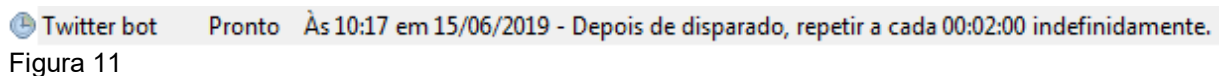
Caso o tweet mencionado tenha "piada" em seu texto, o bot irá tweetar (api.update_status) como resposta (mention.id) ao usuário (mention.user.screen_name) uma resposta aleatória, utilizando da biblioteca random para escolher um item aleatório da lista de charadas.

Os outros comandos irão realizar a mesma tarefa, apenas mudando a palavra-chave procurada dentro do tweet do usuário e a resposta a ser emitida pelo bot.

Na linha 60, no entanto, foi criada uma ação diferente: caso o usuário escreva “eu amo esse bot”, o Bot irá retweetar esta mensagem, como uma demonstração de apreciação.

7. Loop

O programa, no entanto, só irá responder os usuários se o código for executado no prompt de comando. Para automatizar o processo de resposta, foi utilizado o agendador de tarefas do próprio Windows.



A screenshot of the Windows Task Scheduler interface. It shows a task named "Twitter bot" with a status of "Pronto" (Ready). The task is scheduled to run at "Às 10:17 em 15/06/2019" and is configured to "Depois de disparado, repetir a cada 00:02:00 indefinidamente." (After triggered, repeat every 00:02:00 indefinitely).

Dessa forma, o programa será executado de 2 em 2 minutos, independente se o computador está ligado ou não.

O usuário, então, pode receber a resposta em até 2 minutos.

4 APRESENTAÇÃO DOS RESULTADOS

O usuário, ao postar um tweet com alguma das palavras-chave, irá receber automaticamente a sua resposta.



Figura 12



Figura 13

4 CONSIDERAÇÕES FINAIS

Este bot, por apresentar uma proposta nada usual na rede social, é um programa chamativo; usuários curiosos podem sempre testar o bot e se surpreenderem com suas respostas.

4.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

O programa poderia apresentar funções mais diversas. Conseguir ler e compartilhar informações de sites, também chamado de “webscraping”, poderia ser extremamente útil para o programa.

Compartilhar certas notícias sobre a Universidade através do portal da UFPR ou postar quais as refeições que serão oferecidas no RU poderiam ser extremamente úteis para os usuários frequentes do Twitter. Dessa forma, além de proporcionar entretenimento, o bot também seria um bom meio de compartilhamento de informações.

Além disso, ao invés de se utilizar o Gerenciador de Tarefas do Windows, pode se colocar o código em algum compilador online. O compilador online fará a mesma função do Gerenciador de Tarefas, no entanto o computador do programador não precisa estar ligado para que ocorra o loop do programa.

REFERÊNCIAS

<http://docs.tweepy.org/en/latest/api.html#timeline-methods> – *documentação da biblioteca tweepy*

https://aporama.com.br/6-motivos-para-usar-o-twitter/?fbclid=IwAR29mzNUoLvXWBDV0OMJChpRF8FUpWUrejD_wTyC1-Dck6tWOnikVltrVI - *a importância e utilidades do Twitter*