

Construindo um chatBot no telegram

Professor: **Nator Junior Carvalho da Costa**

October 22, 2022

- 1 Como será nossa aplicação;
- 2 O que é um **ChatBot**;
- 3 Como criar uma Bot no telegram;
- 4 O que é uma API;
- 5 Criando a primeira API com FastAPI;
- 6 Criando um serviço que retorna o nome de um servidor público da UFC;
- 7 Criando um serviço que busca um nome de um servidor público da UFC e retorna o vínculo;

Como será nossa aplicação

Criando um chatBot

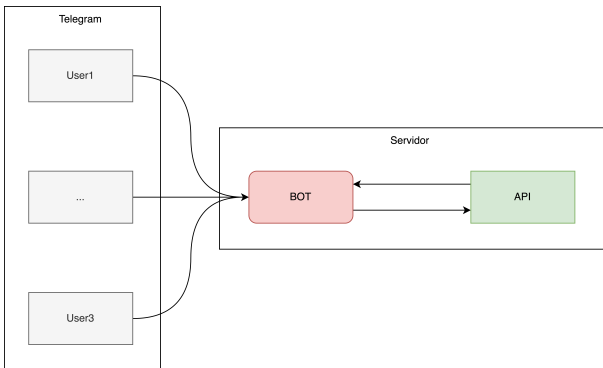
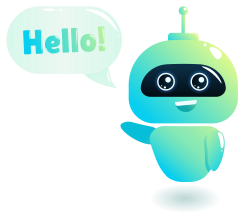


Figure: Interação com dos serviços.

O que é um chatbot

Criando um chatBot

No nível mais básico, um **chatbot** é um programa de computador que simula e processa conversas humanas (escritas ou faladas), permitindo que as pessoas interajam com dispositivos digitais como se estivessem se comunicando com uma pessoa real (Oracle, 2022).

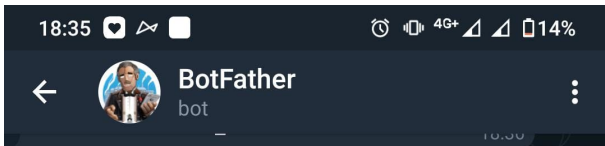




- Os chatbots no Telegram consistem de contas especiais que não requerem um número de telefone para serem configurados;
- Também não é necessária nenhuma autorização prévia para publicar um bot, mas seu funcionamento deve respeitar os Termos de Uso do serviço

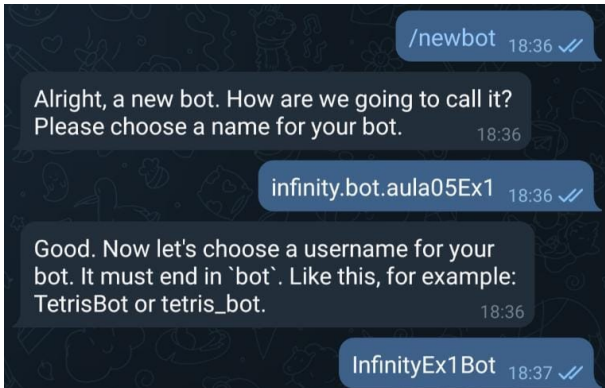


O primeiro passo é buscar o **BotFather** no Telegram, em seguida encaminhar uma mensagem **/start**, em que ele deve mostrar as opções disponíveis.

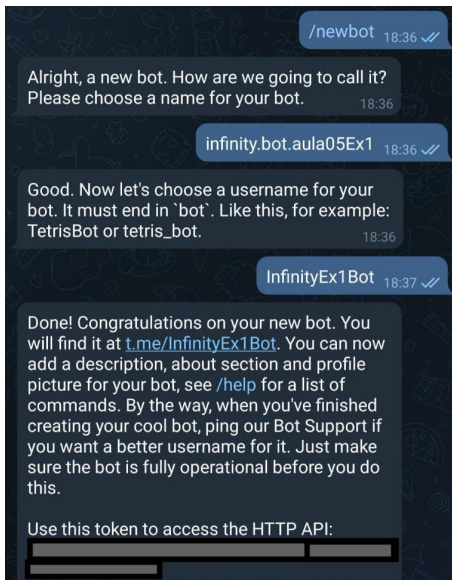




O próximo passo é encaminhar um comando **/newbot**, o BotFhater deverá perguntar o identificador do BOT, em seguida ele deverá perguntar o nome, que deve terminar em Bot, por exemplo **FranciscoBot**.



Criando o Token



Faça a instalação no terminal:

```
1 pip install python-telegram-bot -U --pre
```

Criando uma função para responder o comando **/start**.

```
1 async def start(update, context):  
2     await context.bot.send_message(  
3         chat_id=update.effective_chat.id,  
4         text="Olá, eu sou o BOT, como vai?"  
5     )
```

Criando uma função para responder o comando **/start**.

```
1 import logging
2 from telegram import Update
3 from telegram.ext import *
4 async def start(update, context):
5     await context.bot.send_message(
6         chat_id=update.effective_chat.id,
7         text="Olá, eu sou o BOT, como vai?"
8     )
9 app = ApplicationBuilder().token('TOKEN').build()
10 start_handler = CommandHandler('start', start)
11 app.add_handler(start_handler)
12 app.run_polling()
```

Criando um **BOT** que responde apenas o comando **/start**

```
1 http://dontpad.com/717aula22PrimeiroBot
```

Criando o segundo **BOT** recebe e trata todos os tipos de mensagens:

```
1 http://dontpad.com/717aula22SegundoBot
```



Para essa atividade considere a base de dados disponível em:
<http://dontpad.com/717aula22PandasFunction>

- 1 Retirar linhas duplicadas do dataframe;
- 2 Criar uma função que sorteia aleatoriamente um nome e retorna o histórico do mesmo com um máximo de 5 linhas;
- 3 Criar uma função que busque um nome na base e retorne as **uorg_lotacao**.

Instalando o FastAPI

```
1 pip install fastapi
```

Instalando o uvicorn

```
1 pip install "uvicorn[standard]"
```

crie um arquivo chamado **main.py** e inclua o texto abaixo

```
1 from fastapi import FastAPI
2 app = FastAPI()
3
4 app.get("/inicio")
5 def read():
6     return {"Hello": "World"}
```

Executar no terminal:

```
1 uvicorn main:app --reload
```

Rodar no *browser* **http://127.0.0.1:8000/inicio**

Criando uma API com o FastAPI

Criando uma API com o FastAPI

crie um arquivo chamado **main.py** e inclua o texto abaixo

```
1 from fastapi import FastAPI
2 def retorna_servidor_aleatorio():
3     pass #implementar
4 app = FastAPI()
5 app.get("/retorna_servidor_aleatorio")
6 def retorna_servidor():
7     return retorna_servidor_aleatorio()
```

Executar no terminal:

```
1 uvicorn main:app --reload
```

Rodar no *browser* **http://127.0.0.1:8000/retorna_servidor_aleatorio**

Criando uma API com o FastAPI

Criando uma API com o FastAPI

crie um arquivo chamado **main.py** e inclua o texto abaixo

```
1 from fastapi import FastAPI
2 def busca_servidor(nome):
3     pass #implementar
4 app = FastAPI()
5 @app.get("/retorna_servidor/{nome}")
6 def read_root(nome:str):
7     return busca_servidor(nome)
```

Executar no terminal:

```
1 uvicorn main:app --reload
```

Rodar no *browser* **http://127.0.0.1:8000/retorna_servidor/FRANCISCO**

Obrigado!

Construindo um chatBot no telegram

Professor: **Nator Junior Carvalho da Costa**

October 22, 2022