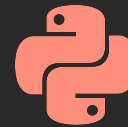




Bots para telegram

com pyrogram

Live de Python # 210



1. Bots e telegram

O que precisamos saber antes de codar?

2. O Pyrogram

Um básico sobre a biblioteca

3. Filtros

Filtrando tipo de mensagens

4. Tipos de mídia

Trabalhando com dados que não sejam texto

5. Decoradores

Nem só de mensagens vivemos

6. Deploy

Usando heroku



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3



Acássio Anjos, Ademair Peixoto, Adilson Herculanu, Alexandre Harano, Alexandre Lima, Alexandre Souza, Alexandre Takahashi, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysson Oliveira, Ana Carneiro, Ana Padovan, Andre Azevedo, André Rafael, André Rocha, Aquiles Coutinho, Arnaldo Turque, Aurelio Costa, Bloquearsites Farewall, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Bruno Ramos, Caio Nascimento, Carlos Alipio, Christiano Morais, Clara Battesini, Daniel Freitas, Daniel Haas, Danilo Segura, David Kwast, Delton Porfiro, Dhyeives Rodovalho, Diego Farias, Diego Guimarães, Dilenon Delfino, Dino Aguilár, Diogo Paschoal, Douglas Bastos, Douglas Braga, Douglas Zickuhr, Emerson Rafael, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Evandro Avellar, Everton Silva, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Felipe Rodrigues, Fernanda Prado, Fernando Rozas, Flávio Meira, Flavkaze Flavkaze, Franklin Silva, Gabriel Barbosa, Gabriel Simonetto, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Guilherme Piccioni, Gustavo Dettenborn, Gustavo Suto, Heitor Fernandes, Henrique Junqueira, Hugo Cosme, Igor Taconi, Israel Gomes, Italo Silva, Jair Andrade, Jairo Lenfers, Janael Pinheiro, João Lugão, João Paulo, João Rodrigues, Joelson Sartori, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jônatas Silva, José Gomes, Joseíto Júnior, Jose Mazolini, Juan Gutierrez, Juliana Machado, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Miranda, Leonardo Mello, Leonardo Nazareth, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Teixeira, Lucas Valino, Luciano Silva, Luciano Teixeira, Luiz Carlos, Luiz Junior, Luiz Lima, Luiz Paula, Luiz Perciliano, Maicon Pantoja, Maiquel Leonel, Marcelino Pinheiro, Marcelo Matte, Márcio Martignoni, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Marina Passos, Mateus Lisboa, Matheus Silva, Matheus Vian, Mírian Batista, Murilo Andrade, Murilo Cunha, Murilo Viana, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Otávio Barradas, Patricia Minamizawa, Paulo Braga, Paulo Tadei, Pedro Henrique, Pedro Pereira, Peterson Santos, P Muniz, Priscila Santos, Rafael Lopes, Rafael Rodrigues, Rafael Romão, Ramayana Menezes, Reinaldo Silva, Renato Veirich, Ricardo Silva, Riverfount Riverfount, Robson Maciel, Rodrigo Alves, Rodrigo Ferreira, Rodrigo Freire, Rodrigo Vaccari, Rodrigo Vieira, Rogério Sousa, Ronaldo Silva, Rui Jr, Samanta Cicilia, Sara Selis, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Curvelo, Thiago Moraes, Thiago Oliveira, Thiago S, Thiago Souza, Tiago Minuzzi, Tony Dias, Valdir Tegon, Victor Wildner, Vinícius Bastos, Vítor Gomes, Vitor Luz, Vlademir Souza, Vladimir Lemos, Walter Reis, Wellington Abreu, Wesley Mendes, William Alves, Willian Lopes, Wilson Neto, Wilson Rocha, Yury Barros



Obrigado você



Estrutuda e
credenciais

Telegram
v3
Bots

Telegram



Telegram é um aplicativo de mensagens, mensageiro instantâneo. Criado por Pavel Durov, lançado em 14 agosto de 2013.

Telegram conta com clientes de código aberto, GPL e um servidor com o código proprietário.

Por conta de seus clientes serem abertos, toda a comunicação com as APIs do telegram é bem documentada e aberta

Bots



Em 2015 o telegram abriu a plataforma para que todas as pessoas que tem uma conta no mensageiro pudesse criar bots.

Os bots pode participar de grupos para moderação, fazer pagamentos, interagir com usuários de forma isolada e etc...

Iniciar a jornada do bot



BORA!



Credenciais para o bot



[Home](#) [FAQ](#) [Apps](#) [API](#) [Protocol](#)



Delete Account or Manage Apps

Log in here to **manage your apps** using Telegram API or **delete your account**. Enter your number and we will send you a confirmation code via Telegram (not SMS).

Your Phone Number

Please enter your number in [international format](#)

Next

<https://my.telegram.org/apps>

Credenciais para o bot



[Home](#) [FAQ](#) [Apps](#) [API](#) [Protocol](#)



Delete Account or Manage Apps

Log in here to **manage your apps** using Telegram API or **delete your account**. Enter your number and we will send you a confirmation code via Telegram (not SMS).

Your Phone Number

+551843874388437 ([Incorrect?](#))

Confirmation code

Confirmation code

☐ Remember Me

Sign In

Credenciais para o bot



[Home](#) [FAQ](#) [Apps](#) [API](#) [Protocol](#)



Delete Account or Manage A

Log in here to **manage your apps** using Telegram or **delete your account**. Enter your number and we will send you a confirmation code via Telegram.

Your Phone Number

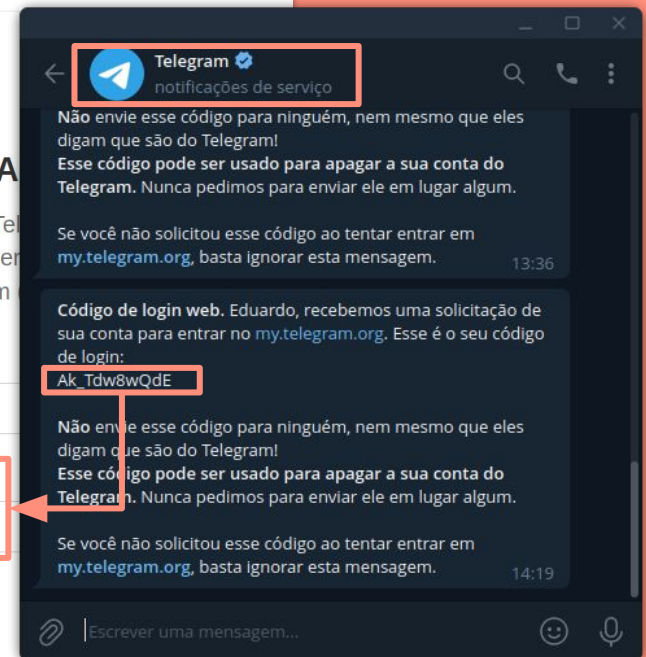
+551843874388437 (Incorrect?)

Confirmation code

Confirmation code

☐ Remember Me

Sign In



Credenciais para o bot



[Home](#) [FAQ](#) [Apps](#) [API](#) [Protocol](#)



Delete Account or Manage Apps

Log in here to **manage your apps** using Telegram or **delete your account**. Enter your number and we will send you a confirmation code via Telegram.

Your Phone Number

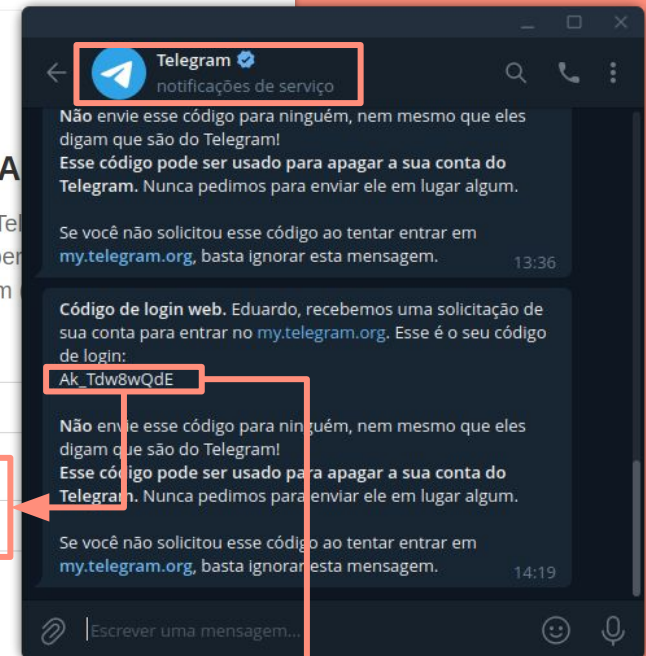
+551843874388437 (Incorrect?)

Confirmation code

Confirmation code

☐ Remember Me

Sign In



Ak_Tdw8wQdE

Credenciais para o bot



[Home](#) [FAQ](#) [Apps](#) [API](#) [Protocol](#)

Create new application

App title:

Short name:

alphanumeric, 5-32 characters

URL:

Platform:

- ☒ Android
- ☐ iOS
- ☐ Windows Phone
- ☐ BlackBerry
- ☐ Desktop
- ☐ Web
- ☐ Ubuntu phone
- ☐ Other (specify in description)

Description:

Create application

Cancel

Suas credenciais (salvem isso)



App configuration

App api_id:

123123123123



App api_hash:

asiudhsaiudhiuashdiusahiudhsaiuhdas



App title:

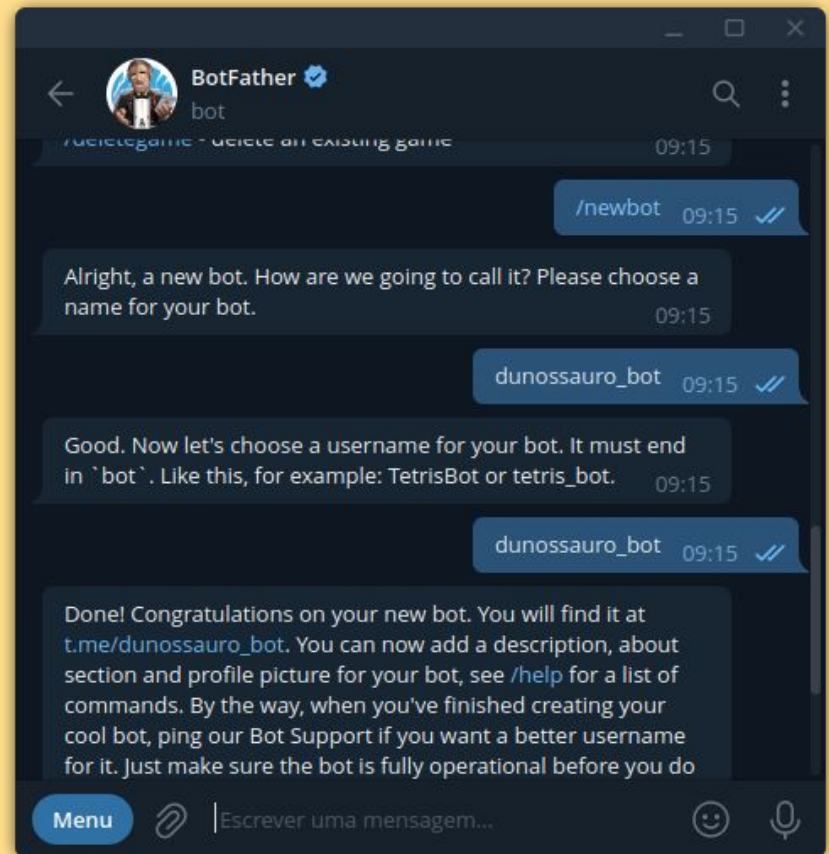
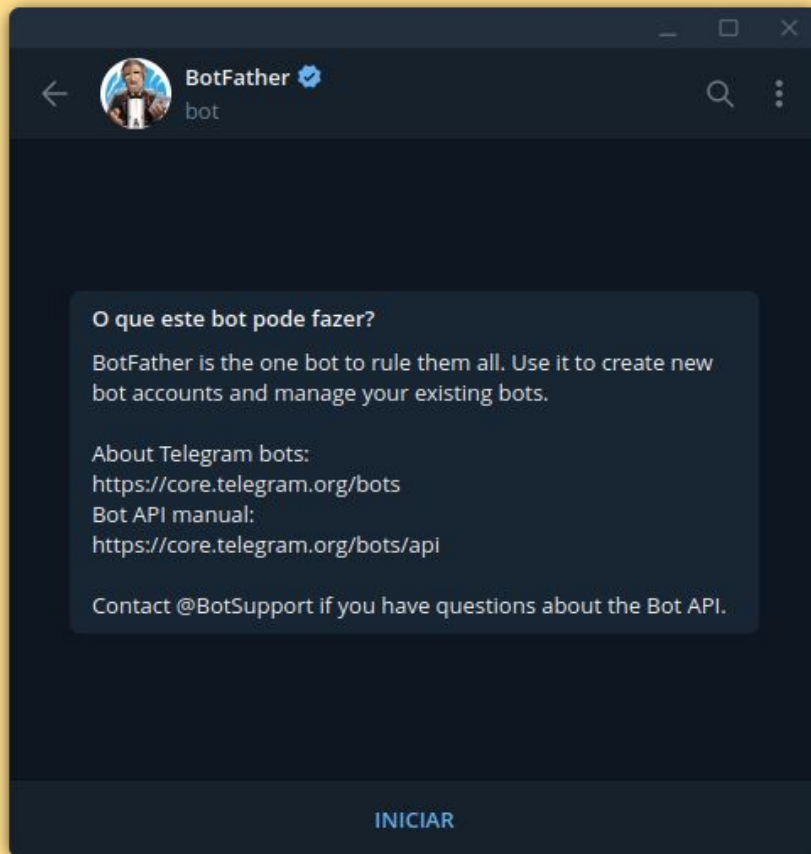
DunoBot

Short name:

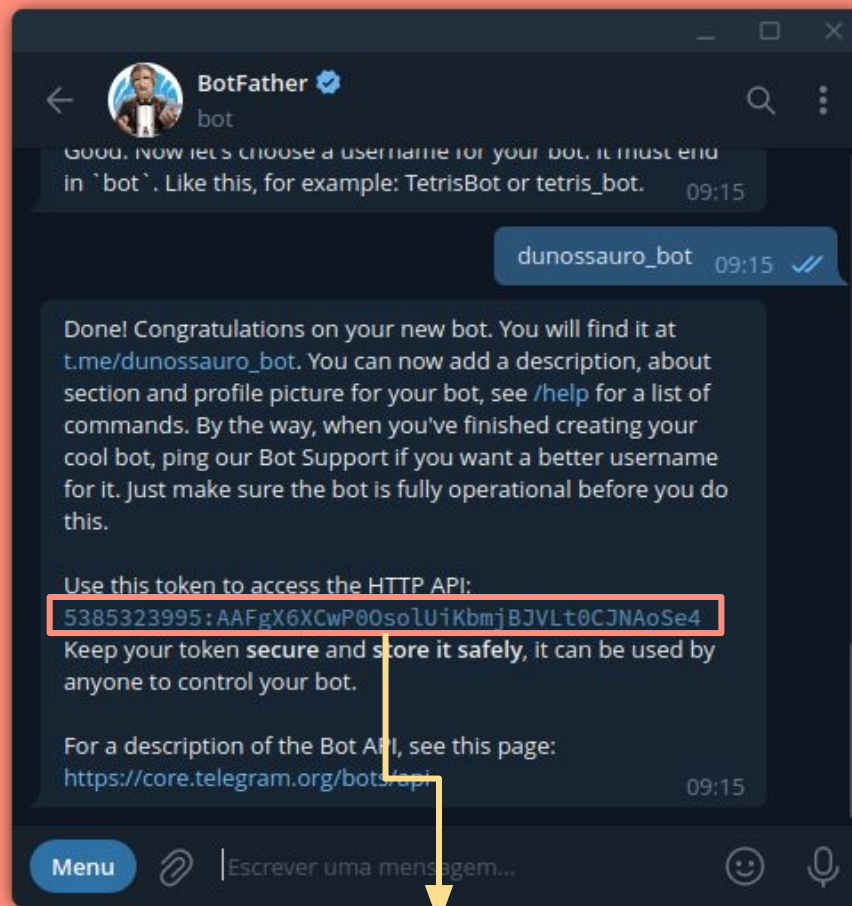
dunobot

alphanumeric, 5-32 characters

0 bot pai! (@BotFather)



A chave do nosso bot (salve isso tbm)



5385323995:AAFgX6XCwP00soLUiKbmjBJVLt0CJNAoSe4

juntando tudo no ambiente



```
# .env
```

```
TELEGRAM_API_ID="1273612783621"
```

```
TELEGRAM_API_HASH="667567sdtavsdhjgasvdymasfd76"
```

```
TELEGRAM_BOT_TOKEN="123123123:SAAS*YHD*I7hiuh3i4eu"
```

Py
rogram

A biblioteca que
vamos usar

Pyrogram

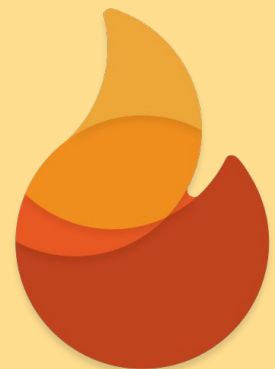


Pyrogram é uma biblioteca em python para criação de bots para telegram.

Inicialmente desenvolvido por @**delivrance** em milão na Italia.

Teve sua release inicial em 2018, em janeiro de 2022, teve sua primeira release estável. A versão 1.0

Atualmente está na versão 2.0.30, a que vamos usar e que foi lançada dia 21 de junho de 2022.



pip install pyrogram



Instalando!



```
pip install TgCrypto
```



Instalando!



tgCrypto é uma biblioteca que reescreve alguns algoritmos de criptografia em C, para ser mais rápido.

Também desenvolvida pelo time do pyrogram

```
pip install TgCrypto
```



Instalando!





`pip install python-dotenv`



Lendo variáveis de ambiente



Passando as chaves para o bot!

```
1  from os import getenv
2  from dotenv import load_dotenv
3  from pyrogram import Client
4
5  load_dotenv()
6
7  app = Client(
8      'dunossauro_bot',
9      api_id=getenv('TELEGRAM_API_ID'),
10     api_hash=getenv('TELEGRAM_API_HASH'),
11     bot_token=getenv('TELEGRAM_BOT_TOKEN')
12 )
```


Mandando a primeira mensagem



```
1  # exemplo_01.py
2  from asyncio import run
3
4  app = Client(...)
5
6  async def main():
7      await app.start()
8      await app.send_message('dunossauro', 'Olá, bb! 🐦')
9      await app.stop()
10
11
12  run(main())
```

Um handle para mensagens



```
1  # exemplo_02.py
2  app = Client(...)
3
4  @app.on_message( )
5  async def messages(client, message: Message):
6      print(message.chat.username, message.text)
7      await message.reply(message.text)
8
9  app.run( )
```

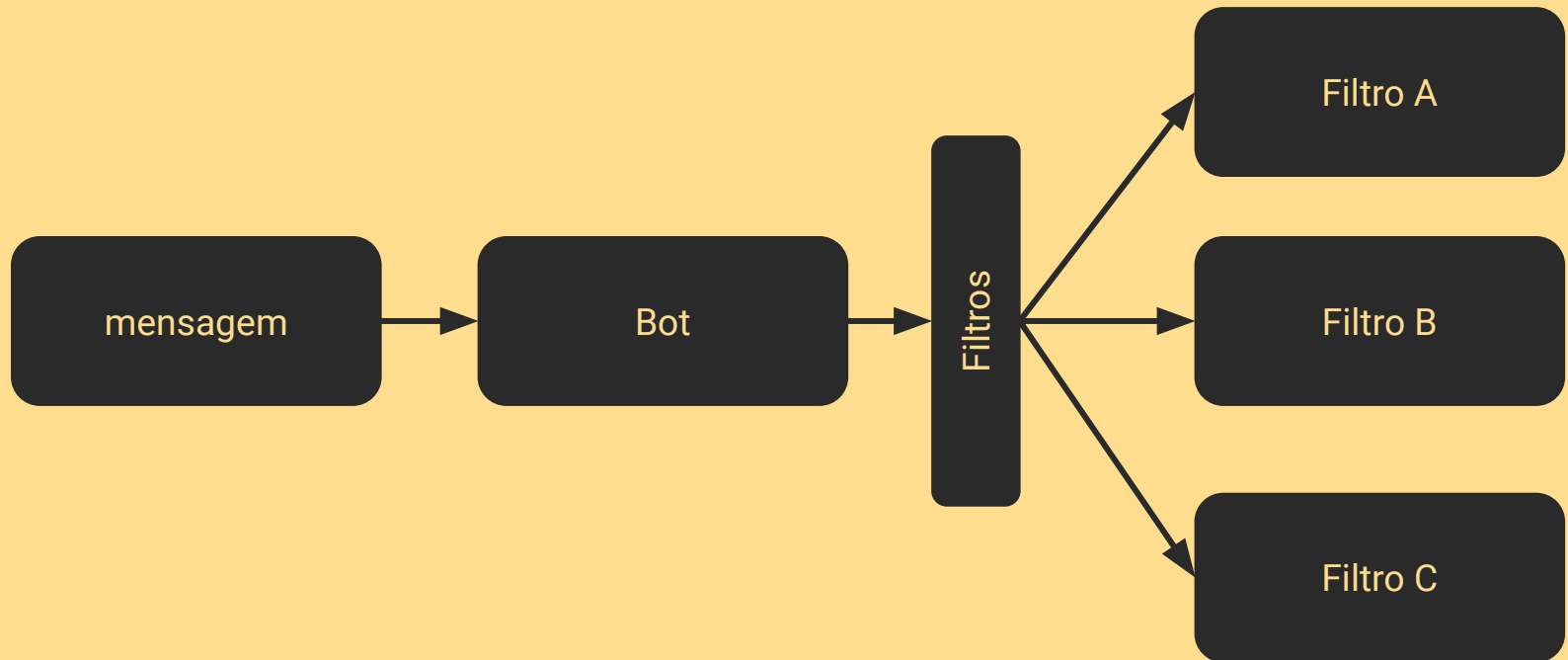
Filtros

0 handler certo,
para mensagem
certa!

Filtros



Os handlers são uma espécie de despachador de mensagens:



Comandos



Os comandos são as mensagens que enviamos para o bot com `/` no início.

```
1  @app.on_message(filters.command('help'))
2  async def help_command(client, message):
3      await message.reply(
4          'Esse é o menu para pedir ajuda!'
5      )
```

Exemplo do /help

Tipos de mídia



O telegram conta com o envio de vários tipos de mídia:

Media	Filtro
Foto	<code>filter.photo</code>
Animações (Gifs)	<code>filter.animation</code>
Documento	<code>filter.document</code>
Stickers / Figurinhas	<code>filter.sticker</code>
Vídeo	<code>filter.video</code>
Audio	<code>filter.audio</code>
Mensagem de voz	<code>filter.voice</code>
...	<code>filter....</code>

<https://docs.pyrogram.org/api/filters#module-pyrogram.filters>

Exemplo



```
1  @app.on_message(filters.voice)
2  async def audio(client, message: Message):
3      await message.reply(
4          'Ah não, já vai começar o podcast!'
5      )
```

Operadores de filtros



Filtros pode conter operadores lógicos

- Ou |
 - `filter.audio | filter.voice`
 - audio ou voz
- E &
 - `filter.sticker & filter.photo`
 - sticker e photo
- Não ~
 - `~ filter.text`
 - Mensagens que não sejam de texto

Exemplo



```
1  @app.on_message(filters.photo | filters.video)
2  async def photo_or_video(client, message: Message):
3      await message.reply('Espero que não seja Nudes!')
```

Privado e grupos



A combinação de filtros pode alterar o comportamento do bot em grupos e no privado.

```
1 @app.on_message((filters.photo | filters.video) & filters.private)
2 async def photo_or_video_private(client, message: Message):
3     await message.reply('Espero que não seja Nudes!')
4
5
6 @app.on_message((filters.photo | filters.video) & filters.group)
7 async def photo_or_video_group(client, message: Message):
8     await message.reply('Quem não encherga não pode fotos, mande texto!')
```

Comandos de teclado



```
1 from pyrogram.types import ReplyKeyboardMarkup
2
3 @app.on_message(filters.command('keyboard'))
4 async def start_command(client, message):
5     await message.reply(
6         'Escolha algo!',
7         reply_markup=ReplyKeyboardMarkup(
8             [
9                 ['a', 'b'],
10                ['c', 'd', 'e', 'f', 'g']
11            ],
12            resize_keyboard=True
13        ),
14    )
```



Mídia

Trabalhando com
coisas que não são
texto

Trabalhando com mídias



Quando o bot recebe a mensagem, ele conta com uma classe da mídia que foi enviada. Por exemplo, se enviarmos um sticker. A mensagem vai ter um objeto Sticker. Que diz qual seu endereço nos servidores do telegram.

```
1  @app.on_message(filters.sticker)
2  async def stickers(client, message: Message):
3      breakpoint()
```

Respondendo mídia



Todas as mensagens que respondemos até agora são mensagens de texto.

Porém, em alguns momentos queremos enviar mídia também. O pyrogram tem interfaces para isso.

Media	Filtro
Foto	<code>app.send_photo()</code>
Animações (Gifs)	<code>app.send_animation()</code>
Documento	<code>app.send_documento</code>
Sticker	<code>app.send_sticker()</code>
...	<code>app.send_...()</code>

<https://docs.pyrogram.org/api/methods/#messages>

Nem só de
mensagens vive um
bot

Decor
adores

Além do `on__message()`



Existem diversos decoradores oferecidos pelo pyrogram. Sendo os principais:

- `on_edited_message()`
 - Para quando mensagens forem editadas
- `on_deleted_message()`
 - Para quando mensagens forem deletadas
- `on_callback_query()`
 - Para quando for preciso continuidade no comando

<https://docs.pyrogram.org/api/decorators>

Callbacks



```
1 from pyrogram.types import InlineKeyboardButton, InlineKeyboardMarkup
2
3 @app.on_message(filters.command('inline'))
4 async def inline(client, message):
5     inline_markup = InlineKeyboardMarkup(
6         [
7             [
8                 InlineKeyboardButton('Callback', callback_data='data'),
9                 InlineKeyboardButton('URL', url='https://docs.pyrogram.org')
10            ]
11        ]
12    )
13    await message.reply('Escolha algo!', reply_markup=inline_markup)
```

Tratamento de callbacks

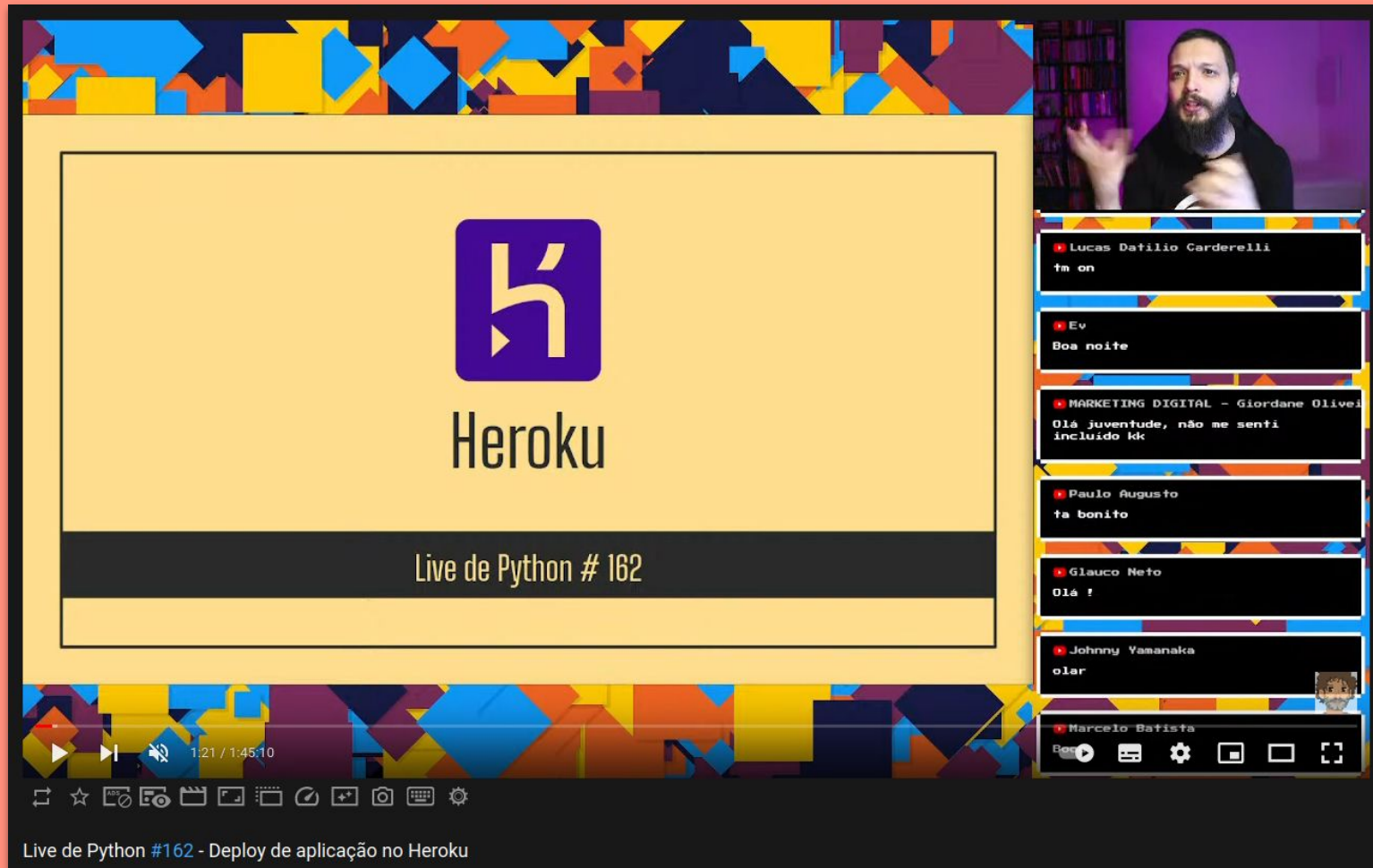


```
1  @app.on_callback_query( )
2  async def callback(client, callback_query):
3      await callback_query.edit_message_text(
4          <mensagem>, reply_markup=<botoes_inline>
5      )
```

Bora botar esse bot
em produção!

Deploy

Caso nunca tenha usado o heroku



<https://youtu.be/fyhZ3a963wl>

Comandos passo a passo [1]



- Criação do arquivo Procfile
 - worker: python bot.py
- Instalar uvloop
 - pip install uvloop (ou poetry add)
- Colocar o uvloop no código
 - import uvloop; uvloop.install()
- Colocar uma versão fixa do python no pyproject.toml
 - 3.10.5
- Criação do projeto no CLI
 - heroku create dunossauro-bot
- Adicionar os buildpacks
 - heroku buildpacks:add
<https://github.com/moneymeets/python-poetry-buildpack.git>
 - heroku buildpacks:add heroku/python

Comandos passo a passo [2]



- Processos do git
 - `git init`
 - `git add .`
 - `git commit -m "Commit inicial"`
- Linka com o repositório do heroku
 - `heroku git:remote -a dunossauro-bot`
- Faz o push no heroku
 - `git push heroku main`
- Inicia o bot no heroku
 - `heroku ps:scale worker=1`



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3



Links de apoio



- <https://docs.pyrogram.org/>
- <https://github.com/BaraniARR/pyrogram-heroku>