

Coleta de dados do *Twitch.tv* utilizando *Python* e a *API* do *Twitch*

Autor: Luiz Carlos Chaves Lima Junior

ROTEIRO

Twitch.tv

- O que é o *Twitch.tv*?
- Transmissão de Jogos no *Twitch*

Utilizando a *New Twitch API*

- Adquirindo o *ID* do cliente

Extração de Dados do *Twitch* em *Python*

- Instalando a biblioteca *twitch-python*
- Exemplos de uso da biblioteca *twitch-python*
- *Twitch Chat Downloader*
- Parâmetros da biblioteca *tcd*

Dados Extraídos

- Dados Disponíveis para Extração
- Extraindo dados do chat de um vídeo do *Twitch*
- Dados Obtidos

REFERÊNCIAS

Twitch.tv

O que é o *Twitch.tv*?

“A *Twitch* é o lugar onde milhões de pessoas se reúnem diariamente para conversar, interagir e criar seu próprio entretenimento ao vivo”

- **Site de streaming**
 - Jogos
 - Música
 - *Talk Shows*
 - Eventos
 - Dentre outros...

Transmissão de Jogos no *Twitch*

Canais Seguidos

- pimentinha (FIFA 20) 402
- jovirone (8 novos videos) Offline
- jukes (7 novos videos) Offline
- Riot Games (3 novos videos) Offline
- C9Sneaky (3 novos videos) Offline

Canais Recomendados

- TrackingThePros (League of Legends) 3,7 mil
- mandiocaa1 (League of Legends) 6,6 mil
- ESL_CSGO (Counter-Strike: Glo...) 4,8 mil

Procurar

Categorias Canais ao vivo

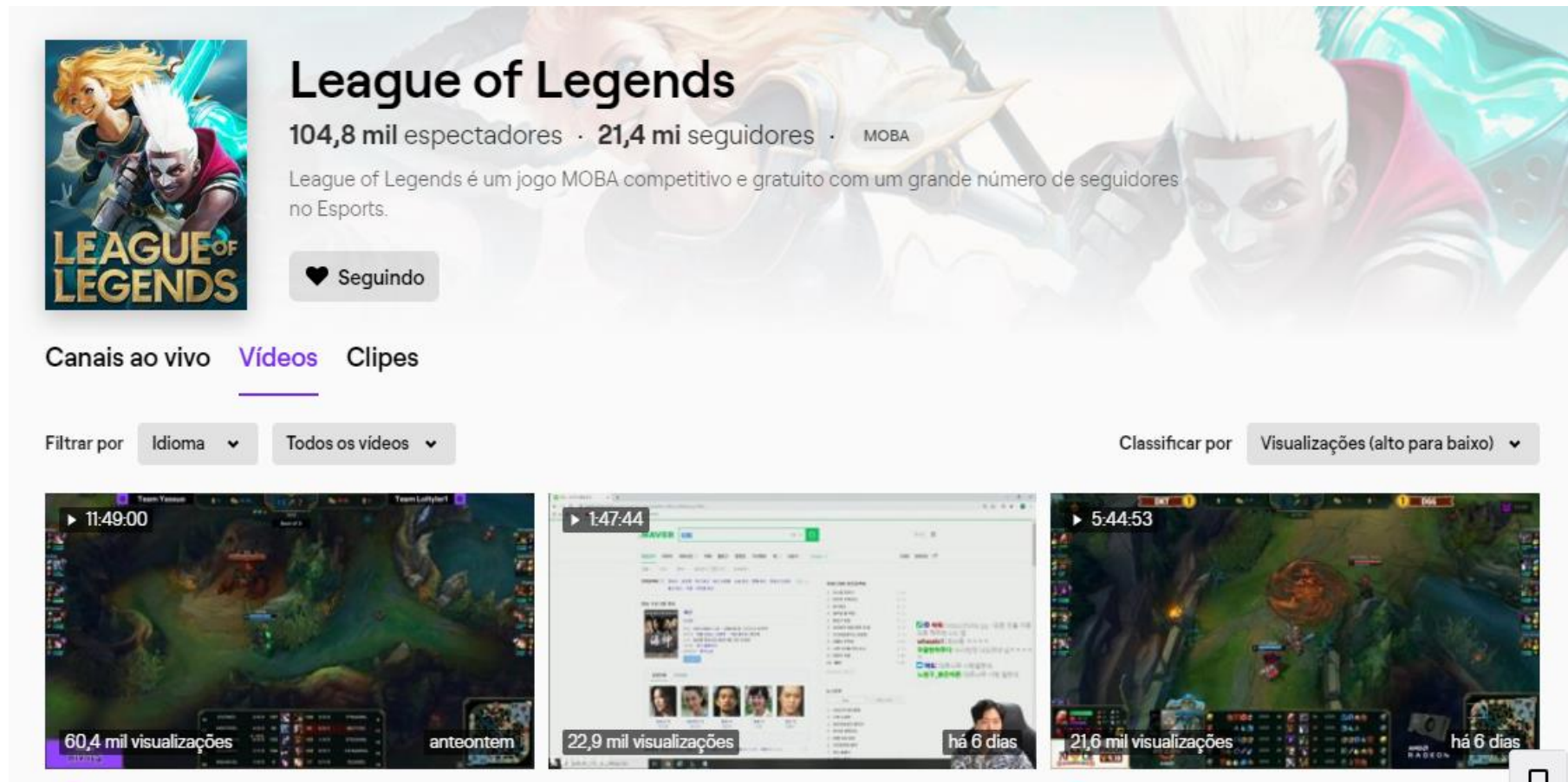
Filtrar por Adicionar filtro +

Classificar por Espectadores (ordem decrescente)

Jogo	Espectadores
League of Legends	107,4 mil espectadores
Só na conversa	90,2 mil espectadores
World of Warcraft	80,2 mil espectadores
FIFA 20	61 mil espectadores
Fortnite	53,9 mil espectadores

Figura 1: Os cinco jogos com maior número de espectadores do site *Twitch.tv*

Transmissão de Jogos no *Twitch*



The screenshot shows the Twitch category page for League of Legends. At the top, there's a banner with the game's logo and a description: "League of Legends é um jogo MOBA competitivo e gratuito com um grande número de seguidores no Esports." Below this, it says "104,8 mil espectadores · 21,4 mi seguidores · MOBA". A "Seguindo" button is visible. The navigation bar includes "Canais ao vivo", "Vídeos" (which is highlighted), and "Clipes". Below the navigation bar, there are filters for "Idioma" and "Todos os vídeos", and a "Classificar por" dropdown set to "Visualizações (alto para baixo)". Three video thumbnails are displayed, each with a play button, a timestamp, and view counts. The first video has a timestamp of 11:49:00 and 60,4 mil visualizações, with a label "anteontem". The second video has a timestamp of 1:47:44 and 22,9 mil visualizações, with a label "há 6 dias". The third video has a timestamp of 5:44:53 and 21,6 mil visualizações, with a label "há 6 dias".

League of Legends
104,8 mil espectadores · 21,4 mi seguidores · MOBA

League of Legends é um jogo MOBA competitivo e gratuito com um grande número de seguidores no Esports.

Seguindo

Canais ao vivo **Vídeos** Clipes

Filtrar por Idioma ▼ Todos os vídeos ▼

Classificar por Visualizações (alto para baixo) ▼

▶ 11:49:00 60,4 mil visualizações anteontem

▶ 1:47:44 22,9 mil visualizações há 6 dias

▶ 5:44:53 21,6 mil visualizações há 6 dias

Figura 2: Categoria do jogo *League of Legends*

Transmissão de Jogos no *Twitch*

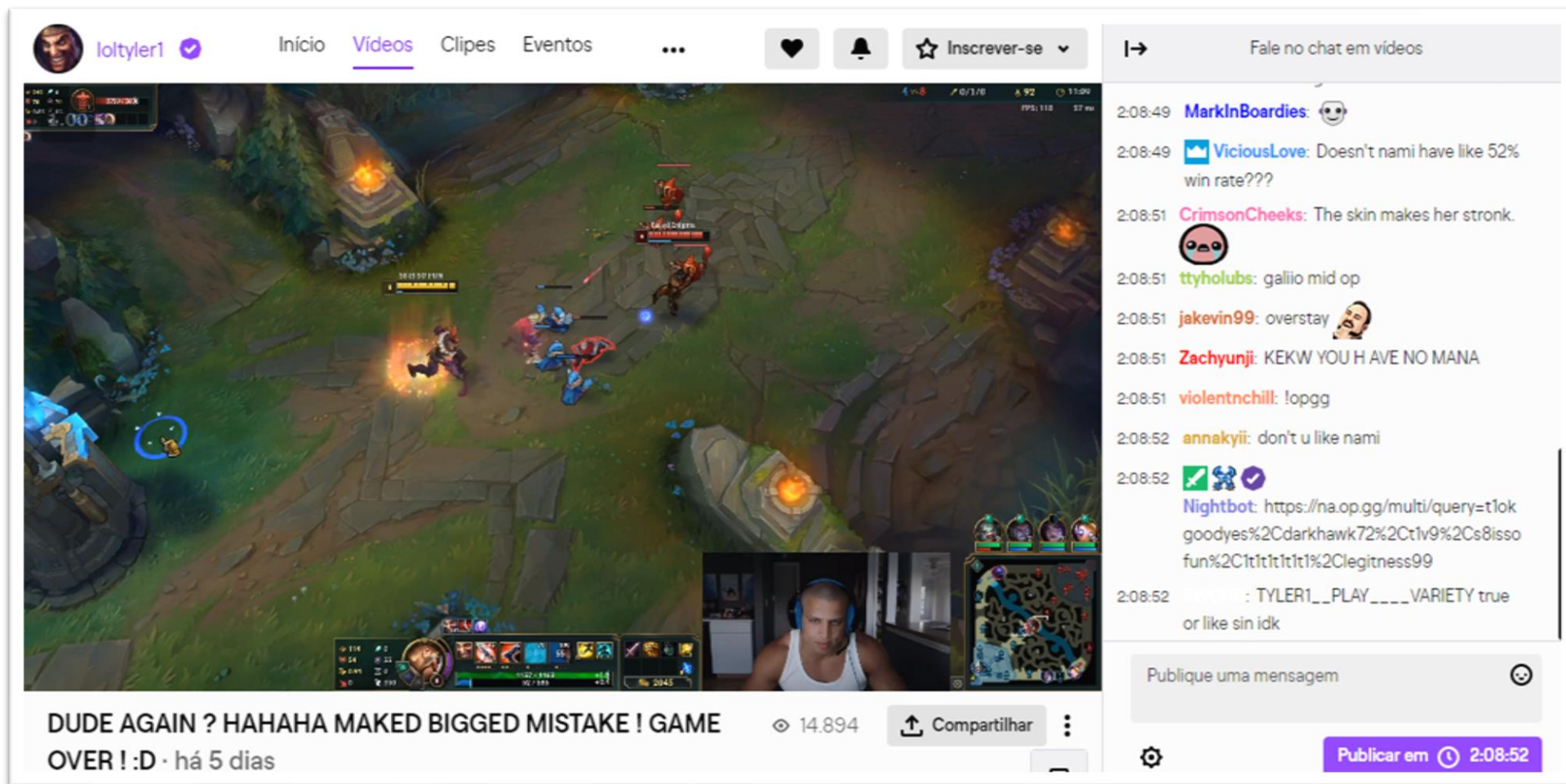


Figura 3: Vídeo de transmissão de um canal de *League of Legends*

Utilizando a *New Twitch API*

Adquirindo o *ID* do cliente

Passo 1: Fazer *login* com sua conta *Twitch* no “*Console*” em <https://dev.twitch.tv/login>



Figura 4: Pagina de *login* para Painel do desenvolvedor do *Twitch (Console)*

Adquirindo o *ID* do cliente

Passo 2: Já dentro do *Console*, selecionar a guia Aplicativos e clicar em “Registre seu aplicativo”

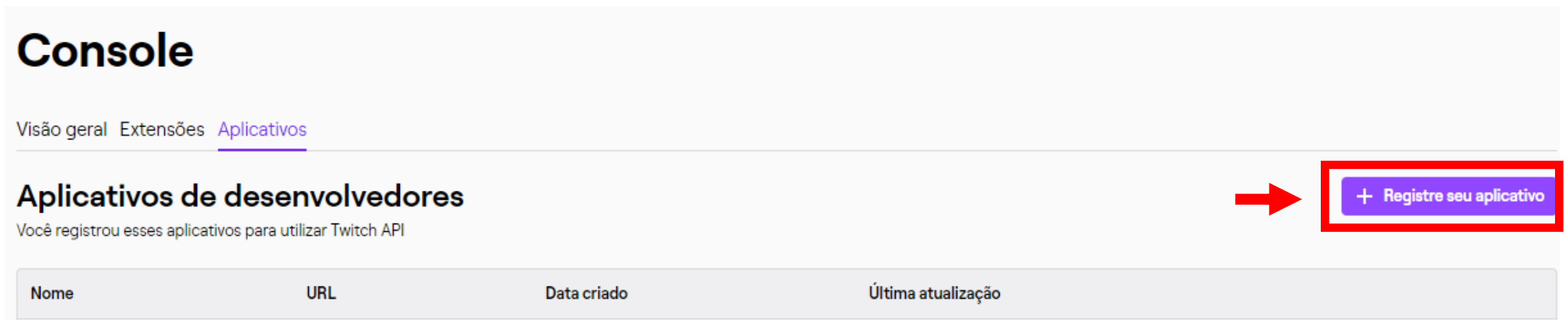
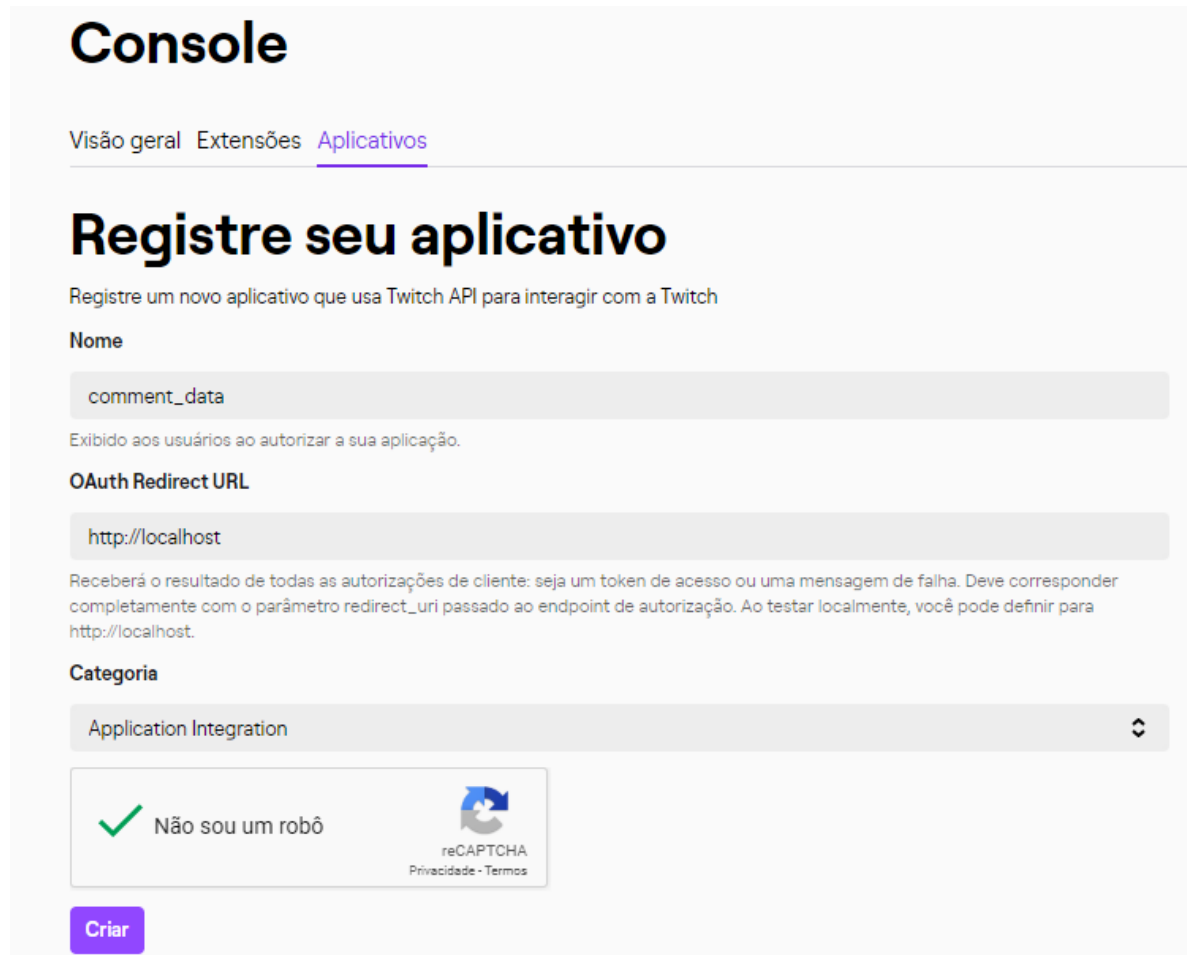


Figura 5: Como registrar aplicativos no *Console* do *Twitch*

Adquirindo o *ID* do cliente



The screenshot shows the 'Console' interface with the 'Aplicativos' tab selected. The main heading is 'Registre seu aplicativo'. Below it, a sub-heading says 'Registre um novo aplicativo que usa Twitch API para interagir com a Twitch'. The form has three main sections: 'Nome' with a text input containing 'comment_data'; 'OAuth Redirect URL' with a text input containing 'http://localhost'; and 'Categoria' with a dropdown menu showing 'Application Integration'. At the bottom, there is a reCAPTCHA widget with a green checkmark and the text 'Não sou um robô', and a purple 'Criar' button.

Figura 6: Registrando aplicativos no *Console* do Twitch

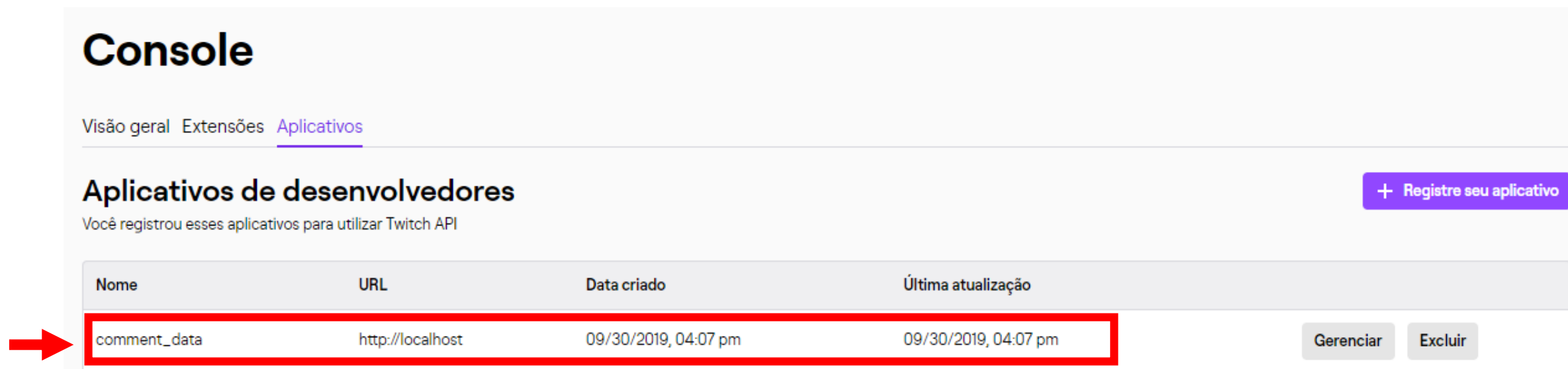
Passo 3: Digitar o nome do aplicativo, a *URI* de redirecionamento do *OAuth*, selecionar uma categoria de aplicativo e, em seguida, clicar em “Criar” para registrar o aplicativo.

Nome: os nomes dos aplicativos não devem incluir, de forma alguma, a palavra “*Twitch*” ou o registro falhará.

OAuth Redirect URL: receberá o resultado de todas as autorizações de cliente: seja um *token* de acesso ou uma mensagem de falha. Deve corresponder completamente com o parâmetro *redirect_uri* passado ao *endpoint* de autorização. Ao testar localmente, você pode definir para <http://localhost>.

Categoria: podem ser *Analytics Tool*, *Chat Bot*, *Game Integration*, *Application Integration*, dentre outros.

Adquirindo o *ID* do cliente



Console

Visão geral Extensões Aplicativos

Aplicativos de desenvolvedores

Você registrou esses aplicativos para utilizar Twitch API

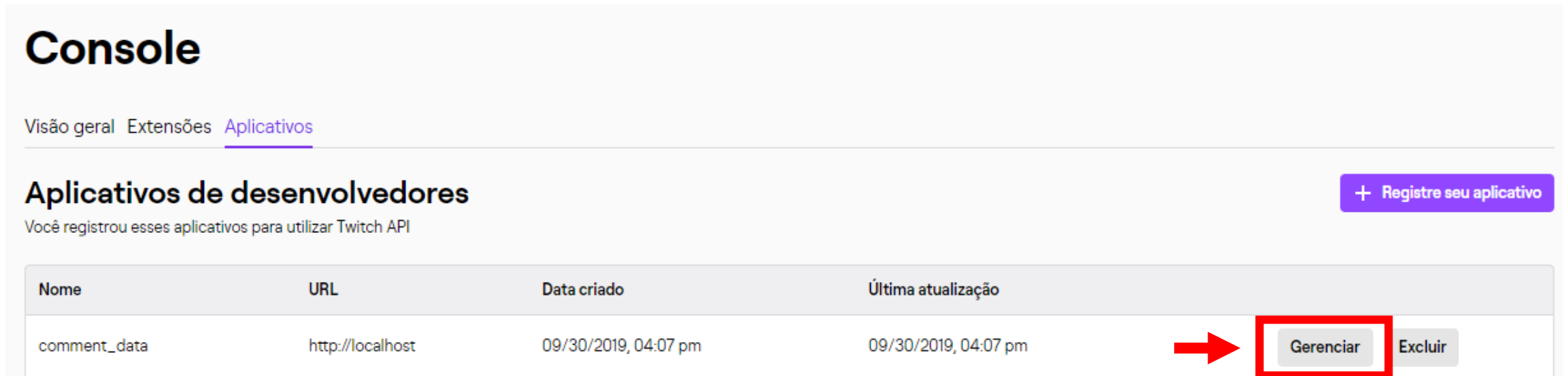
[+ Registre seu aplicativo](#)

Nome	URL	Data criado	Última atualização	
comment_data	http://localhost	09/30/2019, 04:07 pm	09/30/2019, 04:07 pm	Gerenciar Excluir

Figura 7: Aplicativo Registrado no *Console* do Twitch

Adquirindo o *ID* do cliente

Passo 4: Clicar no botão gerenciar para verificar o *ID* do cliente.



The screenshot shows the Twitch Console interface. At the top, there's a 'Console' header with tabs for 'Visão geral', 'Extensões', and 'Aplicativos'. Below this, the section is titled 'Aplicativos de desenvolvedores' with a subtext 'Você registrou esses aplicativos para utilizar Twitch API' and a '+ Registre seu aplicativo' button. A table lists the registered applications. The first application is 'comment_data' with URL 'http://localhost', created on '09/30/2019, 04:07 pm', and last updated on '09/30/2019, 04:07 pm'. To the right of this application, there are two buttons: 'Gerenciar' (highlighted with a red box and a red arrow) and 'Excluir'.

Nome	URL	Data criado	Última atualização	
comment_data	http://localhost	09/30/2019, 04:07 pm	09/30/2019, 04:07 pm	<div><div>Gerenciar</div><div>Excluir</div></div>

Figura 8: Gerenciando aplicativos no *Console* do Twitch

Adquirindo o *ID* do cliente

Visão geral Extensões Aplicativos

Gerenciar aplicativo: comment_data

Registre um novo aplicativo que usa Twitch API para interagir com a Twitch

Nome

comment_data

Exibido aos usuários ao autorizar a sua aplicação.

OAuth Redirect URL

http://localhost

Receberá o resultado de todas as autorizações de cliente: seja um token de acesso ou uma mensagem de falha. Deve corresponder completamente com o parâmetro redirect_uri passado ao endpoint de autorização. Ao testar localmente, você pode definir para http://localhost.

Categoria

Application Integration

☐ Não sou um robô

reCAPTCHA
Privacidade - Termos

ID do cliente

ssnu5ctpovqub5uzn9ca5isb62pg7k

Passado aos endpoints de autorização para identificar o seu aplicativo. Você não pode alterar o ID do cliente do aplicativo.

Passo 5: Por fim, o *ID* do cliente é gerado e encontrado após o campo categoria. Este código será passado aos *endpoints* de autorização para identificar o seu aplicativo.

- **Obs.:** Você não pode alterar o *ID* do cliente do aplicativo.

Figura 9: Obtendo o *ID* do cliente

Extração de Dados do *Twitch* em *Python*

Instalando a biblioteca *twitch-python*

- **Requisitos:**

- *Python* 3.7 ou mais recente
- *ID* de cliente do *Twitch*

- **Instalação pelo pip:**

- `pip install twitch-python`

Exemplos de uso da biblioteca *twitch-python*

```
# Exemplo 1: Extrair dados de Usuários

import twitch

helix = twitch.Helix(' id do client ')

# Users
for user in helix.users(['sodapoppin', 'reckful', 24250859]):
    print(user.display_name)

print(helix.user('zarlach').display_name)
```



Output 1:

- Zarlach
- Reckful
- Sodapoppin

Output 2:

- Zarlach

Exemplos de uso da biblioteca *twitch-python*

```
# Exemplo 2: Extrair dados de Vídeos

import twitch

helix = twitch.Helix('id do client')

# Videos
for video in helix.videos([318017128, 317650435]):
    print(video.title)

print(helix.video(318017128).title)
```



Output 1:

- Overwatch and DbD chill stream
- hating myself

Output 2:

- hating myself

Exemplos de uso da biblioteca *twitch-python*

```
# Exemplo 3: Extrair dados de Comentários

import twitch

helix = twitch.Helix(' id do client ')
# Video Comments (VOD chat)
for comment in helix.video(318017128).comments:
    print(comment.commenter.display_name, '--', comment.message.body)
```



Output:

```
-enTauren -- EPYC sodaH
-WavveS_V2 -- gachiBASS
-Bird_Mannn -- LULW OK NOW HES JUST MAKING SHIT UP
-xDanyPlay -- gachiBASS
-Mahkrn -- PepeS
-SpinningBackKick -- @tuttifrutti_ first of all, I just tuned in, and that doesn't make this retarded
```

Twitch Chat Downloader

- **Requisitos:**

- *Python 3.7* ou mais recente
- *ID* de cliente do *Twitch*

- **Instalação pelo pip:**

- `pip install tcd`

Parâmetros da biblioteca *tcd*

Parâmetro	Função
-h, --help	Show this help message and exit
-v VIDEO, --video VIDEO	Video IDs separated by commas
-c CHANNEL, --channel CHANNEL	Channel names separated by commas
-u USER, --user USER	Messages from users, separated by commas
--first FIRST	Download chat from the last n VODs
--client-id CLIENT_ID	Twitch client ID
-f FORMAT, --format FORMAT	Message format
-o OUTPUT, --output OUTPUT	Output directory
Exemplos	
# Download chat from VODs by video id tcd --video 789654123,987456321 --format irc --output ~/Downloads	
# Download chat from the first 10 VODs from multiple streamers tcd --channel sodapoppin,nymn,lirik --first=10	

Dados Extraídos

Dados Disponíveis para Extração

DADOS DISPONÍVEIS DOS USUÁRIOS		
Nome	Tipo	Descrição
id	string	ID do usuário.
login	string	Nome de login do usuário.
display_name	string	Nome de exibição do usuário.
type	string	Tipo de usuário: "staff", "admin", "global_mod", or "".
broadcaster_type	string	Tipo de emissor do usuário: "partner", "affiliate", or ""
description	string	Descrição do canal do usuário.
profile_image_url	string	URL da imagem do perfil do usuário.
offline_image_url	string	URL da imagem offline do usuário.
view_count	int	Número total de visualizações do canal do usuário
data	dictionary	exibe todas as informações relativas aos usuários

Dados Disponíveis para Extração

DADOS DISPONÍVEIS DOS VÍDEOS		
Nome	Tipo	Descrição
id	string	ID do vídeo.
user_id	string	ID do usuário proprietário do vídeo.
user_name	string	Nome de exibição correspondente a user_id.
title	string	Título do vídeo.
description	string	Descrição do vídeo.
created_at	string	Data em que o vídeo foi criado.
published_at	string	Data em que o vídeo foi publicado.
url	object	URL do vídeo.
thumbnail_url	object	URL do modelo para a miniatura do vídeo.
viewable	string	Indica se o vídeo está visível publicamente. Valores válidos: "public", "private".
view_count	int	Número de vezes que o vídeo foi visualizado.
language	string	Idioma do vídeo.
type	string	Tipo de vídeo. Valores válidos: "upload", "archive", "highlight".
duration	string	Duração do vídeo.
data	dictionary	exibe todas as informações relativas aos vídeos

Dados Disponíveis para Extração

DADOS DISPONÍVEIS DOS COMENTÁRIOS DO CHAT – PARTE 1		
Nome	Tipo	Descrição
id	string	ID do Comentário.
created_at	string	Data em que o comentário foi criado.
updated_at	string	Data em que o comentário foi atualizado.
channel_id	string	ID do canal do usuário.
content_type	string	Tipo de conteúdo exibido. Ex.: “video”.
content_id	string	ID do conteúdo.
content_offset_seconds	float	Deslocamento do conteúdo em segundos.
source	string	Fonte dos dados. Ex. “comment”, “chat”.
state	string	Estado .
more_replies	boolean	Mais respostas para o comentário. “True” ou “False”.
data	dictionary	exibe todas as informações relativas aos comentários.

Dados Disponíveis para Extração

DADOS DISPONÍVEIS DOS COMENTÁRIOS DO CHAT – PARTE 2		
Nome	Tipo	Descrição
message.body	string	Conteúdo da mensagem.
message.fragments	list	Lista contendo detalhes da mensagem Ex.: {'text': 'hello'}.
message.is_action	boolean	“True” ou “False”.
message.user_badges	List	Lista contendo informações sobre o usuário que escreveu a mensagem. Ex.: {'_id': 'subscriber', 'version': '6'}, {'_id': 'partner', 'version': '1'}.
message.emoticons	dictionary	Lista detalhes de emoticons caso o usuários tenha utilizado nas mensagens. Ex.: {'emoticon_id': '90226', 'emoticon_set_id': ''}.
message.user_color	string	Cor utilizada em Hexadecimal. Ex.: #00FF7F.
message.data	dictionary	Contém todas as informações relativas às mensagens.

Dados Disponíveis para Extração

DADOS DISPONÍVEIS DOS COMENTÁRIOS DO CHAT – PARTE 3		
Nome	Tipo	Descrição
commenter.display_name	string	Nome exibido do autor mensagem.
commenter.id	string	ID do autor da mensagem.
commenter.name	string	Nome do autor da mensagem.
commenter.type	string	Tipo de usuário do autor da mensagem.
commenter.bio	string	Autodescrição do autor da mensagem.
commenter.created_at	string	Data em que o autor criou a mensagem.
commenter.updated_at	string	Data em que o autor atualizou a mensagem.
commenter.logo	string	URL do logotipo do autor da mensagem.
commenter.data	dictionary	Contém todas as informações relativas aos autores das mensagens.

Extraindo dados do chat de um vídeo do Twitch

```
import twitch

helix = twitch.Helix('id do client ') # insira o id do client

# Extrai e salva os dados do usuário como um arquivo .csv
with open('TSM_Bjergsen.csv', 'w', encoding='utf8') as f:
    titles =
    f"user_name\tuser_vcount\tvideo_created_at\tvideo_title\tvideo_vcount\tcomment_created_at\tcommenter_name\tmessage\n"
    f.write(titles)
    # Extrai dados dos 5 primeiros videos do usuário 'TSM_Bjergsen'.
    #Para extrair dados de outro usuário basta inserir outro nome de usuário, exatamente da forma como está escrito na
    plataforma twitch
    for user, videos in helix.users(['TSM_Bjergsen']).videos(first=5):
        user_name = user.display_name #nome do usuário
        user_vcount = user.view_count #contagem de visualizações para o canal do usuário
        for video, comments in videos.comments:
            video_title = video.title # título do vídeo
            video_vcount = video.view_count # número de visualizações do vídeo
            video_created_at = video.created_at # data de criação do vídeo
            for comment in comments:
                comment_created_at = comment.created_at #data de criação do comentário
                commenter_name = comment.commenter.display_name # nome do usuário que publicou o comentário
                message = comment.message.body # texto do comentário

f.write(f"{user_name}\t{user_vcount}\t{video_created_at}\t{video_title}\t{video_vcount}\t{comment_created_at}\t{comment
er_name}\t{message}\n")
f.close()
```

Parte dos Dados obtidos

	user_name	user_vcount	video_created_at	video_title	video_vcount	comment_created_at	commenter_name	message
0	TSM_Bjergsen	92608310	2019-10-31T02:30:38Z	Chill stream, playing with a fan at 8 PM PST.	7798	2019-10-31T02:31:02.603Z	Kekkei	hello
1	TSM_Bjergsen	92608310	2019-10-31T02:30:38Z	Chill stream, playing with a fan at 8 PM PST.	7798	2019-10-31T02:31:23.457Z	Tarzan10x	Daddy is live Kappa
2	TSM_Bjergsen	92608310	2019-10-31T02:30:38Z	Chill stream, playing with a fan at 8 PM PST.	7798	2019-10-31T02:31:30.2Z	grouroxd	PogChamp PogChamp PogChamp PogChamp

Figura 10: Alguns dados obtidos do chat

REFERÊNCIAS

- *Twitch*. Disponível em: <https://www.twitch.tv/p/pt-br/about/>. Acesso em: 30 de setembro de 2019.
- *Twitch Developers. Console*. Disponível em: <https://dev.twitch.tv/console>. Acesso em: 30 de setembro de 2019.
- *Twitch Developers. Twitch API Reference*. Disponível em: <https://dev.twitch.tv/docs/api/reference/>. Acesso em: 4 de outubro de 2019
- *Python Package Index. Twitch Python*. Disponível em: <https://pypi.org/project/twitch-python/>. Acesso em: 30 de setembro de 2019.
- *Python Package Index. Twitch Chat Downloader*. Disponível em: <https://pypi.org/project/tcd/>. Acesso em: 30 de setembro de 2019.