

Matheus Adler Soares Pinto

Matrícula: 2112960

**Projeto Final de Programação: Uma ferramenta para
gerar bases de dados textuais a partir de
comentários do YouTube para tarefas de
Processamento de Linguagem Natural**

Rio de Janeiro, Brasil

2022

Matheus Adler Soares Pinto

Matrícula: 2112960

Projeto Final de Programação: Uma ferramenta para gerar bases de dados textuais a partir de comentários do YouTube para tarefas de Processamento de Linguagem Natural

Trabalho apresentado à Prof. Dra. Clarisse Sieckenius de Souza no programa de pós-graduação em informática da PUC-Rio como requisito para obtenção de nota na disciplina INF2102 – Projeto Final de Programação

Pontifícia Universidade Católica do Rio de Janeiro

Departamento de Informática

Programa de Pós-Graduação em Informática

Orientador: Sérgio Colcher

Rio de Janeiro, Brasil

2022

Sumário

1. Especificação do Programa.....	4
1.1. Objetivo	4
1.2. Escopo	4
1.3. Requisitos	4
1.3.1. Requisitos Funcionais	4
1.3.2. Requisitos Não Funcionais.....	5
2. Projeto	5
2.1. Estrutura	5
2.2. Arquitetura	7
2.3. Modelagem.....	8
2.3.1. Diagrama de Casos de Uso.....	8
2.3.2. Diagrama de Atividade	9
3. Testes	10
4. Instruções para o Usuário	11
4.1. Fluxo da aplicação	12
5. Código Fonte	18
6. Colaboração Científica.....	18

1. Especificação do Programa

1.1. Objetivo

O Processamento de Linguagem Natural é uma área de estudo da Inteligência Artificial que visa a compreensão e interpretação da linguagem humana realizada por modelos computacionais. A base de dados é uma das mais importantes etapas para atingir o objetivo de modelar a linguagem e possibilitar que a máquina entenda. Para isso, são necessários pré-processamentos que abstraem e estruturam de forma adequada a língua nessas bases de dados, deixando apenas o que é informação relevante. Esse pré-processamento reduz o vocabulário e torna os dados menos esparsos o que torna a tarefa mais simples para o modelo computacional. Disto isso, o objetivo deste projeto é desenvolver uma ferramenta de geração de bases de dados textuais a partir de comentários de vídeos do YouTube¹ para tarefas de Processamento de Linguagem Natural. Tal ferramenta, a partir de um título de busca, permitirá que o usuário gere uma base pré-processada a partir dos comentários de vídeos encontrados com o título de busca dado.

1.2. Escopo

O escopo do projeto consiste na produção de uma interface gráfica para gerar uma base de dados textuais para tarefas de Processamento de Linguagem Natural. A interface possui o parâmetro de busca para a contextualização da base de dados textuais a partir da necessidade do usuário, otimizando assim o tempo na etapa de extração e pré-processamento da base de dados e também reduzindo assim o contato direto do pesquisador com código para esta tarefa em projetos de machine learning.

1.3. Requisitos

Nessa seção são definidos os requisitos funcionais e não funcionais da ferramenta proposta.

1.3.1. Requisitos Funcionais

Nesse projeto foram definidos quatro Requisitos Funcionais (RF):

- **RF01 – Busca da base de dados:** a ferramenta deve permitir que o usuário informe um título de busca e gere uma base de dados a partir dos vídeos encontrados com essa busca.
- **RF02 – Pré-processamento dos dados:** a ferramenta deve realizar a etapa de pré-processamento nos comentários coletados com as seguintes técnicas:
 - Remover emojis
 - Remover pontuação
 - Remover quebra de linhas
 - Deixar todo texto em minúsculo
 - Remover caracteres especiais
 - Remover números e palavras concatenadas com números
 - Remover e-mails e menções com @

¹ <https://www.youtube.com>

- Remover urls
 - Remover comentários com “kkk”
 - Remover strings com menos de 2 letras e mais de 20 letras
 - Remover stopwords em Português e Inglês
- **RF03 - Gerar arquivos:** a ferramenta deve gerar dois arquivos no formato .csv, um com todos os comentários coletados dos vídeos e outro com esses comentários após o pré-processamento realizado.
 - **RF04 - Salvar base de dados:** a ferramenta deve permitir que o usuário escolha o diretório onde os arquivos gerados serão salvos.

1.3.2. Requisitos Não Funcionais

Nesse projeto foram definidos quatro Requisitos Não Funcionais (RNF):

- **RNF01 – Consistência:** a ferramenta deve assegurar a criação da base de dados a partir do título de busca dos vídeos.
- **RNF02 – Usabilidade:** a ferramenta deve minimizar o contato do usuário com código e ser de fácil uso para gerar uma base de dados.
- **RNF03 – Estabilidade:** a ferramenta deve considerar a busca de vídeos com muitos comentários (máximo de 10.000) e garantir o funcionamento com desempenho satisfatório com um baixo consumo de memória e recursos de máquina.
- **RNF04 – Confiabilidade:** a ferramenta deve garantir que em casos de buscas repetidas e salvas em um mesmo diretório, a última e mais atualizada que permanecerá.

2. Projeto

Esta seção descreve a organização da ferramenta proposta, detalhando sua estruturação, arquitetura e modelagem.

2.1. Estrutura

O projeto está organizado no diretório principal *app*, onde contém todo o código-fonte da ferramenta. Esse diretório é dividido em cinco subdiretórios e um arquivo de configurações. A Figura 1 mostra como está organizado essa divisão.

```

PFP_YouTube_Comments_Dataset_Generator/
|---app/
|   |---css/
|   |   |---app.css
|   |   |---extract-comments.css
|   |   |---preprocessing-comments.css
|   |   |---restart-process.css
|   |   |---search-videos.css
|   |---html/
|   |   |---extract-comments.html
|   |   |---index.html
|   |   |---preprocessing-comments.html
|   |   |---restart-process.html
|   |   |---search-videos.html
|   |---js/
|   |   |---extract-comments.js
|   |   |---index.js
|   |   |---main.js
|   |   |---preprocessing-comments.js
|   |   |---restart-process.js
|   |   |---search-videos.js
|   |---py/
|   |   |---extract_comments.py
|   |   |---preprocessing_comments.py
|   |   |---requirements.txt
|   |   |---search_videos.py
|   |---test/
|   |   |---test.js
|   |   |---test.py
|   |---package.json

```

Figura 1 - Estrutura dos arquivos no diretório app

- **css:** diretório que contém os arquivos .css que estilizam as telas da aplicação.
- **html:** diretório que contém os arquivos.html que estruturam e definem as telas da aplicação.
- **js:** diretório que contém os arquivos .js que gerenciam toda a comunicação e controle das telas.

- **py:** diretório que contém os arquivos .py que controlam as principais funções para a coleta e pré-processamento dos comentários e um arquivo .txt com as bibliotecas necessárias para a execução dos scripts Python.
- **test:** diretório que contém os arquivos de teste das funções JavaScript e Python.
- **package.json:** arquivo de configurações da aplicação que contém dependências de pacotes e informações do autor.

2.2. Arquitetura

A ferramenta foi desenvolvida utilizando o framework Electron², para construir uma aplicação desktop; o framework NodeJS³, para integração e comunicação do ambiente de execução; e o Bootstrap⁴, para estruturação visual da ferramenta. No entanto, o processo principal de geração da base de dados foi implementado utilizando Python⁵.

A implementação seguiu o padrão de desenvolvimento da arquitetura MVC (Model-View-Controller), onde cada um dos frameworks citados desempenha um papel dentro dessa arquitetura. O Model é desempenhado pela unificação do Electron e o NodeJS, View é desempenhado pelo Bootstrap e o Controller da aplicação é realizado pelos scripts Python. A Figura 2 mostra essa relação.

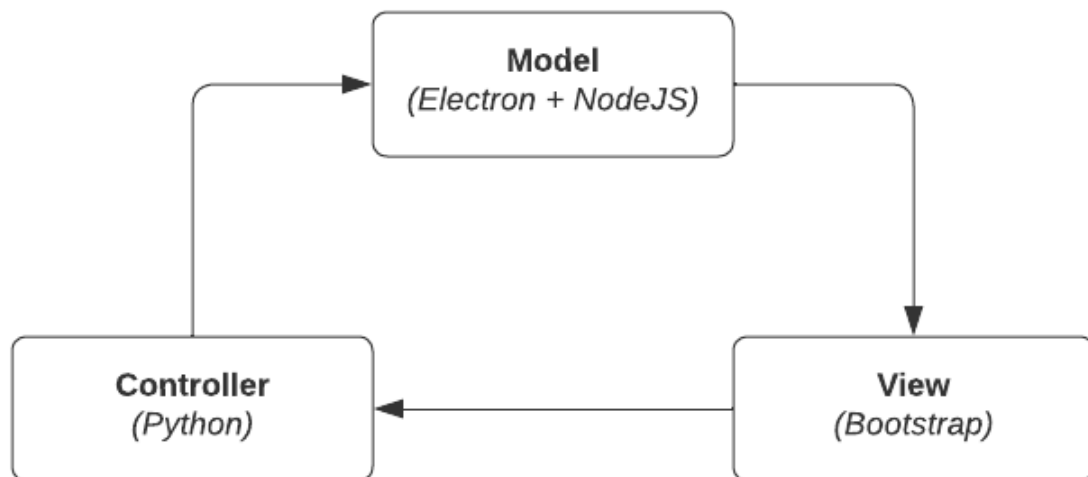


Figura 2- Arquitetura da aplicação

² <https://www.electronjs.org>

³ <https://nodejs.org/en/>

⁴ <https://getbootstrap.com>

⁵ <https://www.python.org>

2.3. Modelagem

Baseado nos requisitos funcionais e não funcionais definidos para o projeto, alguns diagramas da modelagem da ferramenta foram elaborados, tais diagramas permitem uma compreensão melhor do funcionamento esperado da aplicação em termos de funcionalidades e de como utilizá-las.

2.3.1. Diagrama de Casos de Uso

Para a ferramenta desenvolvida, o diagrama de caso de uso mostrado na Figura 3 foi proposto.

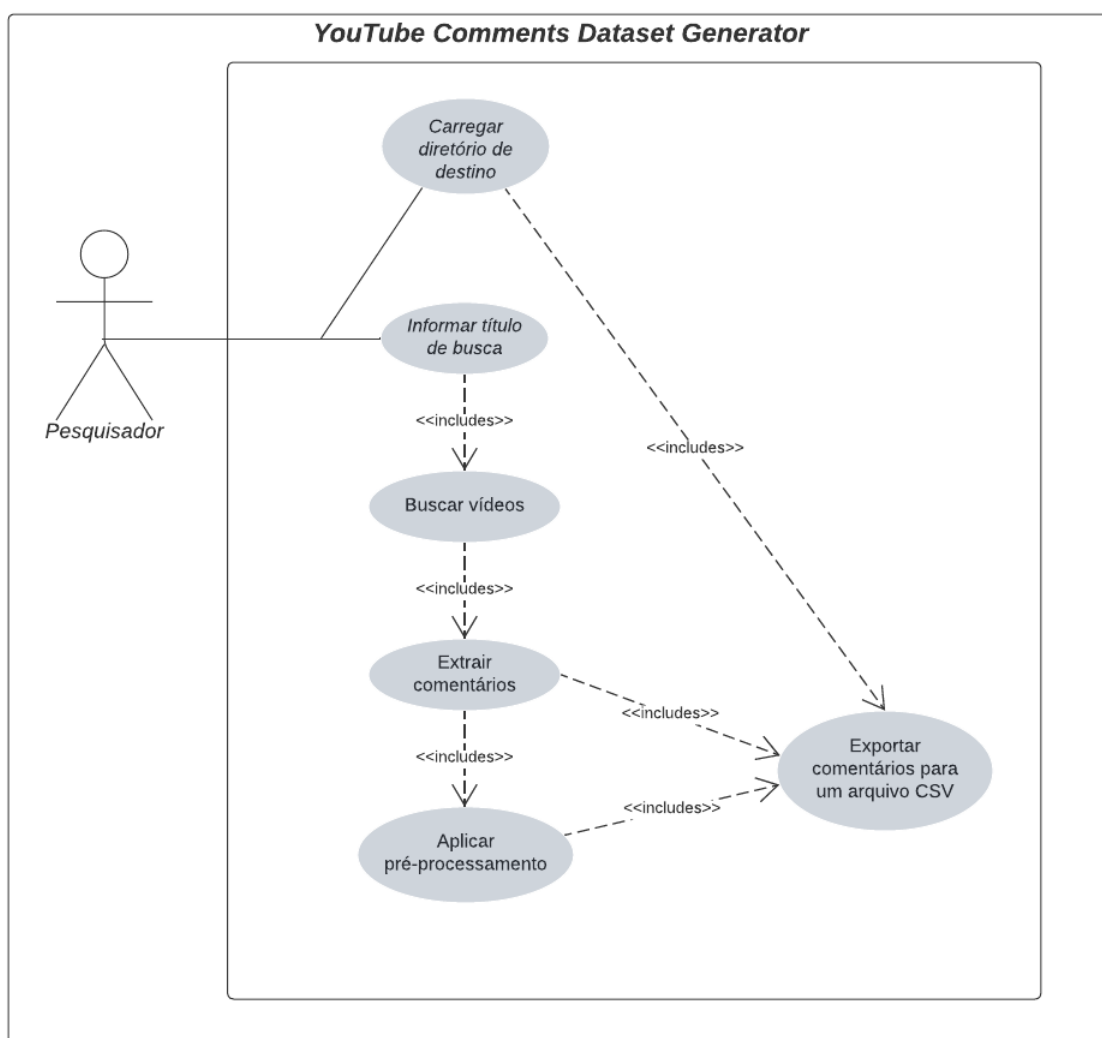


Figura 3 - Diagrama de Caso de Uso

No diagrama, o ator principal é o pesquisador. Ele é responsável por informar o título da busca que será realizada no YouTube e determinar onde os arquivos gerados serão salvos. O pesquisador, no início do processo da tarefa, pode executar as seguintes funcionalidades: **Informar título de busca**, que será o título inserido para realizar as buscas dos vídeos no YouTube; e ele pode **Carregar diretório de destino**, que é o diretório em sua máquina onde os arquivos com os comentários coletados e os comentários coletados após o pré-processamento serão salvos.

2.3.2. Diagrama de Atividade

Os seguintes diagramas descrevem as atividades realizadas em cada etapa do processo de coleta e pré-processamento dos comentários de vídeos do YouTube.

O processo começa com um index, Figura 4, onde seu objetivo é informar um diretório onde os arquivos gerados serão salvos e um título de busca que será a busca realizada no YouTube.

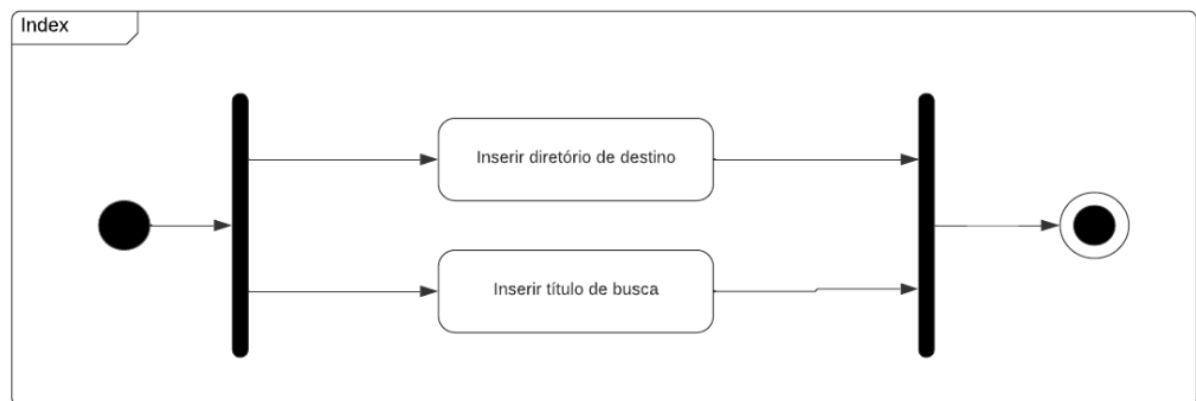


Figura 4 - Diagrama de Atividade da tela Index

A Figura 5 demonstra a etapa de busca pelos vídeos, onde é recebido um título e esse título é a busca realizada no YouTube.

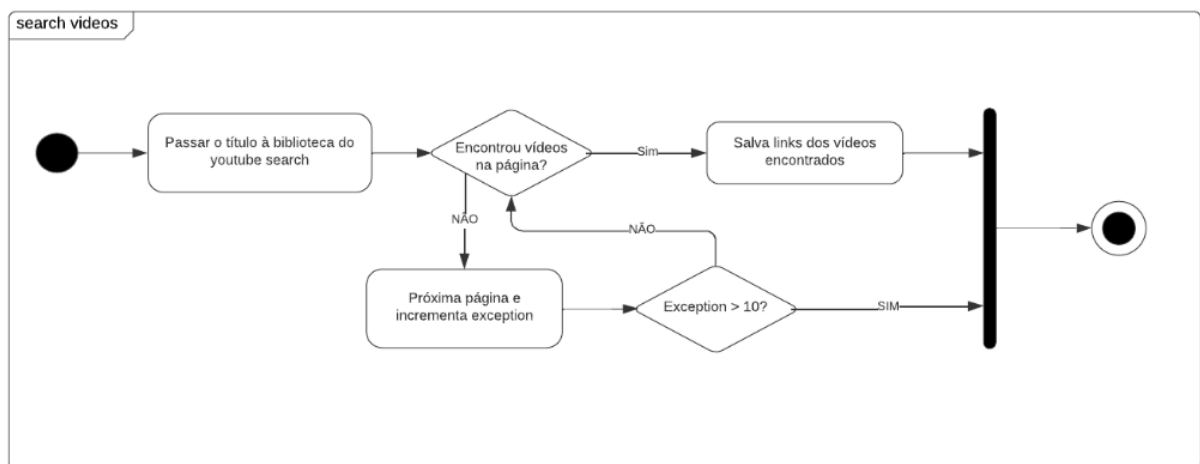


Figura 5 - Diagrama de Atividades da tela de busca dos vídeos

O próximo passo é representado na Figura 6, onde nessa etapa são coletados todos os comentários de cada vídeo encontrado no passo anterior.

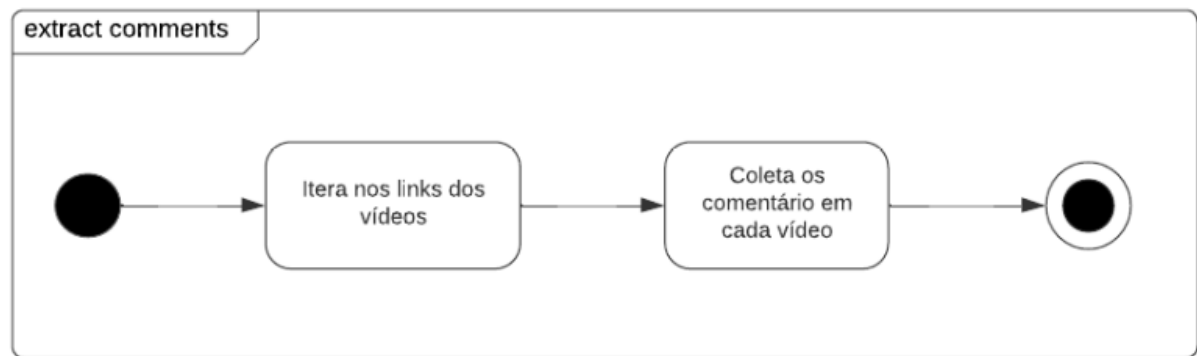


Figura 6 - Diagrama de Atividades da tela de coleta de comentários

Por fim, a última etapa do processo é demonstrada na Figura 7, onde um arquivo csv com todos os comentários é salvo e outro arquivo csv com todos os comentários após uma etapa de pré-processamento também é salvo, ambos no diretório de destino selecionado no primeiro passo.

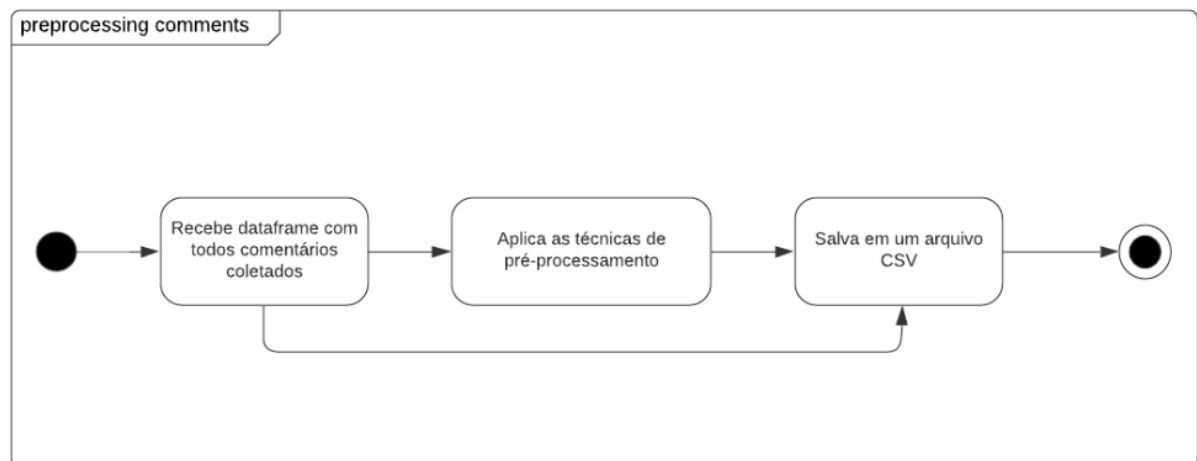


Figura 7 - Diagrama de Atividades da tela de pré-processamento dos comentários

3. Testes

Para assegurar o funcionamento e verificar a qualidade das funcionalidades propostas na ferramenta, testes unitários foram implementados. Foram utilizados o Spectron para testar as funcionalidades do framework Electron e o Unittest para testar as funcionalidades desenvolvidas pelos scripts Python.

A Figura 8 mostra o teste realizado pelo Spectron, nele foram testadas cinco funcionalidades da aplicação Electron.

```
> pfp_youtube_comments_dataset_generator@1.0.0 test
> mocha

YouTube Comments Dataset Generator
✓ Verifica se a janela está visível
✓ Verifica se exibe uma janela inicial
✓ Verifica se o título da aplicação está correto
✓ Verifica se existe o botão "Selecione o diretório"
✓ Verifica se existe o botão "Buscar"

5 passing (52s)
```

Figura 8 - Log de teste Spectron

Agora testando as funcionalidades de controle Python da aplicação, três testes foram implementados. O primeiro teste é o da função de buscar os vídeos no YouTube, o segundo é o teste da função de extrair os comentários de um vídeo e, por fim, o teste da etapa de pré-processamento dos comentários. A Figura 9 mostra o log completo da execução dos testes e os retornos esperados em cada etapa do processo.

```
PS C:\Users\mathe\OneDrive\PUC\PDFP_YouTube_Comments_Dataset_Generator\app\test> python test.py
Retorno esperado da função de extrair os comentários [True]: True
Retorno esperado da função de busca dos vídeos [list]: list
Retorno esperado da função de pré-processamento dos comentários [True]: True
.
-----
Ran 3 tests in 17.275s
OK
```

Figura 9 - Log de teste Unittest

4. Instruções para o Usuário

Nesta seção está descrito o processo de instalação e utilização da ferramenta proposta. As orientações de instalação e uso da aplicação requerem a prévia instalação na máquina do usuário, do Python e do NodeJS. Partindo desse requisito, o usuário pode acessar o link do repositório do projeto no GitHub:

https://github.com/Matheusadler/PFP_YouTube_Comments_Dataset_Generator.

Caso o usuário tenha o Git instalado em sua máquina, ele pode realizar o download simplesmente com o comando:

```
$ git clone https://github.com/Matheusadler/PFP_YouTube_Comments_Dataset_Generator.git
```

Após o download da ferramenta em sua máquina, o usuário deve acessar o diretório raiz do projeto:

```
$ cd ./PFP_YouTube_Comments_Dataset_Generator/app
```

E em seguida, instalar todos os pacotes necessários com o comando:

```
$ npm install
```

Esse comando busca no arquivo package.json todas as dependências do projeto. Por fim, deve-se instalar os pacotes Python que são necessários para executar a aplicação, para isso, o usuário deve executar o comando:

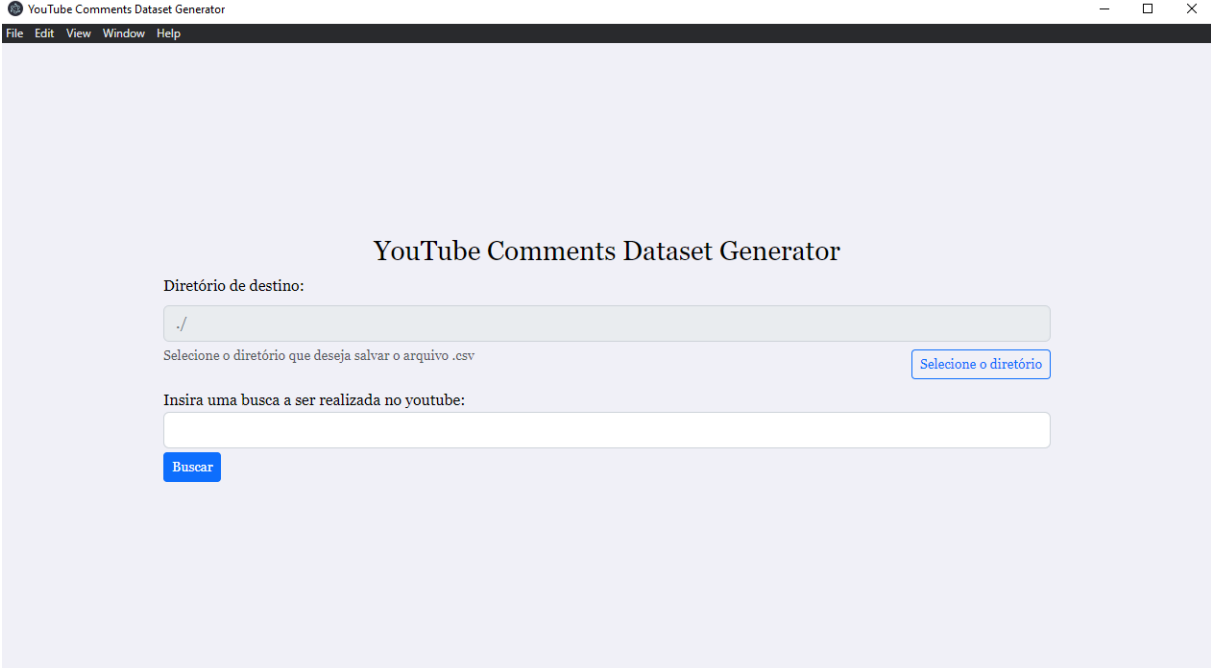
```
$ pip install -qr py/requirements.txt
```

Ao fim de todos esses passos, para iniciar a aplicação com todas as suas funcionalidades, basta executar o comando:

```
$ npm start
```

4.1. Fluxo da aplicação

O fluxo da aplicação começa na tela inicial (Figura 10), onde o usuário deve selecionar um diretório de destino para salvar seus arquivos e um título de busca que será realizado no YouTube para buscar os vídeos de onde os comentários serão extraídos. Caso algum desses campos não seja selecionado, um span de aviso é disparado e mostrado na tela para o usuário (Figura 11).



The screenshot shows a web application window titled "YouTube Comments Dataset Generator". The interface includes a menu bar with "File", "Edit", "View", "Window", and "Help". The main content area has a title "YouTube Comments Dataset Generator" and two input sections. The first section, "Diretório de destino:", contains a text input field with the value "./" and a button labeled "Selecione o diretório". Below this is a label "Insira uma busca a ser realizada no youtube:" followed by a text input field and a button labeled "Buscar".

Figura 10 - Tela Inicial

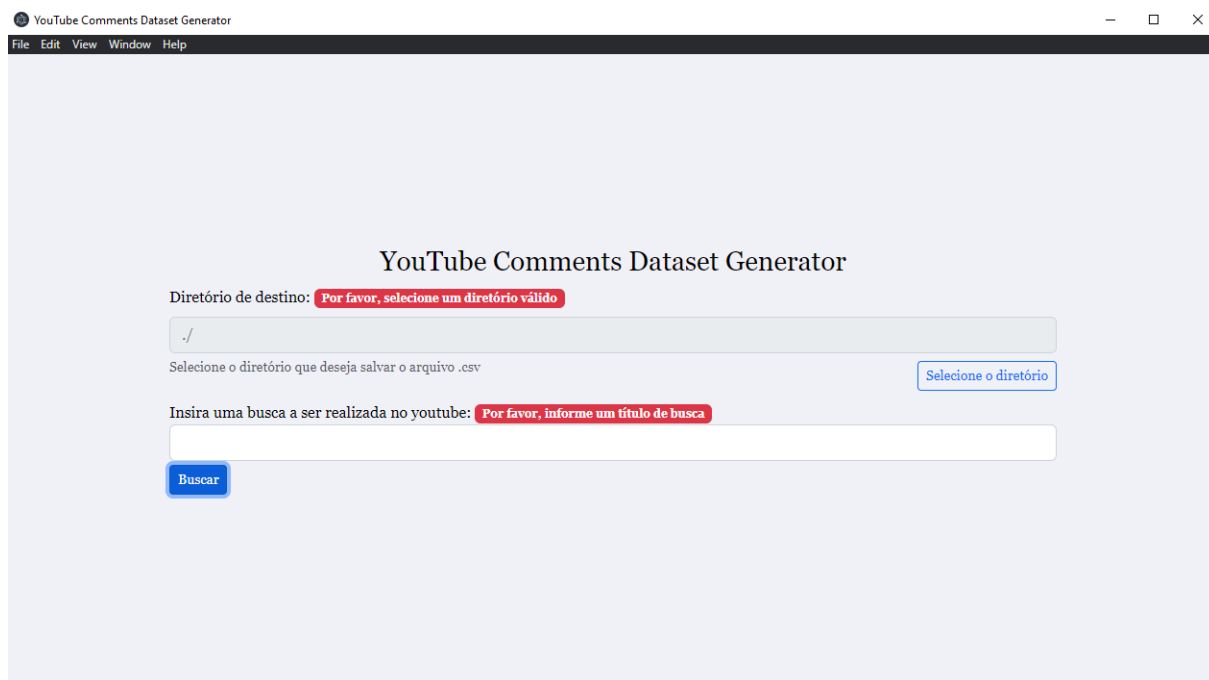


Figura 11 - Tela inicial com os avisos de campos requeridos

Uma vez informado um diretório de destino e um título de busca, a aplicação parte para a próxima etapa do processo que é a de buscar os vídeos no YouTube, a Figura 12 mostra a tela que é mostrada ao usuário enquanto os vídeos estão sendo buscados.



Figura 12 - Tela de carregamento na busca dos vídeos no YouTube

No final desse processo, caso a busca seja realizada corretamente, um modal de sucesso será exibido mostrando quantos vídeos foram encontrados para aquela busca realizada e um botão para avançar no processo (Figura 13). Caso o processo

não seja completado com sucesso, um modal de erro será exibido juntamente com um botão para reiniciar o processo (Figura 14).

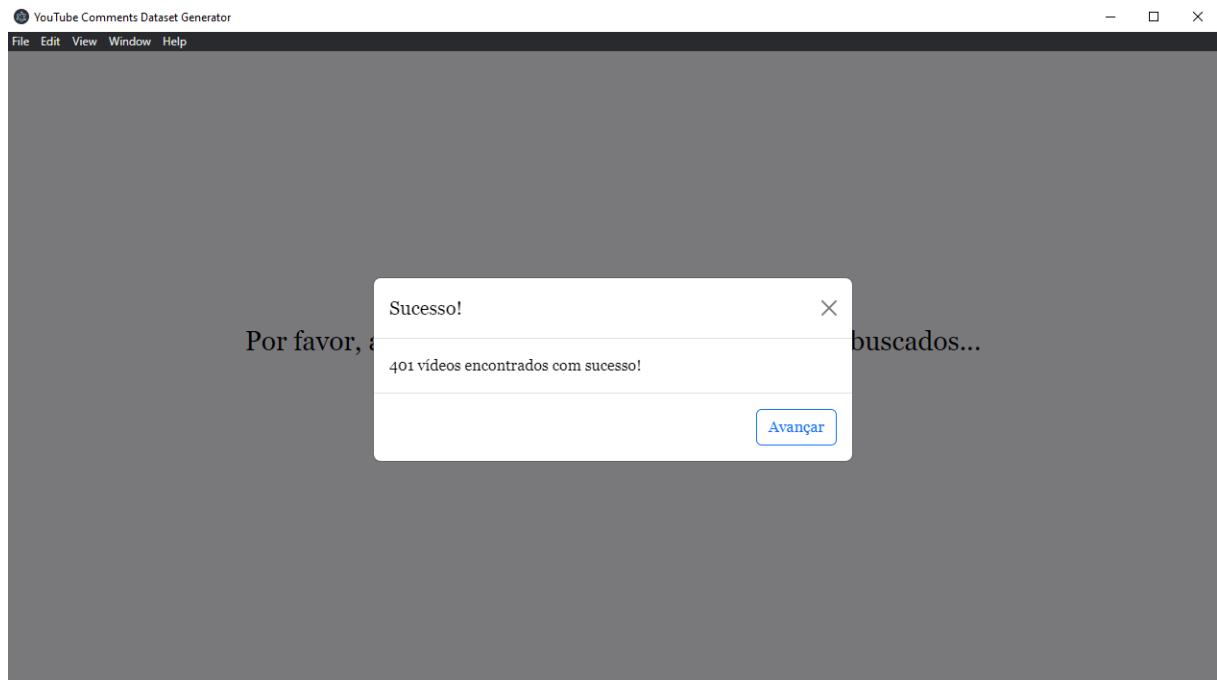


Figura 13 - Modal de sucesso na busca dos vídeos

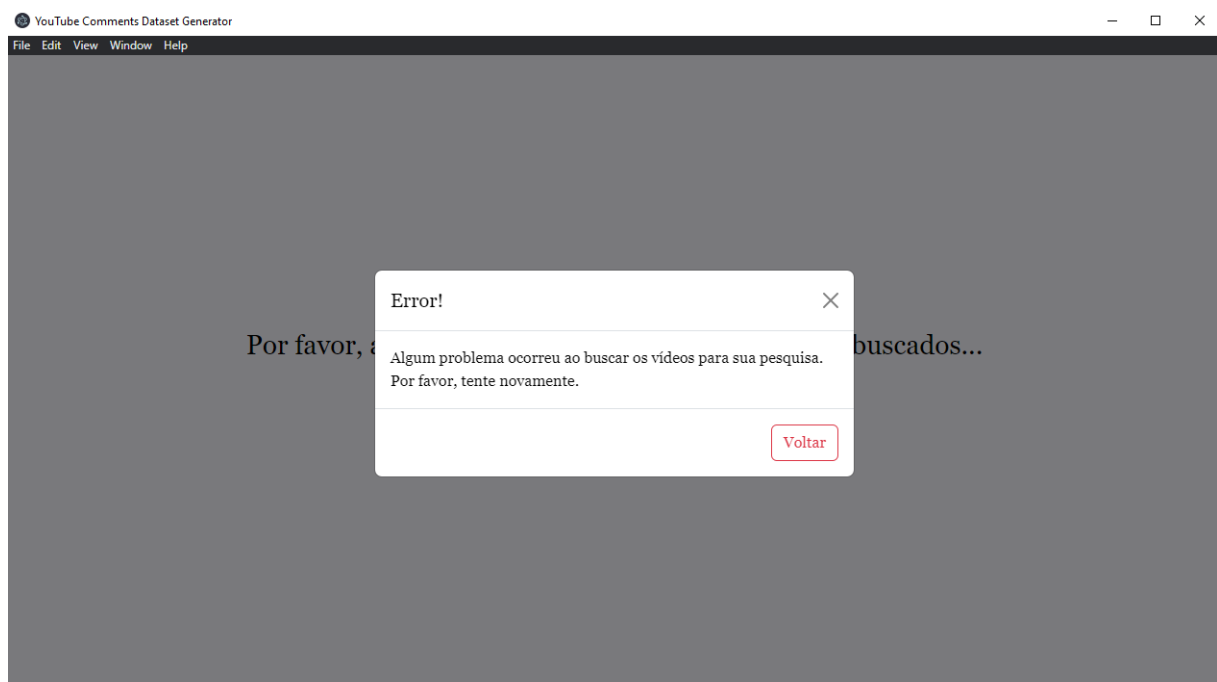


Figura 14 - Modal de erro na busca dos vídeos

A próxima etapa é a de extrair os comentários de cada vídeo, a Figura 15 mostra a tela exibida para o usuário enquanto esse processo de extração está sendo realizado.

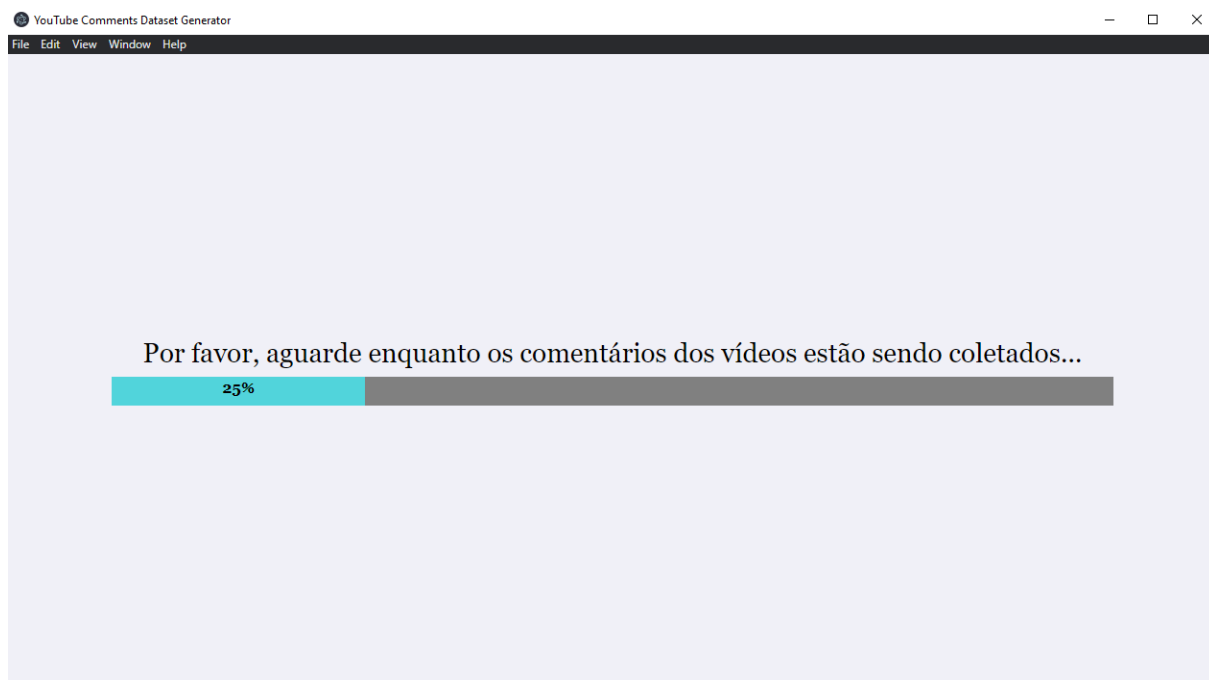


Figura 15 - Tela de carregamento da extração de comentários

No fim desse processo, em caso de sucesso, um modal é mostrado informando quantos comentários foram extraídos e um botão para avançar no processo (Figura 16). Caso haja alguma falha no processo, é exibido um modal de erro com um botão para reiniciar o processo (Figura 17).

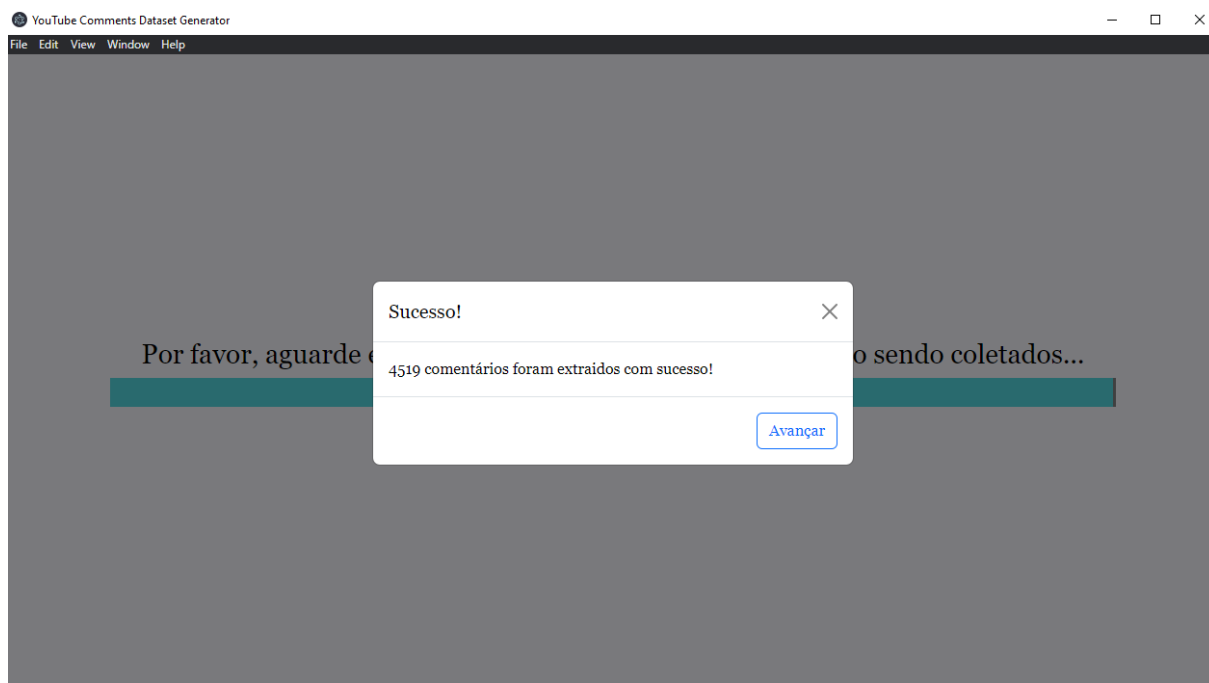


Figura 16 - Modal de sucesso na extração dos comentários



Figura 17 - Modal de erro na extração dos comentários

Após a extração dos comentários, vem a etapa de pré-processamento, nela todos os comentários passam pelas técnicas básicas de pré-processamento que foram especificadas nos requisitos funcionais. Enquanto os comentários estão sendo processados, uma tela de carregamento aparece para o usuário (Figura 18).

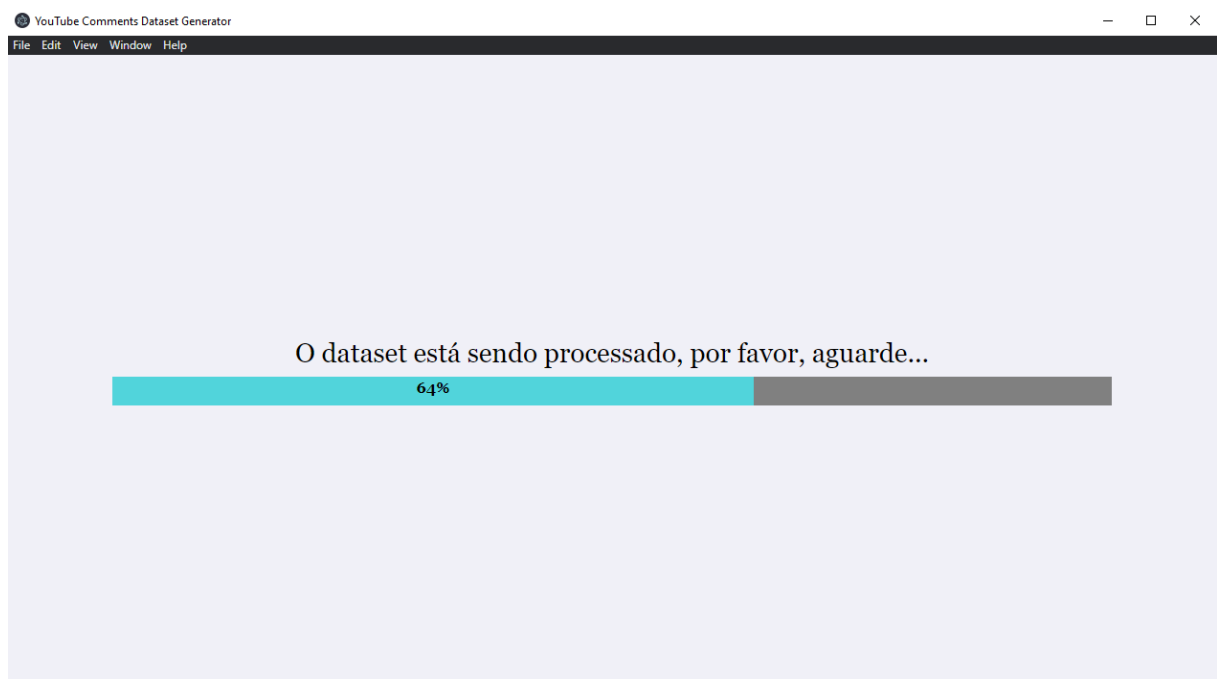


Figura 18 - Tela de carregamento do processamento dos comentários

Após o processamento, um modal de sucesso é mostrado com um botão para a tela final da aplicação (Figura 19).

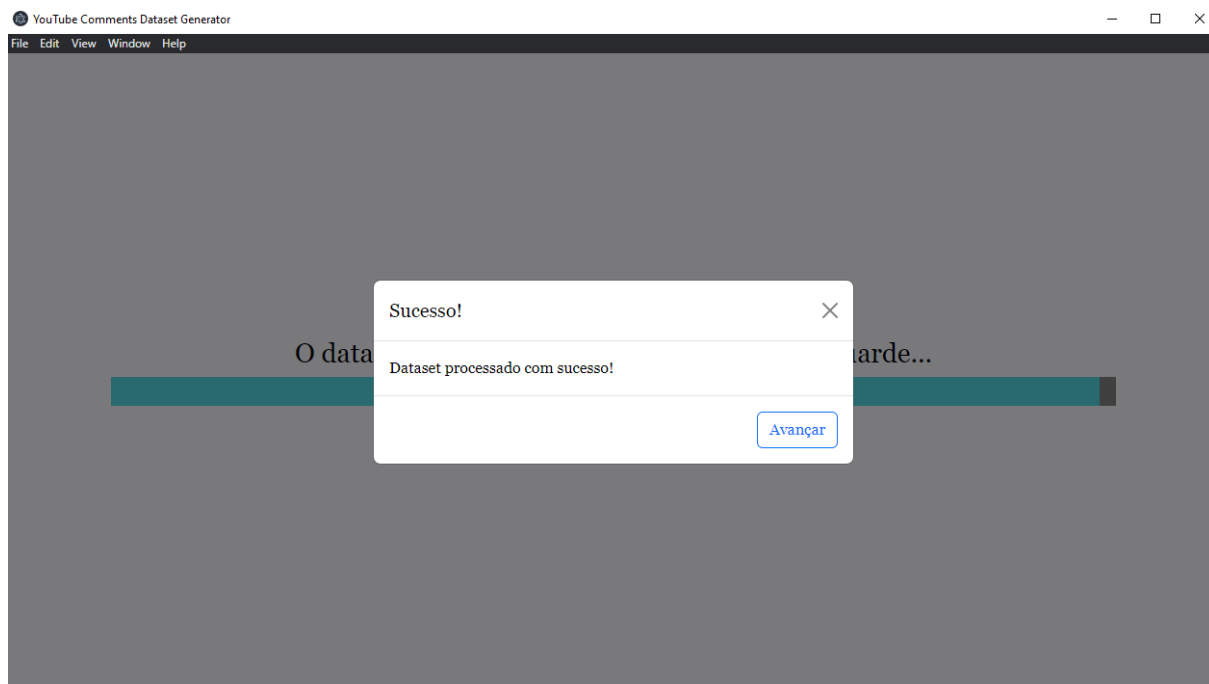


Figura 19 - Modal de sucesso no processamento dos comentários

Caso tudo tenha ocorrido como o planejado, a tela final (Figura 20) mostra algumas informações importantes para o usuário. Primeiro é informado o local onde os arquivos foram salvos, e também são informados os nomes dos arquivos e o que contém em cada um deles. A tela final possui um botão que permite que o usuário repita todo o processo caso deseje realizar uma nova busca.

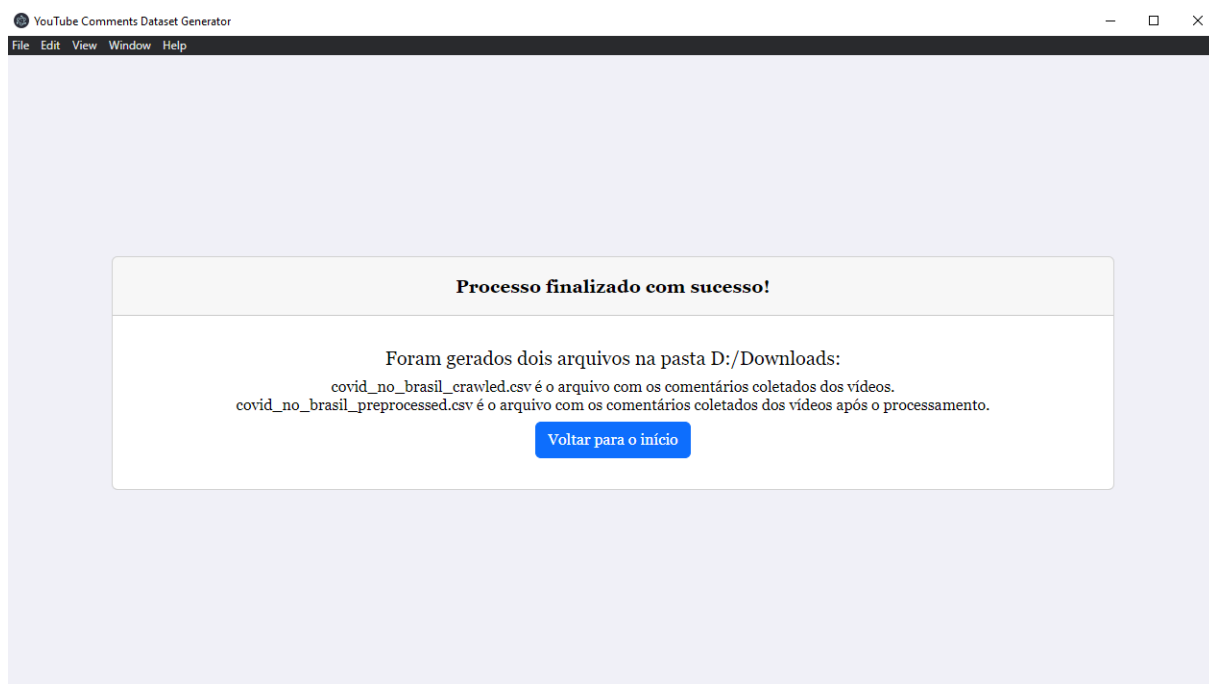


Figura 20 - Tela final com informações para o usuário

5. Código Fonte

Todo o código fonte do projeto está disponível e devidamente comentado no repositório do GitHub⁶ e pode ser acessado a qualquer momento.

6. Colaboração Científica

O tema da minha pesquisa está embasado nas técnicas e modelos de Processamento de Linguagem Natural. Meu objetivo é aplicar clusterização, modelagem de tópicos e análise semântica em conjuntos de dados com muita informação pública, provavelmente em vídeos do YouTube sobre determinado assunto ou conjunto de assuntos.

O *dataset* é uma das mais importantes etapas em qualquer projeto de *Machine Learning*. Quando se trata do contexto de Processamento Textual, essa etapa se dificulta ainda mais, uma vez que coletar comentários de usuários e montar um *dataset* não é uma tarefa simples. Este projeto visa auxiliar pesquisadores na criação e pré-processamento de *datasets* com comentários de vídeos do YouTube para realizar tarefas de Processamento de Linguagem Natural.

A principal contribuição proposta pela ferramenta é o ganho de tempo e comodidade de ter menos contato com códigos nessa etapa de gerar e pré-processar um *dataset* textual. Dessa forma, a ferramenta irá beneficiar outros pesquisadores no estudo de PLN que necessitam gerar *datasets* com comentários do YouTube.

Entretanto, deve-se haver o cuidado com a necessidade dos dados requerido por cada pesquisa. A ferramenta realiza o pré-processamento correto apenas em comentários que tenham sido feitos nos idiomas Português e/ou Inglês, o que dificultaria a etapa de pré-processamento e poderia comprometer os resultados de uma pesquisa de um outro pesquisador.

⁶ https://github.com/Matheusadler/PFP_YouTube_Comments_Dataset_Generator