



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**ANDRÉ GOMES FERREIRA DE OLIVEIRA**

**QUIZZIFY APP:  
UMA FERRAMENTA BASEADA EM PLN PARA O AUXÍLIO NA  
MEMORIZAÇÃO E ASSIMILAÇÃO DE CONTEÚDOS E  
CONCEITOS**

**CAMPINA GRANDE - PB**

**2023**

**ANDRÉ GOMES FERREIRA DE OLIVEIRA**

**QUIZZIFY APP:  
UMA FERRAMENTA BASEADA EM PLN PARA O AUXÍLIO NA  
MEMORIZAÇÃO E ASSIMILAÇÃO DE CONTEÚDOS E  
CONCEITOS**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de **Bacharel ou  
Bacharela** em Ciência da Computação.**

**Orientadora: Fábio Jorge Almeida.**

**CAMPINA GRANDE - PB**

**2023**

**ANDRÉ GOMES FERREIRA DE OLIVEIRA**

**QUIZZIFY APP:  
UMA FERRAMENTA BASEADA EM PLN PARA O AUXÍLIO NA  
MEMORIZAÇÃO E ASSIMILAÇÃO DE CONTEÚDOS E  
CONCEITOS**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de **Bacharel ou  
Bacharela** em Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr.(a.) Fábio Jorge Almeida  
Orientador – UASC/CEEI/UFCG**

**Professora Dr.(a.) Dalton Dario Serey Guerrero  
Examinador – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni  
Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 14 de Fevereiro de 2023.**

## **CAMPINA GRANDE - PB**

### **RESUMO (ABSTRACT)<sup>1</sup>**

The learning process, no matter if it occurs on a human or artificial agent, is not always something fluid and easy and usually takes several training steps. Even so, applications based on artificial intelligence and Natural Language Processing have shown surprising results and often improve through exposure to new situations. Seeking to explore and apply the concepts and advances in this area to human learning, this work aims to propose the development of an application, using NLP techniques, where the user, when prompted to insert a text of his or her interest, will, after doing so, be presented to automatically generated questions, covering the main ideas and key points contained therein. Helping with regard to the fixation and assimilation of contents, assisting in the recollection of topics, definitions and concepts. Herewith, the aim is to streamline the process of studying and memorizing topics of interest to the user, facilitating the process of acquiring and consolidating knowledge.

# Quizzify App: Uma ferramenta baseada em PLN para o auxílio na memorização e assimilação de conteúdos e conceitos

Trabalho de Conclusão de Curso  
André Gomes Ferreira de Oliveira  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil  
[andre.oliveira@ccc.ufcg.edu.br](mailto:andre.oliveira@ccc.ufcg.edu.br)  
Orientação: Fábio Jorge Almeida Moraes

## RESUMO

O processo de aprendizagem, dê-se ele por um agente humano ou artificial, nem sempre é algo fluido e fácil e costuma levar várias etapas de treinamento. Mesmo assim, aplicações baseadas em inteligência artificial e Processamento de Linguagem Natural têm mostrado resultados surpreendentes e costumam melhorar através da exposição a novas situações. Buscando explorar e aplicar os conceitos e avanços desta área ao aprendizado humano, este trabalho visa propor o desenvolvimento de uma aplicação, utilizando técnicas de PLN, onde o usuário, ao inserir um texto do seu interesse, será apresentado a questões geradas automaticamente, abrangendo as principais ideias e pontos-chaves contidos nele. Auxiliando no que diz respeito à fixação e assimilação de conteúdos, assistindo na memorização de tópicos, definições e conceitos. Com isto, visa-se agilizar o processo de estudo e memorização de temas de interesse do usuário, facilitando o processo de aquisição e consolidação de conhecimentos.

## Palavras-Chave

Questões, PLN, Aprendizado.

## Repositórios

- <https://github.com/andregom/quizzify-backend>
- <https://github.com/andregom/quizzify-app-ts>

## 1. INTRODUÇÃO

O processo de aprendizagem nem sempre costuma ser algo fácil<sup>[2]</sup>. Por vezes, nos vemos nos digladiando com a nossa memória para evocar algum conceito que fora aprendido durante uma leitura que ocorreu há

apenas alguns poucos minutos<sup>[3]</sup>. Todavia, revisitar o trecho lido referente a tal ideia ou conceito, costuma, ainda que provisoriamente, resolver o problema e preencher a lacuna recém adquirida.

Isso é ainda mais flagrante quando é maior o tempo entre a leitura ou assimilação da informação e a aferição ou validação do conhecimento aprendido<sup>[4]</sup>. Este fato se dá, pois o esquecimento é algo quase tão presente e, em menor grau, até útil, para o aprendizado quanto o ato de lembrar, auxiliando no processo de generalização<sup>[8]</sup>. Podendo ele ocorrer, inclusive, no aprendizado de máquina<sup>[3]</sup>.

Tendo em vista essa dificuldade, faz-se necessário não somente uma nova passagem pela leitura ou informação que queremos fixar ou interiorizar, mas também um teste a fim de assegurar que o conteúdo que fora aprendido se encontra consistente e em pleno acordo com o material original, que lemos, vimos ou ouvimos<sup>[5]</sup>.

Uma forma bastante utilizada para isso, é a elaboração de quizzes ou questionários baseados nos assuntos estudados, cuja estrutura, composta por questões, nos força a fazer consultas à nossa memória, buscando uma certa informação, com base em uma pergunta ou padrão no qual ela se encaixe. Modalidade esta que foi bastante adotada no ensino remoto.

É sabido que aplicações baseadas em inteligência artificial costumam evoluir e se aprimorarem em suas tarefas conforme vão obtendo mais experiência. E que técnicas de Machine Learning aplicadas a Processamento de Linguagem Natural (PLN) têm mostrado resultados surpreendentes<sup>[1]</sup>.

Imagine, pois, se pudéssemos utilizar o poder dessas descobertas e o avanço nesta área para auxiliar no processo de aprendizagem natural e, ao fim de uma determinada leitura, ter uma forma de perturbar e afixar o que foi lido por nossos olhos, assegurando que nossa mente os acompanhou no processo.

Esta, por fim, é a proposta do presente trabalho. Proporcionar uma aplicação que, recebendo um texto qualquer, por meio das técnicas de reconhecimento de padrões e previsão textual em PLN, elabore um quiz ou questionário sobre os principais termos e definições nele presentes. Tendo como objetivo reduzir o tempo de estudo e facilitar a absorção de um determinado assunto, tornando o processo mais interessante e, até mesmo, gratificante através de algo inovador.

Desta forma, o presente trabalho, faz uso de modelos pré-treinados e seus respectivos tokenizadores para atingir tal objetivo. Empregando-se o framework Python Flask na construção de um backend que compreenderá a lógica de negócio principal, comunicando-se, através de uma API, com um frontend, a ser implementado utilizando-se os frameworks React e Electron a fim de construir uma interface prática e de simples usabilidade. Visa-se proporcionar ao usuário uma aplicação para a geração de questionários automáticos com base em um texto a fim de aferir e auxiliar o entendimento do leitor sobre o material.

## 2. MOTIVAÇÃO

A maioria dos modelos de PLN hoje são voltados para a análise de sentimentos e classificação ou previsão textual. Estas técnicas se mostram muito eficazes para as mais diversas tarefas nos mais variados escopos, como analisar a impressão que um espectador teve em relação a um filme, com base em uma avaliação escrita por ele, ou ainda, se um tweet diz respeito a determinado assunto e qual a opinião de quem tuitou sobre o tema<sup>[4]</sup>.

Uma razão para isso é que automatizar esse processo de análise pode ser muito interessante para empresas que lidam com uma enorme quantidade de dados e desejam extrair deles informações relevantes, como a

impressão de seus clientes sobre um determinado produto, serviço ou assunto.

Outros avanços mais robustos na área, como o framework Generative Pre-trained Transformer 3, ou GPT-3<sup>[28]</sup>, produto dos laboratórios da organização OpenAI<sup>[20]</sup>, mostram todo o poder de técnicas de Deep Learning aplicadas ao PLN<sup>[27]</sup>. Trata-se de um modelo de geração de texto presente na biblioteca transformers<sup>[29]</sup> que se utiliza da arquitetura de mesmo nome.

O que todas essas ferramentas têm em comum é que elas são pensadas para executar tarefas ou responder questões feitas ou requeridas por pessoas. Porém, nenhuma delas, pelo menos não diretamente, se propõe a fazer o contrário, ou seja, questionar a sua contraparte. Mas qual seria a utilidade disso? Como já foi mencionado, aprender não costuma ser um processo fácil, dado que algumas memórias têm um caráter volátil. E, por isso, informações carecem de revisão e manutenção após serem aprendidas.

Como visto no trabalho de Ruger, Henry A. et al. Memory: A Contribution to Experimental Psychology<sup>[19]</sup>, certas informações costumam ter um prazo de validade em nossa memória, denotado por uma taxa de esquecimento, que depende, dentre outros fatores, da dificuldade do material que está sendo aprendido, seu grau de relevância para o aprendente e até o estado físico e mental no qual este se encontra, levando em conta fatores como estresse e sonolência<sup>[4]</sup>.

O conceito de curva do esquecimento é antigo e foi criado pelo psicólogo alemão Hermann Ebbinghaus (1850 – 1909)<sup>[19]</sup>. O autor, por sua vez, ainda conjectura que a taxa de esquecimento basal pode diferir entre indivíduos devido a, dentre outros fatores, suas habilidades de representação mnemônica, por exemplo<sup>[4]</sup>. Ebbinghaus propõe a equação abaixo para modelar a ‘economia’  $b$  em termos de porcentagem em relação a um tempo  $t$  que seria necessário para revisar um assunto<sup>[4]</sup>.

$$b = \frac{100k}{c \log(t) + k}$$

Visto isso, pensou-se em utilizar o poder proporcionado pelos avanços obtidos na área de Machine Learning e PLN aplicados ao processo de aquisição de informações novas, ou seja, aprendizagem. Colocando, agora, o agente artificialmente treinado não na posição de responder perguntas, mas de formulá-las, a fim de aferir e auxiliar no nível de aprendizagem obtido por um leitor frente a um texto qualquer.

### **3. PROBLEMÁTICA E TRABALHOS RELACIONADOS**

#### **3.1. Modelos pré-treinados**

Uma nova tendência tomou espaço no que diz respeito ao treinamento e otimização de modelos em Machine Learning, dando origem ao que chamamos de modelos pré-treinados.

O treinamento de grandes modelos de Machine Learning é algo computacionalmente oneroso. Este, ao lado da mineração e processamento de dados, figura entre os principais contribuidores do setor de tecnologia para o aumento da, já fora de controle, pegada de carbono deixada pelo homem nesta seara. Métrica essa que avalia impacto ambiental em decorrência das atividades humanas<sup>[8]</sup>.

Contudo, grandes empresas têm investido bastante no treinamento de modelos com bases de dados enormes, beirando a ordem de tamanho da informação disponível na internet atualmente<sup>[9]</sup>. Porém, apesar de ainda requererem grandes recursos e consumo de energia para isto, muitas dessas empresas têm, com sucesso, reduzido o consumo de energias não renováveis envolvidas na concepção e aperfeiçoamento de novos modelos<sup>[10]</sup>.

Além de obter avanços no custo computacional necessário ao processo de treinamento, a adoção deste tipo de modelo nos fez repensar o que eles seriam capazes de fazer, particularmente no campo de PLN. Adicionalmente, possibilitou a obtenção de modelos especializados em tarefas bastante complexas, com apenas uma fração do esforço computacional que seria necessário para treiná-los do zero.

Em 2017, com a publicação do artigo “Attention is all you need”<sup>[11]</sup> e o surgimento da biblioteca Transformers, sendo disponibilizada na comunidade acadêmica e open source, ganhou-se um novo entendimento do que estes modelos seriam capazes de realizar.

A utilização destas ferramentas, portanto, se mostrou a abordagem mais promissora no que diz respeito à problemática de obter um algoritmo que, automaticamente, gerasse perguntas com base em um texto de entrada. Tendo em vista que tal tarefa é uma das principais propostas da aplicação idealizada neste trabalho.

#### **3.2. Trabalhos prévios e escolha do modelo**

Como visto anteriormente, hoje temos uma infinidade de modelos disponibilizados gratuitamente na comunidade open source de Machine Learning que podem ser utilizados para as mais diversas funções e um conjunto que chama bastante atenção nesse meio são os modelos pré-treinados. Eles se utilizam de uma técnica chamada transfer learning, o que lhes permite serem reusados para um leque de atribuições, desde que delimitados a um certo escopo e aperfeiçoados de forma adequada para tal.

A este processo de aperfeiçoamento dá-se o nome de fine tuning, termo herdado da Física, que diz respeito ao ajuste preciso das constantes de um modelo teórico para sua validação em observações empíricas. Neste caso, no entanto, o ajuste, em geral, se dá em alguns pesos de algumas camadas de uma rede neural.

Explorando a biblioteca transformers da Hugging Face é possível encontrar diversos modelos pré-treinados voltados para tarefas de análise e classificação textual, que vão desde análise de sentimentos, passando por sumarização e elaboração de respostas com base em uma pergunta e, até mesmo, geração textual em diferentes estilos.

#### **3.3. O modelo T5 da biblioteca transformers**

Por ser mais generalista e ter se mostrado adequado às mais diversas tarefas, o T5<sup>[8]</sup>, um dos modelos disponibilizados na biblioteca transformers da Hugging Face, provou ser a melhor escolha para servir como principal responsável pela lógica de

negócio desta aplicação, dado sua versatilidade e flexibilidade, como visto em outros estudos<sup>[12]</sup>.

O modelo T5 foi proposto em Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer<sup>[9]</sup> por Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu.

Ele funciona bem para uma infinidade de tarefas, como sumarização, classificação textual e mais. Por ser um modelo pré-treinado, ele se utiliza da técnica de transfer learning, sendo enriquecido com uma vasta base de dados, para depois passar pelo processo de fine tuning voltado para uma tarefa específica que se deseja.

Trata-se de um modelo codificador-decodificador multitarefa pré-treinado nas formas tanto supervisionada como não supervisionada, em uma configuração generativa de redução de ruído de texto para texto<sup>[8]</sup>.

Além disso, no seu treinamento é aplicada uma técnica chamada Teacher Forcing, que, em termos gerais, significa que as respostas corretas são reutilizadas para retroalimentar o modelo após suas previsões. Como se houvesse um professor ou tutor guiando o caminho na direção correta.

### 3.4. Base de dados

O "SQUAD 2.0" é um robusto conjunto de dados de compreensão de leitura composto por seções com respostas, contextos e perguntas, retirados das interações entre Crowd Workers da Wikipédia. Ele é disponibilizado atualmente de maneira gratuita através de uma licença Open Source pela Universidade de Stanford<sup>[14]</sup>.

Geralmente, essa base de dados é utilizada para a obtenção de respostas frente a uma pergunta, mas, com algumas adaptações, esta mesma base de dados pode ser utilizada invertendo-se este fluxo, obtendo perguntas com base em um contexto e uma resposta a ser dele extraída<sup>[12]</sup>.

## 4. OBJETIVOS

Com o desenvolvimento desta aplicação, visa-se impactar a aprendizagem numa etapa particular do processo e, possivelmente, no seu ato como um todo, movendo-se num sentido de gamificação e facilitação da tarefa, propondo uma solução, dentre tantas outras possíveis, para o aperfeiçoamento do processo de memorização, assimilação e interiorização de conceitos.

No escopo do usuário, espera-se que seja notado um menor tempo na fixação de tópicos estudados e percebida a praticidade proporcionada pela geração automática de questões, visando a fixação do assunto lido. Com esta finalidade, faz-se necessário, antes de tudo, a escolha de um modelo, base de dados e método de treinamento que possibilite o processamento de um texto qualquer e a obtenção dos resultados desejados, ou seja, perguntas geradas automaticamente a partir dele. A biblioteca Transformers oferece algumas ferramentas e modelos pré-treinados que podem servir a este propósito.

Escolhido o modelo e concluída sua etapa de treinamento, este deverá, então, ser inserido dentro de uma API para que possa se conectar a uma interface simples e intuitiva, possibilitando ao usuário uma interação fácil e direta. Para isto, pretende-se desenvolver uma aplicação desktop utilizando Electron, dado o seu caráter multiplataforma e compatibilidade com o padrão Web, bem como compatibilidade com os principais Sistemas Operacionais, Windows, Linux e Mac, onde, uma vez feito seu deploy específico para o S.O. em questão, poderá ser instalada em qualquer um deles.

Portanto, a aplicação será dividida em dois escopos, adotando uma arquitetura do tipo cliente/servidor e o padrão HTTP como meio de comunicação entre as partes. Visando-se, assim, não restringir a interface do usuário apenas a uma aplicação desktop, como foi inicialmente pensado, podendo esta, futuramente, ser expandida para outros tipos, como Web e Mobile.

## 5. METODOLOGIA

Em geral, nem todas as informações necessárias para se trabalhar com um texto estão em seu corpo. Às vezes, precisamos reescrever, reeditar ou agrupar certos termos ali presentes, ou, até mesmo, buscar exemplos similares a eles em outras fontes. Para que

isso se dê de forma sólida e coesa, porém, faz-se necessário a busca de informações externas, ou o chamado conhecimento de mundo.

Tendo em vista que o presente trabalho visa prover uma modalidade de questionário, mais precisamente o de múltipla escolha. Faz-se necessário, antes de tudo, buscar o conhecimento de mundo, para nele apoiar-se no processo de geração de distratores<sup>[2]</sup>, ou seja, opções falsas que, juntamente com a resposta correta, constituem as alternativas de uma questão de múltipla escolha.

Para isso, buscou-se na Internet ferramentas capazes de relacionar palavras com base no seu significado. Nesta busca nos deparamos com algumas ferramentas capazes de agrupar e classificar palavras quanto a sua formação, tipologia e ou semântica.

No entanto, poucas ou nenhuma ferramentas ou APIs que disponibilizassem tais informações em Português foram encontradas, sendo a maioria delas disponibilizadas em Inglês, sendo as principais delas a Wordnet, Sense2Vec e Conceptnet. Em consequência disso e de outros fatores, a fim de evitar possíveis gargalos ou impedimentos futuros, escolheu-se o Inglês como idioma base a ser utilizado na aplicação.

### 5.1.Implementação

Abaixo, na figura 1, podemos ver um diagrama com os principais componentes e funcionalidades da aplicação:

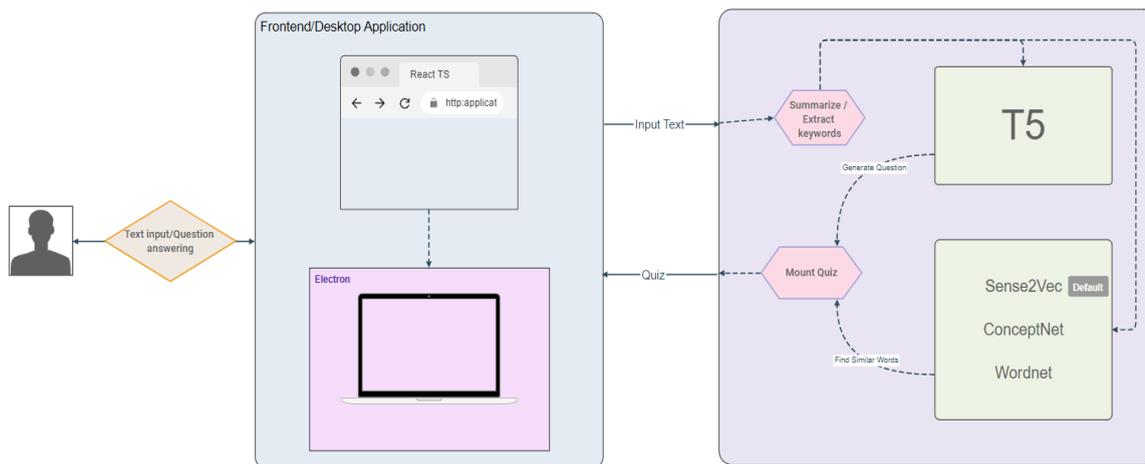


Figura 1: Diagrama de componentes e funcionalidades da aplicação

### 5.2. Backend e lógica de negócio

Já de início, por ter uma sintaxe mais simples e oferecer grande suporte e ferramentas provenientes da comunidade para lidar com análises de dados e textuais, Python foi a linguagem escolhida para implementar toda a lógica de negócio das principais funcionalidades presentes no backend da aplicação. Para a instalação e controle de versionamento de pacotes, decidiu-se utilizar como ferramenta o Pipenv, por prometer uma interface simples, adaptável e compatível com diferentes sistemas para o gerenciamento de ambientes virtuais Python.

### 5.3.Base de dados, modelo e método de treinamento

Com objetivo de fazer o fine tuning do modelo para a tarefa específica de geração de perguntas, faz-se necessário, antes de tudo, a escolha de uma base de dados. Esta deve ser composta por exemplos de perguntas e respostas, para fornecer ao modelo, em quantidade satisfatória, dados de entrada que irão moldar seu comportamento. Adequando-o ao propósito final da tarefa e à obtenção dos resultados desejados, ou seja, uma série de questões geradas de maneira automática a partir de um texto.

Como já visto, uma base de dados que se encaixa e adequa a esses moldes é a Stanford Question Answering Dataset (SQuAD 2.0). Disponibilizada através de uma licença do tipo creative commons pela

Universidade de Stanford. Esta base de dados é composta por três segmentos, consistindo-se em um data frame com dados agrupados em contexto, algumas perguntas sobre ele e suas respectivas respostas.

Já para a escolha do modelo, procurou-se priorizar características como versatilidade e adaptabilidade, baseando-se em estudos prévios<sup>[12]</sup> sobre a utilização de modelos para a tarefa desejada.

Uma vez escolhidos, base de dados e modelo, passamos então à fase de planejamento sobre como se daria o treinamento do mesmo de forma a obter resultados satisfatórios no tocante à lógica de negócio da aplicação.

## 5.4. Treinamento do modelo

Para treinar o modelo é necessário, antes de tudo, fazer a importação da mencionada base de dados e aguardar seu download. Felizmente, isso se dá de maneira bem simples, com apenas algumas poucas linhas de código através da função `load_datasets` da biblioteca `datasets` do Python, seu corpo já vem dividido em dados de treinamento e validação, restando somente agrupá-los em diferentes variáveis. A Figura 2 descreve como esse processo é realizado por meio de um código em python.

```
from pathlib import Path
import pandas as pd
from sklearn.utils import shuffle
from datasets import load_dataset
from pprint import pprint

STORAGE_FOLDER_PATH = Path(__file__).resolve().parents[1].joinpath('storage')
DATASETS_FOLDER_PATH = STORAGE_FOLDER_PATH.joinpath('datasets')

train_dataset = load_dataset('squad', split='train')
valid_dataset = load_dataset('squad', split='validation')

sample_validation_dataset = next(iter(valid_dataset))
pprint(sample_validation_dataset)

context = sample_validation_dataset['context']
question = sample_validation_dataset['question']
answer = sample_validation_dataset['answers'][0]['text']

pd.set_option('display.max_colwidth', None)
df_train = pd.DataFrame(columns = ['context', 'answer', 'question'])
df_validation = pd.DataFrame(columns = ['context', 'answer', 'question'])

print(df_validation)
print(df_train)
```

Figura 2: Download e preparação da base de dados.

Uma vez baixadas as bases de dados, separamo-las em diferentes versões, de treinamento e validação. Tentou-se fazer o fine-tuning do modelo localmente. Contudo, mesmo com uma GPU com 4 Gigabytes de

RAM, a tarefa se mostrou inviável, sempre incorrendo em estouro de memória devido ao tamanho das entradas da mencionada base de dados.

A solução encontrada para isso, foi utilizar um notebook colaborativo com uma placa de vídeo mais robusta. Mesmo assim, para obter um treinamento com 100 epochs, foram necessárias cerca de 4 horas. Finalizado o treinamento, o modelo foi armazenado em um diretório do Google Drive para ser posteriormente utilizado pela aplicação.

No arquivo do Jupyter Notebook, fez-se a preparação e tokenização dos datasets, rotulando os contextos e respostas presentes em seu corpo como entradas (inputs) e questões a eles relacionadas como alvos (targets). Em seguida, usando-se o framework Pytorch Lightning e, adaptando todos elementos envolvidos ao padrão requerido por ele, com as funções de treinamento, passo e validação, fez-se o treinamento do modelo.

## 5.5. Buscando informações online

A Wordnet é uma grande base de dados que reúne grupos de palavras interligadas por meio de relações conceitual-semânticas e lexicais. Cada um dos 117.000 synsets, um conjunto de um ou mais sinônimos que são intercambiáveis em algum contexto sem alterar o valor verdade da proposição na qual estão inseridos<sup>[26]</sup>, presentes na WordNet está vinculado a outros synsets por meio de um pequeno número de “relações conceituais”. Além disso, um synset contém uma breve definição (“gloss”) e, na maioria dos casos, uma ou mais frases curtas que ilustram o uso dos membros do synset<sup>[16]</sup>.

Uma base de dados semelhante foi encontrada em português, como a “WordNet.PT” e “WordNet.Br” mas todas ou a grande maioria não tinha suporte por parte do NLTK, Natural Language Toolkit, uma ferramenta bastante utilizada no tratamento e processamento de dados de linguagem natural.

## 6. Aplicação em uso

Como dito anteriormente, a linguagem de trabalho escolhida para o processamento, entradas e saídas da aplicação foi o inglês. Na Figura 3, podemos ver um exemplo de uma simples interface gráfica provisória

rodando com ajuda da biblioteca Grad.io, para visualização de entradas e saídas, da aplicação em uso.

The screenshot shows a web interface with three main sections. The top section, labeled 'context', contains a Wikipedia-style text block about Alan Turing, mentioning his birth in 1912, his work at King's College, Cambridge, and his role at the Government Code and Cypher School (GC&CS) during World War II. The middle section, labeled 'answer', contains a text input field with the text 'National Physical Laboratory'. Below this are two buttons: 'Clear' and 'Submit'. The bottom section, labeled 'Question', contains a text input field with three questions: '[question: Where did Turing work after the war?', 'question: What was Turing's job after the war?', 'question: What was Alan Turing's job after the war?']

Figura 3: Exemplo de saída do modelo utilizando uma interface gráfica simplificada.

Onde com um contexto retirado da Wikipédia [\[15\]](#) servindo como entrada e a resposta desejada sendo “National Physical Laboratory”, as perguntas obtidas são “Where did Turing work after the war?”, “What was Turing's job after the war?” e “What was Alan Turing's job after the war?”.

Já para o termo numérico “1948”, as perguntas obtidas foram, "What year did Alan Turing join Max Newman's Computing Machine Laboratory?", "When did Alan Turing join Max Newman's Computing Machine Laboratory?", "When did Turing join Max Newman's Computing Machine Laboratory?".

No entanto, como era de se esperar, em algumas entradas, o resultado obtido não é o ideal e chega a não fazer tanto sentido assim.

Outras vezes, como por exemplo, quando apresentado ao seguinte texto:

*“Crowdworkers are a nonrandom sample of the population. Many researchers use crowdsourcing to quickly and cheaply conduct studies with larger sample sizes than would be otherwise achievable.*

*However, due to limited access to the Internet, participation in low developed countries is relatively low.”*

E, tendo o termo “*limited Internet*” como resposta, a saída foi constituída de duas afirmações e uma pergunta. [*question: Due to limited access to the Internet, participation in low developed countries is relatively low*’, ‘*question: Because of limited access to the Internet, participation in low developed countries is relatively low*’, ‘*question: Why is participation in low developed countries relatively low?*’]. Mostrando que na maioria das vezes, pelo menos uma das três amostras de questões é satisfatória. Como podemos ver em outro exemplo, com as seguintes entradas e saídas.

*“Barack Hussein Obama II born August 4, 1961 is an American politician who served as the 44th president of the United States from 2009 to 2017. A member of the Democratic Party, he was the first African-American president of the United States. Obama previously served as a U.S. senator from Illinois from 2005 to 2008 and as an Illinois state senator from 1997 to 2004”, “Barack Hussein Obama”. [*question: Who was the 44th president of the United States?*’, ‘*question: What was the 44th president of the United States?*’, ‘*question: What was the first African-American president of the United States?*’]. Sendo a última mais relevante.*

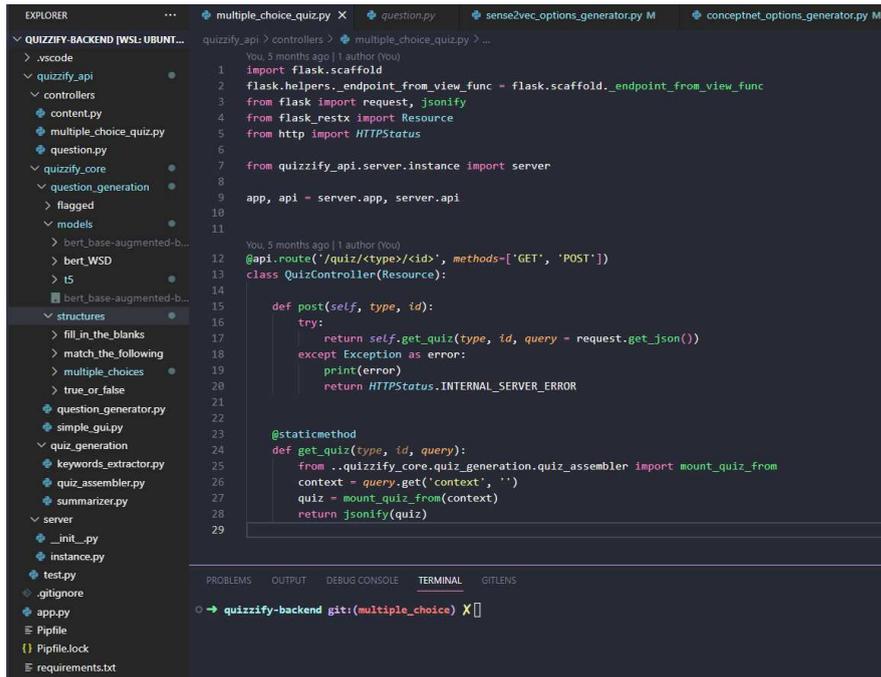
No entanto, como podemos ver, as questões são de caráter mais objetivo, pois assim se encontravam na base de dados, como um breve exame dos exemplos lá presentes pode confirmar.

## 7. Desenvolvimento e integração

### 7.1.1. Lógica de negócio

Dentro da pasta `quizzify_api`, temos os controladores, “`controllers`”, responsáveis por expor os endpoints responsáveis por retornar questões ou quizzes. Estes endpoints, por sua vez, fazem chamadas às funções responsáveis por gerar perguntas ou montar quizzes com base no conteúdo enviado através da requisição. Estas funções, sendo elas “`generate_question`” e “`mount_quiz_from`”, se encontram dentro do diretório `quizzify_core`, lugar que comporta a parte principal de

toda a lógica de negócio da aplicação. Como podemos ver na figura 4.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for a project named 'QUIZZIFY-BACKEND'. The code editor shows the following code:

```
1 import flask.scaffold
2 flask.helpers._endpoint_from_view_func = flask.scaffold._endpoint_from_view_func
3 from flask import request, jsonify
4 from flask_restx import Resource
5 from http import HTTPStatus
6
7 from quizzify_api.server.instance import server
8
9 app, api = server.app, server.api
10
11
12 @api.route('/quiz/<type>/<id>', methods=['GET', 'POST'])
13 class QuizController(Resource):
14
15     def post(self, type, id):
16         try:
17             return self.get_quiz(type, id, query = request.get_json())
18         except Exception as error:
19             print(error)
20             return HTTPStatus.INTERNAL_SERVER_ERROR
21
22
23 @staticmethod
24 def get_quiz(type, id, query):
25     from .quizzify_core.quiz_generation.quiz_assembler import mount_quiz_from
26     context = query.get('context', '')
27     quiz = mount_quiz_from(context)
28     return jsonify(quiz)
```

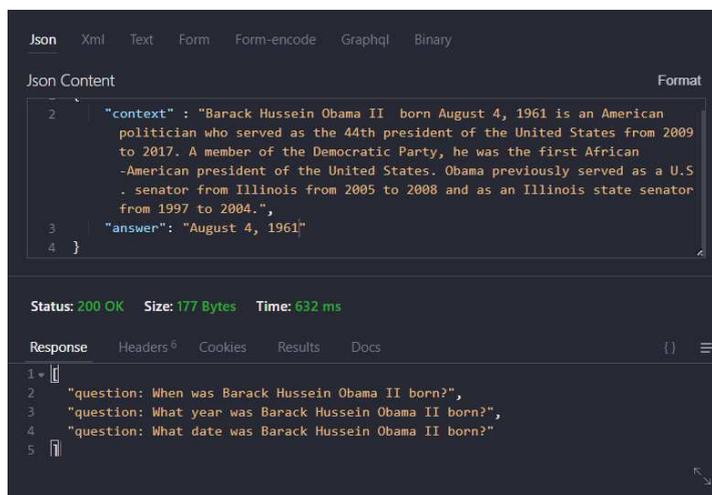
Figura 4: Estrutura de diretórios e endpoint responsável pela geração de um quiz.

A função “mount\_quiz\_from”, por sua vez, presente no arquivo “quiz\_assembler.py”, recebe um parâmetro “context” de caráter textual, do qual irá extrair alguns pontos-chave, os quais utilizará para obter questões de múltipla escolha utilizando duas funções. A “generate\_question”, que fazendo uso do trecho e ponto chave em questão, utilizará predições do modelo treinado para gerar o corpo da questão e a “get\_filtered\_similar\_options\_to” que, fazendo uso de alguma das bases de dados de relacionamento de palavras, gerará os distratores da questão.

Para disponibilizar os dados e fazer a comunicação com o núcleo da aplicação, responsável pela lógica de negócio, resolveu-se utilizar como ferramenta o framework Flask. Por ser leve, simples e estar em uniformidade com a linguagem que vinha sendo usada.

Abaixo, na Figura 5, podemos ver um exemplo de chamada à API requisitando uma pergunta referente a um termo específico.

### 7.1.2. API



The screenshot shows a REST client interface with a JSON response. The response is as follows:

```
{
  "context": "Barack Hussein Obama II born August 4, 1961 is an American politician who served as the 44th president of the United States from 2009 to 2017. A member of the Democratic Party, he was the first African-American president of the United States. Obama previously served as a U.S. senator from Illinois from 2005 to 2008 and as an Illinois state senator from 1997 to 2004.",
  "answer": "August 4, 1961"
}
```

Status: 200 OK Size: 177 Bytes Time: 632 ms

Response Headers Cookies Results Docs

```
1
2 "question: When was Barack Hussein Obama II born?",
3 "question: What year was Barack Hussein Obama II born?",
4 "question: What date was Barack Hussein Obama II born?"
5
```

Figura 5: Exemplo de chamada a API através de requisição HTTP para a obtenção de um conjunto de questões frente a um contexto e uma resposta.

Já abaixo, Figura 6, podemos ver a montagem de um questionário com base no mesmo texto.

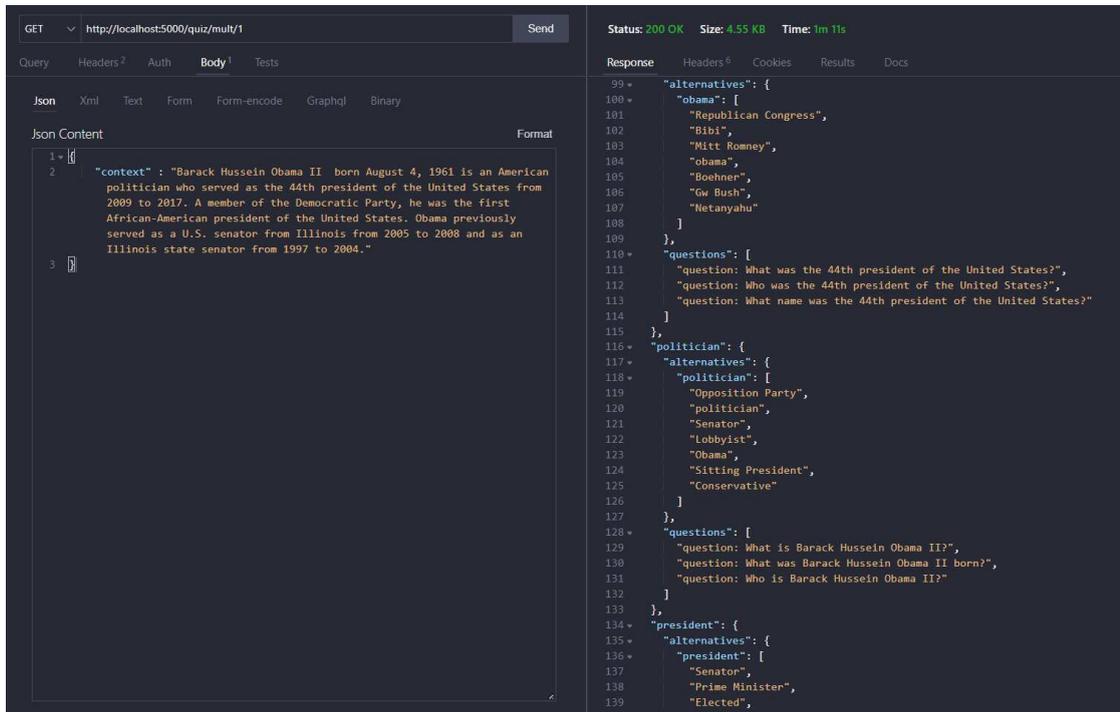


Figura 6: Chamada à API para montagem de questionário.

Como podemos ver, nem todas as saídas (perguntas) são ideais para suas correlatas respostas e alternativas. Uma saída para isso seria implementar uma forma de avaliar e ranquear essas perguntas conforme seus itens relacionados e o seu nível de adequação aos mesmos para, assim, contornar essas incompatibilidades, obtendo um banco de dados apenas com as melhores perguntas filtradas. Mas essa seria uma proposta para outro momento.

### 7.1.3. Interface Frontend

Tendo em vista seu caráter multiplataforma e a agilidade no desenvolvimento proporcionada por estas ferramentas, utilizou-se, em conjunto, os frameworks React<sup>[23]</sup> e Electron<sup>[24]</sup> na construção de uma interface de usuário simples, a fim de criar uma aplicação do tipo desktop compatível tanto com Windows como Mac ou Linux. Possibilitando-se assim, futuramente, a expansão desta interface para as plataformas tanto Web quanto Mobile, uma vez que se usa o padrão HTTP para a comunicação entre as partes.

Procurou-se manter as funcionalidades básicas. Abaixo, nas figuras 7 e 8, algumas telas da interface

gráfica rodando como em uma janela de aplicação nativa do sistema operacional que a comporta.

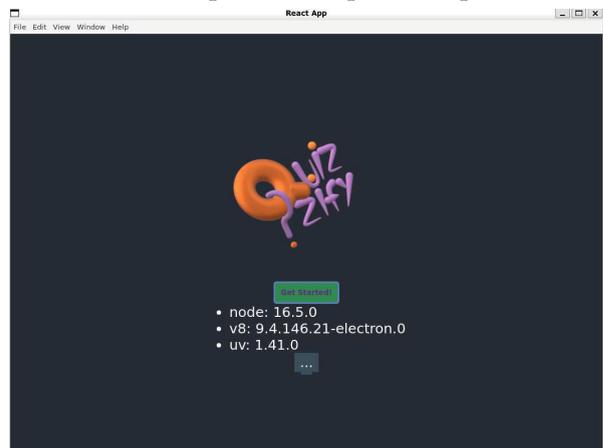


Figura 7: Tela de boas vindas rodando numa janela nativa com algumas informações de tecnologias suas versões.

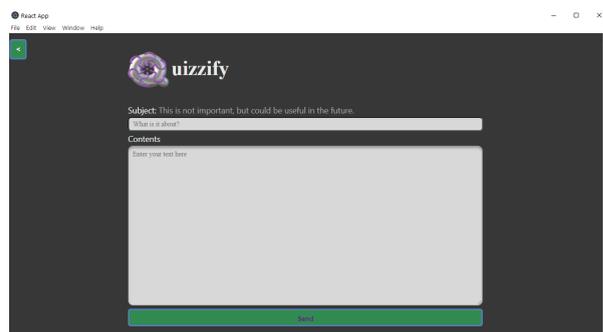


Figura 8: Interface de usuário pronta para receber uma entrada de texto.

## 8. CONCLUSÃO

### 8.1. Resultados alcançados

No que diz respeito a geração de perguntas, pelo modelo, mesmo obtendo resultados um pouco destoantes e poluídos por casos que não faziam tanto sentido assim, ainda podemos considerar satisfatório e, até mesmo, surpreendente o retorno de suas predições. Isso se deve a um fenômeno que é proporcionado pelo aprendizado de máquina, que é obter resultados com poucas ou nenhuma instruções diretas sobre a tarefa a ser executada.

Alguns resultados podem ser observados na seção “Artefatos” ao final do documento.

Apesar das falhas, o desempenho da aplicação se mostrou satisfatório nas principais funcionalidades a que se propôs, sendo ela objetivada para ser antes uma prova de conceito do que qualquer outra coisa.

### 8.2. Sugestões e melhorias futuras

Como pontos de ação para a melhoria das funcionalidades já implementadas, poderia-se agir no problema de redundância que gera perguntas com respostas muito óbvias, às vezes, sinônimos e, até mesmo, palavras homônimas ao que está sendo perguntado.

Isso incorreria em uma investigação acerca do processo de treinamento e execução do modelo e medidas para mitigar o comportamento observado. Ou ainda, como já anteriormente proposto, fazer uma seleção apenas das melhores perguntas geradas, excluindo aquelas redundantes ou que não fizessem tanto sentido assim.

#### 8.2.1. Novas funcionalidades, revisão e o aprendizado como um processo contínuo

Como planos de ação para incrementar e aprimorar o presente trabalho, podemos sugerir a adição de novas funcionalidades que venham a potencializar sua utilização no auxílio da aprendizagem como uma tarefa contínua.

Visto que, a longo prazo, aplicando-se o conceito de curva do esquecimento basal, há uma diminuição no tempo necessário para revisar um dado assunto, encorajaria-se, portanto, o usuário, em intervalos de tempo regulares, a revisar o assunto de seu interesse. Estes intervalos de tempo seriam planejados de forma a otimizar não só a solidez da informação, mas também o tempo de estudo necessário para reavivá-la em futuras revisões, sendo este cada vez menor.

### 8.3. Considerações finais

Por fim, os avanços no meio tecnológico proporcionam várias facilidades e é inegável que, progressivamente, nos aproximamos de avanços surpreendentes que chegam a assustar e colocar a tecnologia num papel de algoz em relação a seus próprios criadores, chegando a ameaçar suas vagas de trabalho e trespassar suas capacidades. No entanto, percebemos que ainda se está um pouco longe disso, embora, talvez hoje, um pouco mais perto que antes.

Contudo, o uso da tecnologia nos trouxe um aumento das nossas potencialidades. Hoje temos um vasto leque de opções seja para crescimento pessoal ou profissional, tudo ao alcance dos nossos dedos.

Com o desenvolvimento desta aplicação, ainda que de forma modesta, pretendeu-se:

- Explorar o uso de técnicas de machine learning, mais precisamente PLN, no processo de desenvolvimento pessoal, profissional, educacional e afins.
- Procurar expandir o uso da tecnologia num sentido de proporcionar mais liberdade e bem-estar ao ser humano e aumentar suas potencialidades.
- Combater o preceito de que o aprendizado tem que ser algo difícil ou oneroso, buscando-se fazer com que mais e mais pessoas tomem gosto pelo processo.
- Proporcionar mais autonomia ao desenvolvimento individual.

## REFERÊNCIAS

- [1] PLN ou do Inglês NLP. Disponível em <<https://developer.ibm.com/br/technologies/natural-la>

[nguage-processing/articles/natural-language-processing-um-pouco-de-historia/](#)>

[2] The forgetting curve explains why humans struggle to memorize. Disponível em <<https://qz.com/1213768/the-forgetting-curve-explain-s-why-humans-struggle-to-memorize/>>

[3] The Psychology of Forgetting and Why Memory Fails. Disponível em <<https://www.verywellmind.com/forgetting-about-psychology-2795034>>

[4] Curva do esquecimento. Disponível em <[https://pt.wikipedia.org/wiki/Curva\\_do\\_esquecimento#cite\\_note-uwaterloo.ca-1](https://pt.wikipedia.org/wiki/Curva_do_esquecimento#cite_note-uwaterloo.ca-1)>

[5] Spaced repetition. Disponível em <[https://en.wikipedia.org/wiki/Spaced\\_repetition](https://en.wikipedia.org/wiki/Spaced_repetition)>

[6] Catastrophic forgetting in neural networks. Disponível em <<https://ieeexplore.ieee.org/abstract/document/323080>>

[7] Significado de Distrator. Disponível em <<https://www.lexico.pt/distrator/>>

[8] T5. Disponível em <[https://huggingface.co/docs/transformers/model\\_doc/t5](https://huggingface.co/docs/transformers/model_doc/t5)>

[9] Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Disponível em <<https://arxiv.org/abs/1910.10683>>

[9] The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations. Disponível em <<https://www.sciencedirect.com/science/article/pii/S2666389921001884>>

[10] The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink. Disponível em <[https://www.techrxiv.org/articles/preprint/The\\_Carbon\\_Footprint\\_of\\_Machine\\_Learning\\_Training\\_Will\\_Plateau\\_Then\\_Shrink/19139645](https://www.techrxiv.org/articles/preprint/The_Carbon_Footprint_of_Machine_Learning_Training_Will_Plateau_Then_Shrink/19139645)>

[11] Attention Is All You Need. Disponível em <<https://arxiv.org/abs/1706.03762?context=cs>>

[12] Deep Learning Based Question Generation Using T5 Transformer. Disponível em <[https://link.springer.com/chapter/10.1007/978-981-16-0401-0\\_18](https://link.springer.com/chapter/10.1007/978-981-16-0401-0_18)>

[13] Item lexical. Disponível em <[https://pt.wikipedia.org/wiki/Item\\_lexical](https://pt.wikipedia.org/wiki/Item_lexical)>

[14] The Stanford Question Answering Dataset. Disponível em <<https://rajpurkar.github.io/SQuAD-explorer/>>

[15] Alan Turing. Disponível em <[https://en.wikipedia.org/wiki/Alan\\_Turing](https://en.wikipedia.org/wiki/Alan_Turing)>

[16] WordNet: A Lexical Database for English. Disponível em <[https://wordnet-princeton-edu.translate.goog/?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=pt&\\_x\\_tr\\_hl=pt-BR&\\_x\\_tr\\_pto=sc](https://wordnet-princeton-edu.translate.goog/?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt-BR&_x_tr_pto=sc)>

[17] Natural Language Toolkit. Disponível em <<https://www.nltk.org/#natural-language-toolkit>>

[18] Barack Obama. Disponível em <[https://en.wikipedia.org/wiki/Barack\\_Obama](https://en.wikipedia.org/wiki/Barack_Obama)>

[19] Ruger, Henry A. (1913). *Memory: A Contribution to Experimental Psychology*. Disponível em <<https://archive.org/details/memorycontributi00ebbiuoft/page/4/mode/2up>>

[20] GPT 3. Disponível em <<https://beta.openai.com/docs/introduction>>

[21] O Teste de Cloze na Avaliação da Compreensão em Leitura. Disponível em <<https://www.scielo.br/pdf/prc/v15n3/a09v15n3.pdf>>

[22] Robust Machine Reading Comprehension by Learning Soft labels. Disponível em <<https://www.aclweb.org/anthology/2020.coling-main.248.pdf>>

[23] React. Disponível em <<https://pt-br.reactjs.org/>>

[24] Électron. Disponível em <<https://www.electronjs.org/docs>>

[25] Teacher Forcing. Disponível em  
<[https://en.wikipedia.org/wiki/Teacher\\_forcing](https://en.wikipedia.org/wiki/Teacher_forcing)>

[26] synset - Wikitionary. Disponível em  
<<https://en.wiktionary.org/wiki/synset>>

[27] Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models. Disponível em <<https://arxiv.org/abs/2102.02503>>

[28] GPT-3. Disponível em  
<<https://en.wikipedia.org/wiki/GPT-3>>

[29] Transformers: State-of-the-Art Natural Language Processing. Disponível em  
<<https://aclanthology.org/2020.emnlp-demos.6/>>

## APÊNDICE

### Artefatos

Abaixo, na Figura 9, alguns exemplos de questionários gerados pela aplicação:



Figura 9: Exemplo de questionário gerado tendo trecho do texto presente em <https://www.biography.com/scientist/alan-turing>.

A seguir, nas Figuras 10, 11,12 e 13, algumas questões selecionadas do questionário gerado na figura acima.

<p>question: What was Turing's major research area? <a href="#">Rephrase</a></p> <ul style="list-style-type: none"><li><input type="radio"/> Probability Theory</li><li><input type="radio"/> Group Theory</li><li><input type="radio"/> Other Sciences</li><li><input checked="" type="radio"/> <b>Mathematics</b></li><li><input type="radio"/> Subfield</li><li><input type="radio"/> Modern Physics</li><li><input type="radio"/> Formal Logic</li></ul> <p><a href="#">check</a></p>	<p>question: What was the foundational research of Alan Turing? <a href="#">Rephrase</a></p> <ul style="list-style-type: none"><li><input type="radio"/> Compsci</li><li><input type="radio"/> Biomedical Engineering</li><li><input checked="" type="radio"/> <b>Computer science</b></li><li><input type="radio"/> Physics Degree</li><li><input type="radio"/> Applied Math</li><li><input type="radio"/> Bioinformatics</li><li><input type="radio"/> Electrical Engineering</li></ul> <p><a href="#">check</a></p>
<p>question: What was the country where Turing was born? <a href="#">Rephrase</a></p> <ul style="list-style-type: none"><li><input type="radio"/> New Zealand</li><li><input type="radio"/> Northern Ireland</li><li><input checked="" type="radio"/> <b>England</b></li><li><input type="radio"/> British Isles</li><li><input type="radio"/> Continental Europe</li><li><input type="radio"/> Bavaria</li><li><input type="radio"/> Cornwall</li></ul> <p><a href="#">check</a></p>	<p>question: What university was Turing a fellow at? <a href="#">Rephrase</a></p> <ul style="list-style-type: none"><li><input checked="" type="radio"/> <b>Cambridge</b></li><li><input type="radio"/> Reading</li><li><input type="radio"/> Chesterfield</li><li><input type="radio"/> Stratford</li><li><input type="radio"/> Loughborough</li><li><input type="radio"/> Newcastle</li><li><input type="radio"/> Trinity College</li></ul> <p><a href="#">check</a></p>

Figuras 10, 11, 12 e 13: Exemplos de questões selecionadas geradas a partir do texto sobre Turing, extraídos a partir de um trecho de uma de suas biografias.

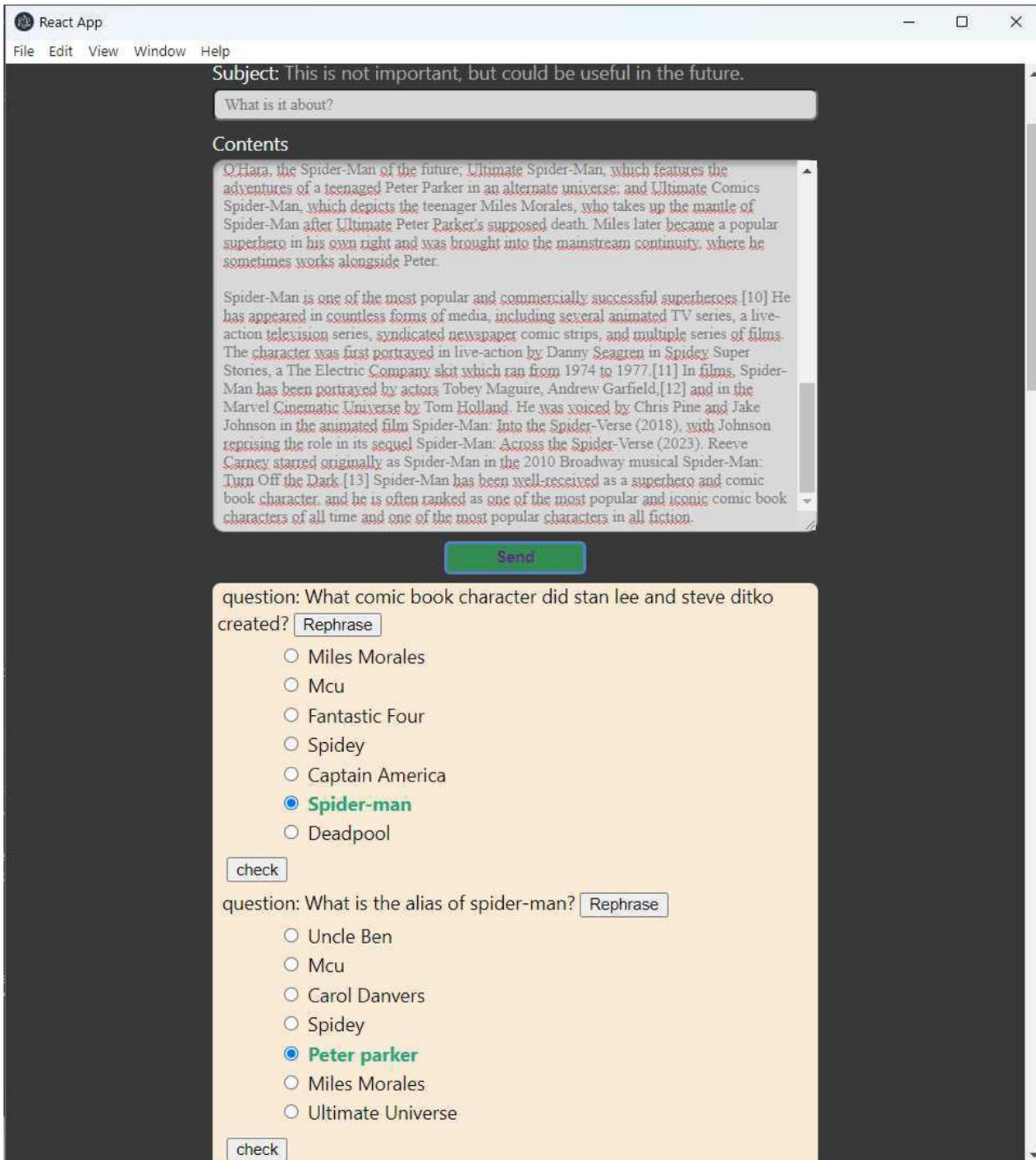


Figura 14: Exemplo de questionário gerado tendo trecho do texto presente em <https://en.wikipedia.org/wiki/Spider-Man>.

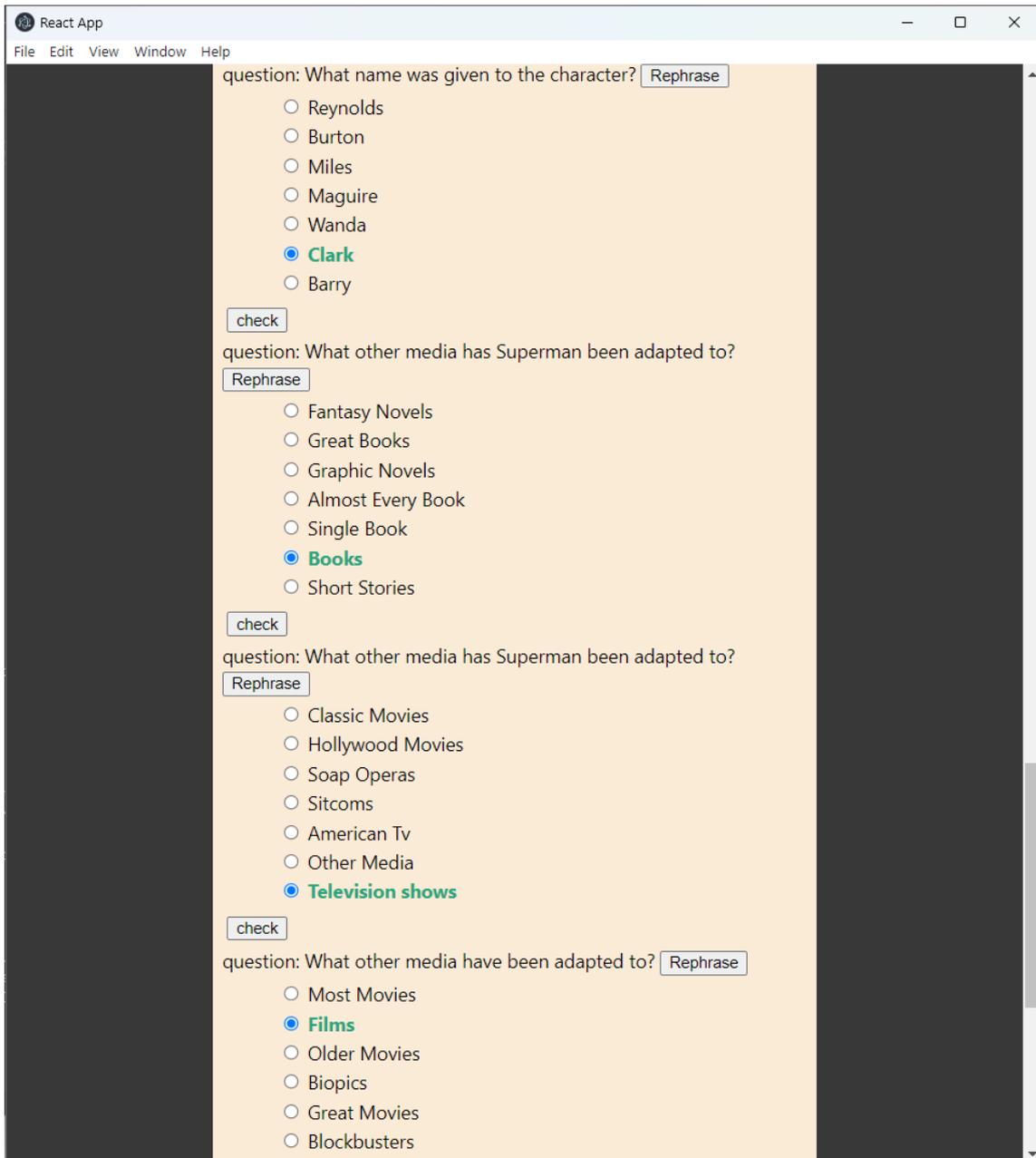


Figura 15: Questões geradas a partir de texto presente na Wikipédia sobre o personagem de histórias em quadrinhos Super-homem.