



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

CAIO FERNANDES MOREIRA

**APLICAÇÃO PARA MODELOS DE INTELIGÊNCIA ARTIFICIAL
QUE REALIZAM GERAÇÃO AUTOMÁTICA DE QUESTÕES A
PARTIR DE TEXTOS**

CAMPINA GRANDE - PB

2023

CAIO FERNANDES MOREIRA

**APLICAÇÃO PARA MODELOS DE INTELIGÊNCIA ARTIFICIAL
QUE REALIZAM GERAÇÃO AUTOMÁTICA DE QUESTÕES A
PARTIR DE TEXTOS**

**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 em
Ciência da Computação.**

Orientador: Professor Dr. Cláudio Elízio Calazans Campelo

CAMPINA GRANDE - PB

2023

CAIO FERNANDES MOREIRA

**APLICAÇÃO PARA MODELOS DE INTELIGÊNCIA ARTIFICIAL
QUE REALIZAM GERAÇÃO AUTOMÁTICA DE QUESTÕES A
PARTIR DE TEXTOS**

**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 em
Ciência da Computação.**

BANCA EXAMINADORA:

**Professor Dr. Cláudio Elízio Calazans Campelo
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Eanes Torres Pereira
Examinador – UASC/CEEI/UFCG**

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

Trabalho aprovado em: 14 de fevereiro de 2023.

CAMPINA GRANDE - PB

ABSTRACT

In recent times, advances in Artificial Intelligence have allowed the development of natural language processing models that may be used in various contexts, like the automation of the process of creating questions about specific themes. Currently, there are models capable of producing questions about any topic after being sent a relevant text as input. Such projects have great potential in the educational context, however, there is still a lack of a system that provides its users with an intuitive and simple way of utilizing these models. This project has as its goal the development of a web application to fulfill that need, incorporating natural language processing models that receive text as input and generate a list of questions relevant to the theme. The application will also bring an opportunity to collect feedback from users about the quality of the generated questions, information that may be used to improve the utilized models.

Aplicação para modelos de inteligência artificial que realizam geração automática de questões a partir de textos

Trabalho de Conclusão de Curso

Caio Fernandes Moreira (Aluno), Cláudio Campelo (Orientador)

Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Paraíba - Brasil

RESUMO

Nos últimos anos, avanços na área de Inteligência Artificial permitiram desenvolvimento de modelos de processamento de linguagem natural em prol de aplicações em diversos contextos, como a automatização do processo de elaboração de questões sobre temas específicos. Atualmente, existem modelos capazes de formular questões sobre um tópico qualquer após receber como entrada um texto relevante. Tais projetos possuem grande potencial auxiliar no contexto educacional, entretanto, ainda existe carência de um sistema que forneça a seus usuários uma maneira fácil e intuitiva de utilizar esses modelos. Esse trabalho tem como objetivo o desenvolvimento de uma aplicação web que supra essa necessidade, incorporando modelos de processamento de linguagem natural que recebem como entrada um texto e geram para o usuário uma lista de perguntas relevantes ao tema. A utilização da aplicação web também traz a oportunidade de obter feedback dos usuários sobre a qualidade das perguntas geradas, informação que pode ser utilizada para retroalimentação e aprimoramento dos modelos utilizados.

PALAVRAS-CHAVE

Processamento de Linguagem Natural, Modelo, Questões

REPOSITÓRIO

<https://github.com/caio-fm/QuestionGeneration>

1 INTRODUÇÃO

Os avanços na área de Inteligência Artificial (IA) dos últimos anos impulsionaram o desenvolvimento de várias ferramentas com intuito de automatizar processos custosos de forma a facilitar nossas vidas. A aplicação dessas novas tecnologias gerou vários produtos bem-sucedidos no mercado como Waze, Alexa e carros autônomos da Tesla. Segundo um relatório da Zion Market Research[2], em 2021 o mercado de Inteligência Artificial foi avaliado em 59,67 bilhões de dólares e as projeções indicam crescimentos anuais de 39,4%, chegando em um valor de 422,37 bilhões de dólares em 2028.

O sucesso alcançado por esses produtos é fruto de décadas de estudos realizados por institutos de pesquisa e universidades ao redor do mundo, que investiram fortemente na IA e suas subáreas,

criando o cenário atual onde suas aplicações estão cada vez mais presentes em nosso cotidiano.

Um dos ramos mais promissores na área de IA é o processamento de linguagem natural (PLN) [16], que permite análise de textos escritos em línguas naturais como português ou inglês. Com modelos de PLN, é possível desenvolver aplicações que conseguem não somente extrair significado desses textos, mas também são capazes de gerar novos textos a partir das informações obtidas.

A geração de textos traz um imenso potencial para automatização de tarefas. Várias equipes de desenvolvedores e pesquisadores ao redor do mundo estão investindo recursos em aplicações diversas como ferramentas de suporte ao cliente e preenchimento automático de formulários de registro de patentes[21].

O potencial dessa tecnologia não está somente em sua versatilidade, mas também na qualidade dos textos gerados. Lançado em novembro de 2022, o modelo ChatGPT[3] foi foco de vários artigos que ressaltam o seu alto nível de habilidade de escrita: o modelo foi capaz de receber notas acima da média em exames de cursos de Direito e Negócios[14] e de escrever um texto de candidatura de emprego que foi avaliado como acima do que 80% dos concorrentes[10].

Frente ao crescente potencial da área, esse projeto foi direcionado para envolvimento em soluções que utilizam modelos de linguagem. Especificamente, ele tem como base um modelo desenvolvido pelo Laboratório de Computação Inteligente Aplicada (LACINA) da Universidade Federal de Campina Grande (UFCG), que explora soluções para problemas utilizando métodos de aprendizagem de máquina, aprendizagem profunda, processamento de sinais digitais, entre outros.

O LACINA desenvolveu um modelo de PLN a fim de trazer esse tipo de solução para o contexto educacional. O modelo em questão recebe um texto e gera uma lista de perguntas sobre o tema, que podem ser então usadas para diversos propósitos, como um auxílio na elaboração de provas por professores ou como uma forma fácil para um aluno criar questões que o ajudem a revisar um assunto. Dessa forma, a automatização do processo de criação de questões tem potencial de liberar mais tempo de docentes e discentes para realização de outras atividades.

O modelo do LACINA foi treinado com um conjunto de questões utilizando a biblioteca Simple Transformers[24] e adotando o conceito da taxonomia de Bloom[22] (ilustrado na Figura 1) para permitir geração de questões que requerem níveis diferentes de engajamento com o tópico. O nível mais básico produz questões que podem ser resolvidas com memorização de fatos sobre o assunto, enquanto níveis acima avaliam habilidades como compreensão e aplicação dos conceitos relacionados ao assunto.

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

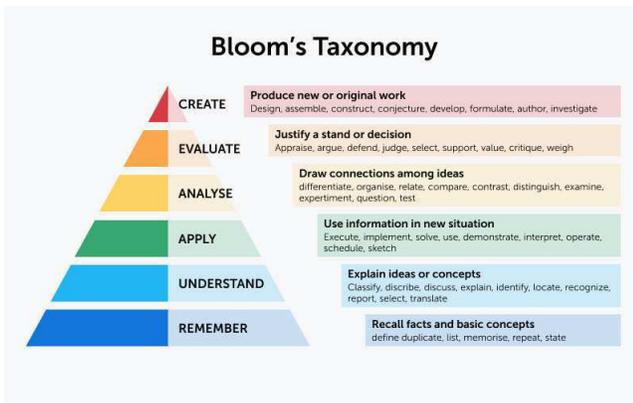


Figura 1: Esquema da taxonomia de Bloom

Atualmente, o modelo ainda não foi integrado em um software utilizável por usuários finais. Face a essa lacuna, o objetivo deste trabalho é desenvolver uma aplicação web intuitiva e responsiva capaz de integrar tais modelos de geração de questões como o que foi desenvolvido pelo LACINA.

A aplicação consiste de uma plataforma onde usuários podem criar uma conta e então utilizar o modelo para gerar as questões a partir de um texto qualquer e escolher quais delas deseja salvar. Questões geradas também podem ser editadas, caso necessitem de, por exemplo, detalhes adicionais ou correções gramaticais. Informações sobre quais questões foram salvas ou editadas são valiosas, uma vez que podem servir como feedback implícito para uso em melhorias dos modelos. Portanto, o sistema também armazena tais informações no seu banco de dados para acesso do seu administrador.

Para ilustrar o funcionamento da aplicação, podemos imaginar um caso em que é enviado para o modelo um texto que detalha o funcionamento do processo de fotossíntese. O modelo então geraria perguntas a partir do que está descrito no texto, como: "Qual é o papel do gás carbônico na fotossíntese?" e "Qual o nome do pigmento responsável por captar luz solar?".

Nas próximas seções, será apresentado em detalhe informações relevantes sobre o sistema e seu desenvolvimento. A Seção 2 discute a arquitetura do projeto, detalhando tecnologias usadas e decisões estruturais do back end e front end. Na Seção 3 está descrito os resultados de uma avaliação feita por usuários do sistema. Por último, a Seção 4 relata como ocorreu o processo de desenvolvimento, além de desafios encontrados e ideias para futuros aprimoramentos do sistema.

2 ARQUITETURA

A aplicação desenvolvida foi estruturada em duas camadas: front end e back end, como é tradicional no desenvolvimento de aplicações web [11]. O back end fica responsável por armazenar, acessar e manipular dados e o front end lida com a apresentação desses dados para o usuário e fornece uma interface que permite a utilização de todas as funcionalidades do sistema pelo usuário de forma acessível e intuitiva.

2.1 Back end

O back end da aplicação foi implementado utilizando a linguagem Java e o framework web Spring Boot [20], uma extensão da tecnologia Spring que adota uma série de configurações e dependências padrões para simplificar o processo de desenvolvimento.

O back end necessita ser capaz de armazenar vários dados, tais como informações de um usuário e suas questões geradas. Para tal função, foi utilizado o banco de dados SQL H2 Database [8] que mantém os dados do sistema em um arquivo. O armazenamento em arquivo também garante persistência dos dados, uma vez que é independente de qualquer execução individual do sistema.

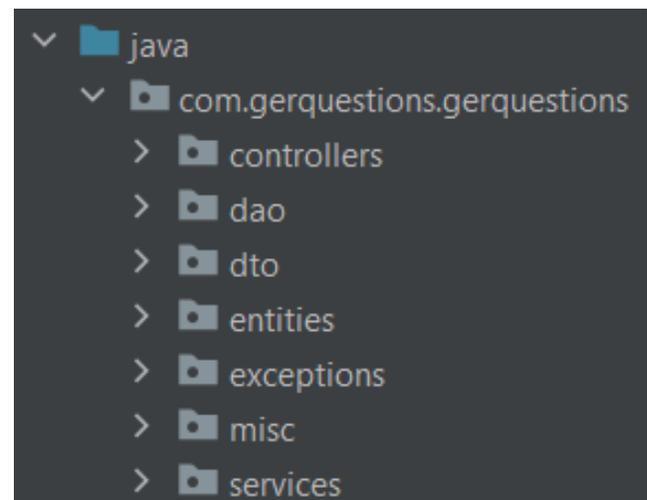


Figura 2: Estrutura do projeto

2.1.1 Estrutura.

A estrutura do back end foi dividida como ilustrado na Figura 2, com camadas DAO[6] e DTO[7] para armazenar e transportar dados e com implementação de métodos divididos entre camadas de controlador e de serviço[5], onde controladores são responsáveis por lidar com as requisições e serviços implementam a lógica dos métodos.

2.1.2 Entidades.

As entidades são as classes responsáveis por representar os objetos que persistem no sistema. Para desenvolver a aplicação como idealizada, foi necessária a utilização de 3 entidades:

- **Question:** Entidade modelada a partir das questões geradas pela aplicação. Cada questão possui um identificador único e o texto gerado pelo modelo de PLN utilizado. Caso a questão gerada não esteja de acordo com as necessidades do usuário, o sistema permite que sejam salvas somente as questões que o usuário desejar, além de possibilitar edição dos textos de qualquer questão.

A entidade também foi modelada tendo em vista o aprimoramento que modelos de PLN podem receber a partir de feedback implícito de usuários. O back end armazena todas as questões geradas, utilizando o atributo "saved" de tipo booleano para indicar se uma questão foi ou não salva

pele usuário. As questões com valor "false" no atributo não são exibidas para os usuários da aplicação, mas ainda estão disponíveis no banco de dados, permitindo uma análise sobre o que tende a levar usuários a salvar ou não uma questão, informação que pode ser bastante útil em melhorias do modelo usado.

Caso o usuário opte por editar uma das questões geradas, o atributo "editedText" é utilizado para exibir a versão editada do texto que será exibida para o usuário, enquanto o texto original é armazenado no atributo "text", de forma a facilitar uma análise sobre o que usuários costumam editar sobre os textos gerados.

```
public class Question {
    @Id
    @GeneratedValue(strategy= GenerationType.IDENTITY)
    private Long id;

    private String text;

    @Column(columnDefinition = "LONGTEXT")
    private String editedText;

    private Boolean saved;
}
```

Figura 3: Código da entidade Question

- *GeneratedQuestions*: Após enviar um texto para o modelo, várias questões são geradas. *GeneratedQuestions* é a entidade que guarda todas as questões geradas a partir de uma execução do modelo. Cada instância dessa entidade tem um identificador único, um atributo "originalText" com o texto utilizado para geração, as questões geradas no atributo "questionList" e o email do usuário que gerou as questões.

```
public class GeneratedQuestions {
    @Id @GeneratedValue(strategy= GenerationType.IDENTITY)
    private Long id;

    1 usage
    @ManyToOne
    @JoinTable(name = "generatedQuestions_questions", joinColumns = {
        @JoinColumn(name = "generatedQuestions_id"), inverseJoinColumns = {
            @JoinColumn(name = "question_id")})
    private List<Question> questionList = new ArrayList<>();

    @Column(columnDefinition = "LONGTEXT")
    private String original_text;

    1 usage
    @ManyToOne
    @JoinColumn(name = "email")
    private AppUser appUser;
}
```

Figura 4: Código da entidade GeneratedQuestions

- *AppUser*: Entidade modelada a partir dos usuários da aplicação. Todo *AppUser* possui um email, que serve como seu identificador único, além de nome, senha e uma lista com todas as entidades *GeneratedQuestions* associadas ao usuário, cada uma criada a partir de uma execução do modelo.

```
public class AppUser {
    @Id
    @NotBlank
    private String email;

    @NotBlank
    private String name;

    @NotBlank
    private String password;

    @OneToMany(mappedBy = "appUser",
        fetch = FetchType.EAGER,
        cascade = CascadeType.ALL,
        orphanRemoval = true)
    @JsonIgnore
    private List<GeneratedQuestions> generatedQuestions = new ArrayList<>();
}
```

Figura 5: Código da entidade AppUser

2.1.3 Login.

Para que a aplicação garanta que usuários tenham acesso aos seus dados, foi estabelecido um sistema de login no back end que gera um JSON Web Token (JWT) [13] utilizando a biblioteca Java JWT [9]. O token identifica o usuário que realizou login e o autoriza o uso de todas as funcionalidades relativas ao usuário. O token expira após 60 minutos, necessitando então de outro login para geração de um novo token.

2.2 Front end

O front end da aplicação foi desenvolvido utilizando a linguagem Javascript devido a abundância de recursos e bibliotecas disponíveis para desenvolvimento web na linguagem.

Para auxiliar o desenvolvimento, foi utilizado a biblioteca React [17], que facilita a criação de websites a partir de diversas funções como a utilização de JSX [12], uma extensão de sintaxe que permite o uso de HTML dentro de código Javascript.

O front end consiste de uma tela para cadastrar usuários, uma tela para login e uma tela principal, onde o usuário pode gerar questões e ter acesso a todas as questões geradas.

A aplicação utiliza os seguintes pacotes no funcionamento do seu front end:

- *Axios* [1]: Responsável por lidar com requisições HTTP para o back end da aplicação.
- *React Router* [18]: Controla o redirecionamento do usuário para diferentes telas a partir da URL acessada.
- *React-cookie* [4]: Possibilita o uso de cookies[25], arquivos que são armazenados localmente no computador do usuário com informações como o token da sessão do usuário[23].

Para lidar com tais dependências do front end, foi utilizado o gerenciador de pacotes npm [15].

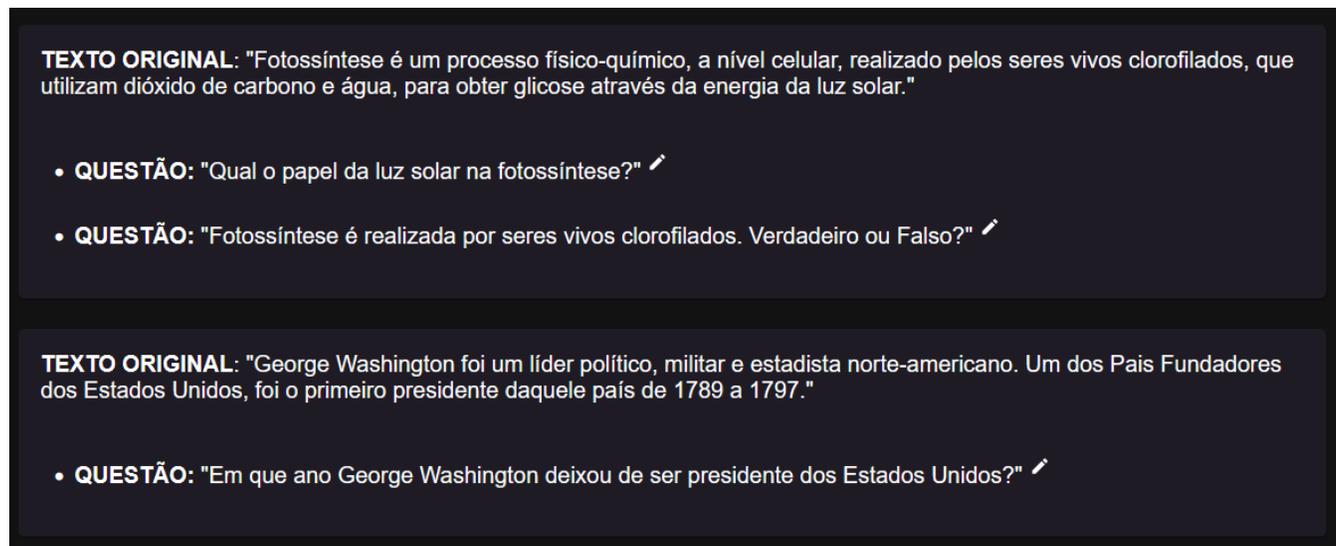


Figura 6: Ilustração de como questões geradas são exibidas para o usuário

3 AVALIAÇÃO

A aplicação foi disponibilizada para 3 estudantes da UFCG para avaliar questões de usabilidade e acessibilidade. Cada usuário recebeu instruções para realizar as seguintes 3 operações e avaliar o quão intuitivo foi o processo:

- Criar uma conta e logar no sistema.
- Gerar questões a partir de um texto.
- Editar questões geradas.

Todos os usuários completaram as operações com sucesso e relataram que o processo foi intuitivo. Após terminar o teste de cada usuário, foi solicitado feedback sobre possíveis melhorias ao projeto. Todos os usuários focaram as críticas no design das páginas, que foi considerado um pouco simples devido a falta de elementos gráficos mais elaborados, como um logo ou uma interface mais interativa.

A ideia geral de uma plataforma capaz de gerar questões a partir de um texto foi recebida com entusiasmo por todos os usuários, especificamente pelo seu potencial como uma ferramenta para auxiliar durante os estudos.

4 EXPERIÊNCIA

Nessa seção será relatado como foi feito o desenvolvimento da aplicação, além das lições aprendidas, desafios encontrados durante o processo e ideias para possíveis trabalhos futuros.

4.1 Desenvolvimento

Após a idealização do sistema, foram separadas suas principais funcionalidades e divididas em sprints de 1 semana inspiradas no método Scrum[19] de desenvolvimento de software. As sprints iniciais foram focadas no back end e o desenvolvimento do front end foi iniciado somente após a implementação das funcionalidades básicas do back end. As sprints foram concebidas de maneira flexível, com as 2 últimas do período letivo sendo reservadas para resolução

de problemas inesperados. Assim, o cronograma foi pensado de maneira a acomodar mudanças perante qualquer tipo de imprevisto, que de fato aconteceram e foram lidados com sucesso durante o processo.

4.2 Desafios

A maior parte dos desafios e obstáculos enfrentados durante o desenvolvimento foram relacionados a falta de experiência com certas tecnologias usadas. De forma geral, o desenvolvimento do front end como todo foi a parte que mais apresentou tal problema, uma vez que todo tempo gasto em desenvolvimento foi acompanhado de bastante tempo estudando materiais sobre as ferramentas utilizadas. Em retrospecto, começar o desenvolvimento do front end mais cedo, em paralelo com desenvolvimento de algumas das funcionalidades mais básicas do back end, poderia ter mitigado o problema, mas os buffers do cronograma foram suficientes para acomodar os atrasos encontrados.

4.3 Trabalhos futuros

Atualmente, o projeto não está integrado a nenhum modelo de geração de questões, logo, ele disponibiliza somente funcionalidades que se encaixam em qualquer modelo do tipo. Para continuar o desenvolvimento da aplicação, seria interessante adotar um modelo e implementar funcionalidades mais específicas dependendo do funcionamento do seu processo de geração de questões, por exemplo, fornecendo opções de criação de diferentes tipos de questões, caso o modelo permita tal escolha.

Após adotar um modelo específico, um possível caminho para continuar o desenvolvimento seria expandir o escopo da aplicação para uma plataforma onde haja interação entre usuários, permitindo que questões geradas sejam marcadas como públicas e acessadas por qualquer usuário. Essas questões poderiam então ser avaliadas e marcadas com tags que indicam informações úteis como o assunto

e tipo da questão. Dessa forma, novos usuários teriam acesso a diversas questões já geradas assim que ingressarem na plataforma.

É possível conceber diversos rumos para evolução do sistema, mas a escolha final do caminho a seguir deve ser feita somente após integração de um modelo e de coleta de feedback de usuários a partir do sistema em uso, pois somente assim serão encontrados os pontos fortes do sistema que devem ser realçados e pontos fracos que devem ser corrigidos.

AGRADECIMENTOS

Agradeço a minha família pelo apoio indispensável fornecido durante toda a minha vida. Agradeço aos professores e alunos de Ciência da Computação da UFCG pelo conhecimento compartilhado durante meus anos no curso. Agradeço aos meus amigos que tornaram momentos difíceis mais suportáveis. Finalmente, agradecimentos especiais para Professor Cláudio Campelo por sugerir a ideia desse trabalho e por atuar como meu orientador.

REFERÊNCIAS

- [1] axios [n. d.]. axios. Retrieved December 14, 2022 from <https://www.npmjs.com/package/axios>
- [2] bloomberg [n. d.]. \$422.37+ Billion Global Artificial Intelligence (AI) Market Size Likely to Grow at 39.4% CAGR During 2022-2028 | Industry. Retrieved December 14, 2022 from <https://www.bloomberg.com/press-releases/2022-06-27/-422-37-billion-global-artificial-intelligence-ai-market-size-likely-to-grow-at-39-4-cagr-during-2022-2028-industry>
- [3] chat [n. d.]. ChatGPT. Retrieved December 14, 2022 from <https://openai.com/blog/chatgpt/>
- [4] cookie [n. d.]. React-cookie. Retrieved December 14, 2022 from <https://www.npmjs.com/package/react-cookie>
- [5] cslayers 2022. Difference Between @Controller and @Service Annotation in Spring. Retrieved December 14, 2022 from <https://www.geeksforgoeks.org/difference-between-controller-and-service-annotation-in-spring/>
- [6] dao 2022. The DAO Pattern in Java. Retrieved December 14, 2022 from <https://www.baeldung.com/java-dao-pattern>
- [7] dto 2022. The DTO Pattern (Data Transfer Object). Retrieved December 14, 2022 from <https://www.baeldung.com/java-dto-pattern>
- [8] h2 [n. d.]. H2 Database Engine. Retrieved December 14, 2022 from <https://www.h2database.com/html/main.html>
- [9] jwt [n. d.]. JSON Web Token Support For The JVM. Retrieved December 14, 2022 from <https://mvnrepository.com/artifact/io.jsonwebtoken/jjwt>
- [10] job [n. d.]. A job application written by ChatGPT fooled recruiters and beat more than 80% of human candidates to an interview, report says. Retrieved January 30, 2023 from <https://www.businessinsider.com/chatgpt-written-application-got-shortlisted-for-interview-recruiters-2023-1>
- [11] Vikram Joshi. 2018. Seven Reasons Why A Website's Front-End And Back-End Should Be Kept Separate. Retrieved December 14, 2022 from <https://www.forbes.com/sites/forbestechcouncil/2018/07/19/seven-reasons-why-a-websites-front-end-and-back-end-should-be-kept-separate/?sh=319bb5924fca>
- [12] jsx [n. d.]. Introducing JSX. Retrieved December 14, 2022 from <https://reactjs.org/docs/introducing-jsx.html>
- [13] jwt [n. d.]. Introduction to JSON Web Tokens. Retrieved December 14, 2022 from <https://jwt.io/introduction>
- [14] law [n. d.]. ChatGPT passes exams from law and business schools. Retrieved January 30, 2023 from <https://edition.cnn.com/2023/01/26/tech/chatgpt-passes-exams/index.html>
- [15] npm [n. d.]. npm. Retrieved December 14, 2022 from <https://www.npmjs.com/>
- [16] pln [n. d.]. O que é processamento de linguagem natural? Retrieved December 14, 2022 from <https://cloud.google.com/learn/what-is-natural-language-processing>
- [17] react [n. d.]. React. Retrieved December 14, 2022 from <https://reactjs.org/>
- [18] router [n. d.]. React Router. Retrieved December 14, 2022 from <https://reactrouter.com/en/main>
- [19] scrum [n. d.]. What is Scrum? Retrieved December 14, 2022 from <https://www.scrum.org/resources/what-is-scrum>
- [20] spring [n. d.]. Spring Boot. Retrieved December 14, 2022 from <https://spring.io/projects/spring-boot>
- [21] startups [n. d.]. 5 Top Emerging Natural Language Generation Startups. Retrieved December 14, 2022 from <https://www.startup-insights.com/innovators-guide/5-top-emerging-natural-language-generation-startups/>
- [22] taxonomy [n. d.]. Bloom's Taxonomy. Retrieved January 30, 2023 from <https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/>
- [23] token [n. d.]. Session tokens overview. Retrieved December 14, 2022 from <https://shopify.dev/apps/auth/oauth/session-tokens>
- [24] transformers [n. d.]. Simple Transformers. Retrieved February 15, 2023 from <https://simpletransformers.ai/>
- [25] whatcookies [n. d.]. Cookies - Informações que os sites armazenam no seu computador. Retrieved December 14, 2022 from <https://support.mozilla.org/pt-BR/kb/cookies-informacoes-sites-armazenam-no-computador>