



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

EMANUELLY LARISSA FREITAS DE MELO

**FUBBY:
UM SISTEMA INFORMATIVO E INTERATIVO PARA O
PLANEJAMENTO ACADÊMICO DO ALUNO DE CIÊNCIA DA
COMPUTAÇÃO NA UFCG**

CAMPINA GRANDE - PB

2024

EMANUELLY LARISSA FREITAS DE MELO

FUBBY:

**UM SISTEMA INFORMATIVO E INTERATIVO PARA O
PLANEJAMENTO ACADÊMICO DO ALUNO DE CIÊNCIA DA
COMPUTAÇÃO NA UFCG**

**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 : Tiago Lima Massoni

CAMPINA GRANDE - PB

2024

EMANUELLY LARISSA FREITAS DE MELO

FUBBY:

**UM SISTEMA INFORMATIVO E INTERATIVO PARA O
PLANEJAMENTO ACADÊMICO DO ALUNO DE CIÊNCIA DA
COMPUTAÇÃO NA UFCG**

**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:

Tiago Lima Massoni

Orientador – UASC/CEEI/UFCG

Dalton Dario Serey Guerrero

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 15 de MAIO de 2024.

CAMPINA GRANDE - PB

RESUMO

Ao ingressar na universidade, muitos alunos desconhecem os múltiplos campos profissionais vinculados ao curso, as horas complementares que devem ser cumpridas e a relevância das disciplinas oferecidas. Atualmente, no Curso de Ciência da Computação da UFCG, mesmo que algumas dessas questões estejam dispostas em portais oficiais do curso e de ser possível alcançá-las por meio do diálogo, elas ainda não são apresentadas de forma simples e centralizada. Esse contexto é problemático por diversos motivos, dentre eles, a disparidade de conhecimento entre as pessoas, a facilidade de propagação de afirmações incorretas e o conseqüente mau planejamento curricular. A fim de contribuir com a diminuição desse cenário de déficit informacional, este trabalho propõe um sistema web que apresenta áreas profissionais de acordo com as disciplinas ofertadas no curso, que interage com o aluno para oferecer informações individuais sobre horas complementares e disciplinas optativas, bem como provê um ambiente colaborativo e saudável de comentários sobre as disciplinas. Os resultados do sistema em uso refletem uma elevada satisfação com a facilidade na execução das tarefas e com a proposta da aplicação, contudo, indicam que a comunicação com o usuário pode ser melhor estabelecida e que são desejadas mais funcionalidades no sistema.

FUBBY:
AN INFORMATIVE AND INTERACTIVE SYSTEM FOR
ACADEMIC PLANNING OF COMPUTER SCIENCE STUDENTS
AT UFCG

ABSTRACT

Upon entering university, many students are unaware of the multiple professional fields linked to their course, the complementary hours that must be completed, and the relevance of the subjects offered. Currently, in the Computer Science program at UFCG, even though some of these questions are available on the course's official portals and can be reached through dialogue, they are still not presented in a simple and centralized manner. This context is problematic for several reasons, among them the disparity in knowledge between individuals, the ease of spreading incorrect statements, and the consequent poor curriculum planning. To help mitigate this scenario of information deficit, this paper proposes a web system that presents professional areas according to the subjects offered in the course, interacts with students to provide individual information on complementary hours and elective subjects, and offers a collaborative and healthy environment for comments about the subjects. The system's results in use reflect high satisfaction with the ease of performing tasks and the application's intended purpose. However, they indicate that communication with the user can be better established and that additional features are desired in the system.

Fubby: um sistema informativo e interativo para o planejamento acadêmico do aluno de Ciência da Computação na UFCG

Emannuely Larissa Freitas de Melo
Universidade Federal de Campina Grande, Campina
Grande, Paraíba, Brasil

emannuely.melo@ccc.ufcg.edu.br

Tiago Lima Massoni
Universidade Federal de Campina Grande, Campina
Grande, Paraíba, Brasil

massoni@computacao.ufcg.edu.br

RESUMO

Ao ingressar na universidade, muitos alunos desconhecem os múltiplos campos profissionais vinculados ao curso, as horas complementares que devem ser cumpridas e a relevância das disciplinas oferecidas. Atualmente, no Curso de Ciência da Computação da UFCG, mesmo que algumas dessas questões estejam dispostas em portais oficiais do curso e de ser possível alcançá-las por meio do diálogo, elas ainda não são apresentadas de forma simples e centralizada. Esse contexto é problemático por diversos motivos, dentre eles, a disparidade de conhecimento entre as pessoas, a facilidade de propagação de afirmações incorretas e o consequente mau planejamento curricular. A fim de contribuir com a diminuição desse cenário de déficit informacional, este trabalho propõe um sistema web que apresenta áreas profissionais de acordo com as disciplinas ofertadas no curso, que interage com o aluno para oferecer informações individuais sobre horas complementares e disciplinas optativas, bem como provê um ambiente colaborativo e saudável de comentários sobre as disciplinas. Os resultados do sistema em uso refletem uma elevada satisfação com a facilidade na execução das tarefas e com a proposta da aplicação, contudo, indicam que a comunicação com o usuário pode ser melhor estabelecida e que são desejadas mais funcionalidades no sistema.

Palavras-chave

Planejamento acadêmico; processamento de linguagem natural; *machine learning*; *website*; usabilidade.

Repositório

<https://github.com/emannuelymelo/FubbyUFCG>

<https://github.com/emannuelymelo/NLP-toxic-detection>

1. INTRODUÇÃO

O surgimento da informática, como o próprio nome sinaliza, teve como um de seus principais objetivos permitir que as pessoas tenham acesso à informação de forma mais acessível e ágil, o que foi concretizado por meio dos dispositivos tecnológicos. Com isso, não apenas indivíduos passaram a buscar o que desejam entender fazendo uso da internet, como também, instituições puderam ter domínio sobre as informações geradas e captadas por elas por meio do processamento e análise de dados, de modo que, em ambos os cenários, o resultado das investigações realizadas

permitem que decisões sejam tomadas com fundamentação e, consequentemente, com maior segurança[17].

Um dos cursos de tecnologia que segue disseminando os benefícios citados é o de Ciência da Computação, o qual, em meio a uma realidade onde há uma busca crescente por automatizar e digitalizar processos de diferentes vertentes organizacionais, mostra-se cada vez mais relevante e necessário economicamente, o que acaba por atrair muitas pessoas para esse ramo de estudo[4]. Contudo, o contexto atual de muitas universidades que oferecem essa formação indica que existe uma deficiência de conhecimento entre os próprios integrantes delas, apesar da proposta inicial da informática ser levar a informação cada vez mais longe.

Diante da pertinência desse cenário, pesquisas recentes que o retratam foram realizadas. Exemplo disso é o estudo sistemático sobre evasão no Ensino Superior oferecido pela PUC por Teixeira, Mentges e Kampff (2019)[19], o qual concluiu que o pouco conhecimento sobre a graduação e as incertezas sobre as possibilidades de carreira são alguns dos principais motivos para evasão, sendo essa realidade mais presente em universidades públicas. Ademais, a Sociedade Brasileira de Computação (SBC) disponibilizou uma pesquisa, desempenhada por Fukao et al. (2023)[6], sobre evasão nos cursos de computação da Universidade Estadual de Maringá, na qual se observou que mais de 50% dos entrevistados evadiram por falta de informações sobre o que é abordado no curso e por falta de orientação acadêmica.

A partir de tal contexto de desinformação como um fator determinante para o descontentamento acadêmico, este trabalho tem como objetivo mitigar os reflexos dessa realidade entre os estudantes de Ciência da Computação da Universidade Federal de Campina Grande (UFCG). Para isso, foi realizado o desenvolvimento de uma aplicação que informa as áreas de atuação possíveis de acordo com as disciplinas ofertadas, que permite a interação com o aluno para processamento e oferecimento de informações sobre horas complementares, bem como sobre as disciplinas optativas do estudante, e, por fim, que disponibiliza um ambiente saudável de comentários sobre essas disciplinas, isto é, um local colaborativo com informações que ajudam nas escolhas acadêmicas do discente e que propõe minimizar a existência de comentários ofensivos, os quais em nada acrescentam na proposta de um ambiente construtivo. Assim, a implementação propõe garantir que conteúdos essenciais e confiáveis sejam acessados de forma simples, a fim de guiar com fundamentação a tomada de decisões e o planejamento dos estudantes ao longo da graduação.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, são descritos elementos técnicos importantes para o entendimento do processo de desenvolvimento do trabalho.

2.1 Análise de Sentimento

Com a evolução do Processamento de Linguagem Natural (PLN), vertente da inteligência artificial que permite pelos computadores a interpretação de texto e de palavras faladas de forma próxima aos humanos [22], foram criadas diferentes formas de lidar com a abrangente linguagem humana, dentre elas o reconhecimento de comandos de voz, a detecção de idiomas, a tradução de textos e a categorização de textos opinativos, também conhecida como análise de sentimento.

O objetivo da análise mencionada é extrair de forma automática a inclinação emocional de um texto. Para isso, utiliza-se processamento de linguagem natural a fim de identificar termos que indicam negatividade, neutralidade ou positividade em frases e, assim, classificar um texto subjetivo dentro de alguma classe, atribuindo a cada palavra uma "pontuação de sentimento" com base em dados anteriormente processados[15]. Uma aplicação da análise de sentimento está, por exemplo, em um texto que faz uso da palavra "lixo", a qual pode receber uma pontuação positiva ou neutra se os dados de base são pautados em conteúdos textuais sobre coleta de resíduos e preservação ambiental, ou pode receber uma pontuação extremamente negativa, considerando como viés uma coleção de comentários com discurso de ódio na internet.

Diante da aplicabilidade dessa forma de detecção, a análise de sentimento ganhou bastante destaque, principalmente, no contexto empresarial, dado que *insights* podem ser gerados a partir da opinião de clientes divulgada, por exemplo, em mídias sociais.

2.1.1 Naïve Bayes

Para efetuar a análise de sentimento, muitas vezes são utilizadas técnicas de *Machine Learning* (ML) e algoritmos de classificação, como é o caso do *Naïve Bayes*. Este é um método de categorização probabilístico simples, porém é considerado bastante eficiente em diversos problemas que buscam estimar a probabilidade de um evento baseado em exemplos prévios de sua ocorrência[5].

O algoritmo é fundamentado pela regra de Bayes com seleção de recursos de forma independente. A condicional envolvida no teorema é um evento de que X vai acontecer, dada a evidência Y. Assim, a regra permite determinar a probabilidade desse evento a partir da chance do contrário acontecer, junto à probabilidade individual dos elementos[21], como pode ser observado na fórmula:

$$P(X|Y) = \frac{P(X)P(Y|X)}{P(Y)} \quad (1)$$

Dessa forma, não há ligação entre uma palavra e outra, logo, o que acontece é uma suposição de independência. Assim, podemos estimar a probabilidade de uma palavra surgir em um contexto positivo ou negativo ao analisar textos previamente classificados dentre essas classes e calcular a frequência com que essa palavra aparece em cada classe, mecanismo muitas vezes comparado a adicionar palavras dentro de um saco e depois contar quantas vezes cada uma apareceu.

Um exemplo mais palpável pode ser observado ao considerar a frase "Gosto demais de morango" como positiva e "Tenho nojo de morango" como negativa. A partir disso, para analisar se uma nova situação como "Gosto de morango" está na classe positiva,

são consideradas a probabilidade de cada palavra aparecer em cada classe e a probabilidade do evento ser positivo ou negativo. Como as duas bases textuais possuem o mesmo tamanho de palavras e apenas a classe positiva possui a palavra "Gosto", a nova frase deve ser classificada como positiva, já que a probabilidade dessa palavra ser negativa é zero.

2.3 Base de Dados

No que diz respeito a algoritmos de *Machine Learning*, principalmente os que utilizam modelos de classificação como o *Naïve Bayes*, o qual computa probabilidades com base na ocorrência de exemplos, é fundamental que tenham a maior quantidade e qualidade de dados possíveis para treinar. Contudo, nem sempre uma fonte completa como essa é acessível. Para atenuar problemas como esse, novas informações podem ser inseridas de forma a gerar resultados promissores por meio de técnicas como *Data Merging* e *Data Augmentation*.

2.3.1 Data Merging

O chamado *Data Merging* se trata da junção de dois ou mais *datasets* em um só. Esse artifício é normalmente aplicado quando dados de diferentes origens possuem conteúdo valioso para ser analisado, sendo conveniente unir as duas amostras. Uma das abordagens mais comuns propõe juntar as linhas de variáveis de natureza semelhante[1], por exemplo, acrescentar linhas de uma tabela que apresenta as alturas de um conjunto de pessoas à coluna de altura presente na tabela original a ser melhorada.

2.3.2 Data Augmentation

Aumento de Dados, ou *Data Augmentation*, é uma técnica que visa expandir a quantidade de dados disponíveis e melhorar as respostas de um modelo[10]. Existem várias formas de colocar esse método em prática, algumas das mais famosas são: substituição por sinônimos e inserção aleatória. A primeira abordagem trata de aumentar as amostras de texto pela substituição de palavras por seus sinônimos, já a segunda propõe usar o sinônimo de uma palavra aleatória da frase e inseri-la em uma posição também aleatória[8].

Além de caminhos mais triviais como os mencionados, a pesquisa sobre aprimoramento no aumento de dados ainda está em evolução, o que abre espaço para a lógica e a criatividade auxiliarem nesse processo[2]. Exemplo disso está na pesquisa de Fadaee et al.(2017)[11] que mostrou uma forma de melhorar resultados a partir da troca de palavras comuns por palavras raras do seu *dataset*, criando melhores contextos sinteticamente.

3. SOLUÇÃO

Como forma de mitigar os problemas levantados relacionados ao ambiente universitário, foi criado o Fubby, um sistema web informativo e interativo sobre as disciplinas e o andamento acadêmico dos alunos do curso de Ciência da Computação da UFUCG.

Para alcançar esse objetivo, o site apresenta aos usuários diversas trilhas de atuação possíveis de acordo com as disciplinas ofertadas, disponibiliza para os estudantes do curso um ambiente de comentários pautado em um modelo de *machine learning* treinado para evitar conteúdos inapropriados sobre as disciplinas e, por fim, permite a interação com o aluno ao calcular e oferecer informações sobre horas complementares, bem como sobre as disciplinas optativas do estudante por meio do envio do documento de histórico.

Dessa forma, o trabalho busca disponibilizar informações indispensáveis para uma graduação bem planejada, incluindo orientações sobre a escolha da carreira, seleção de disciplinas de acordo com sua descrição e feedback de estudantes anteriores, além de detalhes que passam despercebidos em documentos oficiais de regulamentação, como o limite de horas semanais para atividades complementares aproveitáveis. Tudo isso é apresentado de forma acessível e centralizada, visando facilitar a tomada de decisões dos estudantes ao longo de sua jornada acadêmica.

3.1 Funcionalidades

As funcionalidades do sistema foram criadas com base nos estudos mencionados, os quais revelam o grande impacto que a falta de informação e orientação podem causar, no alinhamento de expectativas junto ao professor orientador e, também, foi levada em consideração opiniões de alunos do curso coletadas por meio de entrevistas.

3.1.1 Informativo de Trilhas em Computação

Qualquer visitante do site pode visualizar a descrição de diferentes opções de áreas que o curso de computação pode apresentar para seus alunos a partir das disciplinas que abordam assuntos relacionados a um determinado segmento, como pode ser observado nas Figuras 1 e 2. Um efeito colateral desejado dessa apresentação é que pessoas de fora da graduação passem a ter conhecimento sobre o contexto de atuação do curso e, assim, sintam-se motivadas a aderir a ele.

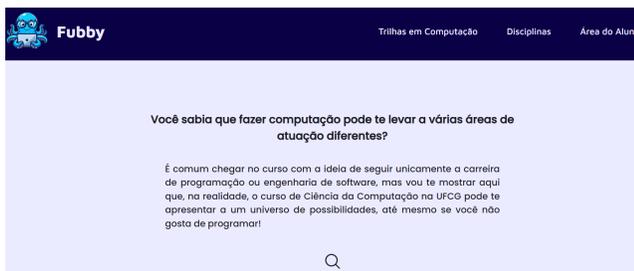


Figura 1 - Introdução à seção de Trilhas em Computação

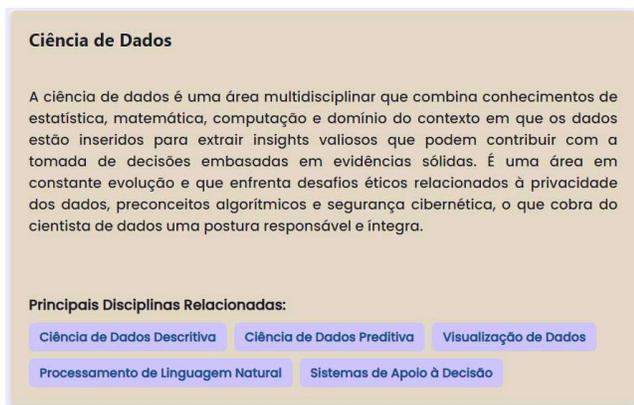


Figura 2 - Descrição de uma das áreas apresentadas

3.1.2 Visualização de Disciplinas

Essa função do sistema também está disponível para qualquer usuário e permite visualizar uma listagem de disciplinas e filtrá-las por nome, como pode ser visto na Figura 3. Ao clicar em uma das opções listadas, o indivíduo é direcionado a uma tela que possui detalhes sobre a disciplina, que é exibida na Figura 4.



Figura 3 - Listagem de Disciplinas

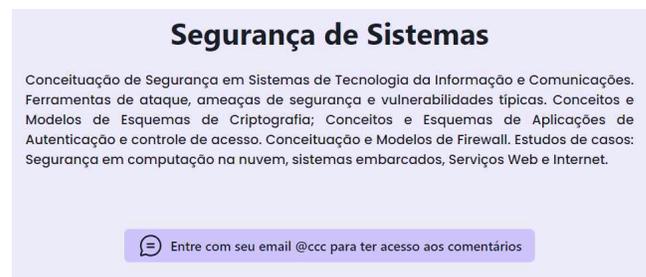


Figura 4 - Descrição da Disciplina

3.1.3 Criação de Comentários

Para ser identificado como um aluno apto a criar comentários, o usuário pode realizar login com o e-mail acadêmico do curso de Ciência da Computação da UFCG. O objetivo dessa funcionalidade é disponibilizar um canal amplo para os alunos comentarem como foi a experiência deles cursando disciplinas, atitude que muitas vezes é feita apenas por conversas com poucas pessoas ou de forma online dentro de grupos limitados. Exemplo do que se espera pode ser visto na Figura 5.



Figura 5 - Criação de comentário com sucesso

Diante da possibilidade de pessoas mal intencionadas tentarem adicionar conteúdos inapropriados, visto que a ação é feita de forma anônima, o comentário passa por um modelo treinado para classificar se o comentário é tóxico ou não, caso seja classificado dessa forma, é retornada para o autor uma mensagem de erro ao tentar efetuar a ação. No exemplo mostrado na Figura 6, ocorreu a tentativa de enviar a frase "O professor dessa disciplina é um lixo!" e a tentativa também foi feita com alfanuméricos como "11x0!", prática recorrente em redes sociais, mas o modelo conseguiu classificá-las como impróprias.



Figura 6 - Tentativa de criar comentário tóxico

3.1.4 Visualização de Comentários

Caso o usuário realize login com o e-mail acadêmico do curso de Ciência da Computação da UFCG, ele é identificado como um aluno apto tanto a criar, como também a visualizar comentários feitos por outras pessoas. Em cada disciplina uma lista de comentários é disposta, como exibe a Figura 5, a fim de que o aluno tenha o maior embasamento possível sobre a metodologia adotada e os assuntos abordados.

3.1.5 Área do Aluno

O usuário que é aluno tem uma parte do site dedicada a lidar com informações mais específicas de acordo com o seu contexto. Como é visto na Figura 7, ele pode escolher seguir dentre as duas abordagens que são retratadas: análise de optativas e horas complementares.



Figura 7 - Introdução à área do aluno

3.1.5.1 Análise de Optativas

Nesta funcionalidade, o aluno pode usar seu histórico em formato PDF, retirado do controle acadêmico, e submetê-lo no site para que seja analisado. A análise busca automatizar a contagem de optativas restantes do aluno, tarefa que até então é feita apenas por contagens a partir de resgate na memória, com auxílio do

fluxograma do curso ou com cálculos manuais. A Figura 7 mostra como o resultado é apresentado ao usuário.



Figura 8 - Análise de optativas via histórico

3.1.5.2 Horas Complementares

3.1.5.2.1 Informativo de Atividades

Esta seção do sistema tem o objetivo de evidenciar respostas para algumas questões que mais geram dúvidas em relação às atividades complementares entre os estudantes, essa informação é pautada em entrevistas feitas com alunos e com base na observação de e-mails prévios enviados pela coordenação na tentativa de alertar os discentes sobre o assunto. Parte das informações passadas pode ser observada na Figura 9.

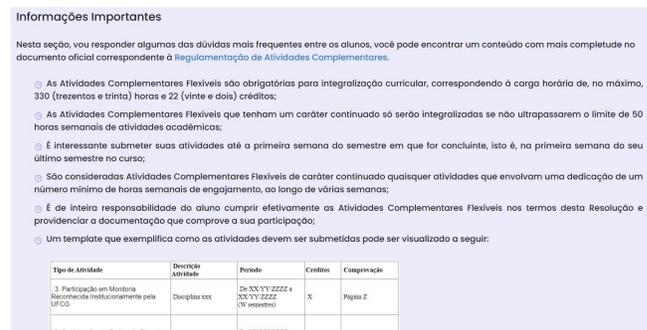


Figura 9 - Informações sobre atividades complementares

3.1.5.2.2 Cálculo de Créditos

A fim de auxiliar o aluno na contagem de créditos obtidos, tendo em vista que cada categoria de atividade possui um requisito diferente de aproveitamento, o site conta com a funcionalidade de calcular horas complementares a partir da categoria, dos meses de atuação e das horas semanais da atividade, dados que devem ser informados pelo usuário. A forma como a calculadora funciona pode ser vista na Figura 10.

Calculadora de Créditos

Já se atrapalhou ao tentar contabilizar suas horas? Posso te ajudar a calcular algumas das principais categorias de atividade complementar. O cálculo será feito de acordo com a [planilha oficial de critérios para aproveitamento](#).

Selecione a categoria de atividade:

2. Participação em Projeto de Pesquisa e Desenvolvimento Reconhecido Institucionalmente pela UFCG, incluindo atividade: ▾

Meses: 12

Horas Semanais: 20

Calcular

Você pode aproveitar até 8 créditos

Figura 10 - Calculadora de Créditos

3.2 Arquitetura

A solução foi estruturada com base na arquitetura cliente-servidor, em que o cliente (*front-end*) solicita serviços e aguarda respostas do servidor (*back-end*), sendo este responsável por oferecer serviços e ficar na espera por solicitações[3]. No contexto do trabalho, foi construído um cliente e dois servidores. Mais detalhes arquiteturais podem ser visualizados na Figura 11.

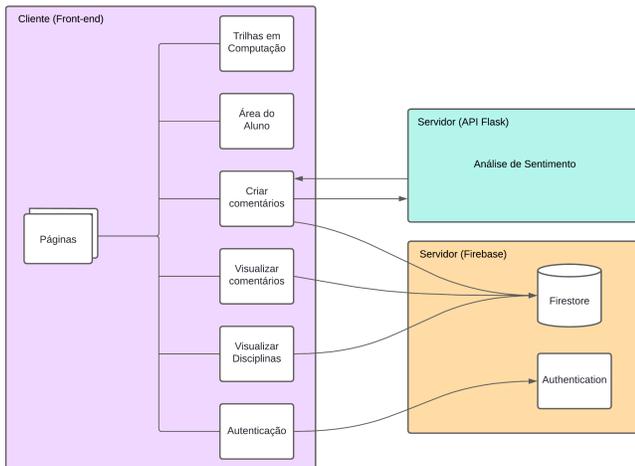


Figura 11 - Diagrama da arquitetura do sistema

3.2.1 Front-end

A parte visual do *site* foi desenvolvida utilizando *React*[29], biblioteca utilizada para gerar interfaces web interativas e que permite a criação de componentes reutilizáveis. Em conjunto dessa tecnologia, a linguagem *TypeScript*[32] foi selecionada para o desenvolvimento, uma vez que ela permite a tipagem dos componentes, o que é interessante para alcançar maior clareza do que é feito no código e, conseqüentemente, evitar faltas que reverberam como erros para o usuário. Para permitir um ambiente de desenvolvimento otimizado utilizando esses recursos, foi integrado o *Vite*[33], uma ferramenta que permite uma inicialização e, conseqüentemente, uma compilação mais rápida, pois o ambiente passa a ser consolidado em módulos que carregam conforme a demanda da página que está sendo acessada.

Além dessas ferramentas, foi incluída no processo o *Chakra UI*[23], uma biblioteca que oferece uma ampla variedade de componentes prontos para uso, como botões, *inputs* e tabelas. O que deixa essa tecnologia mais interessante é que ela é adaptável à solução, uma vez que proporciona diversas opções para

personalizar, permitindo uma flexibilização do estilo dos componentes de acordo com as necessidades do projeto.

A forma de comunicação com os servidores adotada foi a partir de requisições com protocolo HTTP, por meio da troca de dados formatados seguindo o padrão JSON[20].

Quanto à estruturação dos arquivos do cliente, foi decidido atribuir a cada finalidade distinta um diretório diferente, de modo a prezar por distribuir responsabilidades e contribuir para a reutilização de código para, assim, evitar *bad smells*[7].

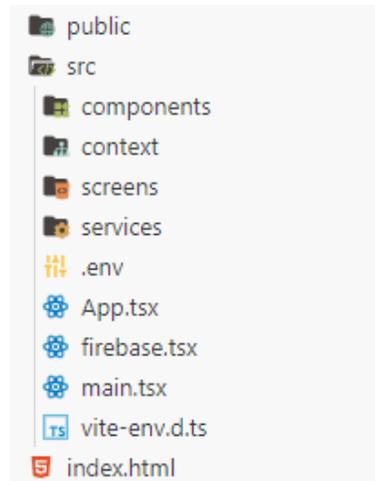


Figura 12 - Estrutura dos diretórios do cliente

3.2.2 Back-end

Como observado na Figura 11, o *back-end* da aplicação está dividido em um servidor utilizando *Flask*[25] e outro utilizando *Firebase*[24].

3.2.2.1 Servidor Flask

Nesse módulo está depositado o modelo de análise de sentimento treinado para avaliar comentários, previamente implementado por meio da plataforma do *Google Colaboratory*[12], a qual facilita o processo de criação por disponibilizar um ambiente de desenvolvimento na nuvem. Para o treinamento do classificador, foi utilizada a linguagem *Python*[28] em combinação, com as bibliotecas: *Natural Language Toolkit* (NLTK)[27] e *Scikit-learn*[31]. Essas tecnologias foram escolhidas devido ao NLTK disponibilizar ferramentas essenciais para o tratamento do texto para análise, como tokenização e remoção de palavras frequentes que não acrescentam valor à diferenciação de classes - *stop words*-, e o *Scikit-learn* oferecer o treinamento de modelos a partir de algoritmos de classificação, como o de regressão linear, de redes neurais profundas e o *Naïve Bayes*, o qual foi selecionado para ser utilizado devido ao entendimento de que um comentário tóxico, em grande parte das vezes, é composto por uma classe de palavras recorrentes, como palavras de baixo calão, que podem ser identificadas pelo modelo de Bayes, de modo já aprofundado inicialmente neste trabalho. Dessa forma, a junção das duas bibliotecas buscou alcançar um pré-processamento adequado para um treinamento eficiente. Além disso, foi utilizada a *Matplotlib*[26], com o objetivo de obter visualizações precisas sobre os dados analisados.

3.2.2.2 Servidor Firebase

O *Firestore* foi selecionado devido ao benefício de *Backend as a Service* (BaaS) que ele oferece, isto é, permite que o desenvolvedor integre a aplicação a um *back-end* com armazenamento em nuvem associado a outras funcionalidades como gerenciamento de arquivos, autenticação de usuários e integração com serviços de redes sociais[16]. A partir disso, foram utilizados os recursos de banco de dados, *Firestore*, e de autenticação, *Firebase Authentication*, a fim de obter atualização dos dados em tempo real e segurança das informações de autenticação de forma simplificada para que o desenvolvimento de outras partes mais complexas do sistema pudesse receber maior atenção.

4. AVALIAÇÃO

Neste tópico do trabalho, é feita a descrição de como foi realizado o acompanhamento do sistema em uso.

4.1 Metodologia

Para coleta de feedbacks, foi criado um formulário de respostas únicas por *e-mail*, por meio da plataforma *Google Forms*, e ele foi repassado para alunos do curso de Ciência da Computação da UFCG. As perguntas foram divididas em duas seções, uma foca na usabilidade e a outra na percepção pessoal da utilidade da ferramenta em relação às experiências que o aluno teve no curso.

As questões que envolvem usabilidade foram criadas com base no questionário PSSUQ (*Post-Study System Usability Questionnaire*)[9], o qual tem o objetivo de avaliar a usabilidade percebida pelo usuário logo após a experiência de uso, além de identificar os quesitos principais em que o sistema apresenta pontos fortes ou a melhorar. Originalmente, esse questionário é composto por 16 perguntas, contudo, para este trabalho foram selecionadas 10 delas, as quais atendem melhor às necessidades do contexto do projeto. A pontuação de cada elemento é feita por uma escala que varia de 1 a 7, em que 1 significa maior discordância com a pergunta e 7 forte concordância com o enunciado.

Enquanto isso, a segunda seção traz perguntas que buscam registrar se a proposta das funcionalidades do site tem relevância e aplicabilidade dentro do contexto acadêmico dos alunos.

4.2 Resultados

Como resultados¹ da análise da aplicação em uso, obtidos por meio de vinte respostas à pesquisa de usabilidade realizada, pode-se observar, pelas médias de pontuação na Tabela 1, que os usuários tiveram uma boa experiência na utilização da ferramenta.

Pergunta	Média
Estou satisfeito com a facilidade em usar o sistema	6,95
Eu poderia ser mais produtivo na conclusão de tarefas utilizando esse sistema	6,85
O sistema me mostra mensagens de erro que claramente explicam como resolver o problema	6,88

¹

<https://docs.google.com/spreadsheets/d/1T02sn3yTiHXg91XqHg4VeEyxRRqx3zfoBxdqBX-UJI4/edit?usp=sharing>

A informação fornecida pelo sistema (como mensagens de aviso na tela) é clara	6,85
Foi fácil encontrar a informação que eu precisava	7
A informação foi efetiva em ajudar a completar tarefas	6,9
A organização da informação na tela do sistema é clara	7
A interface do sistema é agradável	6,95
O sistema tem todas as funções e capacidades que eu esperava que teria	6,8
Estou satisfeito com o sistema	6,95

Tabela 1 - Resultados de Usabilidade baseados no PSSUQ

Além disso, com base nos valores captados, foi possível compreender que os pontos que mais pedem melhorias são os relacionados à mensagens na tela e à maior disponibilidade de funcionalidades.

Em relação aos resultados de percepção pessoal de acordo com as experiências vivenciadas no curso, todos os avaliadores concordaram que: o site pode contribuir positivamente na tomada de decisões ao longo da graduação; o site fornece informações sobre alguma dúvida que já tiveram ao longo da graduação; se depararam com um ambiente colaborativo e saudável na seção de disciplinas. Ademais, as questões evidenciaram que 40% dos usuários não sabem como chegar às informações sobre horas complementares no portal oficial do curso e que todos preferem acessar esses dados por meio do Fubby, o que pode ser analisado na Figura 13.

Você achou mais simples acessar as informações sobre atividades complementares por meio do site em questão ou pelo portal oficial do curso? [Copiar](#)

20 respostas



Figura 13 - Resultado de usabilidade da seção de horas complementares

Por fim, todos que avaliaram afirmaram que recomendariam o site para pessoas do curso ou para pessoas de dentro e fora do curso, o que vai ao encontro do efeito colateral esperado relacionado à apresentação dos campos profissionais da graduação. Esse resultado pode ser observado na Figura 14.

Você recomendaria a utilização do site para um amigo?
20 respostas



Figura 14 - Resultado sobre recomendação do site

5. EXPERIÊNCIA

Nesta seção, são abordados detalhes do desenvolvimento do sistema e os desafios mais relevantes enfrentados nesse processo.

5.1 Processo de Desenvolvimento

De início, foi realizada a elicitação de requisitos do sistema, que teve origem na observação de que existe um déficit de informações sobre o curso de Ciência da Computação da UFCG entre os próprios alunos, uma vez que temas, por exemplo, envolvendo atividades complementares, são recorrentemente comentados em redes sociais e normalmente se concentram nas mesmas dúvidas e nas mesmas respostas. Em seguida, para a elucidação dos requisitos, foram feitas: pesquisas bibliográficas que expuseram a pertinência do problema da falta de conhecimento em relação à graduação em outras universidades; entrevistas semiestruturadas e não estruturadas com discentes do curso; e alinhamento das funcionalidades mais relevantes para definição do escopo do trabalho junto ao professor orientador.

O planejamento das fases de desenvolvimento do sistema teve como base o modelo ágil *Scrum*[18], dado que contou com definição de *backlog* e de *sprints* com tarefas planejadas de modo coerente com o cronograma. A ferramenta que deu apoio à organização das etapas foi o *Notion*, que é personalizável e ajudou na criação de um *Kanban* com tarefas, no acompanhamento das datas limites e no registro dos desafios identificados.

Os primeiros passos da implementação foram voltados ao processamento de linguagem natural para a análise de sentimentos, em que os dados foram devidamente limpos e gerenciados para efetivação do treinamento do modelo que busca verificar se um comentário é saudável ou não. Após alcançar uma acurácia satisfatória de aproximadamente 99,2% com base nos dados utilizados, o código foi exportado e implantado na API *Flask* para atender às requisições de verificação de comentários. As Figuras 15 e 16 podem ser analisadas para extração de mais informações sobre as métricas coletadas.

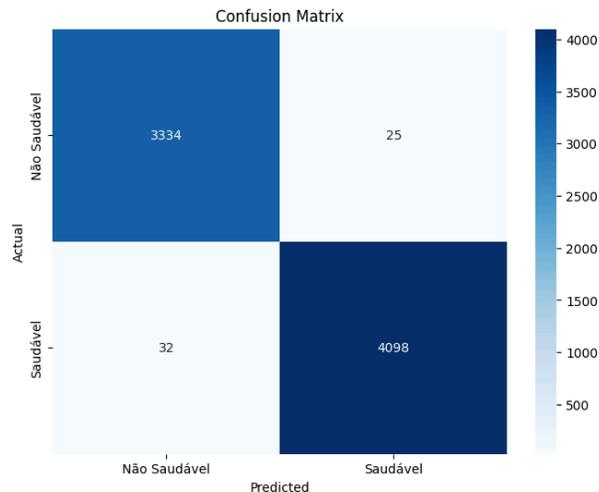


Figura 15 - Matriz de confusão gerada a partir do modelo

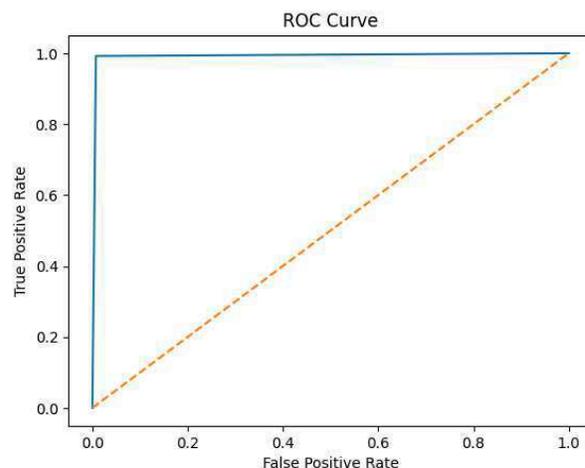


Figura 16 - Curva ROC gerada a partir do modelo

Com isso, de forma sucinta, a matriz de confusão[13] oferece uma visão clara da precisão do modelo de aprendizado de máquina a identificar reais positivos e negativos. Por outro lado, a curva ROC (*Receiver Operating Characteristic Curve*)[14] ilustra visualmente a capacidade do modelo de distinguir entre classes, mostrando a taxa de verdadeiros positivos em relação à taxa de falsos positivos em diferentes pontos de corte. No contexto do treinamento realizado, foi possível observar uma alta precisão e eficácia nos testes, com uma baixa taxa de falsos positivos e negativos, além de uma curva ROC bem posicionada, indicando uma excelente capacidade de discriminação entre classes. Diante desse cenário, foi possível dar continuidade às outras frentes de implementação.

A próxima etapa no desenvolvimento foi destinada ao *front-end*, com priorização inicial da criação de componentes reutilizáveis, como *Header*, *Footer* e caixa de comentários. Em seguida, foram implementadas as telas usando dados de simulação para a preparação da estrutura visual. Mais adiante, o site foi integrado aos servidores *back-end* para que os dados reais, que foram

processados e armazenados, pudessem ser acessados, assim como novos fossem criados pelo mecanismo cliente-servidor.

Por fim, foram realizados testes ponta a ponta no sistema, a fim de validar se o usuário vai estar diante de todos os comportamentos esperados.

5.2 Desafios

Ao longo da criação de um software, é comum o surgimento de falhas no código, incompatibilidades, problemas na configuração do sistema, entre outros obstáculos que podem impactar o fluxo de desenvolvimento. Nesta seção são descritos os principais desafios enfrentados até alcançar a versão atual da aplicação.

A questão mais desafiadora foi criar um modelo de aprendizagem de máquina eficiente na detecção de possíveis conteúdos inapropriados, isso se deu uma vez que não foi encontrada nenhuma base de dados pública em português com qualidade para que fosse realizado o treinamento, fator extremamente importante na classificação baseada no algoritmo de Bayes. O *dataset* utilizado² contém comentários diferenciados entre tóxicos e não tóxicos e os maiores problemas dele que pareciam impedir um resultado satisfatório envolviam: ser composto por frases muito curtas, possuir palavras de baixo calão até nos exemplos não tóxicos e não apresentar palavras que remetessem ao contexto acadêmico. Nas primeiras tentativas de treinamento, essas questões se mostraram prejudiciais para os textos considerados saudáveis, uma vez que acarretou em uma alta taxa de falsos negativos e falsos positivos, isto é, comentários saudáveis classificados como tóxicos e vice-versa.

Com o objetivo de solucionar as duas primeiras problemáticas, foi utilizada a técnica de *Data Merging*, fundindo à tabela original uma coluna de uma base de dados³ com avaliações positivas de *e-commerce* e, dessa forma, apontando como saudáveis esses novos comentários, os quais possuem mais palavras descritivas, que, inclusive, possuem adjetivos compatíveis com o de um comentário sobre uma disciplina, por exemplo "excelente", "ótimo", e frases como "gostei muito".

Para mitigar o problema da falta de palavras com contexto acadêmico, utilizou-se o princípio de *Data Augmentation* ao injetar no *dataset* comentários voltados às disciplinas do curso com e sem conteúdo impróprio e ao substituir as palavras de maior frequência possuindo contexto comercial, como "loja", por diversas palavras relacionadas ao ambiente acadêmico. Dessa forma, em conjunto com a limpeza dos dados, incluindo a remoção de *stop words* com adição à sua lista original palavras recorrentes da coluna nova que estão fora do contexto desejado, como "produto", "entrega" e "aparelho", foi possível focar na qualidade da descrição presente nos textos e, assim, alcançar resultados satisfatórios como os vistos nas Figuras 13 e 14.

Fora do contexto da análise de sentimentos, outro desafio importante foi encontrado na análise de optativas, pois, de início, a biblioteca PDF.js foi a escolhida para extrair e processar o texto do histórico dos alunos, dado que é uma das mais conhecidas para lidar com esse tipo de arquivo, contudo, ao tentar usar essa ferramenta, foram encontradas incompatibilidades no ambiente de desenvolvimento, bem como inconsistências nas chamadas

assíncronas, o que foi responsável por tomar mais tempo do que o previsto na implementação dessa funcionalidade. Para contornar esse problema, foi adotada a biblioteca *React-pdf*[30], que permitiu fazer toda a extração necessária de forma simplificada.

No que diz respeito à integração com os servidores, desafios bastante significativos foram o controle da sincronização dos dados recebidos de forma assíncrona e da ordem de execução dos *hooks*, que são funções do *React* que permitem gerenciar o estado, efeitos colaterais, entre outros recursos do ciclo de vida em componentes de função. Com isso, foi preciso fazer correções da lógica de chamadas para acessar o estado, assim como validar se o problema foi solucionado testando a disposição das informações no site após realizar comandos.

6. CONCLUSÃO E TRABALHOS FUTUROS

Apesar dos desafios apresentados na construção do sistema, foi possível executar com êxito o projeto planejado, com todas as funcionalidades previstas. Além disso, segundo os resultados observados na seção 4.2, o objetivo maior do trabalho, que foi criar uma ferramenta para levar informações essenciais de forma simplificada e interativa para os alunos do curso e, assim, guiar de forma fundamentada as ações deles na graduação, também foi alcançado. Ademais, com a majoritária boa experiência dos usuários utilizando o site, denota-se uma tendência de permanência da utilização da ferramenta, o que abre espaço para realizar melhorias na aplicação.

Diante disso, para trabalhos futuros, é importante que sejam feitas melhorias para suprir as expectativas na clareza de mensagens do sistema apontadas no teste de usabilidade, bem como para otimizar as funcionalidades já existentes. Exemplo de otimização que deve ser feita está no modelo de treinamento, pois, como foi observado dentro do algoritmo *Naïve Bayes*, quanto mais dados de qualidade, maior a chance de eficácia. Dessa forma, é interessante que seja aplicada novamente a técnica de aumento de dados com adição de mais sinônimos dentro do *database*, para que falsos positivos e falsos negativos sejam mais evitados. Além disso, para garantir que o ambiente de comentários seja ainda mais colaborativo e não se limite apenas na detecção automática para evitar conteúdo inapropriado, deve ser permitida a interação dos usuários com os comentários, de modo que um usuário possa apoiá-los por meio de votos e também denunciar alguma publicação, a fim de que rapidamente um comentário falso positivo receba uma solicitação de remoção.

Outro projeto futuro é processar as informações da ementa do curso e integrá-las ao site de modo inteligente. Para isso, será necessário criar um novo modelo de processamento do texto do arquivo para, assim, interpretar informações de todas as disciplinas do curso, como a área delas. A partir desse tipo de automatização, novas disciplinas criadas ou descartadas na ementa poderiam ser adaptadas ao sistema nas seções de "Trilhas em Computação" e de "Disciplinas" de forma mais manutenível.

Por fim, com base nos resultados coletados, também é válido que seja realizada uma nova pesquisa que busque entender quais outras ações os usuários esperam realizar pelo site, a fim de avaliar se novas propostas podem se encaixar ao objetivo do sistema e, intrinsecamente, contribuir no planejamento de mais pessoas da graduação.

²https://www.kaggle.com/datasets/gedorneto/comentarios-toxicos-ptbr?select=comentarios_toxicos_ptBR.csv

³<https://www.kaggle.com/datasets/fredericods/ptbr-sentiment-analysis-datasets>

7. AGRADECIMENTOS

Em primeiro lugar, agradeço à minha família, por todo suporte no decorrer de toda minha vida, em especial à minha mãe, Marbene, e à minha irmã, Emily, que desde muito cedo me mostraram a importância libertadora da educação e da responsabilidade. Ademais, agradeço aos meus amigos por terem me auxiliado sempre que precisei e por terem me proporcionado tantas risadas em meio ao cansaço do cotidiano acadêmico, não faria sentido viver sem as conexões que fazemos ao longo do caminho. Também, gostaria de expressar minha gratidão ao professor Francisco Brasileiro, coordenador durante maior parte da minha graduação, o qual foi essencial na organização da execução curricular dos alunos durante a pandemia iniciada em 2020 e que, com especial paciência, sempre buscou manter os alunos informados, atitude que inspirou a realização deste trabalho. Por fim, deixo meus agradecimentos finais ao professor Tiago Massoni, por ter aceitado me orientar neste trabalho e, mais uma vez, mostrar-se como uma pessoa comprometida em contribuir com a evolução dos alunos.

8. REFERÊNCIAS

- [1] A. De Arriba, M. Oriol and X. Franch, "Merging Datasets for Emotion Analysis," 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), Melbourne, Australia, 2021, pp. 227-231, doi: 10.1109/ASEW52652.2021.00051.
- [2] Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An Easier Data Augmentation Technique for Text Classification. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2748–2754, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- [3] Assunção, Daiane Souza. Arquitetura de Sistemas Cliente-Servidor: O que é e como funciona. Faspec, 2024. Disponível em: <https://blog.faspec.edu.br/arquitetura-de-sistemas-cliente-servidor/>. Acesso em: 09/04/2024.
- [4] Blogs PucPR, 2023. Por que é uma boa escolha estudar Ciência da Computação? Disponível em: <https://blogs.pucpr.br/escolapolitecnica/2023/05/11/por-que-e-uma-boa-escolha-estudar-ciencia-da-computacao/>. Acesso em: 6 mar. 2024.
- [5] C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno and J. Caro, "Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning," IISA 2013, Piraeus, Greece, 2013, pp. 1-6, doi: 10.1109/IISA.2013.6623713.
- [6] FUKAO, Amanda T.; COLANZI, Thelma E.; MARTIMIANO, Luciana A. F.; FELTRIM, Valéria D.. Estudo sobre Evasão nos Cursos de Computação da Universidade Estadual de Maringá. In: SIMPÓSIO BRASILEIRO DE EDUCAÇÃO EM COMPUTAÇÃO (EDUCOMP), 3. , 2023, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2023. p. 86-96. DOI: <https://doi.org/10.5753/educomp.2023.228209>.
- [7] Garcia, J., Popescu, D., Edwards, G., & Medvidovic, N. (2009). Identifying Architectural Bad Smells. 2009 13th European Conference on Software Maintenance and Reengineering. doi:10.1109/csmr.2009.59.
- [8] Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.
- [9] Lewis, James & R., James. (1995). IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. International Journal of Human-Computer Interaction. 7. 57-10.1080/10447319509526110.
- [10] Liu, P., Wang, X., Xiang, C., & Meng, W. (2020). A Survey of Text Data Augmentation. 2020 International Conference on Computer Communication and Network Security (CCNS). doi:10.1109/ccns50731.2020.00049
- [11] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data Augmentation for Low-Resource Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 567–573, Vancouver, Canada. Association for Computational Linguistics.
- [12] Mendes, Daniel. Explorando o Google Colab: Uma Janela para a Programação Colaborativa na Nuvem. Medium, 2024. Disponível em: <https://medium.com/@nielcostadev/explorando-o-google-colab-uma-janela-para-a-programacao-colaborativa-na-nuvem-3d1656c90502>. Acesso em: 11/04/2024.
- [13] Murel, Jacob; Kavlakoglu, Eda. What is a confusion matrix?. IBM, 2024. Disponível em: <https://www.ibm.com/topics/confusion-matrix>. Acesso em: 11/04/2024.
- [14] Nahm FS. Receiver operating characteristic curve: overview and practical use for clinicians. Korean J Anesthesiol. 2022 Feb;75(1):25-36. doi: 10.4097/kja.21209. Epub 2022 Jan 18. PMID: 35124947; PMCID: PMC8831439.
- [15] O que é análise de sentimentos? Amazon Web Services, AWS, 2024. Disponível em: <https://aws.amazon.com/pt/what-is/sentiment-analysis/>. Acesso em: 18/03/2024.
- [16] Rocha, Lucas. Backend as a Service (BaaS). Dio, 2023. Disponível em: <https://www.dio.me/articles/backend-as-a-service-baas>. Acesso em: 10/04/2024.
- [17] SAFFI, Fabiano Chiapinotto. Inteligência estratégica antecipativa : identificação de sinais fracos por meio do Big Data Analytics. Orientador: Janissek-Muniz, Raquel. 2020. 123 p. Dissertação (Programa de Pós-Graduação em Administração) - Escola de Administração, Universidade Federal do Rio Grande do Sul, [S. l.], 2020.
- [18] Scrum: O que é, como funciona e como começar. Atlassian. Disponível em: <https://www.atlassian.com/br/agile/scrum>. Acesso em: 10/04/2024.
- [19] TEIXEIRA, Rita Petrarca; MENTGES, Manuir José; CERVEIRA KAMPFF, Adriana Justin. EVASÃO NO ENSINO SUPERIOR: UM ESTUDO SISTEMÁTICO. Repositório Institucional PUCRS, [S. l.], p. 1-12, out. 2019.
- [20] Valença, Ranieri. A arquitetura Cliente-Servidor e o protocolo HTTP. DEV Community, 2023. Disponível em:

<https://dev.to/ranierivalenca/a-arquitetura-cliente-servidor-e-o-protocolo-http-21nd#json>. Acesso em: 09/04/2024.

- [21] WEIAND , Augusto; WEIAND , Fernanda Rodrigues Ribeiro. Análise de sentimentos do Twitter com Naïve Bayes e NLTK. ScientiaTec, [s. l.], v. 4, ed. 3, 24 abr. 2018.
- [22] What is sentiment analysis? IBM, 2024. Disponível em: <https://www.ibm.com/topics/sentiment-analysis>. Acesso em: 18/03/2024.
- [23] [n. d.]. Chakra UI Documentation. <https://chakra-ui.com/>. Acesso em: 09/04/2024.
- [24] [n. d.]. Firebase Documentation. <https://firebase.google.com/docs?hl=pt-br>. Acesso em: 10/04/2024.
- [25] [n. d.]. Flask Documentation. <https://flask.palletsprojects.com/en/3.0.x/>. Acesso em: 10/04/2024.
- [26] [n. d.]. Matplotlib Documentation. <https://matplotlib.org/3.8.3/index.html>. Acesso em: 10/04/2024.
- [27] [n. d.]. NLTK Documentation. <https://www.nltk.org>. Acesso em: 10/04/2024.
- [28] [n. d.]. Python Documentation. <https://docs.python.org/3/>. Acesso em: 10/04/2024.
- [29] [n. d.]. React Documentation. <https://react.dev>. Acesso em: 09/04/2024.
- [30] [n. d.]. React-PDF Documentation. <https://react-pdf.org>. Acesso em: 13/04/2024.
- [31] [n. d.]. Scikit-learn Documentation. <https://scikit-learn.org/0.21/documentation.html>. Acesso em: 10/04/2024.
- [32] [n. d.]. TypeScript Documentation. <https://www.typescriptlang.org/docs/>. Acesso em: 09/04/2024.
- [33] [n. d.]. Vite Documentation. <https://vitejs.dev>. Acesso em: 10/04/2024.