INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA DO AMAZONAS
CAMPUS MANAUS CENTRO
DEPARTAMENTO ACADÊMICO DE INFORMAÇÃO E COMUNICAÇÃO

VIVIANE DOS SANTOS AMORIM

CONGRADUATIONS: SISTEMA DE RECOMENDAÇÃO ACADÊMICA PARA ALUNOS DE TADS DO IFAM-CMC USANDO PROCESSAMENTO DE LINGUAGEM NATURAL

MANAUS - AM
2025

VIVIANE DOS SANTOS AMORIM

CONGRADUATIONS: SISTEMA DE RECOMENDAÇÃO ACADÊMICA PARA ALUNOS DE TADS DO IFAM-CMC USANDO PROCESSAMENTO DE LINGUAGEM NATURAL

Trabalho de Conclusão de Curso apresentado à banca examinadora Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciências e Tecnologia do Amazonas –IFAM Campus Manaus Centro, como requisito para cumprimento da disciplina TCC II – Projeto de Software.

Orientador: Prof. Dr. Renildo Viana Azevedo

MANAUS - AM
2025

Biblioteca do *Campus* Manaus Centro - IFAM

A524c Amorim, Viviane dos Santos.
Congraduations: sistema de recomendação acadêmica para alunos de TADS do IFAM-CMC usando Processamento de Linguagem Natural. / Viviane dos Santos Amorim. – Manaus, 2025.
73 p.: il. color.

Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas). – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Centro, 2025.
Orientador: Prof. Dr. Renildo Viana Azevedo.

1. Processamento de Linguagem Natural. 2. Sistema de Recomendação. 3. Engajamento Estudantil. I. Azevedo, Renildo Viana. (Orient.). II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.

CDD 005.3

Elaborada por Cybelle Taveira Bentes CRB 11/968

VIVIANE DOS SANTOS AMORIM

CONGRADUATIONS: SISTEMA DE RECOMENDAÇÃO ACADÊMICA PARA ALUNOS DE TADS DO IFAM-CMC USANDO PROCESSAMENTO DE LINGUAGEM NATURAL

Trabalho de Conclusão de Curso apresentado à banca examinadora Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciências e Tecnologia do Amazonas –IFAM Campus Manaus Centro, como requisito para cumprimento da disciplina TCC II – Projeto de Software.

Orientador: Prof. Dr. Renildo Viana Azevedo

Aprovado em _____ de _____ de 2025

BANCA EXAMINADORA

Prof. Dr. Renildo Viana Azevedo (Orientador)
Instituto Federal do Amazonas

Prof. Msc. Valclides Kid Fernandes dos Santos
Instituto Federal do Amazonas

Prof. Msc. Miguel Bonafé Barbosa
Instituto Federal do Amazonas

AGRADECIMENTOS

Gostaria de expressar minha imensa gratidão a minha mãe, Vera, a mais nobre e gentil mulher que sempre foi meu norte, minha sorte e esperança para continuar meu caminho e me tornar quem sou hoje. E ao meu pai, Hernandes, o homem mais forte que me ensinou que o amor reside, principalmente, nas fraquezas. Ressalto minha gratidão por todo cuidado, amor, educação e, principalmente, por terem se abdicado de tanto para que eu pudesse viver. Tudo o que eu tenho, o que sou e o que almejo ser será para sempre refletido em vocês.

Ao meu grande amor, Marcos, por ter sido essencial na minha vida. Por celebrar cada pequena conquista e apoiar nos grandes desafios, sem você, essa monografia sequer existiria. Agradeço sua habilidade de alegrar o dia, por fazer sempre eu me lembrar que a vida presta e, também, pela paciência, companheirismo, risadas e choros que nos guiaram até aqui. Não existem palavras suficientes que sejam capazes de expressar meu amor e admiração pela grande pessoa que você é.

Agradeço a Deus e toda e qualquer força divina onde fiz morada e me apoiei para que pudesse cumprir meus objetivos, mesmo rodeada de incertezas e instabilidade.

A esta instituição, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

A todos que direta ou indiretamente fizeram parte de minha formação, o meu muito obrigado.

RESUMO

Este trabalho aborda a importância da construção de uma trajetória acadêmica sólida durante a graduação, destacando seu papel no desenvolvimento profissional e pessoal de estudantes universitários. A pesquisa centra-se no contexto do curso de Tecnologia em Análise e Desenvolvimento de Sistemas (TADS) do Instituto Federal de Educação, Ciências e Tecnologia do Amazonas Campus Manaus Centro (IFAM-CMC), onde se identificou a dificuldade de acesso de informações organizadas sobre eventos, congressos e programas de pós-graduação como uma barreira significativa para o engajamento estudantil. O objetivo geral da pesquisa é desenvolver um sistema de recomendação, baseado em Processamento de Linguagem Natural (PLN), para incentivar a participação dos alunos em atividades acadêmicas. A metodologia empregada consiste na utilização de técnicas de PLN para analisar as descrições de oportunidades acadêmicas e, com isso, gerar recomendações personalizadas que se alinham aos interesses dos estudantes. Os resultados esperados incluem a criação de um protótipo funcional capaz de conectar os alunos a eventos e programas relevantes através de newsletters, facilitando sua formação e aprimoramento profissional. Conclui-se que a implementação de tal sistema pode mitigar a falta de informações claras, promovendo maior engajamento e qualificando a trajetória acadêmica dos estudantes do curso de TADS.

Palavras-chave: Processamento de Linguagem Natural; Sistema de Recomendação; Engajamento Estudantil.

ABSTRACT

This paper addresses the importance of building a solid academic track record during undergraduate studies, highlighting its role in the professional and personal development of university students. The research focuses on the context of the Technology in Systems Analysis and Development (TADS) course at the Federal Institute of Education, Science, and Technology of Amazonas Campus Manaus Centro (IFAM-CMC), where the difficulty of accessing organized information about events, conferences, and graduate programs was identified as a significant barrier to student engagement. The overall objective of the research is to develop a recommendation system based on Natural Language Processing (NLP) to encourage student participation in academic activities. The methodology employed consists of using NLP techniques to analyze descriptions of academic opportunities and thereby generate personalized recommendations that align with students' interests. The expected results include the creation of a functional prototype capable of connecting students to relevant events and programs through newsletters, facilitating their training and professional development. It is concluded that the implementation of such a system can mitigate the lack of clear information, promoting greater engagement and enhancing the academic trajectory of students in the TADS course.

Keywords: Natural Language Processing; System Recommendation; Student Engagement.

SUMÁRIO

INTRODUÇÃO.....9
PROBLEMATIZAÇÃO.....10
JUSTIFICATIVA.....10
OBJETIVOS..... 13
Objetivo geral.....13
Objetivos específicos.....13
METODOLOGIA.....13
Levantamento de requisitos.....14
Projeto de Sistema.....14
Implementação.....15
Validação de teste.....15
Manutenção..... 15
CAPÍTULO 1: A FORMAÇÃO ACADÊMICA AMPLIADA E SUAS DIMENSÕES FORMATIVAS.....16
1.1 A Formação Acadêmica Além da Grade Curricular.....16
1.2 Relevância do Ensino, Pesquisa e Extensão na Formação Acadêmica..... 19
CAPÍTULO 2: EXPLORANDO OS FUNDAMENTOS TÉCNICOS DO CONGRADUATIONS.. 22
2.1 Sistema de Recomendação.....22
2.1.1 Conceitos e fundamentos..... 22
2.1.1 Abordagens utilizadas.....24
2.2 Processamento De Linguagem Natural.....31
2.2.1 Fundamentos e aplicações..... 31
2.2.2 Pré-processamento de texto.....34
2.2.2.1 Tokenização..... 34
2.2.2.2 Stemming e Lemmatization.....35
2.2.2.3 Remoção de Stop Words e Normalização..... 36
2.2.3 Técnicas Aplicada ao Sistema.....37
2.2.3.1 Bag of Words (BoW)..... 37
2.2.3.2 Term Frequency-Inverse Document Frequency (TF-IDF)..... 37
2.2.3.3 Latent Dirichlet allocation (LDA).....38
2.2.3.4 Agrupamento de K-means ou Clusterização..... 39
2.2.3.5 Modelos Baseados em Embeddings.....40
2.2.3 Aplicações que utilizam processamento de linguagem natural..... 41
2.2.3.1 Aplicação de PLN na recomendação de conteúdo educacional.....41
2.2.3.2 Recommendation System for Students' Course Selection..... 43
2.2.3.3 Agent-Based Recommendation in E-Learning Environment Using Knowledge Discovery and Machine Learning Approaches.....44
CAPÍTULO 3: A CONSTRUÇÃO DO CONGRADUATIONS — DA IDEIA À IMPLEMENTAÇÃO..... 46
3.1 Concepção da Ideia.....46
3.2 Arquitetura e Funcionamento do Sistema.....47

3.3 Produção e Desenvolvimento..... 51
3.3.1 Design do sistema.....51
3.3.2 Desenvolvimento e implementação..... 55
3.3.2.1 Dados.....55
3.3.2.2 Backend..... 59
3.3.2.3 Frontend.....65
CONSIDERAÇÕES FINAIS..... 67
REFERÊNCIAS..... 70

INTRODUÇÃO

O presente trabalho tem como foco o desenvolvimento de uma solução tecnológica que possa conectar alunos do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas (TADS) do Instituto Federal do Amazonas - Campus Manaus Centro (IFAM-CMC) a eventos tech, congressos para publicação de artigos e programas de pós-graduação utilizando técnicas de Processamento de Linguagem Natural (PLN) e envio dessas informações em formato de newsletter. Essa iniciativa busca fomentar a participação ativa dos graduandos em atividades extracurriculares e oportunidades acadêmicas que vão além da sala de aula.

Desenvolver uma boa trajetória universitária traz inúmeros benefícios, como o fortalecimento da rede de contatos profissionais (networking), o aprimoramento de habilidades técnicas e interpessoais, contribui para o avanço do conhecimento e o aumento da visibilidade no meio acadêmico e no mercado de trabalho.

No entanto, a falta de incentivo ou de acesso a informações claras e organizadas sobre essas oportunidades dentro do instituto de ensino, por conta da estrutura e falta de recurso, pode limitar o desenvolvimento pleno dos estudantes do curso de TADS. Graduandos que não participam dessas experiências tendem a enfrentar desafios como a formação de um currículo pouco competitivo, a redução de perspectivas profissionais e uma menor integração com a comunidade acadêmica.

Por outro lado, diante da vasta quantidade de conteúdo disponível, as pessoas frequentemente se sentem perdidas e gastam muito tempo escolhendo algo entre as infinitas opções, conforme explicam Aamir e Bhusry (2015). Por isso, e por várias outras razões, os sistemas de recomendação são essenciais no mundo atual.

Esse cenário motivou a criação de um sistema de recomendação acadêmico chamado Congraduations que utiliza PLN para otimizar a comunicação entre os estudantes de TADS do IFAM-CMC e as oportunidades disponíveis na área da tecnologia, promovendo uma experiência acadêmica mais rica e significativa.

PROBLEMATIZAÇÃO

O ensino superior é amplamente valorizado pela sociedade, sendo considerado, por muitas pessoas, um caminho para alcançar uma vida mais digna e um futuro promissor. Nos dias atuais, houve uma crescente procura por cursos de tecnologia por ser um segmento de mercado com certa abundância de oportunidades empregatícias, além de oferecer boas remunerações. No entanto, a jornada acadêmica vai além de apenas ingressar em uma universidade e frequentar aulas; ela envolve uma série de desafios, especialmente no que se refere à identificação e aproveitamento de oportunidades que complementam a formação, como eventos e congressos de tecnologia e programas de pós-graduação.

No contexto do Instituto Federal do Amazonas - Campus Manaus Centro (IFAM-CMC), esses desafios tornam-se ainda mais evidentes. Apesar de sua relevância no cenário educacional da região, o campus enfrenta problemas estruturais e limitações de recursos que dificultam a divulgação de oportunidades acadêmicas e profissionais para os alunos. A ausência de um sistema centralizado e eficiente para comunicar essas oportunidades, resulta em uma lacuna significativa na formação integral dos estudantes da área de tecnologia. Além disso, há uma escassez de pessoas aptas e dedicadas exclusivamente à tarefa de disseminar essas informações de forma clara e acessível, agravando o problema.

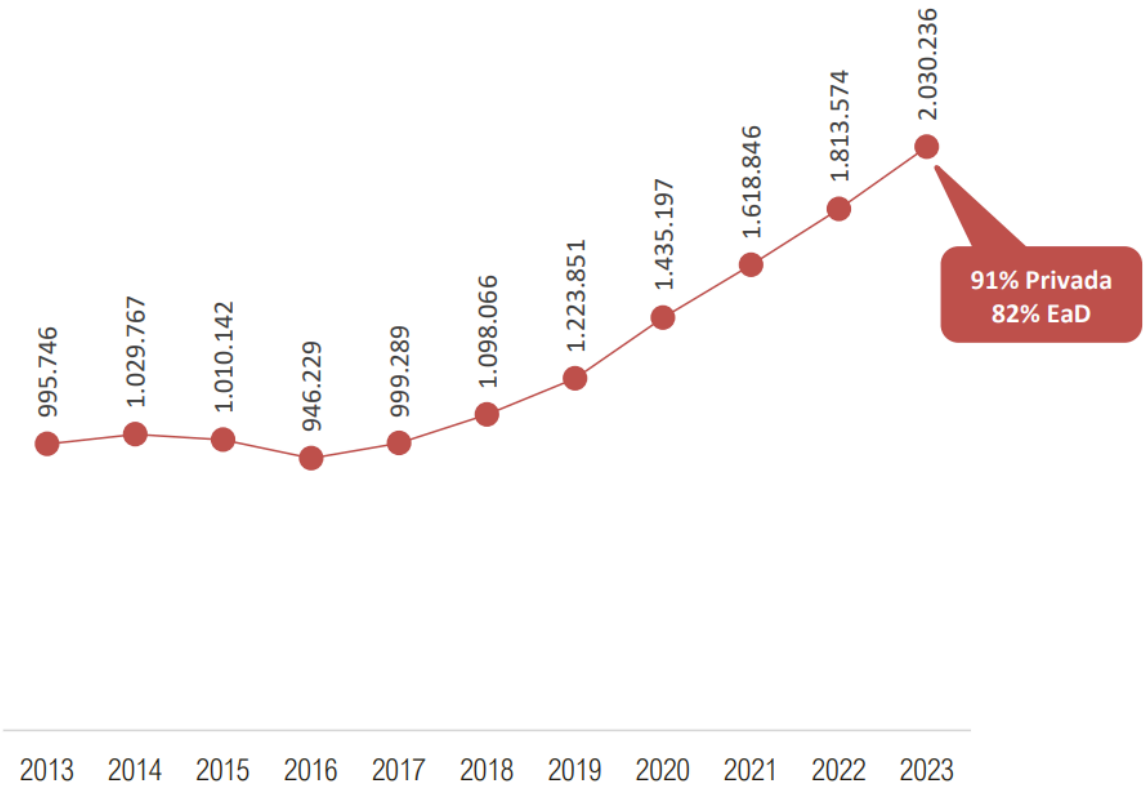
Esse cenário evidencia a necessidade urgente de uma solução que não apenas preencha esse espaço, mas também otimize a comunicação entre o campus e seus alunos, tornando o curso de TADS mais proveitoso e transformador dentro do IFAM-CMC.

JUSTIFICATIVA

O Ministério da Ciência e Tecnologia (BRASIL, 2001) destaca que a integração entre ciência, tecnologia e inovação é essencial para o progresso social e econômico do Brasil. Diversos estudos e relatórios apontam o crescimento nacional do setor tecnológico, o Ministério da Educação (MEC) e o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep) realizam, anualmente, o Censo da Educação Superior com o objetivo de fazer um levantamento de dados e

oferecer informações detalhadas sobre a situação e as tendências do setor, assim como guiar as políticas públicas de educação. No ano de 2023, o censo revelou a ascensão dos cursos de Tecnologia da Informação (TI), evidenciando que o número de ingressantes anuais mais do que dobrou em uma década, como mostra a **figura 1**. Embora a área de TI tenha registrado um crescimento expressivo na quantidade de matrículas nos últimos anos, isso pouco significa se muitos alunos desistirem do curso após os primeiros semestres. De acordo com o 13º Mapa do Ensino Superior no Brasil, o segmento de tecnologia apresenta, consistentemente, taxas de evasão mais altas do que a média das demais graduações ao longo da última década. Esse dado reforça a necessidade de ações e estratégias específicas para melhorar o engajamento e a permanência dos estudantes nesses cursos, garantindo que o crescimento no número de ingressantes seja traduzido em profissionais qualificados no mercado.

Figura 1 - Número de matrículas em cursos de graduação tecnológicos Brasil 2023



Fonte: MEC/Inep; Censo de Educação Superior 2023.

Além disso, a implementação de um sistema de recomendação acadêmica no Instituto Federal do Amazonas - Campus Manaus Centro justifica-se não apenas pelos benefícios diretos aos estudantes, mas também pela consonância com os objetivos delineados pela Lei de Diretrizes e Bases da Educação Nacional (LDB). De acordo com o artigo 43 da LDB (1996), a Educação Superior tem como finalidade estimular o desenvolvimento do espírito científico e do pensamento reflexivo, formar profissionais aptos a contribuir para o progresso da sociedade, incentivar o trabalho de pesquisa e investigação científica, e promover a extensão e a difusão de conhecimentos e benefícios culturais e tecnológicos.

Esses objetivos estão diretamente alinhados com o tripé do ensino superior — ensino, pesquisa e extensão — que, de forma indissociável, busca promover uma formação integral e de qualidade. No entanto, o IFAM-CMC enfrenta desafios estruturais e operacionais que dificultam a divulgação e o acesso a oportunidades acadêmicas e profissionais, como eventos, congressos e programas de pós-graduação. Essa lacuna prejudica o cumprimento pleno dos objetivos da LDB, limitando o potencial de desenvolvimento científico, reflexivo e cultural dos discentes. Ou seja, deve haver a integração e a utilização do tripé, com equilíbrio, pois todas as colunas que o sustentam tem a mesma importância, refletindo o papel da universidade ao longo da história (MOITA, 2009).

Ao conectar estudantes do curso de Análise e Desenvolvimento de Sistemas a atividades acadêmicas e extracurriculares, um sistema de recomendação contribuiria significativamente para fortalecer o ensino, a pesquisa e a extensão no campus. Além de estimular o desenvolvimento do pensamento crítico e a prática científica, essa ferramenta promoveria maior engajamento dos alunos com o ambiente acadêmico, ampliando a visibilidade do IFAM-CMC e reforçando seu compromisso com a excelência educacional e a formação de profissionais capacitados.

Portanto, a criação de um sistema dessa natureza não só atenderia às demandas práticas dos estudantes, mas também estaria em conformidade com os princípios legais e estruturais da Educação Superior, fortalecendo o papel do IFAM-CMC como agente transformador na região e no cenário nacional.

OBJETIVOS

Objetivo geral

Desenvolver um sistema de recomendação acadêmica que seja capaz de conectar os alunos do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal do Amazonas - Campus Manaus Centro a oportunidades extracurriculares como eventos, congressos e programas de pós-graduação na área da tecnologia.

Objetivos específicos

- Identificar e mapear as necessidades dos estudantes do curso de TADS do IFAM-CMC em relação ao acesso a informações sobre oportunidades acadêmicas extracurriculares.
- Projetar a arquitetura de um sistema de recomendação acadêmica utilizando técnicas de Processamento de Linguagem Natural (PLN).
- Desenvolver um protótipo funcional do sistema de recomendação acadêmica, integrando o envio de newsletter com o conteúdo filtrado das áreas tech.

METODOLOGIA

Neste tópico é apresentada a metodologia adotada para o desenvolvimento deste trabalho e da solução proposta. Para atingir os objetivos gerais e específicos do projeto, a metodologia foi estruturada em três etapas principais: Pesquisa, Desenvolvimento e Avaliação.

A etapa de pesquisa teve como finalidade compreender de forma aprofundada as técnicas, abordagens e desafios relacionados ao uso de Processamento de Linguagem Natural (PLN) aplicado a sistemas de recomendação. Para isso, foram realizados levantamentos bibliográficos em bases como Google Scholar, Scopus e IEEE Xplore, além de consultas a artigos científicos e estudos recentes que abordam sistemas de recomendação acadêmica, motores de busca semânticos e uso de embeddings em PLN. O método PICO (Population, Intervention, Comparison e Outcome) foi utilizado como estratégia de revisão

sistemática, garantindo embasamento sólido e evidências científicas para orientar as decisões de modelagem e implementação do sistema.

Na etapa de desenvolvimento foi utilizado o Modelo Cascata como abordagem para o desenvolvimento do sistema de recomendação acadêmica. O Modelo Cascata foi descrito inicialmente por Royce (1970) como uma abordagem sequencial para o desenvolvimento de software, onde cada fase é completada antes de se mover para a próxima. Este modelo promove uma abordagem estruturada e documentada para o desenvolvimento e apesar de suas limitações, a simplicidade do modelo cascata o torna uma escolha popular em projetos educacionais e de aprendizado (PRESSMAN, 2009). As etapas do modelo incluem levantamento de requisitos, análise, projeto, implementação, testes e manutenção.

Na etapa de avaliação, devido a restrições de tempo e de acesso à comunidade acadêmica, a validação não pôde ser realizada com um grupo amplo de alunos do IFAM-CMC. Dessa forma, a avaliação ocorreu em caráter exploratório e pessoal, com testes realizados diretamente pelo pesquisador. As recomendações foram encaminhadas por meio de uma newsletter enviada ao e-mail institucional, o que possibilitou verificar a viabilidade da solução e sua capacidade de gerar recomendações coerentes com o perfil acadêmico esperado. Essa limitação não compromete a relevância do trabalho, mas restringe suas conclusões quanto à experiência de uso e satisfação dos estudantes, que poderão ser exploradas em pesquisas futuras com aplicação em larga escala.

A metodologia da pesquisa deste trabalho foi classificada de acordo com diferentes dimensões. Quanto aos procedimentos, caracteriza-se como pesquisa bibliográfica e documental, pois utilizou artigos, dissertações e documentos acadêmicos para fundamentação teórica, e também como experimental, dado que houve a implementação prática e a realização de testes exploratórios com o sistema de recomendação. Quanto aos objetivos, trata-se de uma pesquisa exploratória, ao investigar o uso de PLN em sistemas de recomendação no contexto educacional, e descritiva, por relatar e avaliar o funcionamento do protótipo desenvolvido. Quanto à abordagem, utiliza métodos quantitativos e qualitativos combinados, ainda que em caráter restrito, pois envolve tanto a análise da coerência das recomendações geradas quanto observações qualitativas do pesquisador sobre o uso do sistema. Quanto à natureza, é uma pesquisa aplicada, visto que busca resolver um problema

prático — auxiliar alunos do curso de TADS na identificação de oportunidades acadêmicas relevantes — com potencial de impacto direto na comunidade acadêmica em etapas futuras.

CAPÍTULO 1: A FORMAÇÃO ACADÊMICA AMPLIADA E SUAS DIMENSÕES FORMATIVAS

Neste capítulo, será explorado a concepção de formação acadêmica ampliada, que transcende os limites da grade curricular tradicional para abarcar um conjunto de experiências e conhecimentos essenciais ao desenvolvimento integral do estudante. Será abordado como a integração indissociável de ensino, pesquisa e extensão é fundamental para qualificar essa formação, preparando os alunos não apenas para os desafios do mercado de trabalho, mas também para uma atuação crítica e proativa na sociedade. A fundamentação teórica aqui apresentada servirá como alicerce para a proposta de um sistema de recomendação acadêmica, desenhado especificamente para os estudantes do curso de Tecnologia em Análise e Desenvolvimento de Sistemas (TADS) do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas Campus Manaus Centro (IFAM-CMC), visando otimizar o acesso a eventos, congressos e oportunidades de pós-graduação.

1.1 A Formação Acadêmica Além da Grade Curricular

A formação acadêmica nas instituições de ensino superior no Brasil tem sido tradicionalmente centrada no cumprimento de grades curriculares rígidamente estruturadas. No entanto, esse modelo, embora necessário para garantir uma base teórica mínima, muitas vezes ignora a importância de experiências extracurriculares que complementam e ampliam o desenvolvimento do estudante, especialmente em cursos de tecnologia e áreas aplicadas, como é o caso da Tecnologia em Análise e Desenvolvimento de Sistemas (TADS).

As atividades extracurriculares, também denominadas complementares ou não obrigatórias, englobam a participação em monitorias, iniciação científica, projetos de extensão, grupos de estudo e pesquisa, envolvimento em órgãos de representação estudantil, congressos e eventos científicos, estágios remunerados ou não remunerados, entre outras (Bardagi & Hutz, 2012; Fior & Mercuri, 2009). Essas experiências desempenham um papel significativo na consolidação da identidade profissional dos estudantes e estão diretamente associadas ao seu nível de comprometimento com a formação superior.

Especificamente, os estágios são frequentemente percebidos pelos estudantes como uma preparação concreta para o mundo do trabalho, além de serem ferramentas para o desenvolvimento do pensamento criativo, das habilidades sociais e do estabelecimento de redes de contato (Bardagi & Hutz, 2012). Por outro lado, as atividades realizadas no próprio ambiente universitário, como projetos de pesquisa, extensão, grupos de estudos e monitoria, favorecem o contato próximo com professores e colegas, ampliam a compreensão do processo científico – da idealização à publicação – e preenchem lacunas curriculares ao proporcionar experiências que não se limitam ao conteúdo das disciplinas obrigatórias (Teixeira et al., 2008). Essas atividades não apenas proporcionam aos alunos a vivência prática do processo científico, como também aprofundam a compreensão sobre a relevância da tríade ensino, pesquisa e extensão no contexto da formação acadêmica. Essa integração é fundamental para aqueles que almejam a continuidade dos estudos em programas de pós-graduação, como mestrados e doutorados, além de representar um diferencial significativo na construção de um currículo profissional mais robusto e competitivo.

De acordo com o Plano Nacional de Educação (PNE 2014–2024), um dos objetivos centrais do ensino superior é “promover a formação integral dos estudantes, fortalecendo a articulação entre ensino, pesquisa e extensão” (BRASIL, 2014). Essa formação integral pressupõe que os alunos tenham acesso a oportunidades que extrapolem o conteúdo programático, como projetos de iniciação científica, participação em eventos científicos e técnicos, produção acadêmica e integração com o mundo do trabalho.

Contudo, em muitas instituições, essas atividades extracurriculares não são tratadas como parte estruturante do processo formativo, o que gera uma lacuna significativa entre a formação ofertada e as competências exigidas pelo mercado ou pela vida acadêmica pós-graduação. Segundo Cunha (2010), a universidade contemporânea enfrenta o desafio de superar a visão de que o ensino se limita à sala de aula, devendo proporcionar vivências que desenvolvam a autonomia intelectual, o espírito crítico e o interesse pela pesquisa. Nesse contexto, estudos como o de Oliveira e Fernandes (2016) também apontam que, embora haja valorização crescente da produtividade acadêmica, a inserção dos alunos em

atividades formativas mais amplas, como grupos de pesquisa e eventos, ainda é limitada por fatores institucionais e culturais.

No caso dos cursos tecnológicos, como os ofertados pelos Institutos Federais, essa situação é ainda mais sensível. Muitos alunos ingressam nesses cursos com o objetivo imediato de inserção no mercado de trabalho, o que não é, em si, um problema, mas conforme destaca Silva (2022), em muitos casos, há uma valorização insuficiente das trajetórias científicas e de pós-graduação, o que pode contribuir para o desestímulo à continuidade da formação acadêmica.

Estudos internacionais também enfatizam a importância do envolvimento em atividades extracurriculares para alavancar a empregabilidade dos jovens (Hu & Wolniak, 2010; Stevenson & Clegg, 2011). Isso porque a participação em atividades complementares demonstra que os estudantes foram proativos na construção de sua trajetória formativa, buscando experiências além daquelas obrigatoriamente ofertadas pela estrutura curricular. Tal postura é interpretada positivamente por empregadores, que passam a enxergar esses indivíduos como profissionais com maior autonomia, iniciativa e capacidade de adaptação — competências amplamente valorizadas em contextos organizacionais contemporâneos.

Stevenson e Clegg (2011) reforçam essa perspectiva ao analisar que, no contexto do ensino superior, o desenvolvimento da "identidade profissional" está cada vez mais ligado à capacidade do aluno de mobilizar recursos externos à sala de aula. Isso inclui, por exemplo, participar de eventos, projetos de pesquisa, redes de inovação e interações com múltiplos agentes sociais. A partir disso, a formação deixa de ser apenas um acúmulo de conteúdos técnicos e passa a ser reconhecida como uma trajetória de experiências vividas, escolhas realizadas e redes construídas — todos elementos fundamentais para a construção de capital social e empregabilidade.

No contexto brasileiro, embora essas evidências sejam igualmente válidas, observa-se uma lacuna entre a teoria e a prática institucional. Em muitos cursos, há poucos estímulos institucionais para que os alunos ampliem sua vivência acadêmica além da sala de aula, o que acaba por restringir o potencial de desenvolvimento pessoal e profissional desses estudantes. Tal descompasso entre o perfil do egresso esperado pelo mercado e a formação efetivamente proporcionada pelas instituições pode comprometer a inserção dos alunos em nichos mais competitivos, como o de

tecnologia e inovação, onde a capacidade de aprender continuamente e interagir com redes de conhecimento é crucial, além de desestimular o estudante a dar continuidade a vida acadêmica.

1.2 Relevância do Ensino, Pesquisa e Extensão na Formação Acadêmica

A educação superior desempenha um papel crucial na formação de profissionais capacitados para enfrentar os desafios contemporâneos e contribuir para o desenvolvimento da sociedade. Nesse contexto, a articulação entre ensino, pesquisa e extensão é fundamental para promover uma formação integral, que alia conhecimentos teóricos, prática profissional e compromisso social. De acordo com Severino (2012), essa integração potencializa a construção de saberes significativos, formando indivíduos mais preparados para enfrentar os desafios complexos de um mundo globalizado e em constante transformação.

O ensino representa o eixo central da educação universitária, pois oferece a base teórica necessária para o desenvolvimento do pensamento crítico e da capacidade analítica. Por outro lado, a pesquisa permite que os estudantes se aprofundem em questões específicas, desenvolvendo soluções inovadoras e contribuindo para a expansão do conhecimento em diversas áreas. A extensão, por sua vez, amplia o impacto da universidade ao aproximar-se da sociedade, promovendo a aplicação prática do conhecimento e contribuindo para o desenvolvimento comunitário.

A integração da tríade ensino-pesquisa-extensão permite que os estudantes desenvolvam diversas competências, como pensamento crítico, resolução de problemas, trabalho em equipe e liderança. Além disso, possibilita que os conhecimentos teóricos adquiridos em sala de aula sejam complementados e aplicados de maneira prática, por meio de projetos e atividades que atendem às múltiplas demandas da comunidade. A exposição a desafios reais durante o processo de formação prepara os estudantes para o mercado de trabalho e para uma atuação profissional responsável e inovadora. Por exemplo, a participação em projetos de pesquisa pode proporcionar uma compreensão mais profunda sobre

questões técnicas, enquanto as atividades de extensão promovem o contato direto com as demandas sociais.

As experiências práticas constituem um elemento indispensável para a formação profissional, pois permitem que os estudantes apliquem na prática os conhecimentos adquiridos em sala de aula. Segundo Kolb (1984), a aprendizagem experiencial é um processo dinâmico em que o conhecimento é criado por meio da transformação da experiência. No âmbito universitário, essas atividades ajudam os estudantes a desenvolver habilidades específicas, como:

- A capacidade de resolver problemas práticos em situações reais;
- Habilidades interpessoais e de comunicação;
- Capacidade de trabalhar sob pressão e em equipes multidisciplinares.

Essas experiências também desempenham um papel importante na transição dos estudantes para o mercado de trabalho, ajudando-os a compreender as demandas e expectativas de suas futuras profissões. Além disso, essas práticas fornecem uma oportunidade para que os estudantes construam suas redes de contatos profissionais e identifiquem áreas de interesse para sua carreira.

Ademais, a participação em eventos científicos e acadêmicos, como congressos, simpósios e feiras de tecnologia, é uma dimensão relevante da formação universitária. Segundo Santos e Almeida (2018), os eventos acadêmicos também desempenham um papel importante na motivação dos estudantes para continuar seus estudos em programas de pós-graduação. Além disso, esses eventos promovem:

- A troca de ideias e conhecimentos entre participantes de diferentes instituições;
- A divulgação de projetos de pesquisa e extensão desenvolvidos na universidade;
- A integração entre estudantes, professores e profissionais do mercado.

A exposição a eventos desse tipo é particularmente importante para estudantes que desejam ingressar no mundo acadêmico ou em carreiras que demandam alto nível de especialização.

Entretanto, apesar dos inúmeros benefícios que a integração entre ensino, pesquisa e extensão pode proporcionar, existem diversos desafios que ainda precisam ser superados. Um dos principais obstáculos é a dificuldade de articulação entre os diferentes setores acadêmicos que, muitas vezes, operam de maneira isolada, sem a comunicação necessária para fomentar a colaboração. Além disso, observa-se uma falta de incentivo para a realização de projetos, que são fundamentais para a construção de um conhecimento mais amplo e abrangente. Contudo, as perspectivas para o futuro são promissoras, uma vez que essa integração tão desejada contribui de maneira significativa para a formação de profissionais mais completos e qualificados, que estejam aptos a atuar de forma ética e responsável na sociedade contemporânea, abordando questões complexas e multidisciplinares.

De acordo com Vasconcelos (1996), a Universidade é um espaço que prioriza, acima de tudo, a transmissão do conhecimento já estabelecido, além de ser um centro criador de novos saberes e uma instituição investigadora que estimula a curiosidade, a ousadia e a iniciativa. Inserida em um contexto histórico, político e social, a Universidade deve atuar e intervir nesse cenário. Nesse sentido, precisamos construir uma Universidade que realmente cumpra seu papel, indo além da formação de profissionais em diversas áreas, que compreendam e atuem na realidade em que estão inseridos. Segundo Menezes (2001), a Universidade não deve ser vista apenas como uma instituição de ensino superior, pois possui um sentido mais amplo. A educação superior está ligada à investigação científica e ao desenvolvimento cultural e científico, focados nos problemas nacionais ou regionais.

CAPÍTULO 2: EXPLORANDO OS FUNDAMENTOS TÉCNICOS DO CONGRADUATIONS

Este capítulo tem como objetivo estabelecer o referencial técnico que fundamenta o desenvolvimento da aplicação Congraduations. Para tanto, serão detalhados os conceitos e as tecnologias essenciais que permitem compreender a arquitetura e o funcionamento do sistema proposto. A jornada técnica inicia-se com uma imersão nos Sistemas de Recomendação, desvendando seu conceito, as principais abordagens de funcionamento e, de forma específica, sua aplicação em contextos educacionais, que é o cenário central do projeto.

Em sequência, o capítulo se aprofundará no Processamento de Linguagem Natural (PLN). Serão apresentados seus fundamentos, as vastas aplicações que permeiam a interação humano-computador, as etapas cruciais de pré-processamento de texto para preparar os dados para análise, e as técnicas específicas de PLN que serão empregadas na construção do sistema.

Por fim, será realizada uma análise de trabalhos relacionados na literatura, que servirá para contextualizar a proposta do Congraduations dentro do panorama de pesquisas existentes, identificando lacunas e contribuindo para a originalidade e relevância do projeto. Este arcabouço técnico é vital para a compreensão das escolhas metodológicas e tecnológicas que guiaram a concepção e a implementação da solução.

2.1 Sistema de Recomendação

Neste capítulo serão apresentados fundamentos teóricos e tecnológicos sobre sistemas de recomendação, explorando conceitos, abordagens e aplicações.

2.1.1 *Conceitos e fundamentos*

Um sistema de recomendação (SR) é um tipo de sistema de aprendizado de máquina que fornece recomendações personalizadas aos usuários com base em seus comportamentos, preferências e padrões anteriores. Os sistemas de recomendação surgiram como uma solução computacional para lidar com a crescente quantidade de informações disponíveis em diversos contextos na internet, ganhando força nos anos 1990. As sugestões fornecidas visam auxiliar os usuários

em vários processos de tomada de decisão e, de acordo com Ricci et al. (2011), esses sistemas têm como objetivo principal filtrar informações relevantes e personalizadas, proporcionando uma experiência de usuário mais eficiente e direcionada. Isso é alcançado por meio da análise de dados históricos, comportamentos e interações dos usuários. Esses sistemas desempenham um papel fundamental em diversas plataformas, como serviços de streaming, e-commerces e redes sociais, facilitando a descoberta de conteúdos, produtos e serviços relevantes.

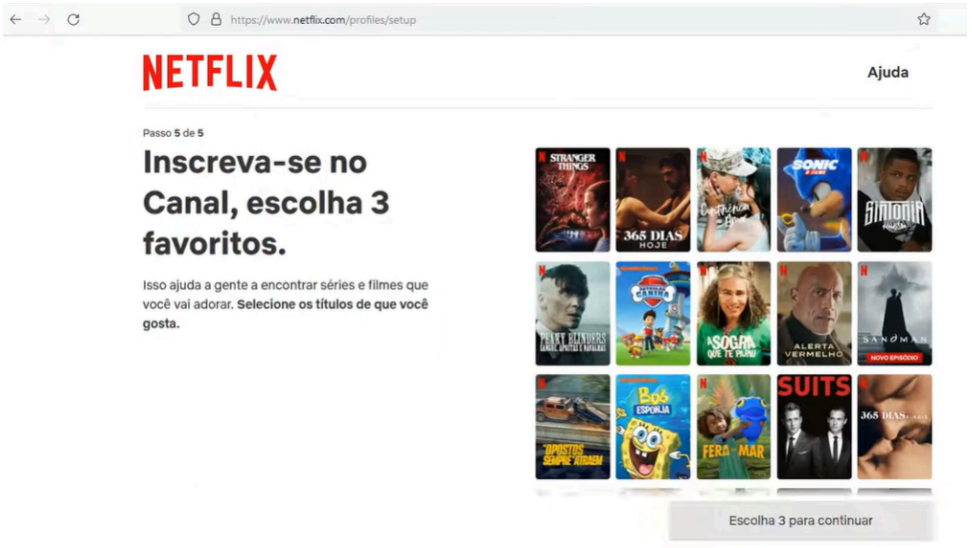
Para o funcionamento correto de qualquer SR, a coleta e o processamento de dados são fundamentais. Sem uma base de dados robusta, o sistema não teria informações para aprender e gerar sugestões relevantes para os usuários. A obtenção desses dados pode ocorrer de duas formas principais: explícita e implícita (Paz e Cazella, 2018). A coleta explícita de dados envolve a obtenção direta de informações fornecidas pelos usuários. Isso pode incluir:

- **Avaliação e classificações:** quando os usuários atribuem notas ou estrelas a itens.
- **Comentários e *feedbacks*:** opiniões textuais sobre os itens.
- **Curtidas e *descurtidas*:** indicações binárias de preferência.
- **Priorização de relevância:** quando o usuário organiza itens com base em seu gosto pessoal.

Um exemplo prático e cotidiano de coleta explícita pode ser encontrado no processo de criação de contas em plataformas de streaming de filmes e séries. Ao se cadastrar, o usuário é frequentemente solicitado a indicar seus gêneros, filmes, ou séries favoritos. Essa etapa, que pode parecer simples, é crucial: ela fornece ao sistema as informações iniciais necessárias para que um algoritmo de recomendação personalizado comece a ser construído.

Com base nos dados fornecidos explicitamente, a plataforma consegue traçar um perfil de interesse inicial do usuário. Por exemplo, se ele indica preferência por ficção científica e dramas históricos, o algoritmo de recomendação pode imediatamente sugerir conteúdos dentro desses gêneros, como ilustrado na Figura 2.

Figura 2 - Tela de configuração inicial da Netflix solicitando a escolha de títulos favoritos para personalização de recomendações.



Fonte: Netflix (2025)

Na modalidade implícita, os dados são coletados a partir de comportamentos do usuário, sem que ele precise fornecer explicitamente a informação. Essa abordagem inclui o rastreamento do histórico de navegação, observando as páginas visitadas; o histórico de compras, analisando o produto adquirido e a frequência dessas aquisições; os cliques e visualizações; e as pesquisas realizadas, identificando os termos que foram buscados.

Além dessas interações diretas com os usuários, os sistemas de recomendação podem ser enriquecidos utilizando outras categorias de dados. Dentre elas, está presente os dados demográficos (como idade, sexo, endereço e etc) e os dados psicográficos (relacionados a interesses e estilos de vida). Adicionalmente, são empregados dados de características dos itens, que descrevem os próprios produtos ou conteúdos, como gênero de um filme, autor de um livro ou tipo de conteúdo. A combinação dessas diversas fontes de dados permite que os sistemas construam perfis mais completos e, por consequência, ofereçam recomendações mais precisas e relevantes.

2.1.1 Abordagens utilizadas

O sistema de recomendação pode ser classificado de acordo com o tipo de informação usada para prever as preferências do usuário. As abordagens para desenvolver sistemas de recomendação podem ser classificadas em três categorias principais: filtragem colaborativa, sistemas baseados em conteúdo e abordagens híbridas. Cada uma possui características e métodos distintos para gerar recomendações.

A filtragem colaborativa (CF - Collaborative Filtering) é uma das abordagens mais populares e amplamente utilizadas. Sua premissa fundamental reside na ideia de que “pessoas que concordaram no passado tendem a concordar no futuro" (RESNICK; VARIAN, 1997, p. 57), ou seja, se baseia na análise de interações passadas de usuários com itens para identificar padrões de comportamento. Essa abordagem não exige conhecimento explícito sobre o conteúdo dos itens ou atributos demográficos dos usuários, baseando-se puramente nas interações entre usuários e itens. Os filtros colaborativos constroem uma base de dados de preferência de itens para usuários e, dessa forma, o usuário é comparado a base para descobrir seus vizinhos, os quais são pessoas que compartilham preferências similares. Os itens de interesse para esses usuários vizinhos são então recomendados ao usuário inicial.

A implementação da Filtragem Colaborativa pode ser realizada por meio de uma variedade de algoritmos sofisticados, que incluem desde redes de crença bayesianas até métodos de clusterização, como o conhecido *k-nearest neighbour* (k-NN), e abordagens baseadas em regressão. A fundamentação do Filtro Colaborativo reside na ideia central de que, se dois indivíduos, X e Y, compartilharam interesses similares ao interagir com um conjunto de itens – por exemplo, dando avaliações parecidas –, é altamente provável que essa semelhança de preferências se estenda a outros itens ainda não explorados por ambos.

Conforme descrito por Sarwar et al. (2001), ela utiliza as seguintes abordagens:

- **Filtragem Colaborativa Baseada em Usuário (UBCF - *User-Based Collaborative Filtering*):** A UBCF opera identificando um conjunto de usuários que compartilham padrões de preferência semelhantes com o usuário-alvo. Uma vez encontrados esses “vizinhos” (também chamados de vizinhos mais próximos), o sistema sugere itens que esses vizinhos apreciaram, mas que o usuário-alvo ainda não consumiu ou avaliou. A similaridade entre os usuários é tipicamente calculada usando métricas como a similaridade de cosseno, a correlação de Pearson ou a distância euclidiana, aplicadas às avaliações ou interações comuns que eles tiveram com os itens. (Ricci et al., 2011).

  Conforme explica Aggarwal (2016), na filtragem colaborativa baseada em usuário, "as avaliações fornecidas por usuários com afinidades semelhantes a um usuário-alvo A são utilizadas para fazer recomendações para A". A ideia central, portanto, é identificar usuários que se assemelham ao usuário-alvo e, a partir daí, estimar as avaliações para os itens não observados por A, calculando médias ponderadas das avaliações desse grupo de pares. Por exemplo, se Alice e Bob avaliaram filmes de maneira similar no passado, as avaliações observadas de Alice sobre um filme como "Exterminador do Futuro" podem ser usadas para prever as avaliações não observadas de Bob para o mesmo filme. Em geral, os *k* usuários mais similares a um determinado usuário podem ser empregados para realizar essas previsões de avaliação, com as funções de similaridade sendo computadas entre as linhas da matriz de avaliações para descobrir os usuários semelhantes (Aggarwal, 2016).

- **Filtragem Colaborativa Baseada em Item (*Item-Based Collaborative Filtering - IBCF*):** Em contraste com a UBCF, a IBCF foca na similaridade entre os itens. Ela recomenda itens ao usuário com base naqueles que o próprio usuário já demonstrou interesse, mas considerando a similaridade entre os *itens*. A ideia é que, se um usuário gostou do item A, e o item A é muito parecido com o item B (com base na forma como outros usuários os avaliaram), então o sistema deve recomendar o item B. Essa abordagem é particularmente vantajosa em cenários com um grande número de usuários,

onde o cálculo da similaridade entre itens pode ser mais estável e escalável do que entre usuários (Sarwar et al., 2001).

O processo se baseia em construir similaridade de itens: pré-computar a similaridade entre pares de itens com base nas avaliações que recebem de outros usuários; prever avaliações; gerar recomendações: selecionar itens com as maiores previsões de avaliação.

Ambas as abordagens da filtragem colaborativa podem enfrentar desafios como o problema da escassez de dados (quando há poucas avaliações) e o problema do *cold start* (dificuldade em recomendar para novos usuários ou itens) (Aggarwal, 2016). No entanto, sua simplicidade conceitual e eficácia em muitos cenários as tornam um ponto de partida fundamental no desenvolvimento de sistemas de recomendação.

Outra abordagem diversamente presente são os **Sistemas Baseado em Conteúdo (*Content-Based System*)**. Esse, por sua vez, opera com uma lógica distinta ao CF, não dependendo das interações de outros usuários para gerar recomendações. Sua principal característica reside em recomendar itens que apresentam características semelhantes àqueles que o próprio usuário já demonstrou preferência em seu histórico de consumo ou avaliação. Para que essa abordagem seja eficaz, é fundamental dispor de duas fontes de informação cruciais: metadados detalhados sobre os itens e um perfil de usuário que reflita de forma precisa seus interesses (Lops et al., 2011).

Os metadados dos itens referem-se a atributos descritivos, como gênero, palavras-chaves, autores ou qualquer informação que caracterize o item. O perfil de usuário, por outro lado, é construído a partir das interações passadas do usuário com os itens, podendo incluir avaliações explícitas ou comportamentos implícitos (como histórico de cliques, tempo de visualização ou compras efetuadas).

O processo de recomendação, nesse contexto, envolve a análise da similaridade entre os itens. Por exemplo, se um usuário demonstrou apreço por um filme de ficção científica dirigido por Christopher Nolan, o sistema baseado em conteúdo poderá sugerir outros filmes do gênero ficção científica ou outros filmes

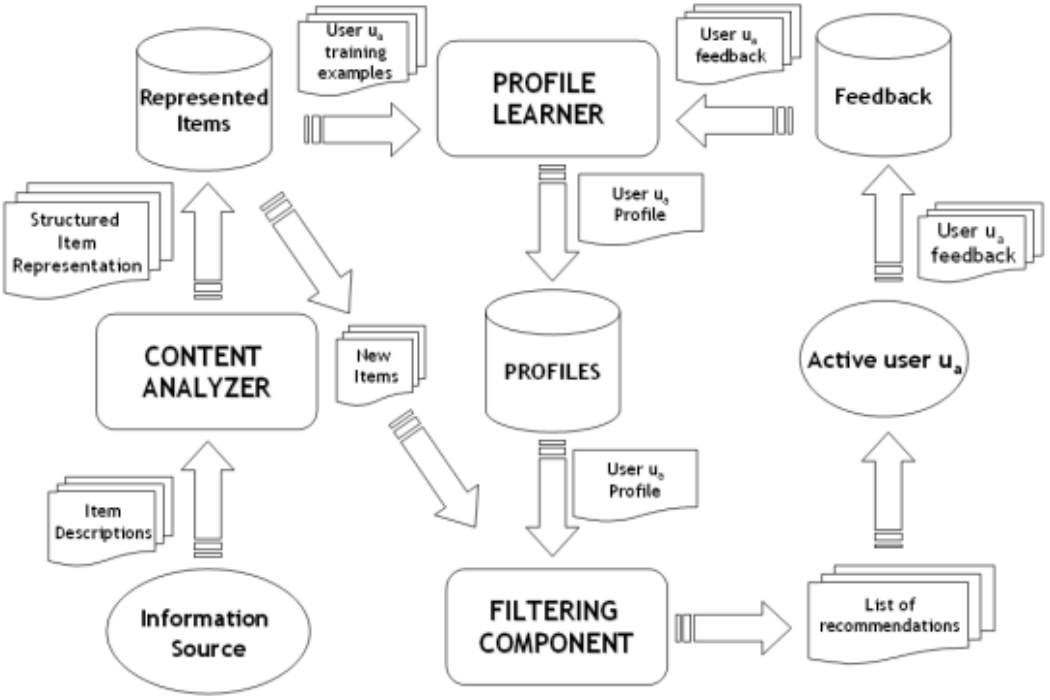
dirigidos por Nolan. Essa similaridade é calculada com base nas características dos itens e nas preferências registradas do usuário, sejam elas obtidas explicitamente (como avaliações diretas) ou implicitamente (inferidas do comportamento) (Lops et al., 2011).

Lops, Gemmis e Semeraro (2007) detalham que os sistemas de filtragem de informação baseados em conteúdos demandam técnicas apropriadas tanto para a representação dos itens quanto para a construção do perfil do usuário, além de estratégias para comparar o perfil do usuário com a representação dos itens. A arquitetura de alto nível de um sistema de recomendação baseado em conteúdo (conforme ilustrado, por exemplo, na Figura 3) envolve um processo de recomendação executado em três etapas distintas, cada uma gerenciada por um componente específico:

- **Analisador de conteúdo (*content analyzer*):** Este componente é crucial quando a informação não possui uma estrutura pré-definida, como textos. Ele realiza uma etapa de pré-processamento para extrair informações relevantes e estruturadas. Sua principal responsabilidade é representar o conteúdo dos itens provenientes de diversas fontes de informação em um formato adequado para as etapas subsequentes. Os itens de dados são analisados por técnicas de extração de características, que transformam a representação do item do espaço de informação original para um espaço-alvo (por exemplo, páginas web representadas como vetores de palavras-chave).
- **Aprendiz de perfil (*profile learner*):** A função deste módulo é responsável por coletar dados que representam as preferências do usuário e, então, generalizar esses dados para construir o perfil do usuário. Geralmente, a estratégia de generalização é implementada por meio de técnicas de aprendizado de máquina, capazes de inferir um modelo dos interesses do usuário a partir de itens que foram apreciados ou não no passado. Por exemplo, o aprendiz de perfil de um recomendador de páginas web pode empregar um método de *relevance feedback*, no qual a técnica de aprendizado combina vetores de exemplos positivos e negativos em um vetor protótipo que representa o perfil do usuário.

- **Componente de filtragem (*filtering component*):** O último módulo utiliza o perfil do usuário construído para sugerir itens relevantes. Isso é feito ao comparar a representação do perfil do usuário com a representação dos itens a serem recomendados, identificando aqueles que melhor correspondem aos interesses do usuário. Resultando em uma lista classificada de itens potencialmente interessantes, calculados usando métricas de similaridade.

Figura 3 - Arquitetura de alto nível de um recomendador baseado em conteúdo



Fonte: Recommender Systems Handbook (2011).

Uma das grandes vantagens dos sistemas baseados em conteúdo é sua capacidade de lidar com o problema do "cold start" para novos usuários, desde que estes expressem alguma preferência inicial ou que o sistema possa construir um perfil básico. Além disso, eles podem recomendar itens muito específicos que talvez não fossem descobertos por abordagens colaborativas, especialmente se esses itens não tiverem muitas interações de outros usuários. Contudo, uma limitação pode ser a falta de diversidade nas recomendações, já que tendem a sugerir itens muito semelhantes aos que o usuário já conhece, o que é conhecido como o problema da "super-especialização" (Aggarwal, 2016).

Por fim, temos a modalidade de **abordagens híbridas** onde combinam elementos da filtragem colaborativa e dos sistemas baseados em conteúdo, visando superar limitações individuais de cada uma. Estas buscam aprimorar a qualidade e robustez das recomendações por meio da combinação inteligente de elementos de ambas as técnicas. O objetivo principal é mitigar as desvantagens individuais, potencializando as forças de cada método (Burke, 2002).

Um exemplo claro dessa complementaridade reside no problema do "cold start". A filtragem colaborativa, por depender de um histórico de interações, apresenta dificuldades em recomendar itens para usuários recém-chegados ou itens recém-adicionados que ainda não possuem um volume significativo de avaliações. Por outro lado, os sistemas baseados em conteúdo, embora capazes de lidar com itens novos, podem pecar na descoberta de itens completamente inusitados ou na capacidade de expandir os horizontes do usuário, oferecendo recomendações que fogem de seu perfil pré-estabelecido (Burke, 2002). As abordagens híbridas visam preencher essas lacunas.

A integração entre as técnicas pode se manifestar de diversas maneiras, conforme categorizado por Burke (2002):

- **Ponderada (ou Mista):** As recomendações geradas por sistemas colaborativos e baseados em conteúdo são calculadas de forma independente e, posteriormente, combinadas por meio de um processo de ponderação. A contribuição de cada abordagem para a recomendação final pode ser ajustada conforme a necessidade ou desempenho em cenários específicos.
- **Sequencial (ou Cascade):** Uma abordagem atua como um refinador dos resultados da outra. Por exemplo, um sistema baseado em conteúdo pode pré-filtrar um grande conjunto de itens, e então a filtragem colaborativa é aplicada sobre esse subconjunto reduzido para gerar as recomendações finais.
- **De *Feature* (ou de Característica):** As características de conteúdo dos itens são integradas diretamente como *features* ou atributos adicionais nos modelos de filtragem colaborativa. Isso permite que os algoritmos

colaborativos considerem não apenas as interações dos usuários, mas também as propriedades intrínsecas dos itens.

- **De Modelo (ou Híbrido Puro):** Um único modelo unificado é construído, integrando aspectos e dados de ambas as abordagens desde a sua concepção. Em vez de combinar saídas ou refinar resultados, o modelo é intrinsecamente híbrido, buscando uma representação conjunta que otimize as recomendações.

A decisão sobre qual tipo de abordagem híbrida implementar é multifatorial. Ela depende da disponibilidade e da natureza dos dados (se há mais informações sobre usuários, itens ou ambos), da necessidade de escalabilidade do sistema para lidar com grandes volumes de dados e usuários, e da eficácia desejada em contornar problemas como o "cold start" para usuários e itens, buscando sempre aprimorar a precisão e a diversidade das recomendações (Burke, 2002).

2.2 Processamento De Linguagem Natural

O Processamento de Linguagem Natural (PLN), do inglês *Natural Language Process* (NLP), é um subcampo da Inteligência Artificial (IA) e Ciência da Computação que utiliza aprendizado de máquina para permitir que computadores entendam e se comuniquem com a linguagem humana. Seu objetivo é capacitar sistemas computacionais a compreender, interpretar e gerar texto ou fala em linguagens naturais. Segundo Silva e Lima (2007), um sistema computacional interpreta uma sentença em linguagem natural, através da análise de informações morfológicas (lexicais), sintáticas (regras gramaticais) e semânticas (significados), armazenadas em um dicionário, juntamente com as palavras que o sistema compreende.

No contexto do projeto Congraduations, o PLN é utilizado para processar automaticamente descrições de eventos acadêmicos, permitindo identificar tópicos, palavras-chave e categorias relevantes para cada oportunidade divulgada. Esse processamento torna possível a criação de recomendações personalizadas de forma automatizada, sem necessidade de categorização manual.

2.2.1 *Fundamentos e aplicações*

Os fundamentos do PLN escoram-se em anos de pesquisa em linguística computacional, inteligência artificial e ciência da computação. Historicamente, a área evoluiu de abordagens baseadas em regras e métodos estatísticos para, mais recentemente, paradigmas dominados por técnicas de aprendizado de máquina e, principalmente, aprendizado profundo (Manning; Schütze, 1999; Jurafsky; Martin, 2023).

A PLN é usada para uma ampla variedade de tarefas relacionadas à linguagem e está presente no cotidiano atual. Dentre as mais relevantes, destacam-se:

- **Análise de sentimentos:** Consiste no processo de inferir a polaridade emocional (positiva, negativa ou neutra) ou o tom expresso em um segmento de texto. Modelos de análise de sentimentos geralmente tomam um texto como entrada e produzem probabilidades associadas a cada categoria de sentimento. Essa tarefa pode ser realizada com base em features estatísticas ou por meio de modelos de aprendizado profundo capazes de capturar dependências sequenciais complexas. Suas aplicações são vastas, abrangendo desde a classificação de avaliações de clientes em plataformas digitais até a detecção de indicadores de transtornos mentais em comunicações online.
- **Classificação de Toxicidade:** Uma especialização da análise de sentimentos, a classificação de toxicidade foca na identificação de intenção hostis, categorizando-as em classes específicas como ameaças, insultos, obscenidades ou discurso de ódio contra identidades. É amplamente empregada na moderação de conteúdo online, auxiliando a detectar e silenciar comentários ofensivos, bem como na verificação de documentos para identificar difamação.
- **Tradução Automática :** Refere-se à automatização da conversão de texto de um idioma de origem para um idioma de destino. Abordagens eficazes nesse domínio são capazes de discernir entre palavras com significados

contextualmente semelhantes e, em alguns casos, realizar a identificação automática do idioma do texto.

- **Reconhecimento de Entidades Nomeadas (*Named Entity Recognition - NER*):** Tem como foco extrair e categorizar entidades específicas de um texto em classes predefinidas, tais como nome de pessoas, organizações, localização geográfica e quantidade. A saída do modelo inclui as entidades detectadas e suas respectivas posições no texto.
- **Detecção de Spam:** Modelos de detecção de spam analisam o texto do e-mail, juntamente com metadados como título e remetente, para determinar a probabilidade de ser uma mensagem indesejada. Provedores de e-mail utilizam amplamente essa técnica para aprimorar a experiência do usuário ao filtrar comunicações não solicitadas.
- **Modelagem de tópicos:** Uma tarefa de mineração de texto não supervisionado que, a partir de um *corpus* de documentos, descobre estruturas temáticas abstratas. A entrada é uma coleção de documentos e a saída consiste em uma lista de tópicos. Presente no Congraduations, algoritmos de Alocação Latente de Dirichlet (LDA) usados nessa área, visualizando documentos como coleções de tópicos e tópicos como coleção de palavras.
- **Chatbots:** Automatizam um lado de uma conversação, permitindo interações com usuários humanos. Podem ser categorizados em consulta a banco de dados, onde permitem que os indivíduos interroguem bancos de dados predefinidos utilizando linguagem natural para obter respostas específicas; também podem ser parte de geração de conversa, onde simulam diálogos mais abertos e fluentes com um humano.
- Recuperação de Informações: Foca em encontrar os documentos mais relevantes para uma consulta específica dentro de uma vasta coleção. Esses sistemas geralmente operam em duas fases: indexação (com modelos como Redes de Duas Torres) e correspondência (utilizando pontuações de similaridade).
- **Sumarização (*Summarization*):** A tarefa de condensar um texto longo para destacar suas informações mais relevantes. Divide-se em dois métodos principais sendo eles a sumarização extrativa onde se identifica e extrai as

frases mais importantes do texto original e as combina para formar o resumo, sem alterar o conteúdo textual; e a sumarização abstrativa que Produz um resumo por meio de paráfrase, reescrevendo o conteúdo de forma mais concisa e podendo incluir palavras e frases que não estavam presentes no texto original, de maneira análoga à escrita de um resumo por um humano. Modelos abstrativos são comumente tratados como tarefas de sequência para sequência.

No contexto específico de sistemas de recomendação, o PLN assume um papel crucial. Em abordagens baseadas em conteúdo, ele possibilita a extração e representação de características detalhadas a partir de descrições textuais de itens (como sinopses de filmes, resumos de livros ou especificações de produtos) e a construção de perfis de usuário baseados em suas interações textuais. Essa compreensão semântica aprimora a capacidade do sistema de recomendar itens que não apenas correspondam a preferências superficiais, mas que também se alinhem a interesses mais profundos e complexos do usuário (Lops; De Gemmis; Semeraro, 2011).

2.2.2 *Pré-processamento de texto*

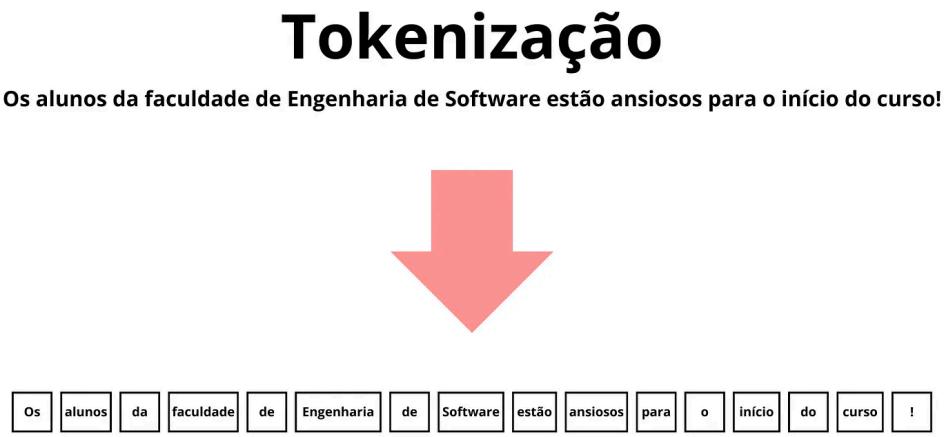
Os modelos de PLN funcionam encontrando relações entre as partes constituintes da linguagem como as letras, palavras e frases encontradas em um conjunto de dados textuais. Antes que as técnicas de PLN possam ser aplicadas, esses conjuntos de dados exigem uma etapa de pré-processamento. Essa fase visa reduzir a complexidade e o “ruído” nos dados, garantindo que as informações mais relevantes sejam preservadas e representadas de forma padronizada (Bird; Klein; Loper, 2009). Assim, o texto fica em um formato que seja mais facilmente compreendido e manipulado pelos algoritmos.

2.2.2.1 Tokenização

No contexto do Processamento de Linguagem Natural (PLN) e da aprendizagem automática, a tokenização é o processo de converter uma sequência de texto em unidades menores, chamadas tokens, como mostra a figura abaixo. Esses tokens podem variar em tamanho, desde caracteres individuais até palavras completas. O objetivo principal da tokenização é representar o texto de forma

significativa para as máquinas, sem perder o contexto. Ao transformar o texto em tokens, os algoritmos conseguem identificar padrões com maior facilidade. Esse reconhecimento de padrões é essencial, pois permite que as máquinas compreendam e respondam às informações humanas de maneira eficaz. De acordo com Manning (2008), a tokenização é o primeiro passo para a maioria das aplicações de PLN, preparando o texto para análises mais complexas.

Figura 4: Exemplo de tokenização básica.



Fonte: Davedovicz (2024).

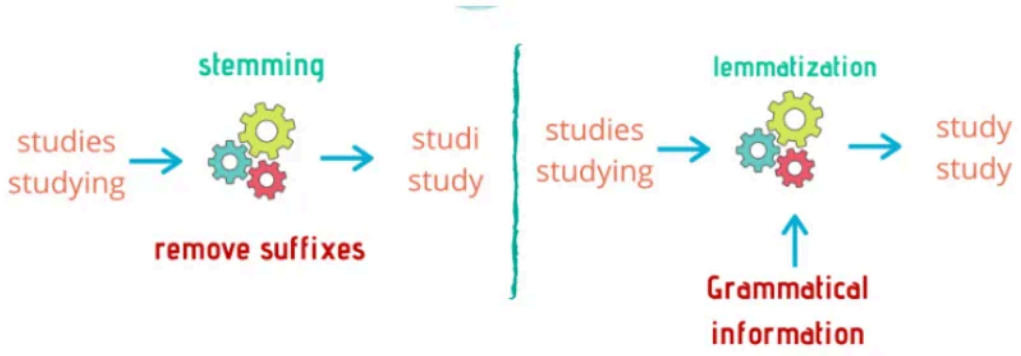
2.2.2.2 *Stemming* e *Lemmatization*

Stemming é uma técnica de processamento de linguagem natural (PLN) que visa reduzir palavras flexionadas ou derivadas ao seu radical ou raiz comum. Por exemplo, palavras como "programação", "programador" e "programas" podem ser reduzidas ao radical comum "programa". Essa técnica é amplamente utilizada para padronizar vocabulários em tarefas de PLN, como recuperação de informações e classificação de texto. Ao reduzir a redundância causada por palavras flexionadas ou derivadas, o stemming melhora a eficiência dos algoritmos ao simplificar a representação textual e identificar relações entre palavras relacionadas (Jurafsky & Martin, 2021). Isso permite que os modelos aprendam de maneira mais eficiente os

padrões associados a diferentes formas de uma palavra dentro de um mesmo contexto.

A *lemmatization* ou lematização é outra técnica PLN que reduz palavras flexionadas ou derivadas à sua forma básica, conhecida como lema. O lema é a forma dicionarizada ou base de uma palavra, determinada com base no seu significado e contexto. Esse processo algorítmico visa identificar a raiz semântica da palavra considerando sua função gramatical e uso contextual (Manning, 2008). Diferentemente do *stemming*, que reduz palavras a radicais muitas vezes incompletos ou sem significado linguístico, a lematização considera aspectos como a parte do discurso (e.g., substantivo, verbo) e o contexto da frase. Isso permite um processamento mais preciso, resultando em lemas que são palavras válidas e semanticamente corretas. Como exemplo, um lematizador mapeia "runs", "running" e "ran" para o lema "run" (Jurafsky & Martin, 2021).

Figura 5 - *Stemming* e *Lemmatization*



Fonte: Think Data Analytics (s.d.).

2.2.2.3 Remoção de *Stop Words* e Normalização

O principal objetivo da remoção de *stop words* é eliminar palavras de alta frequência que possuem baixo valor semântico e que, na maioria dos contextos, não contribuem significativamente para a compreensão do conteúdo principal do texto ou para a distinção entre documentos como artigos, preposições e conjunções (e.g., “o”, “a”, “de”, ”para”), que geralmente não carregam significado semântico para a análise em muitos. A remoção pode ser feita identificando as *N* palavras mais

frequentes em um conjunto de dados de treinamento ou utilizando listas de *stop words* predefinidas (Jurafsky; Martin, 2023).

A normalização visa converter o texto para um formato padrão, como transformar todas as letras em minúsculas, remover pontuações ou caracteres especiais. Muito como não só em atividades de PLN, mas também, como toda aplicação que envolva alguma base textual, seu objetivo é reduzir a variabilidade lexical e padronizar as representações das palavras no texto. Isso significa garantir que diferentes formas de uma mesma palavra ou diferentes representações de um mesmo conceito sejam tratadas de maneira consistente pelo sistema.

2.2.3 *Técnicas Aplicada ao Sistema*

2.2.3.1 *Bag of Words (BoW)*

O modelo Bag-of-Words (BoW) é usado para representar textos de maneira que algoritmos possam processá-los e analisá-los. Essa abordagem é fundamental por sua simplicidade e eficiência em tarefas como classificação de textos, recuperação de informações e análise de sentimentos (Manning,2008).

A principal característica do BoW é que ele ignora a ordem e a estrutura gramatical das palavras no texto, concentrando-se apenas na frequência com que cada termo aparece. O modelo transforma um texto em um vetor numérico, onde cada dimensão do vetor representa uma palavra do vocabulário criado a partir dos textos analisados. O valor em cada dimensão corresponde ao número de vezes que essa palavra ocorre no documento (Jurafsky & Martin, 2021).

2.2.3.2 *Term Frequency-Inverse Document Frequency* (TF-IDF)

Frequência de Termos - Frequência Inversa de Documentos (do inglês, *Term Frequency-Inverse Document Frequency - TF-IDF*), é uma medida estatística amplamente utilizada em processamento de linguagem natural (PLN) e recuperação de informação para avaliar a importância de uma palavra em um documento em relação a um *corpus* de documento. Essencialmente, ele quantifica o quão relevante uma palavra é para um documento específico, levando em consideração sua frequência dentro desse documento e sua raridade em outros documentos do *corpus*.

Como o próprio nome indica, o TF-IDF vetoriza/pontua uma palavra multiplicando a Frequência do Termo (TF) da palavra pela Frequência Inversa do Documento (IDF). Onde:

- **Term Frequency:** TF de um termo ou palavra é o número de vezes que o termo aparece em um documento em comparação com o número total de palavras no documento.

$$TF = \frac{O\ número\ de\ vezes\ que\ o\ termo\ aparece\ no\ documento}{número\ total\ de\ termos\ no\ documento}$$

- **Inverse Document Frequency**: O IDF mede a importância de um termo em todo o corpus. Ele penaliza termos que são muito comuns em muitos documentos, atribuindo-lhes um peso menor, e valoriza termos que são raros e, portanto, mais distintivos para documentos específicos.

$$IDF = log(\frac{número\ total\ de\ documentos\ no\ corpus}{número\ de\ documentos\ no\ corpus\ que\ contém\ o\ termo})$$

  O *log* é geralmente na base *e* (logaritmo natural) ou na base *10*. Adiciona-se +1 ao denominador ou ao numerador em algumas implementações para evitar divisão por zero caso um termo não apareça em nenhum documento, ou para evitar o log de 0.

  O TF-IDF de um termo é calculado multiplicando as pontuações TF e IDF.

$$TF - IDF = TF \times IDF$$

Em termos simples, a importância de um termo é alta quando ele ocorre com frequência em um determinado documento e raramente em outros. Em resumo, a similaridade dentro de um documento medida pelo TF é equilibrada pela raridade entre documentos medidos pelo IDF. A pontuação TF-IDF resultante reflete a importância de um termo para um documento no corpus.

O TF-IDF é útil em muitas aplicações de processamento de linguagem natural. Por exemplo, mecanismos de busca usam o TF-IDF para classificar a relevância de um documento para uma consulta. O TF-IDF também é empregado na classificação e sumarização de textos e na modelagem de tópicos. Além de ser empregado, também, para a filtragem de spam e recomendação de conteúdo, como é o caso do Congraduations.

2.2.3.3 *Latent Dirichlet allocation (LDA)*

O *Latent Dirichlet Allocation (LDA)* é um modelo probabilístico que permite extrair tópicos latentes de uma coleção de documentos. A ideia básica é que os documentos sejam representados como misturas aleatórias sobre tópicos latentes, onde cada tópico é caracterizado por uma distribuição sobre palavras. O principal objetivo do LDA é descobrir os "tópicos latentes" (escondidos) que perpassam uma coleção de documentos. Ele faz isso inferindo as distribuições de tópicos para cada documento e as distribuições de palavras para cada tópico. Sua base é fundamentada, em resumo, em:

- **Documentos são misturas de tópicos:** Um artigo sobre "inteligência artificial" e "aprendizado de máquina" pode ter uma parte de seu conteúdo relacionada a um tópico de "IA" e outra parte a um tópico de "Aprendizado de Máquina".
- **Tópicos são misturas de palavras:** O tópico "IA" provavelmente terá palavras como "algoritmo", "rede neural", "dados", "modelo" com alta probabilidade. O tópico "Aprendizado de Máquina" também pode ter algumas dessas palavras, mas talvez outras como "classificação", "regressão", "treinamento", "validação" sejam mais proeminentes.
- **Tópicos são gerados por um processo probabilístico:** O LDA tenta reverter um processo hipotético de como os documentos são criados, para inferir as estruturas de tópicos subjacentes.

A LDA estima a distribuição tópico-palavra *P(t|z)* e a distribuição documento-tópico *P(z|d)* a partir de um corpus não rotulado, usando "*priors*" de *Dirichlet* para as distribuições com um número fixo de tópicos. Essa distribuição é frequentemente usada para construir um vetor de características (feature vector) *F* para o documento, onde cada elemento representa a probabilidade de um tópico específico estar presente naquele documento:

$$F = (P(z_1|\, d), P(z_2|\, d), \ldots, P(z_k|\, d))$$

Este vetor *F* pode então ser utilizado em outras tarefas de aprendizado de máquina, como classificação ou agrupamento de documentos.

2.2.3.4 Agrupamento de K-means ou Clusterização

O agrupamento de K-means, também conhecido como *K-means clustering,* é um dos algoritmos mais populares da área de aprendizado de máquina não supervisionados. Seu objetivo principal é particionar um conjunto de dados em K grupos distintos (ou *clusters*), com base em características similares. Cada grupo é representado por um centróide que corresponde, de forma simplificada, ao ponto médio de cada cluster e os dados são atribuídos ao grupo cujo centróide estiver mais próximo (MACQUEEN, 1967).

O funcionamento do algoritmo segue um processo iterativo. Inicialmente, são escolhidos K centróides aleatórios. Em seguida, os dados são atribuídos ao centróide mais próximo com base em uma métrica de distância, geralmente a distância euclidiana. Após essa atribuição, os centróides são recalculados como a média dos pontos pertencentes ao respectivo grupo. Esse processo se repete até que os centróides se estabilizem ou até um número máximo de iterações seja atingido (JAIN, 2010). A eficiência e simplicidade do K-means o tornam uma escolha comum em aplicações que envolvem grandes volumes de dados.

No contexto de Processamento de Linguagem Natural (PLN), o K-means pode ser utilizado para descobrir padrões em textos, agrupar documentos semelhantes e organizar conteúdos tematicamente. Ao aplicar vetorização textual (como TF-IDF ou embeddings) a documentos, é possível representar o conteúdo de cada texto como um vetor numérico. Esses vetores podem então ser agrupados com K-means, permitindo, por exemplo, a criação de clusters temáticos de eventos acadêmicos — como “inteligência artificial”, “segurança da informação” ou “desenvolvimento web” — sem a necessidade de etiquetas prévias.

2.2.3.5 Modelos Baseados em *Embeddings*

Os modelos baseados em *embeddings* revolucionaram o campo do Processamento de Linguagem Natural ao introduzirem uma forma densa, contínua e semântica de representar palavras, frases ou documentos como vetores numéricos. Diferentemente das abordagens tradicionais, como o modelo *Bag-of-Words* (BoW), que geram vetores esparsos, os embeddings capturam relações semânticas e

sintáticas entre palavras em um espaço vetorial, facilitando o aprendizado de máquina em tarefas complexas de PLN (Mikolov, 2013).

Os *embeddings* são gerados por meio de algoritmos de aprendizado profundo que mapeiam palavras para vetores de dimensões fixas. O objetivo principal é garantir que palavras semanticamente similares tenham representações vetoriais próximas. Por exemplo, palavras como "rei" e "rainha" terão vetores próximos no espaço vetorial, enquanto "rei" e "banana" estarão distantes (figura 6). Esses modelos aprendem tais relações analisando grandes corpora textuais e utilizando técnicas como o treinamento supervisionado ou não supervisionado (Goldberg, 2016). Modelos como *Word2Vec*, *GloVe* e *FastText* foram pioneiros em criar *embeddings* para palavras, enquanto arquiteturas mais recentes, como *BERT* e *GPT*, expandiram a ideia para capturar o contexto em níveis mais profundos.

Figura 6: Representação de word embeddings.

Fonte: Fonseca (2019).

2.2.3 *Aplicações que utilizam processamento de linguagem natural*

Nesta seção serão apresentadas algumas das aplicações similares que foram retornadas a partir da revisão sistemática da literatura, onde essas aplicações utilizam sistema de recomendação e PNL na educação assim como a solução proposta neste trabalho.

2.2.3.1 Aplicação de PLN na recomendação de conteúdo educacional

Na tese de Filho (2024), é proposto um sistema de recomendação que utiliza técnicas de Processamento de Linguagem Natural (PLN) para ajudar estudantes a organizarem suas trajetórias de estudo, considerando o desafio imposto pela grande quantidade de conteúdo educacional disponível. Segundo o autor, essa abundância de materiais dificulta a identificação de recursos realmente relevantes para o alcance dos objetivos acadêmicos ou profissionais dos alunos. O dataset utilizado para a construção do modelo foi extraído de plataformas renomadas como Udemy e Coursera, conhecidas pela diversidade e qualidade de seus conteúdos educacionais.

O modelo apresentado é um sistema de recomendação híbrido, que combina duas abordagens principais. A primeira é a filtragem colaborativa, que utiliza dados de perfis de usuários para gerar recomendações com base em itens semelhantes aos relacionados ao perfil de entrada. A segunda abordagem é a filtragem baseada em conteúdo, que refina as recomendações priorizando os itens mais relevantes de acordo com as características específicas do usuário.

O funcionamento do modelo é dividido em etapas distintas:

1. **Coleta e Preparação dos Dados:** Nesta etapa, os dados são extraídos, tratados e vetorizados. A vetorização transforma os textos em representações numéricas que podem ser processadas pelo sistema.
2. **Construção da Matriz de Interações:** Uma matriz é criada a partir dos vetores gerados pelos dados tratados, possibilitando a representação das relações entre usuários e itens. Essa matriz será utilizada no cálculo das similaridades.
3. **Cálculo de Similaridade:** O sistema percorre a matriz de interações comparando os vetores para determinar as similaridades entre os itens.
4. **Geração Inicial de Recomendações:** Após identificar os itens mais similares, o sistema gera um conjunto inicial de recomendações. É nesta etapa que as técnicas de PLN são aplicadas, incluindo a transformação de dados textuais, remoção de palavras irrelevantes, redução de palavras ao

radical significativo, eliminação de pontuações e outras técnicas de pré-processamento.

5. **Refinamento com Novo Cálculo de Similaridade:** Na última etapa, o sistema realiza um novo cálculo de similaridade para refinar as recomendações, garantindo maior precisão e relevância no resultado final.

O modelo proposto por Filho (2024) apresenta um fluxo estruturado e eficiente, combinando a força da filtragem colaborativa e baseada em conteúdo com técnicas avançadas de PLN. Esse sistema híbrido se mostra promissor ao priorizar a entrega de recomendações personalizadas e alinhadas às necessidades dos estudantes, permitindo que eles otimizem suas jornadas acadêmicas em meio a uma ampla oferta de materiais educacionais.

2.2.3.2 *Recommendation System for Students' Course Selection*

O estudo realizado por Naren (2020) busca oferecer suporte a estudantes de graduação no processo de escolha de disciplinas a cada semestre, levando em conta suas habilidades, interesses, demandas do mercado e outros fatores determinantes. Para isso, os autores desenvolveram um modelo que integra técnicas de Processamento de Linguagem Natural (PLN) e mineração de dados, implementado por meio de scripts de *Python* para *frontend* e *backend*, com hospedagem em uma plataforma *web*.

No contexto do projeto, a mineração de dados, também conhecida como *Knowledge Discovery from Data* (KDD), foi empregada para reunir informações sobre disciplinas cursadas pelos estudantes, incluindo suas respectivas notas. Já o PLN foi essencial para o processamento dos dados textuais, utilizando métodos como tokenização, remoção de *stopwords*, lematização, e stemização para tornar as informações mais compreensíveis para os algoritmos.

A solução proposta inclui uma plataforma interativa que permite aos estudantes obter recomendações de disciplinas com base em suas experiências acadêmicas prévias. O sistema analisa as notas obtidas em disciplinas já cursadas, calculando sua similaridade com as matérias disponíveis para escolha. Entre essas, o sistema classifica as cinco mais relevantes e, a partir de um cálculo acumulativo de pontuações, recomenda as duas disciplinas mais indicadas ao aluno.

Na implementação técnica, os autores utilizaram o framework Flask para gerenciar o backend da aplicação e processar requisições HTTP via *API REST*. O frontend foi construído com templates HTML gerenciados por Jinja2. Durante os testes, diferentes métricas de similaridade foram avaliadas, como os coeficientes de *Dice*, cosseno, distância Euclidiana, *Manhattan* e coeficiente de *Pearson*. Os resultados mostraram que os coeficientes de *Dice* e cosseno apresentaram o melhor desempenho para medir a similaridade entre disciplinas no contexto textual, sendo o coeficiente de *Dice* identificado como o mais adequado para o sistema.

A análise experimental envolveu a avaliação de 11 disciplinas comparadas com a disciplina de *Python*. Nessa análise, o coeficiente de *Dice* se destacou, confirmando sua eficiência para aplicações baseadas em texto. Outras métricas, como as distâncias Euclidiana e *Manhattan*, não demonstraram resultados satisfatórios no contexto do sistema e foram descartadas. Assim, o trabalho reafirma o valor do coeficiente de *Dice* para análise de similaridade em projetos de recomendação educacional.

2.2.3.3 *Agent-Based Recommendation in E-Learning Environment Using Knowledge Discovery and Machine Learning Approaches*

Um estudo realizado por Shahbazi & Byun (2022) apresenta um sistema de recomendação baseado em agentes projetado para ambientes de *e-learning*. O trabalho explora a combinação de técnicas de descoberta de conhecimento (*Knowledge Discovery*) e aprendizado de máquina para oferecer recomendações personalizadas aos alunos, levando em consideração o comportamento do usuário, histórico de interações e preferências individuais. A proposta busca melhorar a experiência de aprendizado, fornecendo sugestões adaptadas às necessidades específicas de cada estudante.

O sistema utiliza um modelo baseado em agentes, no qual diferentes agentes desempenham papéis especializados, como coleta de dados, análise de comportamento e geração de recomendações. Esses agentes trabalham em conjunto para construir um perfil detalhado de cada aluno, considerando fatores como cursos anteriormente acessados, atividades realizadas e desempenho

acadêmico. Com base nesse perfil, o sistema aplica algoritmos de aprendizado de máquina para prever os cursos ou materiais mais relevantes para cada usuário.

O Processamento de Linguagem Natural (PLN) desempenha um papel central na análise dos dados textuais relacionados ao conteúdo educacional e às interações dos usuários. Técnicas de PLN, como tokenização, análise de sentimentos e categorização de texto, são empregadas para entender melhor as descrições dos cursos e os *feedbacks* fornecidos pelos alunos. Essas análises permitem que o sistema compreenda o significado semântico dos materiais educacionais, garantindo que as recomendações sejam mais precisas e contextualmente relevantes.

Os autores também integram métodos de descoberta de conhecimento para extrair padrões ocultos nos dados coletados, como tendências de aprendizado e preferências gerais de grupos de alunos. Esses padrões são utilizados para aprimorar o sistema de recomendação, tanto em nível individual quanto coletivo. Além disso, algoritmos de aprendizado supervisionado e não supervisionado são aplicados para identificar relações entre alunos e conteúdos, maximizando a eficiência do processo de recomendação.

Os resultados apresentados no estudo indicam que o sistema proposto oferece um desempenho superior em comparação com abordagens tradicionais de recomendação em plataformas de *e-learning*. A integração de técnicas de PLN e aprendizado de máquina permite uma personalização mais rica e um alinhamento maior com as necessidades dos alunos, tornando o aprendizado mais eficiente e satisfatório. O trabalho de Shahbazi e Byun (2022) demonstra o potencial de sistemas baseados em agentes e PLN para transformar a experiência educacional em ambientes digitais.

CAPÍTULO 3: A CONSTRUÇÃO DO CONGRADUATIONS — DA IDEIA À IMPLEMENTAÇÃO

Neste capítulo será abordado o processo de desenvolvimento do sistema de recomendação acadêmica Congraduations que utiliza Técnicas de Processamento de Linguagem Natural (PLN) somado, ao fim, com o envio de newsletters personalizadas para os e-mails acadêmicos dos alunos. O sistema tem como principal objetivo incentivar a participação do discente em eventos acadêmicos, como congressos e seminários, além de promover o interesse pela continuidade dos estudos por meio da recomendação de oportunidades de pós-graduação e produção científica.

3.1 Concepção da Ideia

A escolha do presente tema surgiu diante uma reflexão da minha jornada acadêmica junto de meus colegas dentro do IFAM-CMC. Durante toda a graduação, percebi, junto aos meus colegas, a ausência de estímulos voltados à participação em eventos acadêmicos e científicos, bem como à produção de artigos científicos e a busca por oportunidades nos estudos, além de não haver uma comunidade tecnológica hegemônica e ativa na comunidade tech nacional.

Em nenhum momento, ao longo dos semestres, esses caminhos foram apresentados de forma clara ou incentivados pela Instituição, problemas esses motivados pela falta de estrutura ou gestão. Raramente mencionavam-se congressos, seminários, editais de publicações ou até mesmo a existência de possibilidades de mestrados ou especializações voltadas para egressos. Como resultado, a graduação muitas vezes se resumia à frequência em sala de aula e ao cumprimento da carga horária mínima exigida, sem que os alunos fossem estimulados a explorar o potencial formativo que o ambiente acadêmico pode oferecer.

Essa limitação não apenas empobrece a experiência universitária, como também priva os alunos de vivências que podem ampliar sua visão do mundo, fortalecer o currículo profissional e *networking* na área e abrir portas para a pesquisa, a inovação e o crescimento intelectual. As consequências dessa ausência

de estímulo não recaem apenas sobre os discentes, mas também sobre o IFAM enquanto instituição de ensino superior, que deixa de potencializar seu papel formador, perde visibilidade no cenário acadêmico e deixa de atrair reconhecimento, prestígio e investimentos advindos de políticas públicas e iniciativas privadas.

Foi nesse contexto que surgiu a ideia do Congraduations: um sistema de recomendação acadêmica capaz de conectar os alunos com oportunidades que muitas vezes passam despercebidas, como eventos na área de tecnologia, chamadas para publicação de artigos, cursos de extensão e caminhos para a pós-graduação utilizando Processamento de Linguagem Natural (PLN), área na qual sou fascinada e, para facilitar o acesso à informação, o envio de newsletters com o conteúdo citado anteriormente. A proposta visa, portanto, não apenas informar, mas também inspirar e engajar os estudantes, incentivando uma participação mais ativa e consciente na vida acadêmica.

3.2 Arquitetura e Funcionamento do Sistema

O sistema foi concebido com o propósito de ser uma aplicação leve, acessível e funcional, priorizando a facilidade de uso e acesso para os estudantes. Essa idealização reflete-se na sua arquitetura, que busca minimizar barreiras técnicas.

O frontend foi projetado para oferecer uma experiência amigável e responsiva, facilitando a interação dos alunos com o sistema. A interface principal do sistema é intuitiva e apresenta de forma concisa os objetivos do projeto, guiando o estudante no cadastro de seu e-mail acadêmico para o recebimento das newsletters[1]. Após o cadastro, uma janela de confirmação é exibida, alertando o usuário para verificar sua caixa de entrada. A confirmação do e-mail libera o acesso à newsletter, pois os dados cadastrados são guardados no banco de dados e processados pelo *backend*, onde o envio é disparado em poucos minutos conforme o programado.

[1] Newsletter: Boletim informativo distribuído periodicamente por meio digital (geralmente por e-mail), com o objetivo de comunicar novidades, conteúdos relevantes ou atualizações sobre um tema específico a um público-alvo previamente inscrito.

É importante destacar que uma newsletter é uma ferramenta de comunicação digital enviada por e-mail, com o propósito de manter o público informado sobre atualizações, novidades ou conteúdos relevantes. Com isso, os alunos receberão atualizações diretamente em seus e-mails institucionais, abrangendo temas diversificados como Desenvolvimento de Software, Inteligência Artificial, Segurança da Informação, DevOps[2] e Gestão de Projetos. Além de informar, o sistema visa estimular o engajamento dos estudantes na comunidade tecnológica, promovendo a produção de artigos, a participação em palestras, o estabelecimento de networking e outras iniciativas que contribuam para o crescimento acadêmico e profissional.

A plataforma foi desenvolvida com foco na compatibilidade e integração, optando-se por uma estrutura frontend simples. Uma página web responsiva, construída com React.js que é uma biblioteca JavaScript de código aberto amplamente utilizada para o desenvolvimento de interfaces de usuário (UI). O React permite a criação de componentes que encapsulam lógica, estrutura HTML e estilos CSS, tornando o código mais modular e fácil de gerenciar. Cada componente pode ser visto como um bloco de construção independente, representando partes da interface, como botões, formulários ou tabelas. Com essa abordagem, desenvolvedores conseguem dividir interfaces complexas em partes menores e mais reutilizáveis, o que facilita a manutenção e a escalabilidade de aplicações.

Para o desenvolvimento do *backend* foi utilizado a IDE[3] Rider da JetBrains, uma ferramenta otimizada para o ecossistema .NET. A arquitetura do sistema foi planejada utilizando ASP.NET Core para a construção de APIs RESTful[4], que são a espinha dorsal da comunicação entre os módulos e utilizam o Swagger UI[5], além de ser estruturado com a arquitetura MVC (Model-View-Controller). O ASP.NET Core gerencia toda a lógica de negócio, processando as solicitações vindas do *frontend*, validando os dados fornecidos pelos usuários, suas principais principais responsabilidades incluem receber e validar o e-mail enviado pelo usuário;

[2] DevOps: Prática que integra desenvolvimento e operações para automatizar processos e acelerar entregas.
[3] IDE: Ambiente de Desenvolvimento Integrado, facilita a codificação e depuração de software.
[4] API RESTful: Interface de programação baseada em princípios REST, usada para comunicação entre sistemas via HTTP.
[5] Swagger UI: Ferramenta que documenta e permite testar APIs REST de forma interativa e visual.

armazenar os dados no banco de dados MySQL; gerenciar o status de confirmação de e-mail; controlar o fluxo de envio das newsletters.

A escolha deste *framework* também se deu pelo seu suporte nativo ao envio de e-mails. Através da biblioteca System.Net.Mail, na qual facilita a configuração e a utilização de servidores SMTP (Simple Mail Transfer Protocol). Essa funcionalidade é crucial para o Congraduations, uma vez que as recomendações personalizadas geradas pelo sistema são entregues aos estudantes em formato de newsletter por e-mail, garantindo uma comunicação direta e eficiente.

O core da aplicação, responsável pela coleta, processamento, tratamento de dados e aplicação de técnicas de Processamento de Linguagem Natural (PLN), é desenvolvido utilizando Python e o Jupyter Notebook. O processo se inicia com a coleta automatizada de dados sobre eventos de tecnologia, realizada por meio de *web scraping* em fontes relevantes como o calendário de eventos da Sociedade Brasileira de Computação (SBC) e o repositório “Agenda Tech Brasil” no GitHub[6].

Após a coleta, os dados dos eventos são inicialmente armazenados em um arquivo .CSV[7] e, em seguida, passam por uma etapa de pré-processamento e limpeza de dados. Esse tratamento envolve a remoção de acentos, normalização para minúsculas, exclusão de caracteres especiais, tokenização e filtragem de *stopwords*, com um conjunto específico de palavras adaptadas para o contexto de eventos. O algoritmo RSLP[8] é aplicado para *stemming*, reduzindo as palavras às suas raízes e otimizando a análise.

Para o processamento textual, o sistema emprega a biblioteca spaCy para suporte à Extração de Entidades Nomeadas (Named Entity Recognition - NER), permitindo identificar nomes próprios e outras entidades relevantes nos títulos e descrições dos eventos. Essa biblioteca utiliza modelos pré-treinados baseados em redes neurais para capturar informações semânticas e sintáticas com alta precisão. A classificação temática automática é implementada por uma abordagem baseada em palavras-chave agrupadas por categorias tais como: Inteligência Artificial, Web

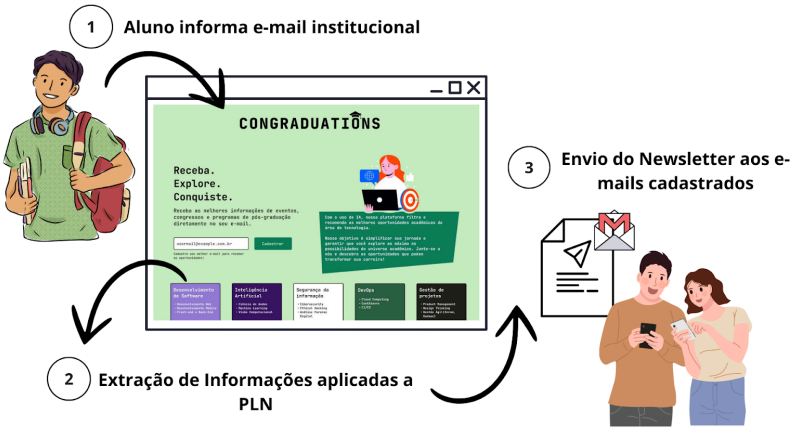
[6] GitHub: Plataforma online de hospedagem e controle de versões de código baseada em Git.
[7] CSV**:** Formato de arquivo de texto usado para armazenar dados tabulares separados por vírgulas.
[8] RSLP: Algoritmo de stemming em português que reduz palavras às suas raízes lexicais.

Mobile, DevOps, Segurança, UX Design, IoT, Programação e Negócios. Avaliando, assim, a similaridade com os nomes dos eventos por meio de pontuação heurística, que considere tanto termos compostos quanto suas partes individuais.

Além disso, a aplicação incorpora métodos de visualização de dados para análise exploratória como nuvens de palavras e gráficos de barra que exibem a distribuição de categorias, utilizando bibliotecas como Matplotlib, Seaborn e WordCloud. Todo o *pipeline* é encapsulado em uma função principal que gerencia desde o carregamento e processamento dos dados até a geração das saídas estruturadas em CSV e dos elementos gráficos. Essa integração de técnicas consagradas de PLN, mineração de texto e machine learning resulta em um sistema robusto de análise automatizada, crucial para aplicações como curadoria de conteúdo, recomendações e a geração de *newsletters* personalizadas.

Desse modo, o fluxo do projeto organiza-se da seguinte maneira: inicialmente, o aluno realiza o cadastro de seu e-mail institucional no sistema. Em seguida, o sistema procede com a extração de informações sobre eventos, congressos e programas de pós-graduação, utilizando *web scraping* para coletar dados relevantes. Após essa etapa, é aplicado o modelo com técnicas de PLN para filtrar e categorizar os eventos. Por fim, as recomendações são formatadas em um template de e-mail e distribuídas automaticamente para os e-mails cadastrados, assegurando uma entrega prática e eficiente das informações. Como mostra o fluxo ilustrado na figura abaixo:

Figura 7 - Fluxo abstrato do projeto



Fonte: Elaborado pela autora (2024).

3.3 Produção e Desenvolvimento

Neste tópico, será detalhado o processo de produção da aplicação proposta, abordando as fases de concepção, desenvolvimento e implementação.

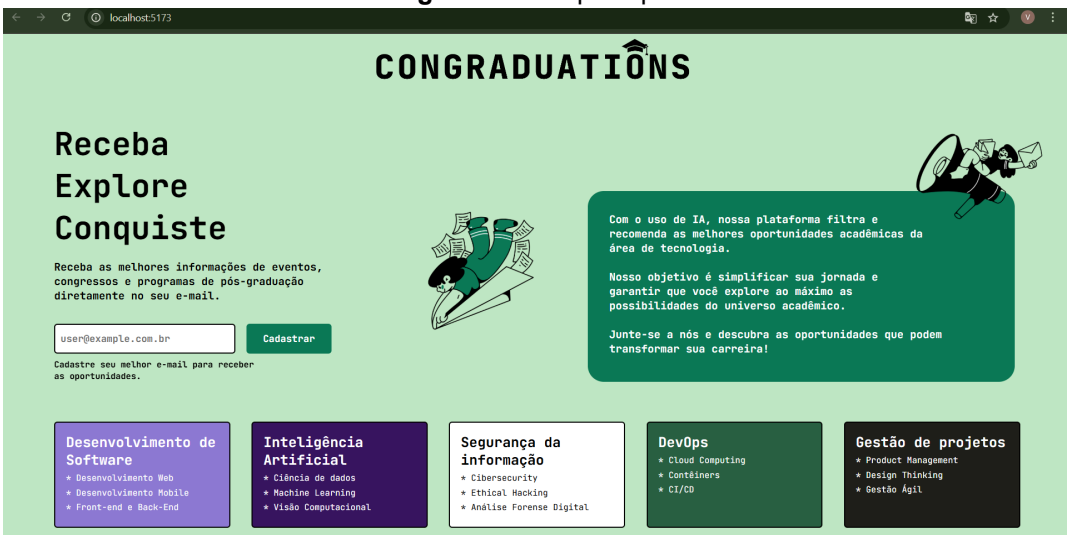
3.3.1 *Design do sistema*

O design da aplicação Congraduations foi concebido com base nos princípios de simplicidade, acessibilidade e objetividade. Por se tratar de um sistema voltado à recomendação de eventos acadêmicos e oportunidades de desenvolvimento estudantil, optou-se por uma interface limpa e funcional, com foco na experiência do usuário (UX) e na clareza das informações apresentadas.

A estética visual verde foi pensada para criar uma identidade acadêmica, moderna e acolhedora. As cores utilizadas seguem uma paleta suave com alto contraste entre texto e fundo, a fim de facilitar a leitura em diferentes dispositivos e condições de iluminação. Os elementos visuais foram distribuídos de forma a guiar o usuário de maneira intuitiva, reduzindo a necessidade de instruções adicionais.

A **tela principal (Figura 8)** apresenta uma descrição resumida do projeto, destacando sua proposta de enviar recomendações personalizadas de eventos e oportunidades acadêmicas por meio de newsletters automáticas. Nela, o usuário é orientado a inserir seu e-mail institucional (terminado em @ifam.edu.br), com o apoio de uma mensagem clara que explica o propósito dessa etapa.

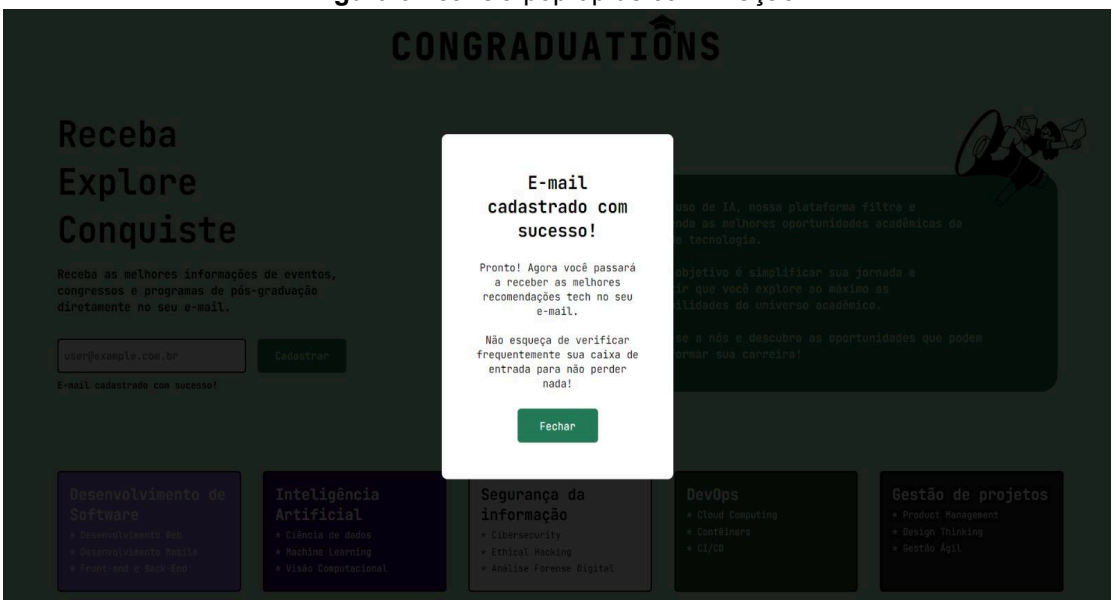
Figura 8 - Tela principal



Fonte: Elaborado pela autora (2025).

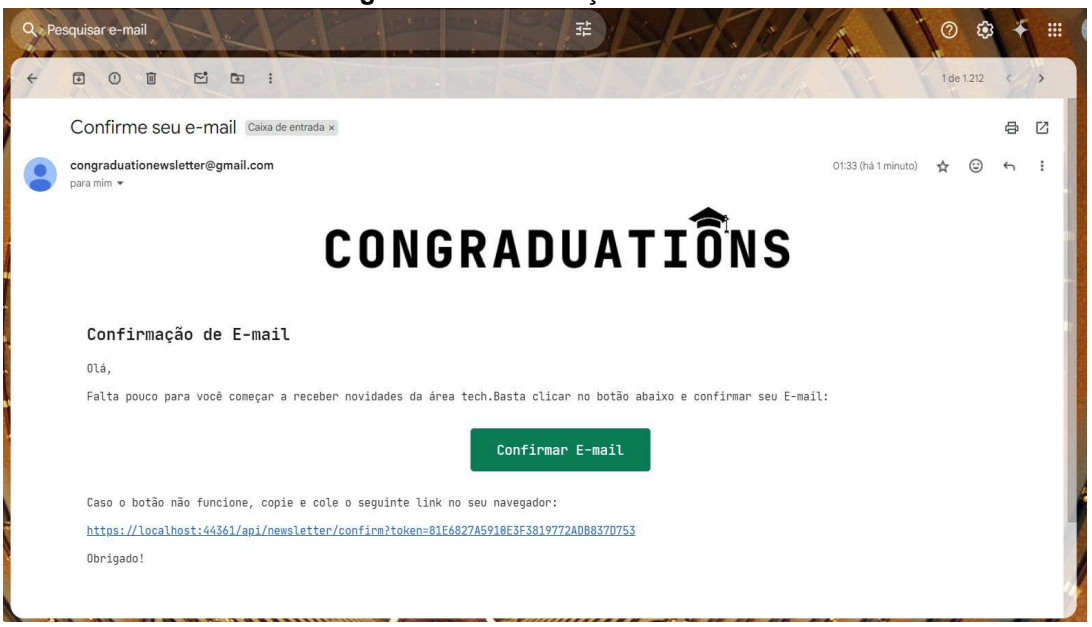
Após o cadastro, uma janela pop-up (Figura 9) é exibida, informando o sucesso do registro e a necessidade de verificar a caixa de entrada para confirmação. A confirmação do e-mail (Figura 10) é realizada através de um e-mail com um botão e de um link enviado ao endereço fornecido, garantindo a autenticidade do usuário. Uma vez confirmado, o e-mail do estudante estará apto para receber as newsletters com as recomendações.

Figura 9 - Janela pop-up de confirmação.



Fonte: Elaborado pela autora (2025).

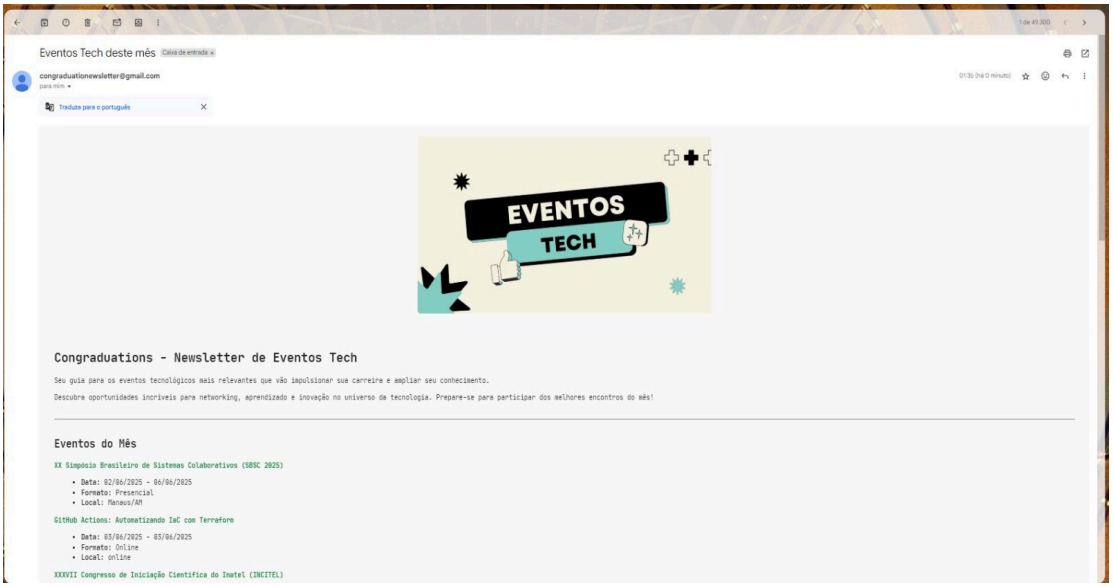
Figura 10 - Confirmação do e-mail.



Fonte: Elaborado pela autora (2025).

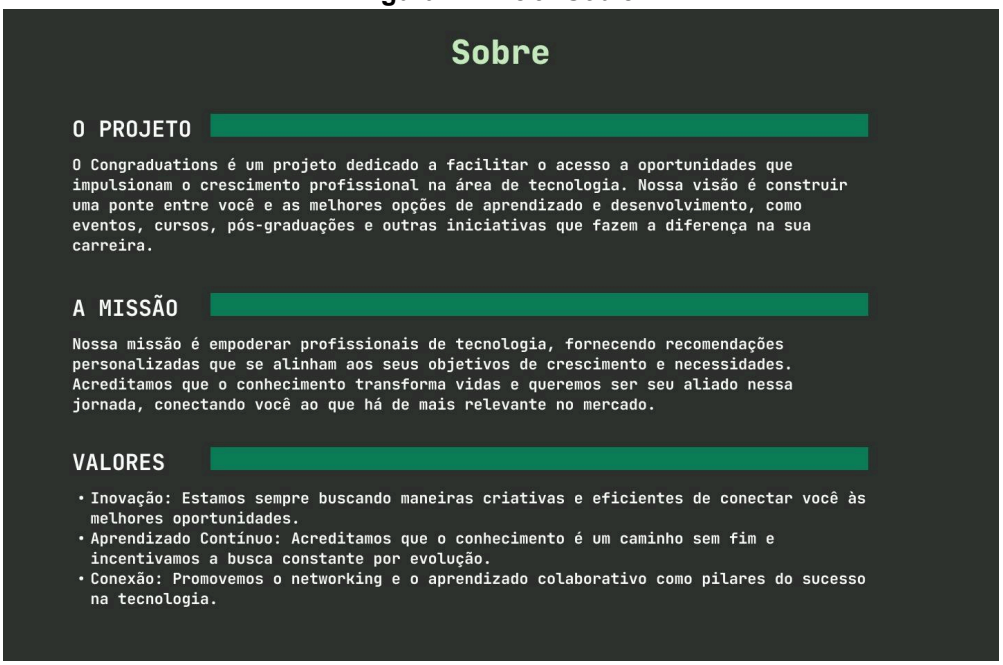
A newsletter (Figura 11), por sua vez, segue a paleta de cor do projeto com tons verdes. Ela contém uma apresentação do projeto Congraduations e, em seguida, são apresentados os eventos e outras oportunidades que acontecerão no mês corrente e nos meses posteriores. Além disso, há também uma tela de “Sobre” do projeto (Figura 12), que complementa a apresentação, detalhando sua missão, visão e valores, reforçando o compromisso com o crescimento acadêmico e profissional dos estudantes.

Figura 11 - O newsletter.



Fonte: Elaborado pela autora (2025).

Figura 12 - Tela “Sobre”



Fonte: Elaborado pela autora (2025).

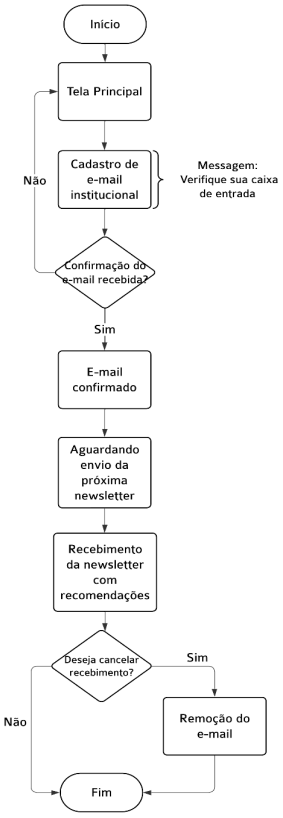
Todos os componentes visuais da aplicação Congraduations foram cuidadosamente prototipados na ferramenta Figma. Essa abordagem permitiu uma validação prévia das telas e do fluxo de navegação antes mesmo de iniciar a implementação em React.js, garantindo que o design atendesse às expectativas e necessidades dos usuários. Para enriquecer a comunicação visual e contribuir para a fluidez da navegação, foram utilizados ícones vetoriais de plataformas renomadas como Flat Icon e Iconscout, que representam ações e elementos do sistema de forma clara e intuitiva.

O layout da aplicação é totalmente responsivo, adaptando-se de forma eficiente a diferentes tamanhos de tela. Isso garante uma excelente usabilidade tanto em computadores desktop quanto em dispositivos móveis, como smartphones e tablets. A consistência visual é uma prioridade: botões, campos de formulário e mensagens de feedback seguem um padrão visual unificado. Essa uniformidade facilita a compreensão do usuário, promovendo uma interação mais fluida e acessível em todas as etapas da jornada.

Além disso, o desenvolvimento do design pautou-se em rigorosos princípios de acessibilidade. Isso inclui a escolha de fontes altamente legíveis, garantindo um tamanho mínimo adequado de texto para facilitar a leitura por diversos públicos, e a inclusão de feedbacks visuais claros para ações importantes, como o envio de e-mail e a confirmação de cadastro. O objetivo primordial do design da aplicação é permitir que o aluno compreenda rapidamente o funcionamento da plataforma e possa utilizá-la com facilidade, mesmo que tenha pouca familiaridade com sistemas web.

O fluxo de utilização completo do sistema, desde o acesso inicial até o recebimento das newsletters personalizadas e a opção de cancelamento da assinatura, é detalhado de forma abrangente no Fluxograma da Figura 13. Este diagrama visual ilustra de maneira clara e concisa todas as etapas de interação do usuário com a plataforma Congraduations, proporcionando uma visão geral do processo.

Figura 13 - Fluxograma do projeto.



Fonte: Elaborado pela autora (2025).

3.3.2 *Desenvolvimento e implementação*

Nesta seção, será detalhado o processo de desenvolvimento e implementação da aplicação Congraduations, com foco nas ferramentas, scripts e *pipelines* construídos. Serão apresentados os principais componentes de código e as etapas de cada processo.

3.3.2.1 Dados

Como dito anteriormente, a *pipeline* de dados é o cerne do sistema. Desenvolvida integralmente em Python e utilizando o Jupyter Notebook, essa *pipeline* se divide em três etapas principais: *web scraping* dos dados, pré-processamento e limpeza, e aplicação de técnicas de Processamento de Linguagem Natural (PLN).

A etapa inicial consiste no *web scraping* automatizado para a coleta de informações sobre eventos de tecnologia. Os dados são extraídos de fontes relevantes e atualizadas, como o calendário de eventos da Sociedade Brasileira de

Computação (SBC) e o repositório "Agenda Tech Brasil" disponível no GitHub. O script de *web scraping* foi projetado para navegar nessas páginas, identificar e extrair metadados dos eventos, tais como nome, data, local, formato (se é presencial, online ou híbrido) e links, no qual pode ser observado nas figuras 14 e 15.

Figura 14 - *Web scraping* - SBC.

```python
# Eventos da SBC (Sociedade Brasileira de Computação)
API_URL = "https://www.sbc.org.br/wp-json/tribe/events/v1/events"

def fetch_sbc_events():
    events = []
    page = 1
    per_page = 10
    total_pages = None

    while True:
        print(f"Buscando página {page} da SBC")

        params = {"page": page, "per_page": per_page}
        response = requests.get(API_URL, params=params)

        if response.status_code == 200:
            data = response.json()
            events.extend(data.get("events", []))

            if total_pages is None:
                total_pages = data.get('total_pages', 1)
                print(f"Total de páginas SBC: {total_pages}")

            if page >= total_pages:
                break

            page += 1
        else:
            print(f"Erro na página {page} da SBC: {response.status_code}")
            break

    return events
```

Fonte: Elaborado pela autora (2025).

Figura 15 - *Web scraping* - GitHub.

```python
# Eventos do GitHub (Agenda Tech Brasil)
def fetch_github_events():
    url = "https://raw.githubusercontent.com/agenda-tech-brasil/agenda-tech-brasil/main/README.md"
    resposta = requests.get(url)
    conteudo = resposta.text
    meses = {···
    ano = "2025"
    eventos = []
    id_counter = 100000  # IDs diferentes dos da SBC

    for mes_nome, mes_num in meses.items():
        padrao_bloco = fr"### {mes_nome.capitalize()}.*?<!-- {mes_nome.upper()}:START -->(.*?)<!-- {mes_nome.upper()}:END -->"
        match = re.search(padrao_bloco, conteudo, re.DOTALL)
        if not match:···
        bloco = match.group(1).strip().split("\n")
        for linha in bloco:
            data_evento = re.match(r"- ([0-9, e]+):", linha)
            link_titulo = re.search(r"\[([^\]]+)\]\(([^)]+)\)", linha)
            localidade = re.search(r"- _([^_]+)_", linha)
            modalidade = re.search(r"!\[([^\]]+)\]", linha)

            if data_evento and link_titulo and modalidade:···
    return eventos
```

Fonte: Elaborado pela autora (2025).

Como resultado da coleta, é gerado um arquivo .CSV unificado, contendo todos os eventos disponíveis em ambas as fontes e que ocorrerão ao longo do ano. No entanto, este arquivo bruto pode apresentar inconsistências, como dados em formatos variados, valores vazios e informações inconsistentes, o que justifica a necessidade da próxima fase da *pipeline*: o pré-processamento dos dados.

Com o arquivo .CSV consolidado, a segunda fase da *pipeline* se concentra no pré-processamento e limpeza dos dados. Esta etapa é fundamental para garantir a qualidade e a padronização das informações antes da aplicação de técnicas de PLN. As principais operações realizadas incluem:

- **Remoção de colunas desnecessárias:** Eliminação de informações que não são relevantes para o objetivo do sistema de recomendação.
- **Renomear colunas:** Padronização dos nomes das colunas para facilitar o manuseio e a consistência dos dados.
- **Preenchimento de valores vazios:** Tratamento de dados ausentes para evitar erros nas etapas subsequentes.
- **Conversão de colunas de data para formato *datetime***: Padronização das datas para permitir operações temporais e filtragens.
- **Filtragem de eventos já ocorridos:** Exclusão de eventos com datas passadas, garantindo que apenas oportunidades futuras sejam consideradas.
- **Ordenação dos eventos por data:** Organização dos dados para facilitar a visualização e o processamento sequencial.

Como resultado deste processo, obtém-se um novo arquivo .CSV limpo e organizado, com os dados prontos para a etapa de PLN, conforme exemplificado na Figura 16.

Figura 16 - DataFrame com os eventos limpos e organizados após o pré-processamento.

	name	start_date	end_date	link	location	format
44	38° Seminário de Física do Inatel (SEFITEL)	2025-06-12	2025-06-14	https://www.sbc.org.br/evento/38-seminario-de-...	Local Santa Rita do Sapucaí – MG	Presencial
45	UNIVERSO TOTVS 2025	2025-06-17	2025-06-17	https://eventos.totvs.com/event/universo-totvs...	São Paulo/SP	Presencial
46	Python Nordeste 2025	2025-06-20	2025-06-20	https://2025.pythonnordeste.org/	Teresina/PI	Presencial
47	cTENcf Santa Catarina: 10 Anos de Cloud Native	2025-06-25	2025-06-25	https://community.cncf.io/events/details/cncf-...	Florianópolis/SC	Presencial
48	Simpósio de Revolução Digital 2025: Concretiza...	2025-06-26	2025-06-27	https://www.sbc.org.br/evento/simposio-de-revo...	Florianópolis/SC	Presencial
...	...	...	...	...	...	...
106	TDC Summit Brasília	2025-11-26	2025-11-26	https://thedevconf.com/tdc/2025/summit-brasilia/	Brasília/DF	Híbrido
107	GambiConf	2025-11-29	2025-11-29	https://gambiconf.dev/	São Paulo/SP	Híbrido
108	XXVIII Simpósio Brasileiro de Métodos Formais ...	2025-12-02	2025-12-05	https://www.sbc.org.br/evento/xxviii-simposio-...	Recife/PE	Presencial
109	XXII Simpósio Brasileiro de Sistemas da Inform...	2026-05-18	2026-05-22	https://www.sbc.org.br/evento/xxii-simposio-br...	Vitória/ES	Presencial
110	XXIII Simpósio Brasileiro de Sistemas da Infor...	2027-05-17	2027-05-21	https://www.sbc.org.br/evento/xxi-simposio-bra...	Campo Grande/MS	Presencial

69 rows × 6 columns

Fonte: Elaborado pela autora (2025).

Antes de prosseguir para a etapa de Processamento de Linguagem Natural (PLN), é realizado um conjunto adicional de operações de pré-processamento específico para otimizar a qualidade textual dos dados. Este passo prepara o texto para uma análise linguística mais eficaz, que compreende:

- **Remoção de acentos:** Padronização do texto para facilitar a correspondência e análise, eliminando variações de caracteres.
- **Normalização para minúsculas:** Conversão de todo o texto para minúsculas, evitando que a mesma palavra seja tratada como diferente devido à capitalização (ex: "Python" e "python").
- **Tokenização:** Divisão do texto em unidades menores (tokens), como palavras e frases, que serão as unidades básicas para a análise subsequente.
- **Filtragem de *stopwords*:** Remoção de palavras comuns (como artigos, preposições e conjunções) que possuem baixo valor semântico para a análise, utilizando uma lista adaptada para o contexto de eventos de tecnologia.
- **Stemming:** Aplicação do algoritmo RSLP para reduzir as palavras às suas raízes (lemmas), otimizando a análise e agrupando variações morfológicas de um mesmo termo (ex: "programação", "programador" para "program").

A etapa final envolve a aplicação de diversas técnicas de PLN para extrair significado e categorizar os eventos. Para isso, utiliza-se a biblioteca spaCy, que oferece suporte robusto para tarefas de PLN. As técnicas empregadas incluem:

- **Extração de Entidades Nomeadas (Named Entity Recognition - NER):** Utilização do spaCy para identificar e classificar entidades nos nomes e descrições dos eventos, como nomes de organizações, locais e datas.
- **Classificação Temática Automática:** Implementação de uma abordagem baseada em palavras-chave pré-definidas e agrupadas por categorias de interesse (e.g., Desenvolvimento de Software, Inteligência Artificial, Segurança da Informação). A similaridade entre os nomes dos eventos e essas palavras-chave é avaliada por meio de uma pontuação heurística, considerando tanto termos compostos quanto suas partes individuais.
- **Vetorização TF-IDF (Term Frequency-Inverse Document Frequency):** Transformação dos textos dos eventos em representações numéricas (vetores), permitindo a aplicação de algoritmos de *machine learning*.
- ***Clustering* K-Means:** Aplicação do algoritmo K-Means para agrupar eventos com base na similaridade textual de seus vetores TF-IDF, identificando automaticamente clusters de temas semelhantes.
- **Extração de Palavras-Chave:** Utilização de TF-IDF ou, alternativamente, LDA (Latent Dirichlet Allocation) para identificar as palavras-chave mais representativas de cada documento ou tópico, fornecendo *insights* sobre o conteúdo dos eventos.

Figura 17 - Clusterização

```python
# Função para clustering de eventos usando K-means
def clustering_eventos(df, coluna_nome='name', n_clusters=5):
    """Agrupa eventos em clusters usando K-means"""
    # Preprocessar os nomes dos eventos - sem stemming para melhor interpretabilidade
    textos_processados = df[coluna_nome].apply(lambda x: preprocessar_texto(x, stemming=False))
    # Vetorizar os textos usando TF-IDF
    tfidf = TfidfVectorizer(max_features=1000, min_df=2)
    # Verificar se há textos válidos
    textos_validos = [texto for texto in textos_processados if texto.strip()]
    if not textos_validos: ···

    X = tfidf.fit_transform(textos_validos)

    # Verificar se temos dados suficientes
    if X.shape[0] < n_clusters: ···

    # Aplicar K-means
    kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10)
    cluster_indices = kmeans.fit_predict(X)

    # Mapear os índices dos documentos válidos para os clusters
    cluster_map = {}
    valid_idx = 0
    for i, texto in enumerate(textos_processados): ···

    df['cluster'] = df.index.map(lambda i: cluster_map.get(i, -1))

    # Identificar palavras-chave para cada cluster
    feature_names = tfidf.get_feature_names_out()
    palavras_chave_clusters = {}

    for i in range(n_clusters): ···

    return df, palavras_chave_clusters
```

Fonte: Elaborado pela autora (2025).

Com os dados processados e classificados, o resultado final da *pipeline* é consolidado em um novo arquivo .CSV. Este arquivo estruturado contém as seguintes colunas: *name, start_date, end_date, link, location, format, categoria, cluster, e palavras_chave.* É a partir deste arquivo que os eventos e oportunidades relevantes serão selecionados para compor o conteúdo da newsletter, sendo posteriormente encaminhado e incorporado pelo *backend* da aplicação para envio aos usuários.

3.3.2.2 *Backend*

O *backend* do sistema Congraduations foi desenvolvido com base nas funcionalidades definidas nas seções anteriores, utilizando o *framework* ASP.NET Core para a construção da API e a IDE JetBrains Rider para o desenvolvimento. Para viabilizar a operacionalização da aplicação, foram implementados 11 *scripts* distintos, conforme detalhado no Quadro 1.

Dentre eles, o *MonthlyNewsletterService* se destaca como o principal responsável pelo funcionamento central do Congraduations, onde também há a

utilização do arquivo .CSV gerado na última etapa da *pipeline* de PLN como fonte de dados.

A API RESTful desenvolvida disponibiliza endpoints específicos para o cadastro, verificação e cancelamento de e-mails, além de uma rota interna dedicada à integração com módulo de recomendação, permitindo a automação do envio de conteúdo relevante aos seus usuários cadastrados.

Quadro 1 - Scripts desenvolvidos no backend da aplicação Congraduations

Script	Descrição
Subscriber.cs	Define o modelo de dados do assinante, com campos para e-mail, confirmação, cancelamento e datas de registro.
TechEvents.cs	Representa os eventos, com informações como título, data, formato, link, local e área temática.
Education.cs	Representa as oportunidades de estudo.
NewsletterRegistrationRequest.cs	Define o DTO utilizado para receber e validar o e-mail no momento do cadastro do usuário na newsletter.
TechEventsCsvModel.cs	Mapeia os dados de eventos tecnológicos importados de arquivos CSV, associando colunas a propriedades do sistema.
NewsletterController.cs	Controla o fluxo de cadastro, confirmação e cancelamento da newsletter. Gerencia tokens de confirmação e envio de e-mails.
CsvImportService.cs	Realiza a importação de eventos a partir de arquivos CSV, convertendo os dados em objetos TechEvent e salvando no banco.
EmailService.cs	Responsável pelo envio de e-mails de confirmação e newsletters, utilizando SMTP com HTML personalizado e dados do arquivo de config.
MonthlyNewsletterService.cs	Gera e envia mensalmente newsletters personalizadas com eventos tech e oportunidades do mês para assinantes confirmados, incluindo link de cancelamento.
ScheduledNewsletterService.cs	Serviço em segundo plano que executa o envio automático da newsletter em intervalos definidos, utilizando o serviço principal
Program.cs	Configura e executa a aplicação web, registrando os serviços, controladores, middleware e executando a importação inicial de

	eventos.

Fonte: Elaborado pela autora (2025).

No desenvolvimento da aplicação, a discussão se concentrará nos scripts mais relevantes para o funcionamento do sistema, com foco na sua construção e operação. Entre eles, o *NewsletterController* desempenha um papel central nas operações de gestão de assinaturas, como inscrição, confirmação e cancelamento.

O script NewsletterController é responsável pelas operações relacionadas a inscrição, confirmação e cancelamento da newsletter. Este controlador interage diretamente com o banco de dados via AppDbContext e utiliza o EmailService para o envio de comunicação aos usuários.

A funcionalidade de registro de novos assinantes é implementada no método Register, que lida com diversos cenários de forma robusta. Quando um usuário tenta se inscrever, o sistema verifica se o e-mail já existe na base de dados. As ações tomadas variam conforme o status do e-mail:

- **E-mail existente e cancelado:** O sistema reativa a inscrição, gera um novo token de confirmação e envia um e-mail para que o usuário possa confirmar a reativação.
- **E-mail existente e ativo:** É retornada uma mensagem de erro, informando que a inscrição já está ativa.
- **E-mail existente e não confirmado:** Um novo token de confirmação é gerado e reenviado, oferecendo ao usuário uma nova chance de validar sua inscrição.
- **E-mail novo:** Um novo assinante é criado no banco de dados com um token de confirmação único. Em seguida, o processo de envio do e-mail de confirmação é iniciado, formalizando a inscrição.

A integração entre o *NewsletterController* e o *MonthlyNewsletterService* não apenas possibilita a gestão eficiente das inscrições, mas também assegura a execução de rotinas automatizadas para a distribuição da newsletter, reduzindo falhas manuais e garantindo que a base de usuários receba informações relevantes e atualizadas. Além disso, o fluxo entre essas duas camadas forma a espinha dorsal da interação do sistema com a comunidade acadêmica, uma vez que centraliza tanto a manutenção dos dados dos inscritos quanto a entrega dos conteúdos recomendados. A confiabilidade desse processo é essencial para a credibilidade do sistema, visto que a proposta da aplicação depende diretamente da regularidade e da pertinência das comunicações enviadas aos alunos do curso de TADS.

Figura 18 - Exemplo do Código NewsletterController

```
namespace Newsletter.Controllers
{
    [ApiController]
    [Route( template: "api/[controller]")]
    public class NewsletterController : ControllerBase
    {
        private readonly AppDbContext _context;
        private readonly EmailService _emailService;
        private readonly IConfiguration _configuration;

        public NewsletterController(AppDbContext context, EmailService emailService, IConfiguration configuration){...}

        // Método auxiliar para gerar token
        private string GenerateConfirmationToken(string email)
        {
            var secretKey :string? = _configuration["TokenSettings:SecretKey"];
            using var md5 = MD5.Create();
            var inputBytes :byte[] = Encoding.ASCII.GetBytes( s: email + secretKey);
            var hashBytes :byte[] = md5.ComputeHash(inputBytes);
            var sb = new StringBuilder();
            foreach (var b :byte in hashBytes)
                sb.Append(b.ToString( format: "X2"));
            return sb.ToString();
        }
```

Fonte: Elaborado pela autora (2025).

Já o script *MonthlyNewsletterService,* é um componente crucial no backend, responsável pela orquestração do envio periódico das newsletters aos assinantes. Este serviço utiliza o AppDbContext para acessar os dados dos eventos da área tech (TechEvents) e as informações dos assinantes (Subscribers), e o EmailService para, de fato, despachar os e-mails.

O método principal, SendMonthlyNewsletter(), é invocado para iniciar o processo de envio. Ele primeiro consulta o banco de dados para recuperar todos os eventos e oportunidades agendadas para o mês atual. Em seguida, busca todos os assinantes ativos e confirmados. Para cada assinante, o serviço constrói uma URL de cancelamento de inscrição única.

A construção do conteúdo HTML da *newsletter* é encapsulada no método auxiliar BuildHtml(). Este método gera dinamicamente o corpo do e-mail, que inclui um cabeçalho com o logotipo do Congraduations, uma introdução sobre o objetivo da *newsletter*, e uma seção detalhada com os eventos do mês. Para cada evento, são exibidas informações como nome (com link para o evento original), datas de início e fim, formato (presencial, online ou híbrido) e local. Caso não haja eventos no mês, uma mensagem apropriada é exibida. O HTML também incorpora um *link* para

cancelamento de inscrição e informações de contato, além de seguir os padrões de design estabelecidos nos tópicos anteriores, como mostra a figura 19.

Figura 19 - Exemplo do HTML da Neswletter

```
private string BuildHtml(List<TechEvent> events, string unsubscribeUrl)
{
    var sb = new StringBuilder();

    sb.AppendLine("<!DOCTYPE html>");
    sb.AppendLine("<html lang=\"pt-BR\">");
    sb.AppendLine("<head>");
    sb.AppendLine("  <meta charset=\"UTF-8\" />");
    sb.AppendLine("  <title>Congraduations - Newsletter de Eventos Tech</title>");
    sb.AppendLine("  <link href=\"https://fonts.googleapis.com/css2?family=JetBrains+Mono&display=swap\" rel=\"stylesheet\" />");
    sb.AppendLine("</head>");
    sb.AppendLine("<body style=\"font-family: 'JetBrains Mono', monospace; margin: 0; padding: 0; background-color: #f8f8f8; color: #222;\">");
    sb.AppendLine("  <div style=\"text-align: center; padding: 20px;\">");
    sb.AppendLine("<img src=\"https://drive.google.com/uc?export=view&id=1SxwZyeYzBCi9AcTbGVJohukm_PDE2JRh\" alt=\"Eventos Tech\" " +
                  "style=\"width: 100%; max-width: 600px; height: auto; display: block; margin: 0 auto; border-radius: 6px;\" />");
    sb.AppendLine("  </div>");
    sb.AppendLine("  <div style=\"padding: 30px;\">");
    sb.AppendLine("    <h1 style=\"font-size: 24px;\">Congraduations - Newsletter de Eventos Tech</h1>");
    sb.AppendLine("    <p>Seu guia para os eventos tecnológicos mais relevantes que vão impulsionar sua carreira e ampliar seu conhecimento.</p>");
    sb.AppendLine("    <p>Descubra oportunidades incríveis para networking, aprendizado e inovação no universo da tecnologia. " +
                      "Prepare-se para participar dos melhores encontros do mês!</p>");
```

Fonte: Elaborado pela autora (2025).

Finalmente, para cada assinante, o EmailService é acionado para enviar a *newsletter* gerada (html) com o assunto "Eventos Tech deste mês", garantindo que as recomendações cheguem diretamente à caixa de entrada dos estudantes interessados. Este serviço automatiza a distribuição de conteúdo relevante, minimizando a intervenção manual e garantindo a periodicidade das informações.

No fluxo de dados do Congraduations, o arquivo .CSV gerado pela *pipeline* em Python precisa ser integrado ao *backend* para que os dados dos eventos e oportunidades fiquem disponíveis para os scripts mencionados anteriormente. Essa integração é realizada por meio do serviço CsvImportService.

O CsvImportService é responsável por ler o arquivo .CSV contendo os dados processados dos eventos e oportunidades e persisti-los no banco de dados da aplicação. Para cada registro lido, o serviço realiza as seguintes verificações e operações:

- **Conversão de Datas:** As datas de início (StartDate) e fim (EndDate) são convertidas de *string* para o tipo DateTime.
- **Verificação de Duplicidade:** É realizada uma verificação no banco de dados para garantir que o evento não seja inserido novamente caso já exista, evitando redundâncias. A duplicidade é avaliada com base no nome do evento, data de início e data de fim.
- **Normalização de Campos:** Antes de persistir, campos como Location e Format são normalizados. Se Location for vazio ou em branco, é definido

como "Online". O método auxiliar CleanFormat() padroniza os valores do formato para "Presencial", "Online" ou "Híbrido", garantindo consistência.

Caso o evento ainda não exista no banco de dados, uma nova instância de TechEvent é criada com os dados limpos e padronizados, e então adicionada ao contexto. Finalmente, as alterações são salvas no banco de dados tornando os eventos acessíveis para outras partes da aplicação, como o MonthlyNewsletterService responsável pelo envio das *newsletters*.

Figura 20 - Exemplo do código de CsvImportService.

```
public class CsvImportService
{
    private readonly AppDbContext _context;

    public CsvImportService(AppDbContext context){...}

    public void Import(string filePath)
    {
        using var reader = new StreamReader(filePath);
        using var csv = new CsvReader(reader, CultureInfo.InvariantCulture);
        var records :List<TechEventCsvModel> = csv.GetRecords<TechEventCsvModel>().ToList();

        foreach (var record :TechEventCsvModel in records)
        {
            var startDate = DateTime.Parse(record.StartDate);
            var endDate = DateTime.Parse(record.EndDate);

            // Verifica se o evento já existe
            bool exists = _context.TechEvents.Any(e :TechEvent =>
                e.Name == record.Name &&
                e.StartDate == startDate &&
                e.EndDate == endDate
            );

            if (exists) continue;
```

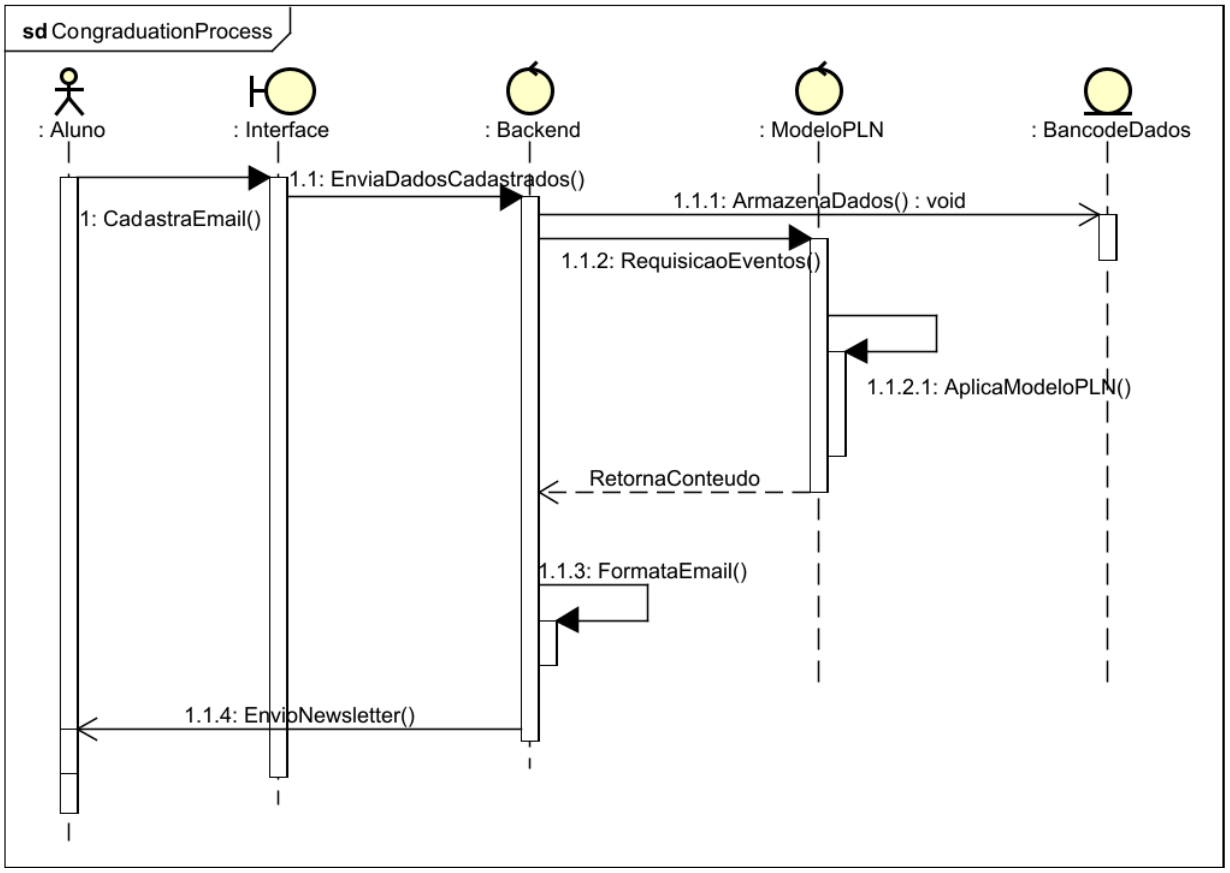
Fonte: Elaborado pela autora (2025).

A integração coesa de todos os componentes do backend é fundamental para o funcionamento eficaz do Congraduations, permitindo uma comunicação fluida e ininterrupta entre as diversas partes do sistema. Essa sinergia garante que cada módulo trabalhe em conjunto, orquestrando as operações necessárias para entregar as funcionalidades propostas.

Para oferecer uma compreensão clara e detalhada da sequência de interação entre os atores (como o usuário) e a aplicação, especialmente em suas operações mais críticas e fundamentais, foi desenvolvido um diagrama de sequência. Este diagrama, apresentado na Figura 21, ilustra visualmente o fluxo de mensagens e as interações temporais para processos essenciais, como o cadastro de um novo usuário na plataforma e o subsequente processo de confirmação de sua conta. Ele desmistifica a complexidade das interações, mostrando passo a passo como cada

componente do backend responde e se comunica para completar essas ações chave.

Figura 21 - diagrama de sequência



Fonte: Elaborado pela autora (2025)

3.3.2.3 *Frontend*

Como dito anteriormente, o frontend foi desenvolvido em React.js, com o propósito de oferecer uma interface de usuário intuitiva, responsável e acessível. A escolha por *styled-components* para a estilização reflete a preocupação com a modularidade e a manutenibilidade do código CSS, permitindo que os estilos estejam definidos diretamente nos componentes React. A comunicação com o *backend* é gerenciada pela biblioteca *axios*, facilitando as requisições HTTP para o registro e a interação com a API de newsletter.

A estrutura visual da aplicação é dividida em seções lógicas, conforme evidenciado pelos componentes estilizados: *HeaderDiv* para o cabeçalho, *MidDiv* que segrega a página em *LeftDiv* (contendo o título principal, textos introdutórios e o formulário de cadastro de e-mail) e *RightDiv* (para elementos gráficos e uma caixa

de texto explicativa sobre o projeto). A área inferior, BaseBoard, apresenta cards (CardDiv) que categorizam os temas abordados pela newsletter, como Desenvolvimento de Software, Inteligência Artificial, Segurança da Informação, DevOps e Gestão de Projetos, demonstrando a amplitude de conteúdo oferecido.

A interatividade é centralizada no formulário de cadastro. O campo EmailInput permite a entrada do e-mail institucional, enquanto o SendButton dispara a função handleSubmit ao ser clicado. Esta função gerencia o estado da aplicação, exibindo mensagens de feedback (Message) em tempo real, indicando o sucesso ou falha do cadastro, e ativando um spinner durante o processamento da requisição, otimizando a experiência do usuário. Em caso de sucesso, um ModalOverlay com um ModalContent é exibido, confirmando o cadastro e instruindo o usuário a verificar o e-mail para a confirmação final.

A aplicação utiliza useState para gerenciar o estado dos componentes, controlando o valor do campo de e-mail, as mensagens de feedback, a visibilidade do modal de sucesso e o estado de carregamento. Recursos visuais como logoImage, iconHomePage e iconHomePage2 são importados e utilizados para enriquecer a identidade visual e o apelo estético da interface. A responsividade é garantida pelas propriedades CSS display: flex e % para larguras e alturas, permitindo que o layout se adapte fluidamente a diferentes tamanhos de tela.

Figura 22 - Exemplo do código do frontend com styled-components.

congraduations_front > src > App.jsx > App

```
1   import styled, { ThemeProvider } from 'styled-components';
2   import { useState } from 'react';
3   import axios from 'axios';
4
5   import { HeaderTitle } from './components/HeaderTitle';
6   import { theme } from './styles/theme';
7   import { GlobalStyle } from './styles/GlobalStyles';
8
9   import logoImage from './assets/Title.png';
10  import iconHomePage from './assets/advice5.png';
11  import iconHomePage2 from './assets/sender5.png';
12
13  const Container = styled.div`
14    background-color: ${({ theme }) => theme.colors.greenBg};
15    font-family: 'JetBrains Mono', monospace;
16    height: 100%;
17    display: flex;
18    flex-direction: column;
19  `;
```

Fonte: Elaborado pela autora (2025).

CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo desenvolver o sistema Congraduations, uma aplicação voltada à recomendação de eventos acadêmicos e oportunidades de desenvolvimento educacional para estudantes do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal do Amazonas (IFAM-CMC). A proposta surgiu a partir da percepção da ausência de estrutura e estímulos à participação discente em congressos, submissão científica e continuidade na pós-graduação dentro das salas de aula, lacuna essa que o sistema busca minimizar por meio da tecnologia.

Para tanto, os objetivos de identificar e mapear as necessidades dos estudantes, projetar arquitetura utilizando técnicas de Processamento de Linguagem Natural (PLN) e desenvolver um protótipo funcional que faça utilização de envio de newsletter foram delineados, e para alcançá-los, foram desenvolvidos etapas que incluíram: levantamento do problema a partir da vivência acadêmica, revisão da literatura sobre sistemas de recomendação e PLN, definição do modelo metodológico com base no ciclo de vida em cascata, modelagem da solução, implementação do frontend em React.js e do backend em ASP.NET Core, construção do módulo de recomendação com Python, além da configuração de rotinas automatizadas para envio das newsletters com base em dados processados a partir de arquivos CSV.

Os resultados obtidos demonstram que é possível, por meio da integração entre técnicas de Processamento de Linguagem Natural e sistemas de recomendação, oferecer aos estudantes sugestões personalizadas de eventos acadêmicos, utilizando um fluxo automatizado, simples e de fácil adesão. A partir da análise realizada durante a implementação e os testes iniciais do protótipo, constatou-se que a estrutura proposta atende aos principais requisitos propostos para o Congraduations: o sistema é capaz de coletar e-mails institucionais dos estudantes, confirmar automaticamente sua inscrição por meio de tokens únicos, e enviar newsletters personalizadas com recomendações de eventos acadêmicos regionais e nacionais, e outras oportunidades, utilizando como base um processo automatizado de análise textual via técnicas de PLN. O módulo de recomendação, por sua vez, permite a extração de informações relevantes dos eventos, com base

em arquivos estruturados em formato CSV, e aplica técnicas como TF-IDF para classificar e organizar os conteúdos a serem enviados.

Essas descobertas contribuem tanto em termos práticos, ao oferecer uma ferramenta funcional que pode ser aplicada em realidades semelhantes à do IFAM (até mesmo dentro da instituição, como em outros cursos), quanto metodológico, ao propor uma solução tecnológica replicável para incentivar o engajamento acadêmico em cursos de graduação de tecnologia. O sistema Congraduations mostra que, mesmo em contextos onde há carência de incentivo institucional à pesquisa e à extensão, é viável desenvolver soluções acessíveis que promovam a aproximação do estudante com o meio científico e a continuidade na vida acadêmica.

Contudo, é importante reconhecer que o presente projeto possui algumas limitações. A principal delas está relacionada à ausência de um mecanismo de coleta direta das preferências individuais dos estudantes. Atualmente, o sistema Congraduations opera com um modelo de recomendação generalista, que envia conteúdos acadêmicos e eventos de forma uniforme para todos os usuários cadastrados, independentemente de seus interesses específicos. Além disso, o sistema possui dependência de arquivos CSV como fonte de dados, o que exige a curadoria manual de eventos acadêmicos em sua fase inicial. A ausência de uma camada de autenticação, personalização por perfil acadêmico e *feedbacks* do sistema também representam um ponto a ser melhorado em versões futuras.

Como propostas de continuidade, recomenda-se a integração do Congraduations com bases de dados acadêmicas externas, como periódicos, plataformas de eventos científicos e diretórios como Lattes e Scielo. Também se sugere a utilização de modelos de NLP mais robustos, como embeddings semânticos (Word2Vec, BERT) ou classificadores baseados em aprendizado profundo, para melhorar a qualidade da recomendação. A criação de dashboards interativos para professores e coordenadores e a expansão do sistema para outros campus do IFAM ou cursos superiores de tecnologia podem ampliar seu alcance e relevância.

Conclui-se, portanto, que o desenvolvimento do sistema Congraduations representa um passo importante na promoção do engajamento acadêmico de estudantes de TADS. Por meio da tecnologia, torna-se possível conectar alunos a oportunidades que antes passavam despercebidas, contribuindo não apenas para

sua formação profissional, mas também para a consolidação de uma cultura institucional voltada à pesquisa, extensão e continuidade na vida acadêmica.

REFERÊNCIAS

AAMIR, M. BHUSRY, M. Recommendation system: state of the art approach. **International Journal of Computer Applications**, v. 120, n. 12, p. 25-32, 2015.

AGGARWAL, Charu C. **Recommender Systems: The Textbook**. Cham: Springer International Publishing, 2016.

BARDAGI, M. P.; HUTZ, C. S. A atividade extracurricular no contexto universitário: contribuições para o desenvolvimento de carreira. **Psicologia: Teoria e Pesquisa**, v. 28, n. 2, p. 193–200, 2012.

BIRD, Steven; KLEIN, Ewan; LOPER, Edward. **Natural Language Processing with Python**: Analyzing Text with the Natural Language Toolkit. Beijing: O'Reilly Media, 2009. Disponível em: https://www.nltk.org/book/. Acesso em: 25 jun. 2025.

BRASIL. **Lei de Diretrizes e Bases da Educação Nacional**, nº 9394 de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional. Artigo 43. Disponível em: https://www.planalto.gov.br/ccivil_03/leis/l9394.htm. Acesso em: 03 dez. 2024.

BRASIL. **Plano Nacional de Educação – PNE 2014–2024**: Lei n. 13.005/2014. Brasília: MEC, 2014.

BRASIL. Ministério da Educação. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep). **Censo da Educação Superior 2023**: resumo técnico. Brasília: MEC/Inep, 2023. Disponível em: https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-escolar/resultados. Acesso em: 12 dez. 2024.

BURKE, Robin. Hybrid Recommender Systems: Survey and Experiments. **User Modeling and User-Adapted Interaction**, v. 12, n. 4, p. 6-25, 2002.

CUNHA, Luiz Antônio. **A universidade temporã**: o ensino superior da colônia à era de Lula. São Paulo: UNESP, 2010.

FIOR, C. A.; MERCURI, E. Formação universitária e flexibilidade curricular: Importância das atividades obrigatórias e não obrigatórias. **Psicologia da Educação**, v. 29, n. 2, p. 191-215, 2009. Disponível em: http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1414-69752009000200010. Acesso em: 30 jun. 2025.

GOLDBERG, Y. A Primer on Neural Network Models for Natural Language Processing. **Journal of Artificial Intelligence Research**, v. 57, p. 345–420, 2016.

HU, Shouping; WOLNIAK, Gregory C. College student engagement and early career earnings: Differences by gender, race/ethnicity, and academic preparation. **Review of Higher Education**, v. 36, n. 2, p. 211-233, 2013.

JAIN, A. K. Data clustering: 50 years beyond K-means. **Pattern Recognition Letters**, v. 31, n. 8, p. 651–666, 2010.

JURAFSKY, Daniel; MARTIN, James H. **Speech and Language Processing**: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 3. ed. Draft. [S. l.]: Stanford University, 2023. Disponível em: https://web.stanford.edu/~jurafsky/slp3/. Acesso em: 25 jun. 2025.

KIM, S. W.; GIL, J. M. Research paper classification systems based on TF-IDF and LDA schemes. **Human-Centric Computing and Information Sciences**, v. 9, n. 30, 2019. Disponível em: https://doi.org/10.1186/s13673-019-0192-7. Acesso em: 30 jun. 2025.

KOLB, D. A. **Experiential Learning**: Experience as the Source of Learning and Development. Englewood Cliffs: Prentice Hall, 1984.

LOPS, P.; GEMMIS, M. de; SEMERARO, G. Content-based Recommender Systems: State of the Art and Trends. In: RICCI, F.; ROKACH, L.; SHAPIRA, B.; KANTOR, P. (org.). **Recommender Systems Handbook**. Boston, MA: Springer, 2011. p. 69-104. Disponível em: https://doi.org/10.1007/978-0-387-85820-3_3. Acesso em: 30 jun. 2025.

MACQUEEN, J. B. Some Methods for Classification and Analysis of Multivariate Observations. In: PROCEEDINGS OF THE FIFTH BERKELEY SYMPOSIUM ON MATHEMATICAL STATISTICS AND PROBABILITY, 1967.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. **Introduction to Information Retrieval**. Cambridge: Cambridge University Press, 2008.

MANNING, Christopher D.; SCHÜTZE, Hinrich. **Foundations of Statistical Natural Language Processing**. Cambridge, MA: MIT Press, 1999.

MENEZES, L. C. de. Políticas de formação de professores: a universidade em questão. In: LISITA, V. M. S. S. (Org.). **Formação de professores**: políticas, concepções e perspectivas. Goiânia: Alternativa, 2001. p. 35-41.

MIKOLOV, T. et al. Efficient Estimation of Word Representations in Vector Space. **arXiv preprint arXiv:1301.3781**, 2013.

MOITA, F. M. G. Da S. C.; ANDRADE, F. C. B. de. Ensino-pesquisa-extensão: um exercício de indissociabilidade na pós-graduação. **Revista Brasileira de Educação**, v. 14, n. 41, maio/ago. 2009.

NAREN, J.; BANU, M. Z.; LOHAVANI, S. Recommendation system for students' course selection. In: SMART SYSTEMS AND IOT: INNOVATIONS IN COMPUTING, 2020. p. 825–834.

OLIVEIRA, M. A.; FERNANDES, M. C. S. G. A atividade discente na universidade: caracterização dos estudantes e impactos da produtividade acadêmica. **Revista Ibero-Americana de Estudos em Educação**, Araraquara, v. 11, n. 3, p. 1423–1440, 2016. DOI: 10.21723/riaee.v11.n3.7179. Disponível em: https://periodicos.fclar.unesp.br/iberoamericana/article/view/7179. Acesso em: 16 maio. 2025.

PERES, Rafael Bruno; OLIVEIRA, Joanne Romão de; MARINHO, Marlon Glauber; MARCHINI, Jhonny Alencar. Ensino, pesquisa e extensão: bases para a formação integral na educação profissional e tecnológica. **Revista Científica Multidisciplinar Núcleo do Conhecimento**, Ano 07, Ed. 12, Vol. 04, pp. 36-51, Dezembro de 2022. DOI: 10.32749/nucleodoconhecimento.com.br/educacao/formacao-integral. Disponível em: https://www.nucleodoconhecimento.com.br/educacao/formacao-integral. Acesso em: 16 maio. 2025.

PRESSMAN, R. S. **Engenharia de Software**: Uma abordagem profissional. 7. ed. São Paulo: McGraw-Hill, 2009.

REATEGUI, Eliseo Berni; CAZELLA, Sílvio César. Sistemas de recomendação. In: **XXV Congresso da Sociedade Brasileira de Computação**. 2005. p. 306-348.

RESNICK, Paul; VARIAN, Hal R. Recommender systems. **Communications of the ACM**, v. 40, n. 3, p. 56-59, mar. 1997.

ROYCE, W. W. **Managing the Development of Large Software Systems**. In: PROCEEDINGS OF IEEE WESCON, 1970. p. 1–9.

SANTOS, B. S.; ALMEIDA, J. P. **Ciência, Tecnologia e Sociedade**: Uma Perspectiva Histórica. Lisboa: Edições 70, 2018.

SARWAR, Badrul et al. Item-based collaborative filtering recommendation algorithms. In: PROCEEDINGS OF THE 10TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB. Hong Kong: ACM, 2001. p. 285-295.

SEMESP. **Mapa do Ensino Superior no Brasil**: 13ª edição. São Paulo: Semesp, 2021. Disponível em: https://www.semesp.org.br/mapa/edicao-13/. Acesso em: 10 dez. 2024.

SEVERINO, A. J. **Metodologia do Trabalho Científico**. São Paulo: Cortez, 2012.

SHAHBAZI, Z.; BYUN, Y.-C. Agent-Based Recommendation in E-Learning Environment Using Knowledge Discovery and Machine Learning Approaches. **Mathematics**, v. 10, n. 7, p. 1192, 2022. Disponível em: https://www.mdpi.com/2227-7390/10/7/1192. Acesso em: 25 jan. 2025.

SILVA, Mirna Ribeiro Lima da; TREVIZAN MISSAKI, Andressa Christina; BUENO, Belmira Amélia de Barros Oliveira. Formação, trabalho e carreira: a docência nos Institutos Federais de Educação, Ciência e Tecnologia. **Revista Internacional de Formação de Professores**, Itapetininga, v. 7, p. e022005, 2022. Disponível em: https://periodicoscientificos.itp.ifsp.edu.br/index.php/rifp/article/view/710. Acesso em: 30 jun. 2025.

SILVA, R. R.; LIMA, S. M. B. **Consultas em Bancos de Dados Utilizando Linguagem Natural**. [S. l.: s. n.].

STEVENSON, J.; CLEGG, S. Possible selves: Students orientating themselves towards the future through extracurricular activity. **British Educational Research Journal**, v. 37, n. 2, p. 231–246, 2011. DOI: 10.1080/01411920903540672.

VASCONCELOS, M. L. M. C. **A formação do professor de 3º Grau**. São Paulo: Pioneira, 1996.