



UNIVERSIDADE ESTADUAL DO PIAUÍ
CAMPUS DRA. JOSEFINA DEMES
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

BRUNA CRISTINA BRANDÃO E SILVA

**MONITRACK: SISTEMA DE GERENCIAMENTO DE OPORTUNIDADES DE
MONITORIA NA UNIVERSIDADE ESTADUAL DO PIAUÍ COM DISTRIBUIÇÃO
EQUILIBRADA DE VAGAS**

FLORIANO

2025

BRUNA CRISTINA BRANDÃO E SILVA

MONITRACK: SISTEMA DE GERENCIAMENTO DE OPORTUNIDADES DE
MONITORIA NA UNIVERSIDADE ESTADUAL DO PIAUÍ COM DISTRIBUIÇÃO
EQUILIBRADA DE VAGAS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciências da Computa-
ção da Universidade Estadual do Piauí, como
requisito parcial à obtenção do grau de bacharel
em Ciências da Computação.

Orientadora: Suzana Matos França de
Oliveira.

FLORIANO

2025

Monitrack: Sistema de Gerenciamento de Oportunidades de Monitoria na Universidade Estadual do Piauí com Distribuição Equilibrada de Vagas

Bruna Cristina Brandão e Silva¹, Suzana Matos França de Oliveira¹

¹ Ciência da Computação – Universidade Estadual do Piauí (UESPI)
Florianópolis – PI – Brasil

brunachbesilva@aluno.uespi.br, suzana.matos@frn.uespi.br

Abstract. *Academic monitoring is an essential activity in supporting the learning process in higher education. However, the unequal distribution of interested students among disciplines can compromise the use of available positions and the effectiveness of the program. This work presents the development of Monitrack, a web-based system designed to manage academic monitoring opportunities at the State University of Piauí (Universidade Estadual do Piauí - UESPI), with the goal of balancing the distribution of vacancies. The system was built using technologies such as Java, Spring Boot, Thymeleaf, and MySQL, adopting the MVC architecture. Functionalities such as user registration with different access profiles, expression and cancellation of student interest, visualization of competition among disciplines, and report generation were implemented. The results demonstrated that the system is functional and feasible for managing academic monitoring activities.*

Keywords: *Web system, academic monitoring, spring boot.*

Resumo. *A monitoria acadêmica é uma atividade essencial no apoio ao processo de aprendizagem no ensino superior. No entanto, a distribuição desigual de interessados entre as disciplinas pode comprometer o aproveitamento das vagas e a eficácia do programa. Este trabalho apresenta o desenvolvimento do Monitrack, um sistema web voltado para o gerenciamento de oportunidades de monitoria na Universidade Estadual do Piauí (UESPI), com o objetivo de equilibrar a distribuição das vagas. O sistema foi construído utilizando tecnologias como Java, Spring Boot, Thymeleaf e MySQL, adotando a arquitetura MVC. Funcionalidades como cadastro de usuários com perfis distintos, manifestação e cancelamento de interesse pelos alunos, visualização da concorrência entre disciplinas e geração de relatórios foram implementadas. Os resultados demonstraram que o sistema é funcional e viável para a gestão das monitorias.*

Palavras-chave: *Sistema web, monitoria acadêmica, spring boot.*

1. Introdução

A monitoria acadêmica desempenha um papel fundamental no apoio pedagógico aos estudantes universitários, ao promover oportunidades de aprofundamento de conteúdo, desenvolvimento da autonomia e estímulo à colaboração entre alunos e professores (VICENZI et al., 2016). Como estratégia de ensino complementar, a monitoria contribui significativamente para o processo formativo dos discentes, tanto monitores quanto monitorados

(NETO; ANDRADE, 2016). No entanto, para que seu impacto seja positivo, é necessário que o processo de distribuição de vagas seja equilibrado de maneira justa e eficiente.

Na Universidade Estadual do Piauí (UESPI), o gerenciamento das vagas de monitoria frequentemente enfrenta dificuldades relacionadas à distribuição desigual de candidatos entre as disciplinas. Algumas áreas apresentam alta concorrência, enquanto outras permanecem com vagas ociosas. Essa má distribuição compromete o aproveitamento das bolsas disponíveis, podendo afetar negativamente a qualidade do apoio pedagógico, especialmente em disciplinas que não atraem candidatos para preencher a vaga.

Diante dessa problemática, este trabalho propõe o desenvolvimento do *Monitrack*, um sistema *web* voltado para o gerenciamento de oportunidades de monitoria acadêmica na UESPI. A plataforma tem como objetivo principal equilibrar a distribuição das vagas por meio da visualização de interesse dos alunos. Dessa forma, espera-se reduzir a concentração de interessados em poucas disciplinas e estimular uma distribuição mais justa entre as áreas ofertadas.

Além disso, o *Monitrack* visa facilitar o acesso às informações por parte de administradores, professores e alunos, oferecendo recursos como: cadastro de disciplinas, registro e cancelamento de interesse por parte dos discentes, geração de relatório geral de interesses por parte dos administradores.

Este trabalho está dividido em outras quatro seções. Na Seção 2, tem-se a apresentação de sistemas e tecnologias que serviram como base para o desenvolvimento da solução proposta. A Seção 3 descreve as etapas de construção do sistema, com detalhamento dos requisitos funcionais e não funcionais, regras de negócios e diagramas de modelagem. A Seção 4 apresenta as funcionalidades implementadas, destacando como cada recurso contribui de forma prática para solucionar o problema identificado. Por fim, na Seção 5, são apresentadas a conclusão do trabalho, análise dos resultados alcançados e ideias de melhorias futuras.

2. Trabalhos Relacionados

Esta seção apresenta os trabalhos que foram analisados para fundamentar o desenvolvimento do *Monitrack*. Os mesmos abordam tanto o gerenciamento de monitorias quanto tecnologias aplicadas à construção de sistemas *web* voltados para instituições de ensino.

2.1. Funcionamento Atual: SIGAA na UESPI

Atualmente, o gerenciamento de vagas de monitoria na UESPI é realizado por meio do Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA). A inscrição dos alunos é feita manualmente durante os processos seletivos abertos por cada curso. No entanto, o sistema não fornece mecanismos de visualização da concorrência entre disciplinas, nem permite que os estudantes analisem estratégias para distribuir melhor suas escolhas. Essa ausência motivou o desenvolvimento de uma solução que promova maior equilíbrio entre as vagas ofertadas e o interesse estudantil.

2.2. Iniciativas para Monitoria Acadêmica

Os primeiros trabalhos são voltados à organização de programas de monitoria e enfatizam o aspecto pedagógico da experiência. Natário e Santos (2010) propuseram um

programa de formação e acompanhamento de monitores em uma universidade particular do Estado de São Paulo, com foco em esclarecer o papel do monitor e na preparação para a carreira docente. Essa abordagem, apesar da sua contribuição na formação acadêmica dos monitores, não explorava aspectos tecnológicos de automação ou gestão digital.

Com o avanço das tecnologias educacionais, soluções mais voltadas à eficiência do gerenciamento surgiram. [Ribeiro e Santos \(2018\)](#) desenvolveram um software para gerenciar e executar monitorias nos cursos de Computação, oferecendo funcionalidades como cadastro de monitores, controle de atividades e emissão de certificados. Já o sistema GEM foi criado para Institutos Federais e incorporou recursos como gerenciamento de editais, homologação de relatórios e acompanhamento de alunos ([LIMA et al., 2019](#)).

2.3. Tecnologias Aplicadas em Soluções Semelhantes

Trabalhos abordaram o uso de tecnologias específicas em sistemas acadêmicos. [Alves \(2017\)](#) utilizou o padrão *openEHR* com a arquitetura MVC (*Model-View-Controller*) para desenvolver uma aplicação de gestão de consultas médicas, demonstrando como a separação entre camadas facilita a manutenção e escalabilidade. [Júnior \(2011\)](#) apresentou um sistema que integrou tecnologias móveis com *web*, utilizando *Java*, *.NET* e SMS para notificação de eventos, o que evidenciou a importância de linguagens orientadas a objetos em sistemas integrados.

Já o aplicativo Akademic trouxe uma interface amigável para o gerenciamento de atividades complementares no ensino superior, com foco na usabilidade do estudante ([NASCIMENTO, 2023](#)). Embora não focado em monitorias, sua abordagem em tempo real e interativa inspira soluções similares.

2.4. Projetos com Arquitetura Moderna e Integração de Dados

[Alves e Costa \(2022\)](#) construíram uma aplicação para gerenciar o consumo de bens de forma eficiente utilizando *Spring Boot* e *ReactJS*, demonstrando o uso de APIs REST para integração entre *frontend* e *backend*. De forma semelhante, [Pereira \(2024\)](#) propôs uma aplicação para georreferenciar contadores de eletricidade, também baseada em microserviços, *Spring Boot* e *Thymeleaf*, validando a arquitetura moderna como caminho viável para soluções acadêmicas e institucionais.

3. Processo Metodológico

Nesta seção, serão abordados os processos de construção do sistema *web* (*Monitrack*), que visa auxiliar na distribuição de vagas de monitorias na Universidade Estadual do Piauí, e conta com uma arquitetura MVC (Figura 1) desenvolvida utilizando o *framework Spring Boot*. Segundo [Santos et al. \(2010\)](#), o padrão MVC ajuda a criar aplicações de maneira mais eficiente, facilitando a programação e garantindo a manutenção de softwares depois de concluídos.

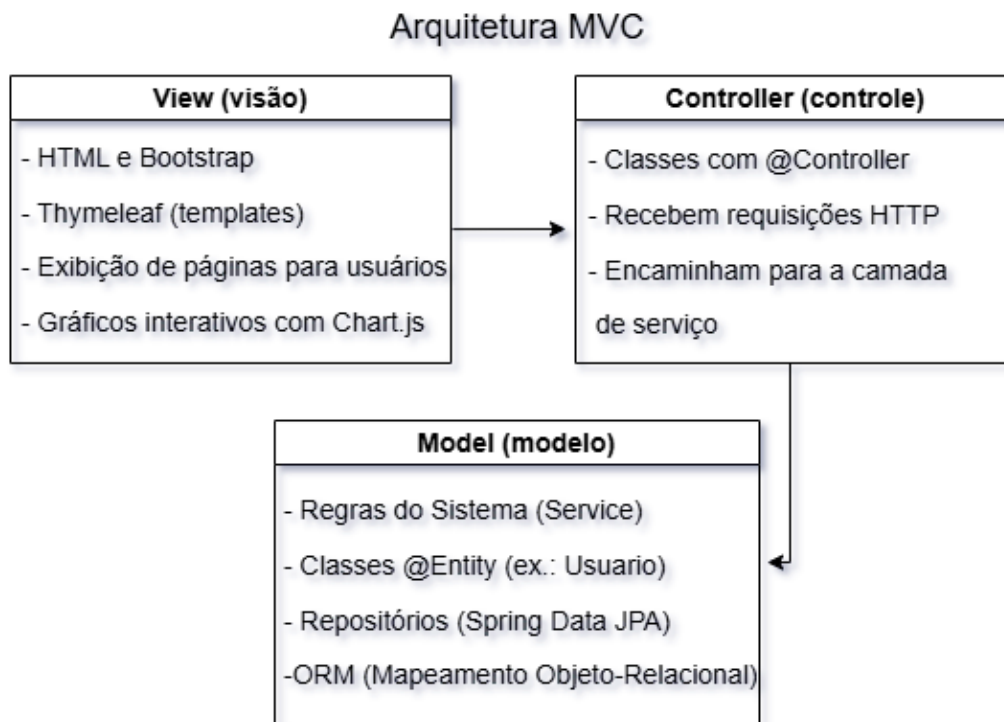


Figura 1. Arquitetura do sistema Monitrack utilizando o padrão MVC. O sistema foi dividido em camadas *View*, *Controller* e *Model*.

3.1. Requisitos do Sistema

Para a implementação do sistema, foi necessário mapear os requisitos para que atendessem aos objetivos principais do projeto, que foram divididos entre: Requisitos Funcionais e Não Funcionais e Regras de Negócio, além da modelagem com diagramas de caso de uso e diagramas de classes.

3.1.1. Requisitos Funcionais

Os Requisitos Funcionais descrevem as funcionalidades essenciais do sistema (FIGUEIREDO, 2011). No contexto do *Monitrack*, os principais requisitos funcionais implementados foram:

- **RF01:** Permitir o cadastro e autenticação de usuários com perfis distintos (administrador, professor e aluno).
- **RF02:** Permitir que administradores cadastrem disciplinas e associem professores responsáveis.
- **RF03:** Permitir que alunos manifestem interesse em monitorias por disciplina, e cancelar se for necessário.
- **RF04:** Permitir que alunos manifestem interesse em no máximo 3 disciplinas.
- **RF05:** Permitir que os Administradores gerem relatórios sobre o interesse dos alunos em cada vaga de monitoria.

- **RF06:** Controlar o acesso às funcionalidades com base no tipo de usuário autenticado.
- **RF07:** Exibir painel personalizado com informações relevantes para cada tipo de usuário.

3.1.2. Requisitos Não Funcionais

Os Requisitos Não Funcionais dizem respeito às qualidades do sistema e restrições técnicas, como segurança e desempenho do sistema (FIGUEIREDO, 2011). No *Moni-track* os requisitos não funcionais foram:

- **RNF01:** O sistema deve ser acessado via navegador *web* e utilizar a arquitetura cliente-servidor.
- **RNF02:** A aplicação deve ser implementada em Java, utilizando o *framework Spring Boot*.
- **RNF03:** As interfaces devem ser renderizadas com uso da engine de *templates Thymeleaf*.
- **RNF04:** O sistema deve armazenar os dados em banco de dados relacional com suporte a JPA (MySQL).
- **RNF05:** A autenticação deve ser feita com base no *Spring Security*, garantindo o controle de acesso por perfil, e senhas salvas com criptografia no banco de dados.
- **RNF06:** O sistema deve ser responsivo, permitindo o uso em diferentes tamanhos de tela.
- **RNF07:** Os dados do sistema devem ser inseridos manualmente no banco de dados.

3.1.3. Regras de Negócio

Segundo Dallavalle e Cazarini (2000), as regras de negócio ajudam a integrar uma dinâmica de organização aos sistemas de informação, sendo um pré-requisito essencial para o sucesso de um projeto de desenvolvimento de sistemas.

- **RN01:** Professores só podem visualizar disciplinas pelas quais são responsáveis.
- **RN02:** Alunos só podem demonstrar interesse uma única vez por disciplina.
- **RN03:** Professores ou Administradores podem editar e deletar disciplinas ofertadas.

3.1.4. Diagrama de Caso de Uso

O diagrama de caso de uso mostra a representação externa das funcionalidades do sistema, assim como mostra a Figura 2. Essa é uma ferramenta importante para poder entender o comportamento do sistema de forma visual (JÚNIOR, 2020).

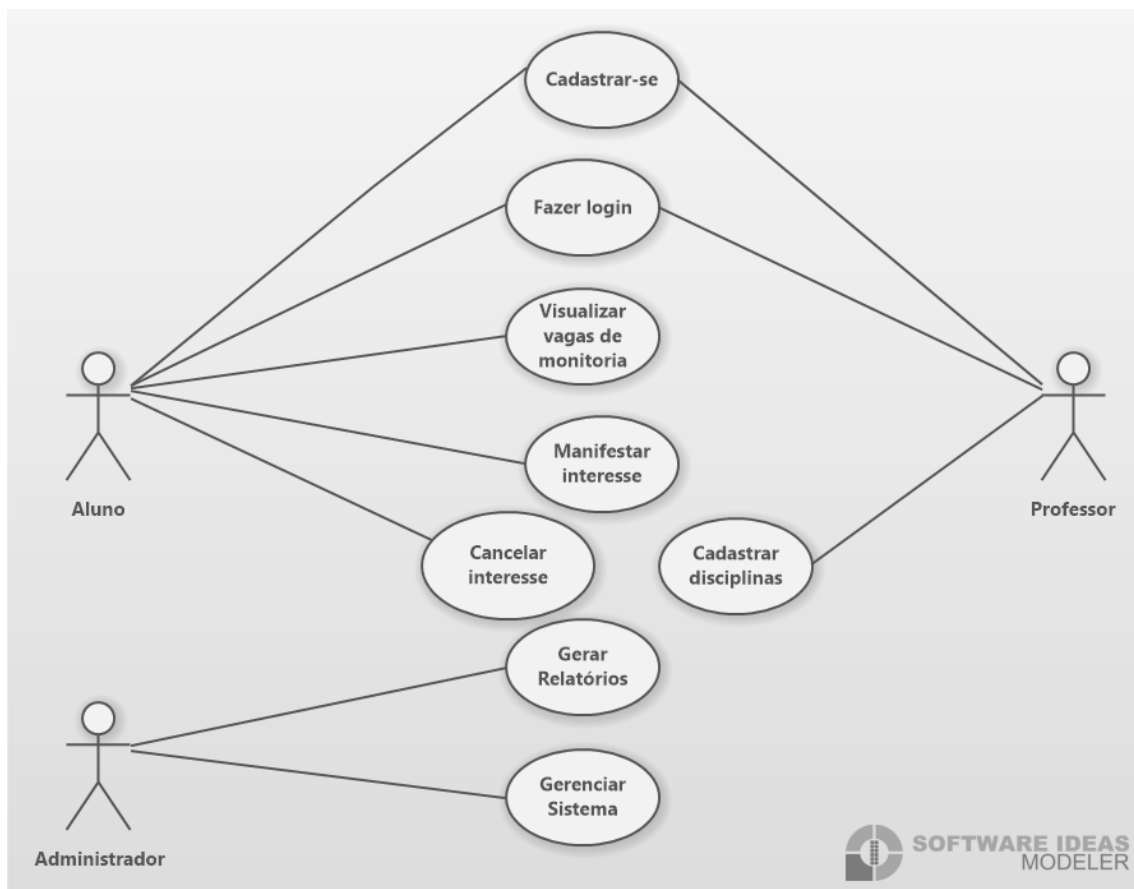


Figura 2. Diagrama de caso de uso. Representa os principais atores (aluno, professor e administrador) e suas interações com as funcionalidades do sistema, como manifestação de interesse, cadastro de disciplinas e geração de relatórios.

3.1.5. Diagrama de Classes

O diagrama de classes é importante para a demonstração da estrutura de classes de qualquer sistema orientado a objetos (JÚNIOR, 2020). A Figura 3 mostra uma representação das partes mais importantes da estrutura do *Monitrack*.

3.2. Tecnologias

Para o desenvolvimento do sistema *Monitrack*, foram utilizadas algumas tecnologias que oferecem robustez, escalabilidade e facilidade de manutenção. A arquitetura da aplicação foi dividida em duas principais camadas: *Frontend* e *Backend*.

3.2.1. Frontend

O *frontend* do sistema é responsável pela camada de apresentação, ou seja, por aquilo que o usuário vê e interage diretamente. Para isso, foi utilizada a biblioteca de templates *Thymeleaf*, que permite a renderização dinâmica de páginas *HTML* com dados vindos do *backend*. Além disso, foram utilizados:

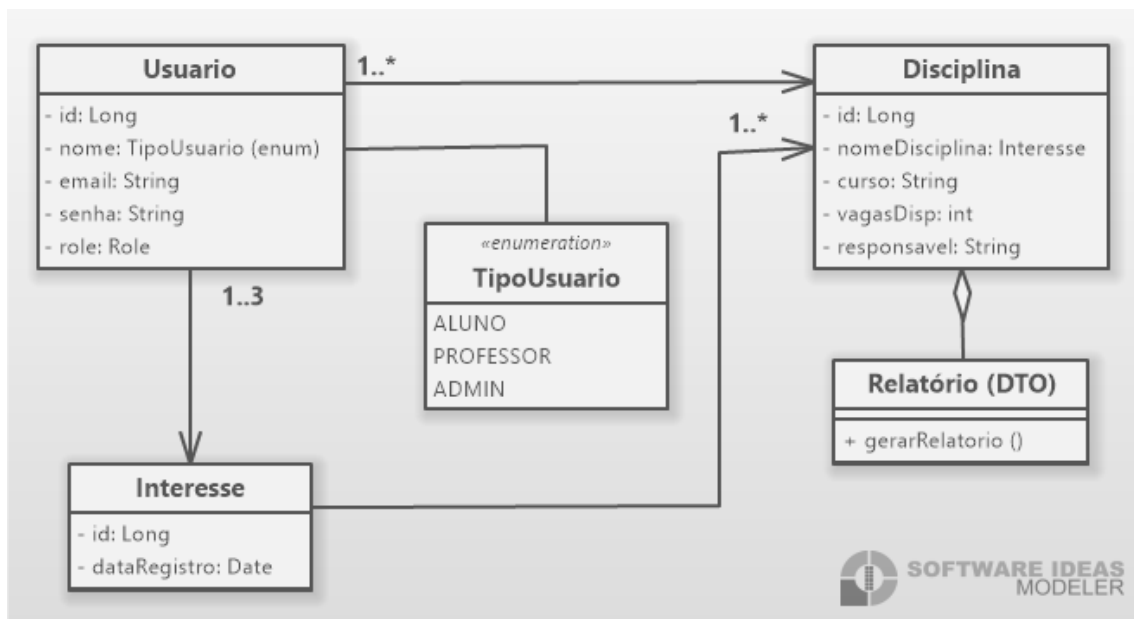


Figura 3. Diagrama de classes. Apresenta a estrutura orientada a objetos do sistema, incluindo entidades como Aluno, Disciplina, Professor, Interesse e seus relacionamentos.

- **HTML5 e CSS:** Estruturação e estilização das páginas.
- **Bootstrap:** Para garantir um layout responsivo e compatível com diferentes tamanhos de tela.
- **JavaScript:** Para a implementação de interações mais avançadas com o usuário junto com biblioteca *Chart.js* que foi incorporada para gerar gráficos interativos no painel do aluno, permitindo a visualização do nível de concorrência entre as disciplinas de forma clara e responsiva.
- **Thymeleaf:** Facilita a integração entre os dados *Java* e as *views HTML*, mantendo a lógica no servidor.

3.2.2. Backend

O *Backend* foi construído com o *framework Spring Boot*¹, que facilita a criação de aplicações *Java* baseadas em microserviços e arquitetura MVC. Ele é responsável por toda a lógica de negócio, controle de acesso e persistência de dados. As principais tecnologias utilizadas no *backend* incluem:

- **Spring Boot:** *Framework* principal da aplicação, oferece suporte a MVC, injeção de dependência, segurança e muito mais.
- **Spring Data JPA:** Facilita a persistência de dados no banco de forma orientada a objetos.
- **Spring Security:** Controla autenticação e autorização de acesso às funcionalidades.
- **MySQL:** Banco de dados relacional para armazenar informações sobre usuários, disciplinas e interesses.
- **Maven:** Gerenciador de dependências e *build* da aplicação.

¹Site oficial do *spring boot* (<https://spring.io/projects/spring-boot>)

4. Resultados

No desenvolvimento do *Monitrack*, foi possível validar de forma prática as funcionalidades propostas para o gerenciamento de vagas do programa de monitorias na UESPI. Nesta seção serão apresentados os resultados alcançados no sistema desenvolvido.

4.1. Funcionalidades implementadas

O *Monitrack* foi desenvolvido para auxiliar os alunos a escolherem a melhor vaga de monitoria para concorrer, fazendo com que seja possível ter uma concorrência mais equilibrada dessas vagas. Entre os principais recursos implementados, incluem-se a separação por perfis de usuários:

- **Administrador:** possui acesso ao sistema sendo responsável por gerenciar usuários e gerar os relatórios com base na manifestação de interesse dos alunos (Figura 4). O relatório apresenta todas as informações centralizadas, como as vagas ofertadas em cada disciplina, professor responsável e o total de interessados em cada uma delas.



Relatório de Interesses em Monitorias			
Disciplina	Professor	Interessados	Vagas
TCC II	Bia Silva	1	1
Grafos	Rute Silva	3	1
Estrutura de Dados	Bia Silva	2	2
Fisica II	Lia Silva	0	1
Programação III	Bia Silva	0	3

Figura 4. Visualização do relatório. Exibe o painel do administrador com os dados consolidados das oportunidades de monitoria, disciplinas, professores responsáveis e quantidade de alunos interessados.

- **Professor:** tem permissão para cadastrar oportunidades de monitoria, consultar os alunos interessados e editar ou excluir suas respectivas ofertas como mostra a Figura 5.

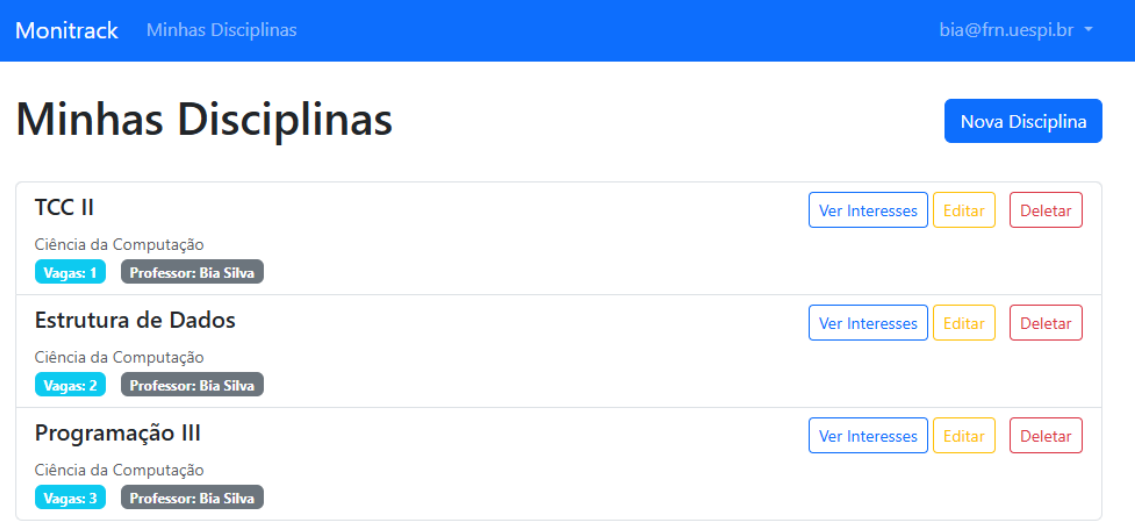


Figura 5. Tela dos professores. Permite o cadastro, edição e exclusão das vagas de monitoria sob sua responsabilidade, bem como a visualização de alunos interessados.

- **Aluno:** tem acesso a um painel onde pode visualizar as oportunidades de monitoria que estão em andamento, registrar seu interesse em até três disciplinas distintas, e também pode cancelar interesse caso desejem modificar suas escolhas, de acordo com a Figura 6.

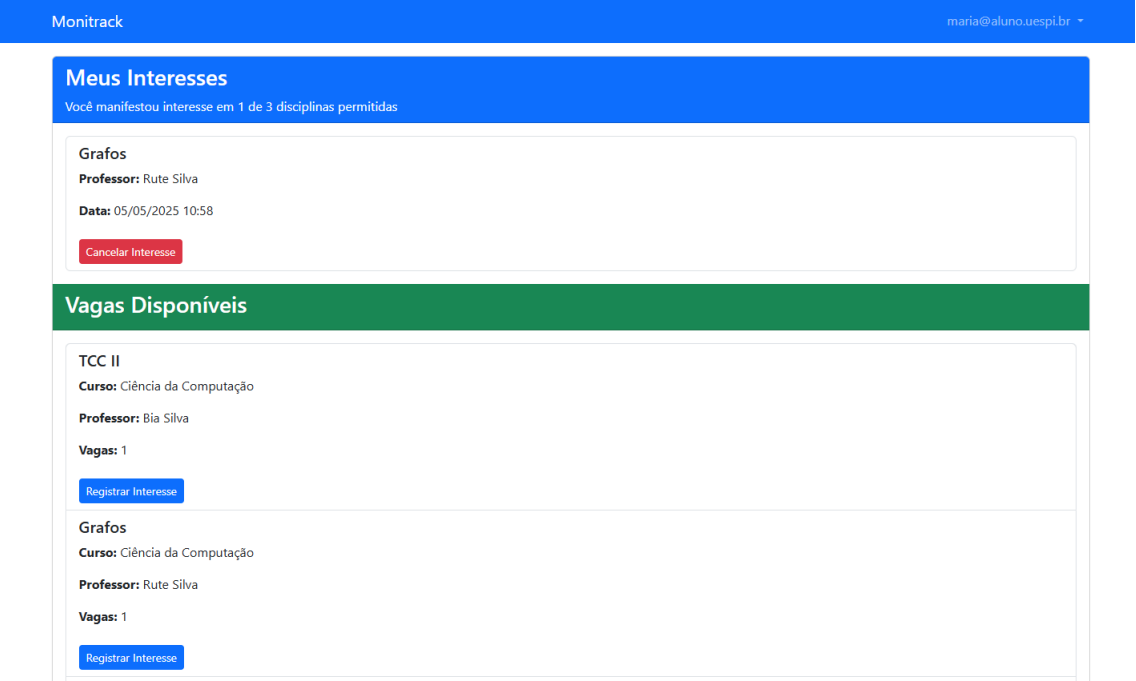


Figura 6. Painel para registrar e cancelar interesses. Exibe as disciplinas disponíveis, permite registrar ou cancelar o interesse e consultar o número de interessados em cada uma.

Na mesma página, o aluno pode consultar a concorrência das disciplinas; na tabela, possui a contagem de quantos alunos manifestaram interesse em cada vaga ofertada, e o sistema organiza iniciando das menos concorridas para as mais concorridas (Figura 7). Com isso, o aluno pode analisar qual seria sua melhor opção para se inscrever na vaga.

Disciplinas Prioritárias	
Veja quais disciplinas têm menos concorrência	
Menos concorridas	Mais concorridas
Estrutura de Dados Interessados: 0/ 2 vagas Professor: Bia Silva	TCC II Interessados: 1/ 1 vagas Professor: Bia Silva
Física II Interessados: 0/ 1 vagas Professor: Lia Silva	Grafos Interessados: 1/ 1 vagas Professor: Rute Silva
Programação III Interessados: 0/ 3 vagas Professor: Bia Silva	

Figura 7. Visualização de concorrência de vagas. Mostra a quantidade de alunos interessados por disciplina em ordem crescente, auxiliando o aluno a tomar uma decisão estratégica.

Além disso, o aluno pode escolher visualizar a concorrência com um gráfico de barras exemplificado pela Figura 8.

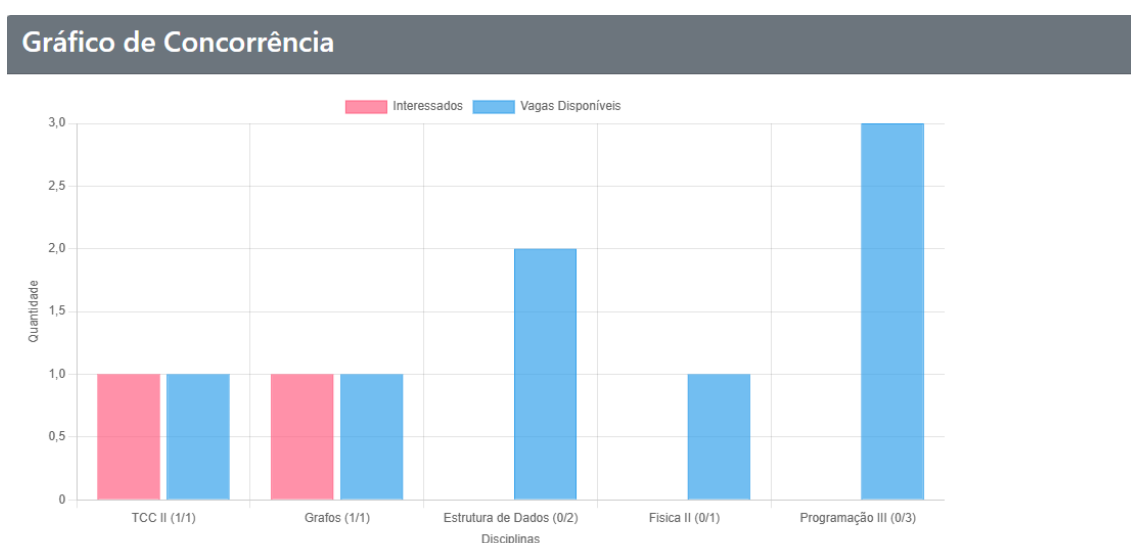


Figura 8. Quantidade de interesses por vagas. Representação visual interativa que ajuda o aluno a identificar rapidamente as vagas mais e menos concorridas.

5. Conclusão

Este trabalho apresentou o desenvolvimento do sistema *web* Monitrack, um protótipo voltado para o gerenciamento de interesses em vagas de monitoria na UESPI. Com objetivos propostos a partir da identificação de problemas como a má distribuição de interessados nas vagas de monitoria, o sistema foi implementado para oferecer uma solução prática para essa demanda na instituição.

Dentre os resultados alcançados no desenvolvimento, destacam-se: o cadastro e autenticação de usuários com perfis distintos, o registro e cancelamento de interesse por vagas de monitoria por parte dos alunos, a visualização da concorrência entre as disciplinas e a geração de relatórios para os administradores — sendo esses os principais pontos positivos do sistema. Esses recursos foram implementados com o objetivo de minimizar o desequilíbrio na distribuição dos candidatos entre as disciplinas, oferecendo aos alunos uma visualização clara da concorrência e facilitando decisões mais estratégicas na escolha de vagas. Entretanto, o protótipo possui algumas limitações: na versão atual, não é possível exportar o relatório e não há armazenamento de histórico de uso do sistema ao longo do tempo.

O Monitrack contribui como uma ferramenta auxiliar para a gestão de monitorias acadêmicas. Ao centralizar informações relevantes sobre as vagas disponíveis e os níveis de concorrência, melhora o processo de seleção de monitores, tornando o preenchimento das vagas mais equilibrado.

Apesar de o Monitrack ter alcançado seu principal objetivo, existem possibilidades de melhorias para trabalhos futuros, como a implementação de notificações automáticas (por e-mail ou sistema) para alertas de disciplinas com baixa concorrência, além da inclusão de uma análise histórica que permita aos professores acessar dados anteriores e avaliar quais disciplinas são interessantes para continuar ofertando vagas.

O repositório do protótipo pode ser acessado por meio do seguinte link: <https://github.com/bruna-crist/Prototipo1.0.git>.

Referências

- ALVES, B.; COSTA, R. M. de R. Gerenciamento da aquisição de bens de consumo não duráveis através de uma aplicação web. *Caderno de Estudos em Sistemas de Informação*, v. 8, n. 2, 2022. 3
- ALVES, J. M. Utilização do modelo mvc no desenvolvimento de aplicações e-saúde utilizando o padrão openehr. *GRAD - Cursos de Graduação*, 2017. 3
- DALLAVALLE, S. I.; CAZARINI, E. W. Regras do negócio, um fator chave de sucesso no processo de desenvolvimento de sistemas de informação. *Anais do XX ENEGEP-Encontro Nacional de Engenharia de Produção*. São Paulo, 2000. 5
- FIGUEIREDO, E. Requisitos funcionais e requisitos não funcionais. *Icex, Dcc/Ufmg*, v. 14, 2011. 4, 5
- JÚNIOR, E. S. *Diagrama de Caso de Uso*. [S.l.]: IFTM, 2020. 5, 6

- JÚNIOR, V. F. Desenvolvimento web comunicação móvel aplicados em um sistema de gerenciamento de compromissos. <https://www.researchgate.net/publication/280082811>, 2011. [3](#)
- LIMA, F. A. d. et al. Gem: sistema de gerenciamento de programas de monitoria. *Tecnólogo em Análise e Desenvolvimento de Sistemas*, Recife, 2019. [3](#)
- NASCIMENTO, I. J. M. d. Desenvolvimento de uma aplicação móvel para gerenciamento de atividades complementares em instituições de ensino superior. *Engenharia de Software - Quixadá - TCC*, 2023. [3](#)
- NATÁRIO, E. G.; SANTOS, A. A. A. d. Programa de monitores para o ensino superior. *Estudos de Psicologia (Campinas)*, SciELO Brasil, v. 27, p. 355–364, 2010. [2](#)
- NETO, J. d. C. S.; ANDRADE, I. L. A contribuição da monitoria acadêmica para o incentivo a docência. *Revista Interfaces: Saúde, Humanas e Tecnologia*, v. 4, n. 12, p. 93–99, 2016. [2](#)
- PEREIRA, F. L. P. *Sistema de informação georreferenciado para contadores de electricidade nos pontos de consumo*. Dissertação (Mestrado) — Universidade de Évora, 2024. [3](#)
- RIBEIRO, A. R. P.; SANTOS, W. I. d. Desenvolvimento de protótipo de um software para gestão de monitoria usando processo ágil. *Trabalhos de Conclusão de Curso - TCC's (Engenharia de Computação)*, 2018. [3](#)
- SANTOS, I. et al. Possibilidades e limitações da arquitetura mvc (model–view–controller) com ferramenta ide (integrated development environment). *RE3C-Revista Eletrônica Científica de Ciência da Computação*, v. 5, n. 1, 2010. [3](#)
- VICENZI, C. B. et al. A monitoria e seu papel no desenvolvimento da formação acadêmica. *Revista Ciência em Extensão*, v. 12, n. 3, p. 88–94, 2016. [1](#)