

Projeto Horário

Daniel Lins Silva Andrade
Depto. de engenharias e tecnologia
UFERSA

Pau dos Ferros, Brasil
daniel.andrade@alunos.ufersa.edu.br

José Claudio Alves Sobrinho
Depto. de engenharias e tecnologia
UFERSA

Pau dos Ferros, Brasil
jose.sobrinho52296@alunos.ufersa.edu.br

Kennedy Alves Pereira
Depto. de engenharias e tecnologia
UFERSA

Pau dos Ferros, Brasil
kennedy.pereira@alunos.ufersa.edu.br

Resumo—O atual processo de alocação de recursos e organização das atividades acadêmicas enfrenta complexidades, especialmente no que diz respeito à consideração da disponibilidade dos professores. O problema central enfrentado é a complexidade na administração dos horários das turmas do BTI, evidenciando a dificuldade em otimizar a alocação de recursos e evitar conflitos de horários, impactando negativamente na experiência dos professores e na eficiência administrativa do curso. A proposta apresenta relevância ao buscar a maximização do aproveitamento dos recursos disponíveis, a redução de conflitos de horários, a melhoria da experiência dos professores e o aumento da eficiência administrativa. A automação e aprimoramento do gerenciamento de horários contribuirão para uma melhor organização e planejamento do curso de BTI. A principal contribuição do projeto é o desenvolvimento de um sistema que atenda às necessidades específicas do curso, considerando a disponibilidade de professores. Ao oferecer uma solução customizada, o sistema busca facilitar a administração dos horários, proporcionando benefícios tangíveis, como a otimização de recursos e a melhoria da experiência acadêmica. O projeto visa concluir com a implementação bem-sucedida de um sistema de gerenciamento de horários que efetivamente atenda às demandas do curso de BTI. Espera-se que a automatização contribua para a redução de conflitos de horários, maximize a eficiência administrativa e proporcione uma experiência mais fluida para professores e estudantes. Em última análise, é esperada uma gestão de horários mais eficaz e uma melhor organização do curso.

Index Terms—Alocação de recursos, Organização acadêmica, Eficiência administrativa, Automação, Otimização de recursos

I. INTRODUÇÃO

O projeto proposto visa desenvolver um sistema de gerenciamento de horários para turmas do curso do Bacharelado em Tecnologia da Informação (BTI) do Campus Pau dos Ferros (CMPF) da Universidade Federal Rural do Semi-Árido (UFERSA).

Buscando otimizar a alocação de recursos e a distribuição eficiente das atividades acadêmicas, o sistema será projetado para atender às necessidades específicas do curso, considerando a disponibilidade de professores.

O uso dessa solução visa diversos benefícios, como a maximização do aproveitamento dos recursos disponíveis, a redução de conflitos de horários, a melhoria da experiência dos professores ao evitar sobreposições indesejadas e o aumento da eficiência administrativa do curso.

II. FUNDAMENTAÇÃO TEÓRICA

Durante a execução deste projeto, primou-se pela adoção dos princípios da Engenharia de Software a fim de promover

uma organização e planejamento mais eficazes. Essa abordagem, conforme apresentada por (PRESSMAN; MAXIM, 2021), destaca cinco atividades genéricas fundamentais na aplicação da Engenharia de Software: comunicação, planejamento, modelagem, construção e implantação.

Os princípios da Engenharia de Software, conforme preconizados por (PRESSMAN; MAXIM, 2021), incluem a definição de processos de desenvolvimento bem estruturados, a gestão de projetos de software, a garantia de qualidade, a documentação adequada e a consideração dos aspectos de manutenção ao longo do tempo. Além disso, a Engenharia de Software promove a colaboração eficaz entre membros da equipe e a comunicação transparente com os stakeholders.

Neste contexto, a Engenharia de Requisitos emerge como uma disciplina fundamental no processo de desenvolvimento de software, concentrando-se na identificação, análise, especificação e gerenciamento dos requisitos de um sistema de software (PRESSMAN; MAXIM, 2021). Garantir que o software seja desenvolvido conforme as expectativas do cliente, com requisitos documentados de forma clara e precisa, é crucial para o desenvolvimento do software em questão.

A especificação formal, notação Z, se apresenta como uma ferramenta valiosa neste processo. A notação Z permite uma representação matemática e precisa dos requisitos, contribuindo para a clareza e a formalidade na comunicação entre os membros da equipe de desenvolvimento. Essa abordagem formal facilita a detecção de inconsistências e ambiguidades nos requisitos, promovendo assim a qualidade desde as fases iniciais do desenvolvimento. (SPIVEY, 1988)

III. TRABALHOS RELACIONADOS

A otimização da distribuição de cargas horárias em ambientes acadêmicos é uma área de pesquisa crucial, dada a complexidade inerente a esse processo. No estudo conduzido na Universidade de Passo Fundo (UPF), os coordenadores de área são incumbidos da distribuição de disciplinas entre os professores, visando evitar conflitos de horários. Este desafio é tratado como um problema de otimização combinatória, destacando a importância de estratégias eficientes para a alocação de recursos. (KRIPKA; KRIPKA; ORO,)

Enquanto o estudo na UPF aborda a otimização de distribuição de disciplinas entre os professores, o projeto proposto vai além, concentrando-se na automatização e aprimoramento do gerenciamento de horários para benefícios

específicos. Tais benefícios incluem a maximização do aproveitamento dos recursos disponíveis, a redução de conflitos de horários, a melhoria da experiência dos professores ao evitar sobreposições indesejadas e o aumento da eficiência administrativa do curso. Dessa forma, o projeto não apenas enfrenta os desafios gerais da otimização, mas também aborda demandas específicas do contexto do BTI na UFERSA.

IV. METODOLOGIA

O desenvolvimento do software de gerenciamento de horários foi conduzido seguindo uma abordagem ágil, com ênfase na metodologia Kanban. Esta escolha se justifica pela flexibilidade e adaptabilidade que o Kanban oferece, permitindo ajustes contínuos de prioridades e requisitos ao longo do ciclo de desenvolvimento. (UNIVERSITY; ANDERSON; CARMICHAEL, 2016)

O Kanban é uma metodologia ágil que se baseia em princípios de visualização do trabalho, limitação de trabalho em progresso e melhoria contínua (PRIKLADNICKI; WILLI; MILANI, 2014). Para este projeto, utilizamos um quadro Kanban virtual para visualizar as etapas do desenvolvimento, desde a concepção até a implementação e testes.

A escolha da linguagem Java proporcionou uma base sólida para o desenvolvimento, oferecendo portabilidade e ampla comunidade de suporte. A orientação a objetos em Java contribuiu para a organização modular do código.

JavaFX foi empregado para criar uma interface gráfica intuitiva e de fácil interação. A capacidade de JavaFX de integrar facilmente com a lógica do Java proporcionou uma experiência de usuário coesa.

Foi utilizado um banco de dados relacional para armazenar informações relevantes sobre horários, professores e cursos. A escolha do banco de dados foi baseada na capacidade de lidar eficientemente com grandes volumes de dados e garantir a integridade das informações.

A combinação de uma metodologia ágil como o Kanban com tecnologias robustas como Java, banco de dados relacional e JavaFX foi crucial para o sucesso do projeto, garantindo não apenas a eficiência do desenvolvimento, mas também a qualidade e adaptabilidade do software.

V. CONSIDERAÇÕES FINAIS

Durante o desenvolvimento do software, aprendemos valiosas lições sobre a dinâmica e os desafios inerentes à gestão de horários em um ambiente acadêmico. A compreensão aprofundada das necessidades específicas do curso BTI foi crucial para o sucesso do projeto. A interação constante com os stakeholders, incluindo professores e administradores, permitiu uma adaptação contínua do sistema para atender às expectativas e requisitos em constante evolução.

Ao longo do processo de desenvolvimento, enfrentamos desafios significativos, como a integração de dados provenientes de diferentes fontes, a otimização dos algoritmos de alocação e a garantia de usabilidade para usuários com diferentes níveis de familiaridade com tecnologia. A superação desses desafios fortaleceu não apenas a robustez técnica do software, mas

também a colaboração entre a equipe de desenvolvimento e os usuários finais.

Para os próximos passos deste projeto, sugerimos algumas áreas promissoras de pesquisa e desenvolvimento, considerando essencial a migração do software para uma plataforma web, proporcionando maior acessibilidade e flexibilidade. Isso permitirá que os usuários acessem o sistema de qualquer local e dispositivo, facilitando a colaboração remota e otimizando ainda mais o gerenciamento de horários. Assim como uma refatoração abrangente dos métodos do software, visando aprimorar a eficiência e a escalabilidade do sistema. Isso inclui a otimização de algoritmos de alocação, a implementação de técnicas mais avançadas de gerenciamento de dados e a adoção de práticas de desenvolvimento mais eficientes.

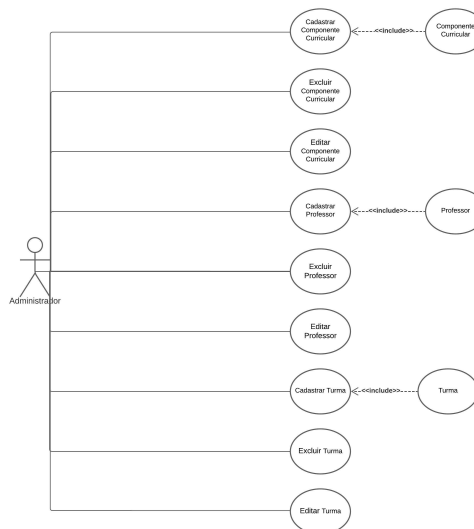
REFERÊNCIAS

- KRIPKA, M.; KRIPKA, R. M. L.; ORO, N. T. Aplicação de técnicas de otimização na determinação da distribuição de cargas horárias na universidade de passo fundo. *Gestão*, p. 45.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software-9*. [S.l.]: McGraw Hill Brasil, 2021.
- PRIKLADNICKI, R.; WILLI, R.; MILANI, F. *Métodos ágeis para desenvolvimento de software*. [S.l.]: Bookman Editora, 2014.
- SPIVEY, J. M. *Understanding Z: a specification language and its formal semantics*. [S.l.]: Cambridge University Press, 1988. v. 3.
- UNIVERSITY, L. K.; ANDERSON, D. J.; CARMICHAEL, A. *Kanban esencial condensado*. Kaban University, 2016.

VI. ARTEFATOS

A. Diagrama de caso de uso

Figura 1. Caso de uso.



B. Notação Z

Utilizando a tática de notação Z, foram elaboradas as seguintes tabelas de esquemas de demonstração inicial das funcionalidades funcionais do projeto. Abaixo você poderá visualizar os esquemas.

Entidade: Professor
<i>matricula</i> : STRING($1 \leq \text{length}(\text{matricula}) \leq 10$)
<i>nome</i> : STRING
<i>email</i> : STRING
<i>titulacao</i> : STRING

Tabela I

TABELA PARA A ENTIDADE PROFESSOR

CadastrarProfessor
$\Delta \text{CadastroProfessor}$
<i>nome?</i> , <i>email?</i> , <i>titulacao?</i> , <i>matricula?</i> : STRING
<i>nome'</i> = <i>nome?</i>
<i>email'</i> = <i>email?</i>
<i>titulacao'</i> = <i>titulacao?</i>
<i>matricula'</i> = <i>matricula?</i>

Tabela II

TABELA PARA CADASTRAR PROFESSOR

EditarProfessor
$\Delta \text{CadastroProfessor}$
<i>n?</i> , <i>e?</i> , <i>t?</i> , <i>m?</i> : STRING
<i>nome'</i> = <i>n?</i>
<i>email'</i> = <i>e?</i>
<i>titulacao'</i> = <i>t?</i>
<i>matricula'</i> = <i>m?</i>

Tabela III

TABELA PARA EDITAR PROFESSOR

ExcluirProfessor
$\Delta \text{CadastroProfessor}$
<i>matricula?</i> : STRING
<i>matricula'</i> = <i>matricula?</i>

Tabela IV

TABELA PARA EXCLUIR PROFESSOR

VisualizarListaProfessores
$\Xi \text{CadastroProfessor}$

Tabela V

TABELA PARA VISUALIZAR LISTA DE PROFESSORES CADASTRADOS

Entidade: ComponenteCurricular
<i>codigo</i> : STRING($\text{length}(\text{codigo}) = 5$)
<i>nome</i> : STRING
<i>tipo</i> : STRING($\text{tipo} \in \{\text{'obrigatória'}, \text{'não obrigatória'}\}$)
<i>cargaHoraria</i> : INT($\text{cargaHoraria} \in \{30, 60, 90\}$)

Tabela VI

TABELA PARA A ENTIDADE COMPONENTE CURRICULAR

CadastrarComponenteCurricular
$\Delta \text{CadastroComponenteCurricular}$
<i>componenteCurricular?</i> : ComponenteCurricular
<i>codigo'</i> = <i>componenteCurricular?.codigo</i>
<i>nome'</i> = <i>componenteCurricular?.nome</i>
<i>tipo'</i> = <i>componenteCurricular?.tipo</i>
<i>cargaHoraria'</i> = <i>componenteCurricular?.cargaHoraria</i>

Tabela VII

TABELA PARA CADASTRAR COMPONENTE CURRICULAR

ExcluirComponenteCurricular
$\Delta \text{CadastroComponenteCurricular}$
<i>codigo?</i> : STRING
<i>codigo'</i> = <i>codigo?</i>

Tabela VIII

TABELA PARA EXCLUIR COMPONENTE CURRICULAR

VisualizarListaComponentesCurriculares
$\Xi \text{CadastroComponenteCurricular}$

Tabela IX

TABELA PARA VISUALIZAR LISTA DE COMPONENTES CURRICULARES CADASTRADOS

Entidade: Turma
<i>componenteCurricular</i> : ComponenteCurricular
<i>capacidadeMaxima</i> : INT($1 \leq \text{capacidadeMaxima} \leq 50$)
<i>professor</i> : Professor
<i>horario</i> : STRING($\text{length}(\text{horario}) \in \{4, 6, 8\}$)
<i>cargaHoraria</i> : INT($\text{cargaHoraria} \in \{30, 60, 90\}$)

Tabela X

TABELA PARA A ENTIDADE TURMA

CadastrarTurma
$\Delta \text{CadastroTurma}$
<i>turma?</i> : Turma
<i>componenteCurricular'</i> = <i>turma?.componenteCurricular</i>
<i>capacidadeMaxima'</i> = <i>turma?.capacidadeMaxima</i>
<i>professor'</i> = <i>turma?.professor</i>
<i>horario'</i> = <i>turma?.horario</i>
<i>cargaHoraria'</i> = <i>turma?.cargaHoraria</i>

Tabela XI

TABELA PARA CADASTRAR TURMA

EditarTurma
Δ CadastroTurma turma? : Turma remove($\{turma \in$ CadastroTurma $ turma.componenteCurricular = turma?.componenteCurricular$ $\wedge turma.professor = turma?.professor \wedge$ $turma.horario = turma?.horario\}$) <hr/> $componenteCurricular' = turma?.componenteCurricular$ $capacidadeMaxima' = turma?.capacidadeMaxima$ $professor' = turma?.professor$ $horario' = turma?.horario$ $cargaHoraria' = turma?.cargaHoraria$

Tabela XII

TABELA PARA EDITAR TURMA

ExcluirTurma
Δ CadastroTurma componenteCurricular?, professor?, horario? : STRING remove($\{turma \in$ CadastroTurma $ turma.componenteCurricular = componenteCurricular? \wedge$ $turma.professor = professor? \wedge$ $turma.horario = horario?\}$)

Tabela XIII

TABELA PARA EXCLUIR TURMA

VisualizarTurmas
Ξ CadastroTurma

Tabela XIV

TABELA PARA VISUALIZAR TURMAS CADASTRADAS

ExibirHorario
Δ ExibicaoHorario semestre?, professor? : STRING

Tabela XV

TABELA PARA EXIBIR HORÁRIO POR SEMESTRE OU PROFESSOR