

DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA TIMETABLING UTILIZANDO HTML, CSS E JAVASCRIPT: uma solução prática e acessível para a organização de horários

AMARAL, Víctor Paiva da Silva¹; PEREIRA, Ana Amelia de Souza²;
TREVIZANO, Waldir Andrade²; BAÍA. Joás Wesley²



victoramara951@outlook.com
ana.amelia@unifagoc.edu.br

¹ Graduando em Ciência da Computação - UNIFAGOC

² Docente do curso de Ciência da Computação - UNIFAGOC

RESUMO

O *timetabling*, ou programação de horários, é um desafio complexo que afeta várias áreas, incluindo instituições de ensino. Este artigo tem por objetivo desenvolver uma aplicação web, utilizando a IDE Vscode, que seja capaz de resolver os problemas de timetabling. Para isso, foi criada uma interface utilizando a linguagem HTML, na qual os usuários poderão interagir e inserir as informações necessárias para o escalonamento. As funcionalidades da aplicação foram implementadas em JavaScript, visando realizar os cálculos necessários para a geração dos horários. A combinação dessas tecnologias permitiu a criação de uma ferramenta prática e acessível, para auxiliar na resolução dos desafios enfrentados pelas instituições de ensino no processo de organização de horários e alocação de recursos. O projeto inclui o cadastro de professores, disciplinas, salas, dias da semana e horários de aula, todos armazenados de forma organizada. A visualização é simplificada em uma grade, tornando as alocações claras. A aplicação permite o armazenamento local dos dados, mas a migração para um banco de dados é recomendada em ambientes de produção. Além disso, a aplicação é disponibilizada online para fácil acesso. Os testes incluíram a avaliação no navegador Opera e a análise da responsividade em dispositivos móveis. O destaque foi um teste com uma grade curricular real da instituição UNIFAGOC. A aplicação inclui uma função de validação de horários para evitar conflitos. Para projetos futuros, são sugeridas melhorias no layout da grade curricular e a implementação de um banco de dados. O código-fonte do projeto está disponível no GitHub. Este projeto oferece uma solução para as instituições de ensino que buscam otimizar a organização de horários e alocação de recursos, contribuindo para um processo de timetabling eficiente e satisfatório.

Palavras-chave: Timetabling. Grade de horário. Detecção de conflitos.

INTRODUÇÃO

O *timetabling*, ou o problema de programação de horários, é uma questão complexa que afeta várias áreas, como instituições de ensino, empresas e organizações que lidam com a alocação de recursos e agendamento de atividades (Borges, 2003).

A elaboração de um cronograma eficiente envolve a consideração de diversos fatores, como a disponibilidade de recursos, as restrições de horários, as preferências dos

envolvidos e a otimização de critérios específicos. O objetivo principal é encontrar uma solução que minimize os conflitos de horários, maximize a utilização dos recursos disponíveis e atenda às necessidades e preferências dos envolvidos (Hime, 2003).

A área de pesquisa em *timetabling* tem sido objeto de interesse para a comunidade acadêmica e profissional, impulsionada pela necessidade de soluções eficientes e automatizadas para esse problema.

Considerando a diversidade de problemas de *timetabling* existentes, como a programação de horários em instituições de ensino, a alocação de recursos em empresas e a organização de eventos, é fundamental compreender as peculiaridades de cada domínio e identificar os requisitos específicos de cada contexto.

O objetivo geral deste trabalho consiste em desenvolver uma aplicação web, utilizando a IDE Vscode, que seja capaz de resolver os problemas de *timetabling*. Para isso, foi criada uma interface utilizando a linguagem HTML, na qual os usuários poderão interagir e inserir informações necessárias para o escalonamento. As funcionalidades da aplicação foram implementadas em JavaScript, visando realizar os cálculos necessários para a geração dos horários. A combinação dessas tecnologias permitiu a criação de uma ferramenta prática e acessível, para auxiliar na resolução dos desafios enfrentados pelas instituições de ensino no processo de organização de horários e alocação de recursos.

O trabalho tem como objetivos específicos:

- Realizar o levantamento dos requisitos: Identificar e compreender as necessidades e exigências da instituição de ensino UNIFAGOC em relação ao *timetabling*, incluindo restrições de horários, recursos disponíveis, preferências dos envolvidos, entre outros.
- Projetar a interface do usuário: Desenvolver uma interface intuitiva e amigável utilizando a linguagem HTML, que permita aos usuários inserir e gerenciar os dados necessários para a geração dos horários, como disciplinas, salas, professores, turmas, entre outros.
- Implementar as funcionalidades: Desenvolver as funcionalidades da aplicação, de forma a realizar os cálculos e otimizações necessárias para gerar os horários de forma eficiente, levando em consideração as restrições e preferências definidas.
- Realizar testes e validação da aplicação: Validar os resultados obtidos através do feedback do usuário e das restrições apresentadas no projeto.
- Documentar o processo de desenvolvimento: Registrar detalhadamente todas as etapas do desenvolvimento da aplicação, incluindo as decisões de projeto, as funcionalidades implementadas, as técnicas utilizadas e os resultados obtidos.

REFERENCIAL TEÓRICO

Serão discutidas as tecnologias HTML e JavaScript, que foram escolhidas para a implementação da aplicação web de *timetabling*. Além disso, será abordado o método de desenvolvimento adotado, destacando a abordagem prática e acessível utilizada para a criação da solução.

Restrições

No presente trabalho, serão exemplificadas algumas das restrições fundamentais no processo de escalonamento de aulas. Para embasar a discussão, serão utilizadas as contribuições apresentadas por Suzan Kelly Borges em seu artigo intitulado "Resolução de Timetabling Utilizando Algoritmos Genéticos e Evolução Cooperativa" (Borges, 2003).

De acordo com Borges (2003), o processo de escalonamento como os de provas ou exames apresenta restrições fundamentais que devem ser consideradas. Dentre essas restrições, destacam-se duas principais: a primeira é que nenhum estudante pode realizar mais de uma prova simultaneamente. A segunda restrição consiste em assegurar que nenhuma escola ou universidade atenda a um número de alunos superior à sua capacidade suportada, de forma a garantir um ambiente adequado para a realização das avaliações.

Tecnologias

O desenvolvimento de um algoritmo eficiente para resolver o problema de timetabling desempenha um papel fundamental na otimização da organização de horários e alocação de recursos em instituições de ensino. Um código de *timetabling* bem elaborado é capaz de lidar com as complexidades inerentes a esse desafio, considerando as restrições e requisitos específicos de cada contexto (Cotta, 2014). O projeto será desenvolvido utilizando as tecnologias JavaScript. O JavaScript é uma linguagem de programação amplamente adotada, conhecida por sua flexibilidade e capacidade de interação com o navegador. Ao optar pela utilização do JavaScript, é possível desenvolver uma aplicação web robusta e de fácil utilização para a resolução do problema de *timetabling*.

IMPLEMENTAÇÃO

A implementação do algoritmo ocorrerá em etapas para garantir um desenvolvimento eficiente e progressivo. Inicialmente, será realizada a definição dos requisitos e a modelagem do problema de *timetabling*. Isso envolverá a identificação das restrições e objetivos do escalonamento, bem como a definição das informações necessárias para gerar os horários. Com base nesses requisitos, será feita a implementação das funcionalidades em JavaScript. Serão criados os componentes e a lógica necessária para interagir com a interface web. Em seguida, o algoritmo será responsável por processar essas informações e realizar os cálculos necessários para a geração dos horários. Durante todo o processo de implementação, serão realizados testes e validações para verificar a correta funcionalidade do algoritmo e identificar possíveis erros ou melhorias. Essa abordagem iterativa permitirá ajustes e refinamentos ao longo do desenvolvimento, garantindo um algoritmo eficiente e confiável. Por fim, a interface web será responsável por apresentar os horários gerados de forma clara e organizada aos usuários, facilitando a visualização e interação com as informações. Através desta aplicação, será possível otimizar a organização de horários e alocação de recursos em instituições de ensino, contribuindo para uma gestão mais eficiente e satisfatória.

METODOLOGIA

A metodologia envolve a definição de requisitos, a modelagem do problema, a implementação das funcionalidades e a realização de testes e validações ao longo do processo. Essa abordagem permite uma maior flexibilidade na adaptação do código às necessidades específicas do *timetabling*, bem como a identificação e correção de possíveis erros ou melhorias ao longo do desenvolvimento. É importante destacar que o uso do JavaScript no desenvolvimento do código de *timetabling* contribui para a criação de uma aplicação web interativa, intuitiva e de fácil utilização. A combinação dessas tecnologias permite a criação de uma interface amigável, na qual os usuários podem inserir as informações necessárias e visualizar os horários gerados de forma clara e organizada.

Levantamento dos requisitos: Realizar um estudo das necessidades e exigências das instituições de ensino em relação ao *timetabling*, identificando as principais restrições de horários, recursos disponíveis, preferências dos envolvidos e demais aspectos relevantes.

Projeto da interface do usuário: Utilizar a linguagem HTML para projetar a interface do usuário, desenvolvendo uma interface intuitiva e amigável que permita aos usuários inserir e gerenciar os dados necessários para a geração dos horários, como disciplinas, salas, professores, turmas, entre outros.

Implementação das funcionalidades em JavaScript: Desenvolver as funcionalidades da aplicação utilizando a linguagem JavaScript, aproveitando as bibliotecas e recursos disponíveis para manipulação de dados, cálculos e otimizações necessárias para a geração dos horários de forma eficiente.

Testes e validação da aplicação: Realizar testes minuciosos na aplicação, verificando a correta funcionalidade de todas as características implementadas. Testar diferentes cenários, considerando as diversas restrições e preferências definidas, a fim de garantir a precisão e a eficiência na geração dos horários.

Documentação do processo de desenvolvimento: Registrar todas as etapas do processo de desenvolvimento da aplicação, incluindo as decisões de projeto, a configuração do ambiente de desenvolvimento, a estruturação do código e a integração dos componentes front-end e back-end. Essa documentação servirá como referência para futuras atualizações e manutenção da aplicação.

Disponibilização da aplicação: Hospedar a aplicação em um servidor web ou plataforma online, garantindo que ela esteja acessível para as instituições de ensino. Configurar o ambiente de produção e realizar todos os procedimentos necessários para o correto funcionamento da aplicação em um ambiente de produção.

RESULTADO

Nesta seção, serão apresentados o resultado do software desenvolvido e as suas funcionalidades.

Cadastros

O programa oferece a funcionalidade de cadastro conforme exemplo apresentado na Figura 1, permitindo que professores, matérias, salas, dias da semana e horários de aula sejam registrados. Esses registros são armazenados de forma organizada em uma tabela, o que proporciona uma visão completa de todas as informações.

Figura 1 - Tela de cadastro de professores

Cadastro de Professores

Nome do Professor:

Disciplina:

Sala:

Dias de Aula:

Horário de Início:

--:-- -- (L)

Horário de Fim:

--:-- -- (L)

Cadastrar

Fonte: elaborada pelos autores.

O programa permite o cadastro de informações, incluindo detalhes de professores, disciplinas, salas, dias da semana e horários de aula. Os registros são armazenados em uma tabela que exibe todas as informações de maneira unificada, facilitando a gestão global dos dados.

Visualização em grade

Após o cadastro, o sistema organiza as informações conforme o exemplo da Figura 2. Os dados são separados e apresentados na grade, levando em consideração os dias da semana. Isso resulta em uma representação visual clara e organizada do horário, onde as aulas são distribuídas nos dias correspondentes. Essa abordagem aprimora a legibilidade e a acessibilidade das informações, tornando fácil para os usuários identificar os horários das aulas em dias específicos.

Figura 2 - Tela de grade curricular

Grade Curricular

Segunda-feira
Terça-feira
Quarta-feira
Quinta-feira
Sexta-feira

Fonte: elaborada pelos autores.

Armazenamento

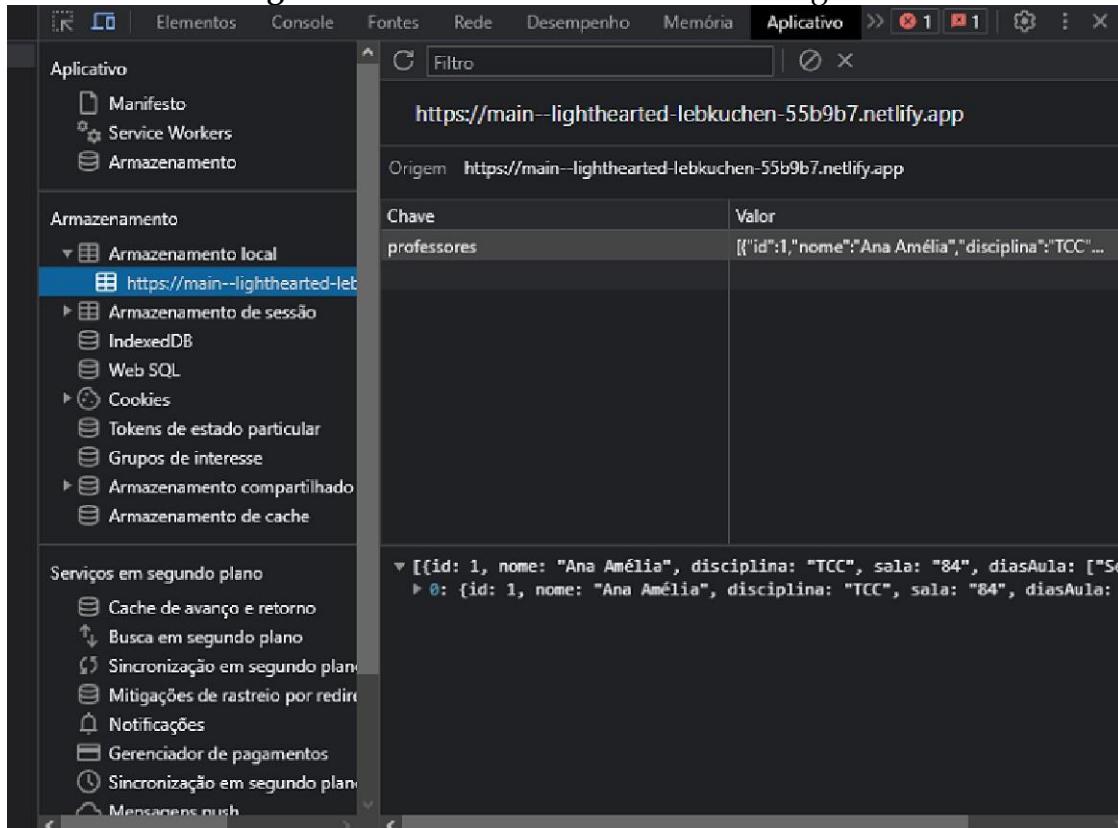
Os registros gerados no sistema são armazenados no armazenamento local do navegador, conhecido como "localStorage" conforme o exemplo da Figura 3. No entanto, vale destacar que, em ambientes de produção real, a persistência de dados em um banco de dados robusto é uma escolha preferencial e recomendada. Embora o armazenamento local seja uma solução prática para fins de demonstração e testes, como é o caso deste projeto, ele não é apropriado para o armazenamento de dados em larga escala e ambientes de produção.

Sistemas de armazenamento de dados robustos, como bancos de dados SQL e NoSQL, oferecem segurança, escalabilidade e capacidade de gerenciar grandes volumes de informações com eficiência. Portanto, ao implementar essa solução em um ambiente de produção, é altamente aconselhável migrar para um banco de dados apropriado para garantir a integridade, a disponibilidade e a escalabilidade dos dados.

A escolha de usar o armazenamento local neste projeto serviu para fins de demonstração e testes, facilitando o desenvolvimento e a validação das funcionalidades. No entanto, ao implantar essa aplicação em um ambiente de produção real, é fundamental considerar a migração para um sistema de gerenciamento de banco de dados que atenda às necessidades de armazenamento, recuperação e segurança de dados em larga escala.

No contexto do nosso projeto, é relevante ressaltar a importância do armazenamento local por meio do Local Storage. Na imagem abaixo, podemos visualizar o comportamento dos dados nesse armazenamento, o qual oferece a praticidade de armazenar informações localmente no navegador. Além disso, ele permite que os dados sejam modificados diretamente no Local Storage, facilitando os testes e experimentações.

Figura 3 - Armazenamento local do navegador



Fonte: elaborada pelos autores.

Essa funcionalidade do Local Storage torna os testes mais simples e flexíveis. Os desenvolvedores podem facilmente modificar os registros e observar como as mudanças afetam a aplicação, agilizando o processo de desenvolvimento e depuração.

Acessibilidade online

Outro aspecto significativo é que a aplicação foi hospedada online, proporcionando fácil acesso a todos os interessados. Isso significa que instituições de ensino, professores e alunos podem acessar a aplicação de qualquer lugar, a qualquer momento, para gerenciar seus horários de maneira eficaz.

Ambiente de Teste e Responsividade

Durante o desenvolvimento deste projeto, foi realizada uma bateria de testes para garantir a robustez e eficiência do software. O sistema foi avaliado nos navegadores Opera, Google-Chrome e Microsoft Edge. Além de ser submetido a uma série de testes de responsividade para garantir que a aplicação seja acessível e funcione perfeitamente em uma variedade de dispositivos, incluindo aparelhos móveis. Essa abordagem assegura que a aplicação seja utilizável e esteja disponível para uma ampla gama de usuários, independentemente do dispositivo ou navegador que escolham.

Um dos testes incluiu a utilização de uma grade curricular real do curso de Computação da instituição Unifagoc conforme é mostrado na Figura 4. Esse teste prático possibilitou a avaliação do software em um cenário real, simulando uma das necessidades de alocação de recursos enfrentadas pelas instituições de ensino.

Figura 4 - Grade curricular com dados reais preenchidos

Grade Curricular

Segunda-feira	Aula: Professor 1, Disciplina: Fundamentos matematica, Sala: sala 1, Inicio: 19:00, Termino: 20:40	Aula: Professor 2, Disciplina: Sistemas de informação, Sala: sala 1, Inicio: 21:00, Termino: 22:40	Aula: Professor 4, Disciplina: Redes, Sala: sala 2, Inicio: 20:00, Termino: 20:40
	Aula: professor 2, Disciplina: S.O, Sala: sala 2, Inicio: 19:00, Termino: 19:50	Aula: Professor 4, Disciplina: Redes, Sala: sala 2, Inicio: 21:00, Termino: 22:40	
Terça-feira	Aula: Professor 1, Disciplina: Fundamentos matematica, Sala: sala 1, Inicio: 19:00, Termino: 20:40	Aula: professor 3, Disciplina: Estrutura de dados, Sala: sala 1, Inicio: 21:00, Termino: 22:40	Aula: professor 6, Disciplina: organização, Sala: sala 2, Inicio: 21:00, Termino: 22:40
	Aula: professor 5, Disciplina: Estatística, Sala: sala 2, Inicio: 19:00, Termino: 20:40	Aula: professor 6, Disciplina: organização, Sala: sala 2, Inicio: 21:00, Termino: 22:40	

Fonte: elaborada pelos autores.

Além disso, a aplicação inclui um mecanismo de validação que impede o registro de horários conflitantes para um mesmo professor conforme é mostrado na Figura 5. Por exemplo,

se houver uma tentativa de agendar um professor para o mesmo horário de outra disciplina no mesmo dia, o software irá barrar esse registro, garantindo a integridade da grade horária.

Figura 5 - Simulação de conflito de horário

Cadastro de Professores

Nome do Professor:
professor 2

Disciplina:
tcc

Sala:
sala

Dias de Aula:
Segunda-feira

Horário de Início:
01:00 AM

Horário de Fim:
02:00 AM

Cadastrar

O professor já possui uma aula agendada no mesmo horário ou sala.

Fonte: elaborada pelos autores.

Codificação da validação de horários

A função verificarConflitoHorario exemplificada no Código 1, desempenha um papel fundamental no sistema, garantindo que não ocorram conflitos indesejados durante o cadastro de horários de aula. Através dessa função, é possível verificar se existem situações em que os horários de dois professores se sobreponem.

- Ela recebe como entrada o nome do professor, a sala onde a aula será ministrada, os dias da semana em que a aula ocorrerá, bem como o horário de início e término da aula.
- Em seguida, a função itera sobre a lista de professores já cadastrados (professores) para verificar se existe algum professor que já tenha sido cadastrado e cujos horários colidem com os novos horários informados. Isso é realizado através de uma série de condições.
 - A função verifica se o nome do professor ou a sala de aula coincidem com os já cadastrados e se há dias da semana em comum entre o novo horário e os horários já existentes do professor.
 - Além disso, a função também verifica se há sobreposição nos horários. Ou seja, ela analisa se o horário de início do novo cadastro está dentro do intervalo de horários já cadastrados do professor, se o horário de término está dentro desse intervalo ou se o novo horário abrange totalmente o horário já cadastrado do professor.

Código 1 - Trecho de código da validação de horários

```

function verificarConflitoHorario(nome, sala, diasAula, inicio, fim) {for (const professor of professores) {
if (professor.id !== undefined && (
(professor.nome === nome || professor.sala === sala)
&&
professor.diasAula.some(dia => diasAula.includes(dia))
&&
((inicio >= professor.inicio && inicio <
professor.fim) ||
|| professor.fim))
)) {

(fim > professor.inicio && fim <= professor.fim)

(inicio <= professor.inicio && fim >=
return true;
}
}
return false;
}

```

Fonte: elaborada pelos autores.

Se qualquer uma dessas condições for satisfeita, a função retorna *true*, indicando que há um conflito de horário. Caso contrário, se nenhuma correspondência for encontrada e nenhum conflito for identificado, a função retorna *false*, indicando que o novo horário pode ser cadastrado sem problemas

MELHORIAS

Para projetos futuros, tanto para quem deseja utilizar este artigo como referência quanto para aqueles que desejam continuar o desenvolvimento da aplicação, existem

algumas áreas que podem ser aprimoradas:

Layout da Grade Curricular: Uma das áreas que podem ser aprimoradas é o layout da grade curricular. É possível explorar opções de design e apresentação que tornem a visualização dos dados ainda mais organizada e intuitiva. Uma representação mais atraente e informativa dos horários das aulas pode facilitar a compreensão e o uso da aplicação.

Persistência de Dados em um Banco de Dados: Embora o armazenamento local (LocalStorage) tenha sido utilizado neste projeto como uma solução prática para demonstração e testes, a transição para um sistema de gerenciamento de banco de dados é uma melhoria importante para um ambiente de produção real. O uso de um banco de dados robusto oferece vantagens em termos de segurança, escalabilidade e capacidade de gerenciar grandes volumes de informações de forma eficiente. A migração para um banco de dados também permite a recuperação e a persistência de dados confiáveis em cenários de uso real.

Essas possíveis melhorias visam tornar a aplicação mais adequada para ambientes de ensino reais, onde a otimização de horários e alocação de recursos é uma prioridade constante. Portanto, os desenvolvedores que desejarem utilizar este projeto como ponto de partida têm a oportunidade de aprimorar a aplicação e adaptá-la às necessidades específicas de suas instituições de ensino ou contextos de uso.

CONCLUSÃO

Neste projeto, foi desenvolvida uma aplicação web para solucionar os desafios do *timetabling* em instituições de ensino. Utilizou-se das tecnologias HTML, CSS e JavaScript para criar uma ferramenta prática e acessível que permitisse a organização eficiente de horários e a alocação de recursos.

O desenvolvimento desta aplicação web para timetabling representa um passo significativo na resolução dos desafios enfrentados pelas instituições de ensino na organização de horários e alocação de recursos. Ao longo deste projeto, foram estabelecidos objetivos específicos para criar uma ferramenta prática e acessível que atendesse às necessidades específicas das instituições educacionais.

O sistema de cadastro de professores, disciplinas, salas, dias da semana e horários de aula oferece aos usuários a possibilidade de inserir dados essenciais para a criação dos horários de forma flexível e amigável. A tabela de registros e a grade de horários resultante tornam a visualização e o acompanhamento das alocações mais simples e eficientes.

A escolha de tecnologias como HTML, CSS e JavaScript permitiu o desenvolvimento de uma aplicação web de fácil utilização. A combinação dessas tecnologias possibilitou uma interface interativa e intuitiva para os usuários, enquanto o armazenamento de dados no local storage do navegador garante que as informações estejam acessíveis a qualquer momento.

Além disso, a disponibilização da aplicação online¹ torna o acesso e a utilização da ferramenta acessíveis a todos os interessados, independentemente da localização geográfica.

O código-fonte do projeto foi disponibilizado em um repositório público no GitHub, incentivando a colaboração, revisão e aprendizado contínuo. Desenvolvedores e pesquisadores interessados podem acessar o código-fonte em Repositório GitHub² para obter informações detalhadas sobre a implementação.

Em resumo, a aplicação web para *timetabling* é uma solução valiosa para as instituições de ensino que desejam otimizar a organização de horários e alocação de recursos. Esperamos que este projeto possa contribuir para tornar o processo de *timetabling* eficiente e satisfatório, beneficiando tanto os educadores quanto os alunos.

REFERÊNCIAS

BORGES, Suzan Kelly. **Resolução de timetabling utilizando algoritmos genéticos e evolução cooperativa**. Disponível em: <https://core.ac.uk/download/pdf/147519892.pdf>. Acesso em 19 jun. 2023.

COTTA, Kleyton Pontes. **Ferramenta de detecção de conflitos e gestão para o timetabling problem**. Rio de Janeiro: Universidade Federal Rural do Rio de Janeiro, Instituto Multidisciplinar. Disponível em: <https://www.cc.ufrrj.br/wp-content/uploads/2021/12/TCC-Kleyton-Pontes-Cotta1.pdf>. Acesso em: 14 jun. 2023.

HIME, Rodrigo; ALBUQUERQUE, Jones; GONÇALVES, Glauco. **Uma aplicação da programação inteira no school timetabling problem**. Disponível em: <http://200.17.137.109:8081/novobsi/bsina-ufrpe-recife/trabalhos-de-conclusao-de-curso/2014.2/Hime.pdf>. Acesso em: 8 jun. 2023

¹ <https://main--lighthearted-lebkuchen-55b9b7.netlify.app>

² <https://github.com/Fri5Day/TCC>