

João Vítor Fernandes Dias

***Timetabling Problem: desafios no
desenvolvimento de um sistema de decisão
voltado ao problema de organização de tabela
de horários no ensino superior***

Campos dos Goytacazes, RJ

10 de maio de 2024

João Vítor Fernandes Dias

Timetabling Problem: desafios no desenvolvimento de um sistema de decisão voltado ao problema de organização de tabela de horários no ensino superior

Monografia apresentada ao Curso de Graduação em Ciência da Computação da Universidade Estadual do Norte Fluminense Darcy Ribeiro, sob orientação do Prof. Dr. Fermín Alfredo Tang Montané

Orientador: Fermín Alfredo Tang Montané

Campos dos Goytacazes, RJ

10 de maio de 2024

João Vítor Fernandes Dias

Timetabling Problem: desafios no desenvolvimento de um sistema de decisão voltado ao problema de organização de tabela de horários no ensino superior

Monografia apresentada ao Curso de Graduação em Ciência da Computação da Universidade Estadual do Norte Fluminense Darcy Ribeiro, sob orientação do Prof. Dr. Fermín Alfredo Tang Montané

Trabalho aprovado. Campos dos Goytacazes, RJ, XX de Junho de 2024:

Fermín Alfredo Tang Montané
Orientador

Professor
Convidado 1

Professor
Convidado 2

Campos dos Goytacazes, RJ
10 de maio de 2024

Dedico este trabalho a todos aqueles que me apoiaram, incentivaram e cobraram pela conclusão do meu TCC. Àqueles que vieram antes de mim, Sânya e Ricardo, que inicialmente pesquisaram e desenvolveram seus trabalhos sobre o mesmo tema. À UENF, ao CCT e ao Curso de Ciência da Computação, que me guiaram na jornada de aprendizado e crescimento acadêmico e que poderão usufruir das semestres que planei, caso vinguem. Mas principalmente dedico a todos os futuros alunos e pesquisadores, que inconformados com o *status quo*, buscam sempre melhorar a si mesmos e ao mundo ao seu redor.

AGRADECIMENTOS

Agradeço aos meus pais que se dedicaram para que eu pudesse estar cursando esta graduação, assim podendo completar mais uma etapa da minha vida. Sem o apoio, conselhos, carinho e amor, nada disso seria possível. Sou eternamente grato por tudo que vocês fazem e sempre fizeram para que minha vida fosse especial.

Agradeço aos meus amigos e colegas de curso que compartilharam comigo bons momentos, risadas e refeições no RU. Em especial, aos meus grandes amigos Daniel Brito e José Lucio, que unidos pela Iniciação Científica e pela progressão contínua, seguimos juntos até o fim, também aos meus amigos João Víttor e Adriano, que entraram comigo no curso e que, apesar dos percalços, seguem sendo grandes companhias pra vida.

Agradeço ao meu orientador, Prof. Dr. Fermín Alfredo Tang Montané, que se dispôs a me ouvir reclamar e propor ideias. Que se dispôs a me orientar mesmo que em suas férias, e que sempre esteve disponível para me ajudar.

*“Quando você não sabe alguma coisa,
procura quem sabe e pergunta.
Pra não fazer feio, nem passar vergonha.*

Vovô Nando, 01/12/2023

RESUMO

Esta monografia apresenta um estudo sobre a criação de grades horárias em universidades, com foco na Universidade Estadual do Norte Fluminense, e o desenvolvimento de um sistema de suporte à decisão para a criação de grades horárias. Este processo é complexo e envolve diversos fatores, como a disponibilidade de salas e professores, a quantidade de alunos e a demanda por disciplinas. Como uma das maiores dificuldades no ramo acadêmico da criação de grades horárias é a modelagem do problema, este trabalho se propõe a estruturar as informações referentes à instituição estudada através de pesquisas e entrevistas. A partir disso, foi possível identificar a sequência de criação das grades e as restrições impostas pela instituição, podendo assim propor um sistema que auxilie o processo. Para tanto, o sistema foi desenvolvido utilizando JavaScript e a biblioteca React, sendo hospedado no GitHub Pages e utilizando da Amazon Web Services para a execução de seu *backend* e armazenamento de dados. Suas funcionalidades incluem as funcionalidades CRUD (criar, ler, atualizar e deletar) de professores, disciplinas, salas, alunos, turmas e horários, a criação automatizada de uma grade horária inicial para o curso de Ciência da Computação da UENF, a visualização de conflitos entre os recursos alocados e a análise história dos dados armazenados no sistema.

Palavras-chave: Tabela de Horários. Agendamento de Aulas Universitárias. Heurísticas. Organização do Conhecimento. Interação Humano-computador.

ABSTRACT

Esta monografia apresenta um estudo sobre a criação de grades horárias em universidades, com foco na Universidade Estadual do Norte Fluminense, e o desenvolvimento de um sistema de suporte à decisão para a criação de grades horárias. Este processo é complexo e envolve diversos fatores, como a disponibilidade de salas e professores, a quantidade de alunos e a demanda por disciplinas. Como uma das maiores dificuldades no ramo acadêmico da criação de grades horárias é a modelagem do problema, este trabalho se propõe a estruturar as informações referentes à instituição estudada através de pesquisas e entrevistas. A partir disso, foi possível identificar a sequência de criação das grades e as restrições impostas pela instituição, podendo assim propor um sistema que auxilie o processo. Para tanto, o sistema foi desenvolvido utilizando JavaScript e a biblioteca React, sendo hospedado no GitHub Pages e utilizando da Amazon Web Services para a execução de seu *backend* e armazenamento de dados. Suas funcionalidades incluem as funcionalidades CRUD (criar, ler, atualizar e deletar) de professores, disciplinas, salas, alunos, turmas e horários, a criação automatizada de uma grade horária inicial para o curso de Ciência da Computação da UENF, a visualização de conflitos entre os recursos alocados e a análise história dos dados armazenados no sistema.

This monograph presents a study on the creation of timetables in universities, focusing on the Universidade Estadual do Norte Fluminense, and the development of a decision support system for the creation of timetables. This process is complex and involves several factors, such as the availability of classrooms and teachers, the number of students and the demand for disciplines. As one of the greatest difficulties in the academic field of creating timetables is the modeling of the problem, this work proposes to structure the information regarding the institution studied through research and interviews. From this, it was possible to identify the sequence of creating the timetables and the restrictions imposed by the institution, thus being able to propose a system that assists the process. For this, the system was developed using JavaScript and the React library, being hosted on GitHub Pages and using Amazon Web Services for the execution of its backend and data storage. Its features include the CRUD (create, read, update and delete) functionalities of teachers, disciplines, classrooms, students, classes and schedules, the automated creation of an initial timetable for the Computer Science course at UENF, the visualization of conflicts among the allocated resources and the historical analysis of the data stored in the system.

Keywords: Timetabling. University Class Scheduling. Heuristics. Integer Programming. Knowledge Organization. Human Computer Interaction.

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Resumo de trabalhos, parâmetros, dimensões, tempo e técnicas	20
Figura 2.2 – Comparação entre artigos que solucionam o problema de grade horária	21
Figura 2.3 – Análise de publicações aceitas	21
Figura 2.4 – Grade horária do LECIV para 2017.2 e 2018.2	24
Figura 2.5 – Disciplina atribuída no sistema acadêmico à determinada hora e local . .	25
Figura 2.6 – Falha de alocação na grade horária do CCT de 2023.1	26
Figura 4.1 – Estágios para a obtenção de grade horária ótima	41
Figura 4.2 – Etapas do Design de Interação	42
Figura 4.3 – Diagrama Conceitual do banco de dados	44
Figura 5.1 – Andamento do aluno no Sistema Acadêmico	48
Figura 5.2 – Protótipos de cartões de turma	52
Figura 5.3 – Protótipos de caixas de seleção	52
Figura 5.4 – Protótipo da página principal do sistema	53
Figura 5.5 – Protótipo da página de seleção	54
Figura 5.6 – Protótipo da página de salas	55
Figura 5.7 – Protótipo da página de alunos	55
Figura 5.8 – Protótipo da página de disciplinas	56
Figura 5.9 – Protótipo da página de professores	57
Figura 5.10–Protótipo da página de turmas	57
Figura 5.11–Recursos usados para o desenvolvimento do sistema	58
Figura 5.12–Diagrama da progressão funcionamento da permanência dos dados . . .	59
Figura 5.13–Diagrama do armazenamento preliminar dos dados	60
Figura 5.14–Tabela de tarefas do GitHub Projects	62
Figura 5.15–Comparação entre bancos de bados da Versão 2.0	63
Figura 5.16–Diagrama inicial das tabelas de dados SQL	65
Figura 5.17–Logomarcas do sistema	66
Figura 5.18–Logomarca oficial	67
Figura 5.19–Marcos concluídos do GitHub Projects	69
Figura 5.20–Marcos abertos do GitHub Projects	70
Figura 5.21–Etiquetas do GitHub Projects	71
Figura 5.22–Gráfico de Marco <i>versus</i> quantidade de tarefas separadas por etiqueta .	71
Figura 5.23–Diagrama da progressão funcionamento da permanência dos dados . .	72
Figura 5.24–API REST no AWS	73
Figura 5.25–Novo diagrama de banco de dados	79
Figura 5.26–Paleta de cores do sistema	84

Figura 5.27–Exemplo de conflito nulo	84
Figura 5.28–Exemplo de conflito de alocação de professor	85
Figura 5.29–Exemplo de conflito de alocação de sala	86
Figura 5.30–Exemplo de conflito de capacidade na sala	87
Figura 5.31–Avisos flutuantes dos conflitos de disciplinas	88
Figura 5.32–Diagrama do fluxo de obtenção de dados	89
Figura 5.33–Diagrama entidade relacionamento final	91
Figura 6.1 – Página inicial do sistema	97
Figura 6.2 – Página de multiturmas com filtros	98
Figura 6.3 – Página de multiturmas com conflitos	98
Figura 6.4 – Página de multiturmas com disciplinas pendentes	99
Figura 6.5 – Página de grade de horários	99
Figura 6.6 – Página de turmas	100
Figura 6.7 – Página de professores	100
Figura 6.8 – Página de salas	101
Figura 6.9 – Página de disciplinas	101
Figura 6.10–Página de alunos	102

LISTA DE TABELAS

Tabela 6.1 – Calendário Acadêmico da SECACAD de 2023.1 (simplificado)	94
Tabela 6.2 – Calendário Acadêmico da SECACAD de 2023.2 (simplificado)	94
Tabela 6.3 – Calendário Acadêmico de 2023 (simplificado)	95
Tabela 6.4 – Calendário Acadêmico de 2023 - Alterado	96
Tabela A.1–Número de respondentes por curso	112
Tabela A.2–Número de respondentes por ano	112
Tabela A.3–Ocorrência de experiências acadêmicas	113
Tabela A.4–Preferências por distribuição de disciplinas em um mesmo dia	115
Tabela A.5–Preferências por distribuição de disciplinas em um mesmo dia	115
Tabela A.6–Critérios para a escolha de disciplinas conflituosas	115
Tabela A.7–Tempo de atraso em disciplinas	116
Tabela A.8–Notas dadas às características da distribuição de disciplinas	117

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Problemáticas	14
1.2	Hipótese	15
1.3	Objetivos	15
1.4	Justificativas	16
1.5	Metodologia	16
1.6	Organização	17
2	MARCO TEÓRICO	18
2.1	Definição de termos	18
2.2	Métodos de resolução	19
2.3	Desafios recorrentes	22
2.4	<i>Timetabling</i> na UENF	22
2.4.1	Sânya Carvalho dos Santos	22
2.4.2	Ricardo Couto da Silveira	23
2.4.3	LECIV	23
2.5	Exemplo de erros humanos	24
3	ESTRUTURA DA INSTITUIÇÃO	27
3.1	A UENF	27
3.1.1	Responsáveis pela criação das grades horárias	29
3.2	Entrevistas	30
3.2.1	Direção do CCT	30
3.2.2	Desenvolvedor do Sistema Acadêmico	32
3.2.3	Chefia de Laboratório de Matemática	33
3.2.4	Responsável pela Secretaria Acadêmica (SECACAD)	34
3.2.5	Coordenação de Computação	35
3.2.6	Entendimento geral das entrevistas	36
3.3	Sequência de criação das grades horárias	37
3.4	Formulário quantitativo aos discentes	38
3.5	Relações entre as variáveis	38
4	MODELAGEM GERAL DO SISTEMA	41
4.1	Estágios de execução	41
4.2	Iteração	42
4.3	Funcionamento	42

4.4	Modelo de banco de dados	44
4.5	API REST	45
4.6	Diferença dos trabalhos anteriores	45
5	DESENVOLVIMENTO DO SISTEMA DE SUPORTE À DECISÃO	47
5.1	Projetos anteriores	47
5.1.1	Andamento dos alunos	48
5.1.2	Cálculo de demanda	49
5.2	Acesso aos dados acadêmicos da instituição	50
5.3	Prototipagem	51
5.3.1	Protótipos de componentes	51
5.3.2	Protótipos de páginas	53
5.4	Programação do sistema	58
5.4.1	Versão 1.0	59
5.4.2	Versão 2.0	61
5.4.3	Versão 3.0	68
5.5	Solução inicial	80
5.5.1	Heurística de solução inicial	80
5.6	Detectção e alerta dos conflitos	82
5.6.1	Típicos conflitos atípicos	82
5.6.2	Conflitos tratados pelo sistema	83
5.7	Preenchimento de dados	88
5.8	Próximos desenvolvimentos	90
5.8.1	Funcionalidades preparadas	91
6	RESULTADOS	93
6.1	Alternativas burocráticas	93
6.1.1	Tempo de elaboração das grades	93
6.1.2	Alteração forçada de horários	94
6.1.3	Aplicação prática dos métodos burocráticos	94
6.2	Sistema desenvolvido	96
6.2.1	Páginas	96
6.2.2	Funcionalidades	102
7	CONCLUSÕES	104
7.1	Trabalhos futuros	105
	REFERÊNCIAS	108
	APÊNDICE A – RESULTADOS DO FORMULÁRIO	111

APÊNDICE B – FORMULÁRIO DE PESQUISA	120
APÊNDICE C – CÓDIGO-FONTE DA MONOGRAFIA	124
APÊNDICE D – EXEMPLO DE <i>TEMPLATE.YAML</i>	125

1 INTRODUÇÃO

No ensino superior brasileiro, cada curso de uma instituição de ensino tem em seu projeto pedagógico, ou seja, no documento que rege quais as atribuições e justificativas de existência do curso, uma listagem de disciplinas a serem ministradas em cada semestre ao longo de sua duração esperada. Disciplinas estas que para serem cursadas os discentes precisam cumprir determinados requisitos.

Embora haja o planejamento de duração do curso, diversos fatores podem influenciar a previsão, dentre eles podemos citar eventos como: quebra de pré-requisitos, trancamento de matrícula, transferências, reprovações, indisponibilidade de professores, greves, dentre tantos outros.

Estes eventos tendem a, no geral, aumentar o tempo médio para conclusão do curso. Situação em sua maioria indesejada tanto pelos alunos, que mesmo durante seu estudo já visam o mercado de trabalho, quanto pelos professores e pela instituição, visto que a evasão do ensino superior brasileiro é um problema existente e estudado a fim de ser minimizado.

Com isso, é esperado que a instituição busque alternativas para tornar mais dinâmica e atrativa a experiência dos discentes durante sua jornada. Uma dessas formas é tentando minimizar o impacto que os atrasos na grade causam nos semestres consecutivos. Para isso sendo então necessária uma análise das disciplinas que devem ser oferecidas no próximo semestre, sendo então necessário definir **quais, quando, onde, por quem e para quem** serão ministradas. Esta tarefa, entretanto, não é trivial.

1.1 Problemáticas

Embora seja um problema atualmente, isso não significa que seja recente. Desde 1978 ([BARHAM; WESTWOOD, 1978](#)) o termo *timetabling* encontra-se no meio acadêmico como o termo referente ao tabelamento de grade horária, sendo assim, é este o termo que será principalmente utilizado neste trabalho. Neste artigo de 1978 já se propunha uma forma para que se obtivesse um tabelamento otimizado, e demonstrava que o método utilizado gerava bons resultados.

Outra característica é informada por [Thomas, Khader e Belaton \(2009\)](#) que fala sobre a multidimensional do problema de *timetabling*. Por causa dessa questão há uma complexidade elevada para conseguir conceber visual e mentalmente de que forma os dados relacionados ao problema se estruturam, assim dificultando a elaboração de sistemas computacionais que auxiliem nessa tarefa.

Também segundo [Miranda, Rey e Robles \(2012\)](#), embora o problema de atribuição de salas não seja novo e tenha extensa literatura a seu respeito, são poucos os que de fato implementaram um sistema para suporte de decisões. Isso se dá por diversos fatores, também listado pelo autor fazendo referência a trabalhos anteriores, sendo alguns deles a resistência organizacional a mudanças e adoção de novas tecnologias, nível de dificuldade do problema, dentre outros.

Algumas outras características que se apresentam como problemas são a falta de otimalidade das grades horárias desenvolvidas em boa parte das instituições de ensino superior e a quantidade de tempo necessária para a criação dessas grades não-ótimas.

Considerando que situações como a descrita acima são passíveis de ocorrer, e que a tarefa de criação de grades horárias é recorrente, um sistema de suporte à decisão que supra às necessidades dos seus usuários se faz necessário.

1.2 Hipótese

Dada as características intrínsecas ao problema de agendamento de grade horária, é esperado que os *softwares* atualmente existentes que lidam com este problema não apresentem completas capacidades de se moldar ao caso de uma instituição específica.

E, caso a primeira hipótese se apresente correta, o *software* a ser desenvolvido, assim como seus similares, se apresentará como uma solução plausível para a resolução do problema proposto embora ainda apresente melhorias possíveis a serem implementadas. O *software* se apresentará de tal forma que os *stakeholders* que, esperadamente, decidirem não o utilizar não causarão a impossibilidade do uso do sistema.

1.3 Objetivos

Os objetivos desta monografia podem ser divididos entre gerais e específicos, não havendo relação de superioridade de um em relação ao outro, visto que ambos igualmente nortearão o desenvolvimento da pesquisa.

Como **objetivos gerais**, espera-se conseguir desenvolver um sistema de suporte à decisão tal que aumente a eficiência, eficácia e efetividade do processo de criação de grades horárias que semestralmente demandam extensa quantidade de tempo dos coordenadores de curso na UENF e não alcançam a otimalidade. Nesse processo, também é esperado que as grades horárias finais tragam benefícios aos alunos como forma de mais disciplinas à sua disposição. Visto que estes muitas vezes lidam com grades horárias que não contemplam suas reais demandas. Dessa forma aumentando a satisfação de todos os participantes do processo, desde os coordenadores de curso até os alunos.

Como **objetivos mais específicos**, podemos listar os seguintes:

- Entender de que forma os setores administrativos da UENF atualmente lidam com a questão do *timetabling*;
- Obter os aprimoramentos desejados pelos responsáveis na criação de grades horárias;
- Modelar o sistema de *timetabling* de acordo com os requisitos demandados;
- Incentivar o uso de uma ferramenta centralizada para a otimização do *Timetabling Problem*.

1.4 Justificativas

Levando em conta a problemática evidenciada e os sucessos prévios dos artigos anteriores, vê-se grande potencial de auxílio e aumento na satisfação de todos os que utilizarem os métodos propostos. Não havendo um sistema geral que solucione todos os casos como evidenciado pelos pesquisadores da área, resta aos interessados rumarem em busca de uma solução entalhada nos moldes de sua instituição específica. Considerando que é um problema existente atualmente e que uma solução está disponível, o que se torna necessário é realizar o esforço inicial suficiente para que ocorra a quebra da inércia em que se encontram os processos ineficientes usuais para assim alcançar um melhor. Sendo assim, faz-se válida a pesquisa e desenvolvimento de um *software* que vise tal propósito.

1.5 Metodologia

Considerando as dificuldades encontradas em trabalhos anteriores, entende-se que o maior desafio será superar as especificidades que serão encontradas durante a modelagem da universidade em questão. Para isso, será inicialmente necessária uma pesquisa bibliográfica com foco no estudo das abordagens qualitativas realizadas anteriormente que obtiveram sucesso em eliciar os requisitos adequados para as instituições de ensino.

Com este conhecimento, um material inicial para a pesquisa exploratória e qualitativa deve ser desenvolvido levando em conta as questões próprias da universidade em questão, visando também coletar dados relevantes para uma futura pesquisa com maior enfoque em características emergentes que a pesquisa anterior pode levantar, similar a como foi proposto e realizado por [Andre e Dinata \(2018\)](#). Esta pesquisa exploratória sobre a realidade da instituição se subdivide em três frentes: o estudo sobre a **documentação teórica, entrevista** com os *stakeholders* e **formulário** direcionado aos alunos.

Sendo compreendido então o cenário atual da universidade, será então necessário modelar o sistema de suporte à decisão de acordo com as especificidades encontradas.

Por fim, será apresentado o processo do desenvolvimento do sistema, quais foram as suas versões, quais funcionalidades foram desenvolvidas e quais foram as tecnologias utilizadas.

1.6 Organização

Este trabalho abordará capítulos que de forma resumida lidam com os seguintes tópicos:

- O Capítulo 1 na página 14 de introdução traça informações gerais sobre o assunto do trabalho, elaborando mais detalhadamente quanto à sua problemática, hipótese, objetivos, justificativas, a metodologia escolhida e a organização de suas informações;
- O Capítulo 2 na página 18 de revisão literária informa mais detalhadamente sobre os problemas de agendamento, suas categorias, soluções, desafios e definições de termos;
- O Capítulo 3 na página 27 de contexto da instituição apresenta a Universidade Estadual do Norte Fluminense Darcy Ribeiro (UENF), suas características, estrutura organizacional, entrevistas com os *stakeholders* relacionados à criação de grades horárias e a sequência de passos para a criação de uma grade horária;
- O Capítulo 4 na página 41 de modelagem, apresenta-se a conceitualização macro de como o sistema deve se comportar, quais são as suas funcionalidades e quais são os seus objetivos;
- O Capítulo 5 na página 47 de desenvolvimento, pode ser visto o processo de criação do sistema, partindo dos projetos anteriores, passando pela prototipação, para então chegar ao desenvolvimento em três etapas do sistema final;
- O Capítulo 6 na página 93 de resultados demonstra um apanhado geral do que foi aprendido e que se obteve como produto final da elaboração deste trabalho;
- O Capítulo 7 na página 104 da conclusão e trabalhos futuros finaliza o presente trabalho com os pensamentos gerais sobre a pesquisa e desenvolvimento, apresentando as características não abordadas e indicando caminhos a serem seguidos por pesquisadores posteriormente.

2 MARCO TEÓRICO

Antes de prosseguirmos com o desenrolar deste trabalho, é adequado que primeiro definamos alguns parâmetros para o melhor entendimento do que está por vir.

2.1 Definição de termos

Ao longo dos anos de desenvolvimento acadêmico, diversos assuntos vão se aprofundando e se tornando mais específicos, assim, os estudiosos acabam cunhando novos termos que o auxiliam a desvencilhar as novas áreas específicas das suas áreas originárias. Porém, existe o potencial de que haja um crescimento desestruturado destes novos termos, assim vários termos diferentes podem se referir a um mesmo conceito, enquanto que um mesmo termo pode se referir a vários conceitos diferentes de acordo com o autor.

Segundo [Wren \(1996\)](#), podemos definir *timetable* como uma estrutura que mostra quando que eventos ocorrerão, não havendo necessariamente a alocação de recursos. Vale ressaltar que este termo não tem seu uso limitado para os fins desta pesquisa, sendo também usado para problemas de alocação de enfermeiros, esportes, funcionários e transportes ([ARRATIA-MARTINEZ; PADRÓN; TORRES, 2021](#)). Entretanto, neste trabalho, abordaremos principalmente os termos relacionados ao que pode ser chamado de *Educational Timetabling* (Ed-TT) ([ALENCAR et al., 2019](#)), que é o que tende a envolver um conjunto específico de recursos relacionados à educação.

[Wren \(1996\)](#) também define os conceitos para *class timetable*, *university examination timetable* e *university class timetable*, tendo relevância apenas o último, que considera a disponibilidade de professores e salas, a quantidade de alunos e os requisitos que determinada disciplina exige.

Assim como feito por [Wren \(1996\)](#), definiremos os conceitos dos termos que serão usados ao longo deste trabalho. Sendo assim, usaremos a definição do termo *university class timetable* de forma simplificada, sendo chamada apenas de *timetable*, “grade horária” ou “tabela horária”.

Aqui, visto que uma solução final envolverá várias dimensões (Professores × Disciplinas × Sala × Alunos × Horários × Dias), consideraremos *timetable* como esse pacote de valores distribuídos em uma só estrutura. Para que esses valores sejam distribuídos, daremos o nome de **alocação** ao ato de criar qualquer relação entre as dimensões. Como a relação de horários e dias será considerada fixa, a **alocação** se referirá à atribuição como a de professores a disciplinas, disciplinas a salas, disciplinas a um determinado padrão de dias e horários, etc.

Para que esta alocação ocorra, é necessário atender a certos critérios, e aí entra o “problema de organização de grade horária”, também chamado de *timetabling problem*. Esta é uma subcategoria do **problema de agendamento** (*scheduling Optimization Problem*) (ALENCAR et al., 2019) que por sua vez é definido por Wren (1996) como sendo:

Resolver problemas práticos relacionados à alocação, sujeito a restrições, de recursos a objetos sendo colocados no espaço-tempo, usando ou desenvolvendo quaisquer ferramentas que possam ser apropriadas. Os problemas irão frequentemente se relacionar à satisfação de certos objetivos.

Outro termo relevante a se pontuar são as *hard and soft constraints* que podemos chamar de restrições rígidas e flexíveis. Alencar et al. (2019) as define dizendo que as restrições rígidas são de atendimento obrigatório, enquanto as restrições flexíveis são opcionais, mas convenientes para melhorar a qualidade da solução obtida.

Exemplo de restrição rígida: nem professores nem alunos podem ser alocados simultaneamente a duas salas ou disciplinas simultaneamente. Uma solução que viole esta restrição se torna automaticamente inviável.

Exemplo de restrição flexível: professor J. prefere não dar aulas nas tardes de sexta-feira, e prefere dar aula nas manhãs da segunda-feira. Uma solução que viole esta restrição não se torna inviável, porém tende a ter menos valor neste critério do que uma solução que siga as preferências definidas.

Alguns outros termos similares a este campo de pesquisa encontrados na literatura são *periodic event scheduling problem*, *timetable scheduling*, *class scheduling*, *student scheduling*, *university course timetabling*, dentre outros.

2.2 Métodos de resolução

Existem diversas implementações já realizadas, utilizando uma miríade de métodos. Em seu trabalho, Miranda, Rey e Robles (2012) informam sobre diversos sistemas baseados em computador para auxiliar na tarefa de agendamento. Miranda, Rey e Robles (2012) também citam um dos métodos de resolução como sendo o **modelo de programação inteira e heurísticas**.

Outros trabalhos buscaram condensar em forma de tabela as informações encontradas. Abaixo estão dispostas algumas das tabelas encontradas durante o estudo bibliográfico e que foram elaboradas por diversos autores.

Na Figura 2.1, Poulsen e Bandeira (2012) traça a relação entre os diversos autores, ano de sua publicação e seu país de origem com os dados encontrados em seus trabalhos quanto aos parâmetros utilizados na elaboração da grade horária, quão grandes eram cada

um de seus parâmetros, quanto tempo foi necessário para achar uma solução e quais foram as técnicas utilizadas.

Figura 2.1 – Resumo de trabalhos, parâmetros, dimensões, tempo e técnicas.

Autores	Ano	País	Parâmetros				Dimensões			Técnica								
			Turmas	Disciplinas	Professores	Locais de aula	Turmas	Professores	Locais de aula	Aulas	MIP	Heurísticas	TS	GA	GRASP	SA	CLP	NN
1 Gotlieb	1962	Canadá	✓				111	93	107	NI	NI	✓						
2 Lawrie	1969	Inglaterra	✓	✓	✓		NI	74		NI	NI	✓						
3 De Werra	1970	Canadá	✓	✓	✓		48	84		NI	50 min*	✓						
4 Gans	1981	Holanda	✓	✓	✓	✓	NI	NI	NI	NI	NI	✓						
5 Abramson	1991	Austrália	✓		✓	✓	101	37	24	3030	14 h				✓			
6 Alvarez-Valdes <i>et al</i>	1996	Espanha	✓		✓	✓	26	60	NI	1100	NI		✓					
7 Wright	1996	Inglaterra	✓	✓	✓		NI	80		NI	NI	✓	✓					
8 Birbas <i>et al.</i>	1997	Grécia	✓	✓	✓		21	46		721	NI	✓						
9 Colomi <i>et al.</i>	1998	Itália	✓	✓	✓		10	24		300	8 h	✓	✓	✓	✓			
10 Schaerf	1999b	Itália	✓		✓		38	61		1368	4,5 h		✓					
11 Smith <i>et al.</i>	2003	Austrália	✓		✓	✓	8	8	8	240	7,2 min							
12 Valouxis; Housos	2003	Grécia	✓	✓	✓		9	23		315	1 h							
13 Carrasco; Pato	2004	Portugal	✓		✓	✓	92	107	27	626	8,6 min							
14 Santos; Ochi; Souza	2004	Brasil	✓		✓		20	33		500	NI	✓						
15 Moura <i>et al.</i>	2004	Brasil	✓		✓		NI	NI		NI	4 min	✓	✓	✓	✓			
16 Avella <i>et al.</i>	2007	Itália	✓		✓		43	82		1548	20 min				✓			
17 Marte	2007	Alemanha	✓		✓	✓	**	91	NI	1157	NI							
18 Jacobsen <i>et al.</i>	2007	Alemanha	✓		✓	✓	**	91	NI	1157	100 s	✓	✓	✓	✓			
19 Santos; Souza	2007	Brasil	✓		✓		20	33		NI	NI	✓	✓	✓	✓			
20 Belingiannis <i>et al.</i>	2008	Grécia	✓	✓	✓		13	35		455	45 min				✓			
21 Birbas <i>et al.</i>	2008	Grécia	✓	✓	✓		21	48	NI	404	4 min	✓						
22 Belingiannis <i>et al.</i>	2009	Grécia	✓	✓	✓		13	35		455	45 min				✓			
23 Zhang <i>et al.</i>	2010	***	✓	✓	✓		13	35		455	3,5 min				✓			

NI: Não Informado

* 5% não solucionado

** No modelo alemão, alunos - e não turmas - são alocados

*** Os autores são de diversos países. O estudo não aponta o país em que o modelo foi baseado

Fonte: Poulsen e Bandeira (2012)

Na Figura 2.2, Arratia-Martinez, Padrón e Torres (2021), apresentam uma comparação similar à anterior, porém não separada em categorias, apenas categorizando entre verdadeiro e falso algumas características como alocação de salas, professores, nível institucional e método exato ou inexato.

Na Figura 2.3, Alencar et al. (2019) exploram uma categoria mais específica do problema, que é a característica da interatividade das interfaces desenvolvidas. Este apresenta características qualitativas quanto aos métodos, os dados dispostos, as técnicas de interação e o método utilizado para solucionar o problema de grade horária educacional. Nesta figura, os autores usam “Y” para simbolizar “Sim”, “N” para “Não” e “-” para “Inconclusivo”.

Figura 2.2 – Comparaçao entre artigos que solucionam o problema de grade horária

A comparison of similar papers that solve the university timetabling problem.

Characteristics	MirHassani [2]	Dimopoulos and Miliotis [9]	Philips et al. [11]	Oladje et al. [12]	Aladag and Hocaoglu [13]	Abuhamadah and Ayob [14]	Lu and Hao [15]	Abdullah et al. [19]	Chen and Shih [20]	Bolaji et al. [21]	Di Gaspero and Shaerf [25]	Qu and Burke [26]	Veenstra and Vis [27]	Lemos et al. [28]	Our work
Course timetabling	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Professor assignment	✓	✓	—	—	✓	—	—	—	—	—	—	—	✓	—	✓
Schedule	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Classroom/space assignment	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
Department level	—	✓	—	—	✓	—	—	—	—	—	✓	—	—	✓	✓
Institutional level	✓	✓	✓	✓	—	—	—	—	—	—	—	—	✓	—	—
Exact method	✓	✓	✓	✓	—	—	—	—	—	—	—	—	✓	✓	✓
Nonexact method	—	✓	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—

Fonte: [Arratia-Martinez, Padrón e Torres \(2021\)](#) - editado

Figura 2.3 – Análise de publicações aceitas

Analysis of the Accepted Publications. Some features are marked as [Y]es, [N]o or [–] for inconclusive.

Reference Identification	IV Method Applied	Displayed Data Elements by IV Method	Interaction types by IV Technique(s)						Application and Solution Techn. Appl.	
			OF	Constr.	Optimiz. Techn.	Manual Solution	Select Area	Appli.	Method(s) Used to Solve the Ed-TTP	
Piechowiak and Kolski (2004)	2D-table and time chart.	timetable, resources x time.	N	Y	N	Y	N	Y	Manual with constraint-based reasoning.	
Thomas et al. (2008)	Oriented cluster graph drawing.	classes and students enrolled.	—	Y	N	Y	N	N	Manual or by any automatic scheduler.	
Thomas et al. (2009b)	Directed graph drawing, histogram, daisy chart, tree view	pre-processing data (raw input data).	N	N	N	N	N	Y	There is no attempt to solve the problem, just processing/visualizing raw input data.	
Thomas et al. (2009a)	2D-table, oriented cluster graph drawing, histogram and tree representation	timetable (complete) and pre-processing data (raw input data).	N	Y	N	N	Y	Y	Constraint Satisfaction Program.	
Thomas et al. (2010b)	2D-table, graph drawing (2D, 3D).	timetable (complete), constraints and conflicts.	—	Y	N	Y	Y	Y	Constraint Satisfaction Program (in a constraint network, with backtracking) with user collaboration.	
Thomas et al. (2010c)	2D-table, graph drawing (2D, 3D).	timetable (complete), constraints and conflicts.	—	Y	N	Y	Y	Y	Constraint Satisfaction Program (in a constraint network) with user collaboration.	
Thomas et al. (2010a)	2D-table, graph drawing, tree representation	timetable (complete), constraints, conflicts.	N	Y	N	Y	Y	Y	Visual analysis heuristics and evolutionary algorithms.	
Abdelraouf et al. (2011)	Undirected graph drawing (representing peoples, courses, ...)	timetable with day/time, graphs and text	N	Y	N	Y	N	Y	Constraint satisfaction problem solving.	
Thomas et al. (2011)	Parallel coordinates (for uni/multi dimensional variables).	timetable (complete).	N	N	N	N	Y	Y	There is no resolution of the problem, just processing raw data.	
Thomas et al. (2012)	2D-table, graph drawing (2D, 3D), parallel coordinates.	timetable (complete), constraints and conflicts.	—	Y	—	Y	—	Y	Manual and user-driven problem solving environment, with clashes reconciliation (AI Techniques).	

Fonte: [Alencar et al. \(2019\)](#) - editado

2.3 Desafios recorrentes

Apesar da vasta quantidade de trabalhos realizados com este fim, o *Timetabling Problem* segue sendo uma área sem uma solução definitiva.

[Murray, Müller e Rudová \(2007\)](#) trazem a questão da modelagem como um dos maiores obstáculos. À medida em que a complexidade aumenta, se torna cada vez mais difícil desenvolver uma solução efetiva. Assim fazendo com que a solução para uma universidade possa não ter utilidade para outras, ou até mesmo não seja capaz de lidar com todos os problemas de uma mesma universidade.

Apesar do contrafluxo encontrado na resolução desse problema, [Murray, Müller e Rudová \(2007\)](#) citam que, apesar da complexidade, é sim possível desenvolver soluções que tenham uso prático, mesmo que não seja um processo fácil. As ferramentas existem e estão disponíveis. Restando então considerar e resolver as preocupações dos usuários às questões, visto que as técnicas de resolução já se encontram vastamente documentadas.

Com isso, entramos também no ramo da Interação Homem-Máquina, ramo abordado por [Andre e Dinata \(2018\)](#) que visaram em seu desenvolvimento a criação de uma interface focada no usuário. Assim minimizando o atrito na abordagem desse problema complexo. Também sendo área de enfoque de [Alencar et al. \(2019\)](#) em sua revisão literária

2.4 *Timetabling* na UENF

A instituição não se mostra desprovida de histórico na tentativa de resolução deste problema. Sânya e Ricardo, ambos estudantes de Ciência da Computação na UENF, já realizaram trabalhos com o mesmo fim, porém com abordagens diferentes da atual proposta. Em seus trabalhos, [Santos \(2013\)](#) e [Silveira \(2014\)](#) abordaram a criação de tabelas horárias através de métodos de otimização, utilizando heurísticas e meta-heurísticas para este fim. A seguir são descritos seus trabalhos anteriores e suas respectivas abordagens.

2.4.1 Sânya Carvalho dos Santos

Em seu trabalho, [Santos \(2013\)](#) aborda o problema de Programação de Horários de Disciplinas em Universidades, tendo como foco o curso de Ciência da Computação da UENF. Sua abordagem foi a de desenvolver um software que fosse capaz de gerar uma grade horária ótima para o curso, levando em conta as restrições impostas pelo curso. Para isso, Sânya explicou diversos métodos possíveis para se alcançar a solução desejada, passando inicialmente pelos métodos construtivos, seguido de métodos refinamento, podendo essas heurísticas serem utilizadas em conjunto com meta-heurísticas.

Por fim, utilizou uma heurística que consistia em respeitar a uma matriz de preferência para a distribuição das disciplinas. Seguindo com o uso do *Simulated Annealing*

para a otimização da solução inicial.

2.4.2 Ricardo Couto da Silveira

Em seu trabalho, [Silveira \(2014\)](#) aborda também o Problema de Programação de Horários (PPH) em instituições de ensino superior. Ele explora os diversos métodos heurísticos para resolução deste problema que visa encontrar a alocação ótima dos horários em suas grades. Ele teve como objetivo de seu trabalho a implementação de um software que fosse capaz de resolver o PPH do curso de Ciência da Computação da UENF, utilizando os métodos heurísticos Construção Gulosa e Busca Local e os métodos metaheurísticos *Simulated Annealing* e Busca Tabu. Para este fim, ele utilizou a linguagem de programação C para a implementação dos métodos e a linguagem de programação Java para a implementação da interface gráfica do software. Com isso conseguiu elaborar uma ferramenta automatizada para a geração de quadros de horários, alocando aulas e professores em dias e horários disponíveis na semana.

Ele descreve também como possibilidade de trabalho futuro o aperfeiçoamento do banco de dados da ferramenta desenvolvida para que se possa armazenar mais informações pertinentes ao problema, para que o usuário possa realizar modificações no quadro de horários, sendo guiado pelo retorno do Software que informa a viabilidade da alteração, assim gerando maior flexibilidade à aplicação.

2.4.3 LECIV

Através de pesquisas no [site oficial da UENF¹](#), mais especificamente na página de horários do LECIV para 2017.2 e 2018.2 foram encontrados alguns PDFs que continha a grade horária do curso de Engenharia Civil para o período de 2017 e 2018. A [Figura 2.4](#) ilustra a página referente às disciplinas do sexto período.

Quanto a análise desses documentos sobre esta grade algumas conclusões podem ser obtidas: 1. Esta forma de criação de grade foi mantida até no mínimo a data de 31/08/18 que consta no nome de um dos arquivos; 2. Foi utilizado o conceito de curva de nível com uma barra deslizante para visualizar a grade horária de cada período; 3. Com a legenda, entende-se que já se trabalhava com a ideia de maleabilidade de horários e também a ideia de alteração iterativa da grade; 4. Não se sabe ao certo se esta grade fora criada através de um software de criação de planilhas ou se foi criado através de um site.

Conclui-se então que esforços para se criar formas grades horárias intuitivas e maleáveis não estão limitados apenas ao curso de Ciência da Computação, também se estendendo a outros cursos da UENF.

¹ <https://uenf.br/portal>

Figura 2.4 – Grade horária do LECIV para 2017.2 e 2018.2

Horário - Engenharia Civil - 2017-02 e 2018-02 (vs.5.2)						
6º Período Civil		Selecione o período desejado através da barra de rolagem				
Horário		segunda	terça	quarta	quinta	sexta
7 - 8						
8 - 9	CIV01331	CIV01436	CIV01434	CIV01329	CIV01436	
9 - 10	CIV01331	CIV01436	CIV01434	CIV01329	CIV01436	
10 - 11	CIV01329	CIV01447	CIV01331	CIV01447	CIV01434	
11 - 12	CIV01329	CIV01447	CIV01331	CIV01447	CIV01434	
12 - 13	CIV01329		CIV01331		CIV01434	
13 - 14						
14 - 15						
15 - 16						
16 - 17		CIV01433		CIV01433		
17 - 18		CIV01433		CIV01433		
18 - 19						
19 - 20						
20 - 21						
21 - 22						

(*) Disciplinas sujeitas a mudanças de horários devido à disponibilidade de professores
(**) Disciplinas cujas turmas e horários estão sendo definidos

Disciplinas	Turma	Sala	Professor
1 - CIV01447 Saneamento e Engenharia Ambiental	A	Gustavo de Castro Xavier	
2 - CIV01434 Concreto Armado I	A	Sergio Luis González Garcia	
3 - CIV01436 Construção Civil	A	Jonas Alexandre	
4 - CIV01433 Análise Estrutural III	A	Vania Jose Karam	
5 - CIV01329 Mecânica dos Solos II	A	Fernando Saboya Albuquerque Junior	
6 - CIV01331 Estradas e Pavimentação	A	Paulo César de Almeida Maia	

Fonte: Haddad (2018)

2.5 Exemplo de erros humanos

Dada a grande quantidade de variáveis interconectadas e as características específicas de cada instituição (MIRANDA; REY; ROBLES, 2012), a organização destas informações buscando a melhor solução possível apresenta-se como um desafio. Principalmente se considerarmos que esta solução é, muitas vezes, buscada manualmente, estando também passível de erros humanos como ilustram a Figura 2.5 e Figura 2.6.

Nestas imagens, fica exemplificado um dos possíveis problemas que podem ocorrer durante a criação de grades horárias, que é, mesmo quando uma seção da universidade (o Sistema Acadêmico, ilustrado pela Figura 2.5) aloca uma turma a uma determinada sala, outra seção da mesma instituição (o Centro de Ciência e Tecnologia, ilustrado pela Figura 2.6) pode não estar ciente do mesmo, ou, mesmo estando ciente, pode acabar não delimitando aquela lacuna de tempo como ocupada, assim estando passível de uma segunda alocação naquele período de tempo naquela sala, assim gerando problemas.

Figura 2.5 – Disciplina atribuída no sistema acadêmico à determinada hora e local

 **Turma (Graduação)**

Dados Básicos

Código: EP

Disciplina: PRO01122 - Metodologia Científica para Engenharia de Produção - HT: 68, HP: 0, HE: 0

Calendário: 2023/1

Vagas: 30

Local: Sala 105/CCT

Máximo de faltas (horas): 17

Coordenador da turma: Manuel Antonio Molina Palma, D.Sc. (42528755)

Direcionamentos

Engenharia de Produção (Bacharelado / Presencial)

Ciência da Computação (Bacharelado / Presencial)

Horários

Dia	Horário inicial	Horário término
Terça	16:00	18:00
Quinta	16:00	18:00



Professores

Matrícula	Nome	Carga Horária
42528755	Manuel Antonio Molina Palma, D.Sc.	68

Fonte: autoria própria

Figura 2.6 – Falha de alocação na grade horária do CCT de 2023.1

SALA 105 Prédio do CCT -1º. Semestre 2023						
Capacidade: 35 alunos						
Horário	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA	SÁBADO
07:00 - 08:00						
08:00 - 09:00	Mecânica dos Solos I Prof. Tibana	Métodos Estatísticos para Engenharia Prof. Arica	Introdução à Computação Prof. Rivera	Métodos Estatísticos para Engenharia Prof. Arica	Fenômenos de Transporte Prof. Djalma	
09:00 - 10:00		Fotoquímica, fotofísica e fotobiologia Prof. Sergio		TEG-Mecânica dos solos... Prof. Rodrigo	Geotecnica de Barragens de terra... Prof. Saboya	
10:00 - 11:00						
11:00 - 12:00			Ética Profissional Profa. Maria Alves			
12:00 - 13:00						
13:00 - 14:00	Química Geral II Prof. Sergio		Química Geral II Prof. Nagipe			
14:00 - 15:00		Administração de Operações e serviços Prof. André		Administração de Operações e serviços Prof. André	Matemática Discreta Prof. Rafael	
15:00 - 16:00						
16:00 - 17:00	Física II Prof. Everton	Metodologia Cient. para Eng. De Produção Prof. Molina	Física II Prof. Everton			
17:00 - 18:00						
18:00 - 19:00	Cálculo I Profa. Ana Senna	Matemática Básica Prof. Thallis	Cálculo I Profa. Ana Senna	Cálculo I Profa. Ana Senna	Matemática Básica Prof. Thallis	
19:00 - 20:00						
20:00 - 21:00	Estratégias de Ensino III Prof. Roberto Faria	Mecânica Prof. Leonardo	Tópicos em Ciências, Ensino e Sociedade Prof.	Mecânica Prof. Leonardo		
21:00 - 22:00						

Legendas: PG=Pós-Graduação; LF=Licenciatura em Física; LM=Licenciatura em Matemática; LQ=Licenciatura em Química

Fonte: autoria própria

3 ESTRUTURA DA INSTITUIÇÃO

O problema de criação de grade horária tem forte influência da estrutura organizacional da instituição. Por tal motivo, este capítulo tem como objetivo apresentar a estrutura organizacional da Universidade Estadual do Norte Fluminense Darcy Ribeiro (UENF)¹ e como se dá o processo de criação de grades horárias na mesma. Essa apresentação se mostra necessária para que se possa entender melhor o problema e as possíveis soluções.

A apresentação se dará em três partes principais: a explanação sobre como a UENF está definida segundo seus documentos oficiais, a análise qualitativa das entrevistas realizadas com os principais envolvidos no processo de criação de grades horárias e a análise quantitativa das respostas do questionário aplicado aos alunos da UENF. Os dois primeiros tópicos convergem na criação de uma sequência de criação de grades horárias. Por fim, será apresentada a análise das relações entre as variáveis envolvidas no processo de criação de grades horárias.

3.1 A UENF

A UENF conta com diversos documentos que visam regulamentar e explicar seu funcionamento. Os documentos vão desde o mais alto nível onde se atribui as funções da UENF, até o menor nível onde se mostra a Grade Curricular para o curso de Ciência da Computação. Desses documentos, podemos listar o Estatuto da UENF (UENF, 2002), o Plano de Desenvolvimento Institucional (UENF; CONSUNI, 2023), o Regimento Geral da UENF (UENF; CONSUNI, 2006), o Regimento Geral da Graduação (UENF, 2012), o Regimento da Câmara de Graduação (UENF et al., 2019), as Normas da Graduação (UENF; PROGRAD, 2019), os Projetos Pedagógicos do Curso de Ciência da Computação (UENF, 2005; UENF, 2015b; UENF, 2022) e a Matriz Curricular do Curso de Ciência da Computação (UENF, 2015a).

Segundo o estatuto, a UENF comprehende:

- Órgãos da Administração Superior de política, gestão e supervisão;
- Unidades universitárias de ensino, pesquisa e extensão;
- Órgãos e serviços especiais, destinados a auxiliar na administração e a suplementar as atividades de ensino, pesquisa, extensão e apoio técnico.

¹ <https://uenf.br/portal>

Quanto aos órgãos da Administração Superior devemos enfocar o órgão executivo, constituído unicamente pela reitoria, cujos órgãos auxiliares englobam a Secretaria Acadêmica, que por sua vez tem como algumas de suas atribuições as seguintes:

1. Coordenar a **divulgação do horário escolar dos vários cursos da UENF**, de modo a **otimizar os recursos humanos, ampliar as opções de disciplinas para os alunos** e tornar acessíveis os dados escolares;
2. **Centralizar os serviços de registro da vida escolar dos alunos**, compreendendo **inscrição, admissão, matrícula, créditos, opções**, transferências, promoções, graduações e preparação dos respectivos diplomas, dentro das normas estabelecidas.

Já quanto as unidades universitárias de ensino, temos no estatuto que “as unidades universitárias de ensino, pesquisa e extensão, definidas por áreas de conhecimento, são constituídas em Centros, que por sua vez congregam Laboratórios afins” e que “o Laboratório é a menor parte da estrutura universitária para todos os efeitos de organização administrativa, didático-científica, distribuição de pessoal e de representação nos órgãos colegiados da UENF”.

A administração do Centro é da competência do Diretor e seu Conselho. Os Laboratórios, por sua vez, são administrados pelos Chefes de Laboratório.

O Conselho de Centro, tem como uma de suas atribuições, descrito no inciso XVII do artigo 34 do estatuto, a seguinte: **designar, semestralmente, os professores responsáveis pelas disciplinas dos Cursos de Graduação e Programas de Pós-Graduação, ouvidos os respectivos Laboratórios, os Colegiados de Curso e Comissões de Coordenação.**

Atualmente, segundo o site da UENF, a universidade possui 4 Centros, sendo eles:

1. Centro de Ciências do Homem - **CCH²**;
2. Centro de Ciência e Tecnologia - **CCT³**;
3. Centro de Biociências e Biotecnologia - **CBB⁴**;
4. Centro de Ciências e Tecnologias Agropecuárias - **CCTA⁵**.

E também existem 8 **laboratórios⁶** vinculados ao Centro de Ciência e Tecnologia (CCT), sendo eles:

² <https://uenf.br/cch>

³ <https://uenf.br/cct>

⁴ <https://uenf.br/cbb>

⁵ <https://uenf.br/ccta>

⁶ <https://uenf.br/cct/administracao/laboratorios>

1. Laboratório de Meteorologia – LAMET⁷;
2. Laboratório de Ciências Físicas – LCFIS⁸;
3. Laboratório de Engenharia Civil – LECIV⁹;
4. Laboratório de Ciências Químicas – LCQUI¹⁰;
5. Laboratório de Materiais Avançados – LAMAV¹¹;
6. Laboratório de Ciências Matemáticas – LCMAT¹²;
7. Laboratório de Engenharia de Produção – LEPROD¹³;
8. Laboratório de Engenharia e Exploração de Petróleo – LENEP¹⁴.

Os Laboratórios englobam os Cursos de Graduação e Pós-Graduação, que são administrados pelos Coordenadores de Curso.

Além disso, o LCMAT mantém dois cursos de graduação e um programa de pós-graduação stricto sensu. Sendo eles:

1. Licenciatura em Matemática¹⁵;
2. Bacharelado em Ciência da Computação¹⁶;
3. Mestrado Profissional em Matemática¹⁷ – PROFMAT¹⁸ / SBM¹⁹.

3.1.1 Responsáveis pela criação das grades horárias

As atribuições de cada setor estão, em sua maioria, descritas nas documentações encontradas. De forma resumida suas as atribuições são:

- **Secretaria Acadêmica:** elaborar e divulgar o Calendário Acadêmico; otimizar os recursos humanos; ampliar a oferta de disciplinas;

⁷ <https://uenf.br/cct/lamet>

⁸ <https://uenf.br/cct/lcfis>

⁹ <https://uenf.br/cct/leciv>

¹⁰ <https://uenf.br/cct/lcqui>

¹¹ <https://uenf.br/cct/lamav>

¹² <https://uenf.br/cct/lcmat>

¹³ <https://uenf.br/cct/leprod>

¹⁴ <https://uenf.br/cct/lenep>

¹⁵ <https://uenf.br/posgraduacao/licenciatura-matematica>

¹⁶ <https://cc.uenf.br>

¹⁷ <https://uenf.br/posgraduacao/matematica/apresentacao>

¹⁸ <https://uenf.br/posgraduacao/programas/pos-graduacao>

¹⁹ <https://profmat-sbm.org.br>

- **Câmara de Graduação:** aprovar e modificar o calendário acadêmico; sugerir vagas de bolsistas;
- **Direção de Centro:** designação semestral de professores responsáveis pelas disciplinas, após ouvir os Laboratórios, Colegiados e Coordenações;
- **Chefia do Laboratório:** atribuir carga horária didática aos docentes do laboratório e aos bolsistas; designar docente responsável por disciplina ofertada;
- **Coordenação do Curso:** coordenar a distribuição de estudantes do curso; indicar à chefia de laboratório as disciplinas a serem ofertadas ao curso coordenado;
- **Sistema Acadêmico:** articula parcialmente as informações entre os setores, como a oferta de disciplinas e a distribuição de estudantes.

3.2 Entrevistas

Como forma de entender melhor a percepção real daqueles que recorrentemente lidam com a tarefa de criação da grade horária, diversas entrevistas foram feitas com o intuito de analisar qualitativamente quais são as opiniões, pedidos, reclamações e pensamentos de diferentes níveis organizacionais da UENF. Essas análises têm como objetivo principal averiguar se o processo de criação de grades horárias realizada na UENF se apresenta como uma tarefa problemática e se a solução proposta neste trabalho se mostra como uma solução viável.

Pois, como é informado por Pierre, Fairley e Society (2014), uma das fontes de requisitos é o ambiente organizacional e como o *software* muitas vezes visa auxiliar em algum processo da instituição, processo este já condicionado à sua estrutura, cultura e políticas externas, o engenheiro de *software* precisa estar atento a elas, visto que o novo *software* não deve forçar mudanças não planejadas em processos de negócios.

3.2.1 Direção do CCT

O primeiro entrevistado foi o atual Diretor do CCT. Ele atualmente estrutura a relação de disciplinas ofertadas pelo CCT em Excel e as publica [em formato PDF no site do CCT²⁰](#). Seu trabalho auxilia os Chefes de Laboratório e Coordenadores de Curso a visualizarem quais são as salas disponíveis e em quais horários cada professor está alocado.

Um dos tópicos dialogados, foi quanto às categorias das disciplinas, ou seja, quais características notáveis as disciplinas poderiam ter. Com isso podemos listar as seguintes categorias de disciplinas:

²⁰ <https://uenf.br/cct/secretaria-academica/distribuicao-das-salas-de-aula-do-cct>

- **Anuais:** disciplinas que ocorrem apenas uma vez no ano;
- **Ímpares:** disciplinas que são ofertadas no primeiro semestre letivo;
- **Pares:** disciplinas que são ofertadas no segundo semestre letivo;
- **De serviço:** disciplinas ofertadas para mais de um curso simultaneamente;
- **Ciclo básico:** disciplinas oferecidas para todas as engenharias;
- **Repetentes:** turmas criadas especialmente para repetentes.

As disciplinas ímpares e pares geralmente estão atreladas à expectativa de que os alunos progredirão sequencialmente sem reprovação alguma. Entretanto, caso uma quantidade de alunos considerável de alunos reprove em determinada disciplina, é possível que estes se enquadrem na criação de uma turma especial para repetentes, ou não.

Uma sugestão de utilidade para o software é a de permitir que as “disciplinas de serviço” sejam fixas, visto que estas são as que têm maior complexidade de manejamento de horário posteriormente, justamente por geralmente abrangerem muitos alunos e de diversos cursos diferentes.

Uma outra característica notável é a repetição de atribuições de disciplinas em pares regulares, ou seja, alocadas no mesmo período de horário com um dia de intervalo entre elas. Um exemplo desse tipo de alocação recorrente seria “14 às 16 horas de segunda e quarta feira”.

Com isso, surge a dúvida: há uma preferência ativa por aulas alocadas com este padrão? A resposta dada é que não. O que se mostra como uma restrição a menos na hora de se alocar as turmas.

Outro caso notável é a existência majoritárias de turmas criadas com dois períodos de duas horas, entretanto existem algumas que fogem deste padrão e possuem três horas de duração. A solução encontrada pelo Diretor é a de colocar esta disciplina começando às 10h, o que faz com que se alongue até as 13h, período geralmente usado pelos estudantes e servidores para se alimentar, e justamente por isso evitando que atrapalhe a distribuição das salas. Outra alternativa é alocar esta turma para as 13h, fazendo com que finalize às 16h, horário em que as disciplinas com duas horas de duração geralmente terminam.

Segundo ele, saber a demanda máxima possível seria bom, visto que podem haver casos de solicitações de vagas para disciplinas de serviço que extrapolam a quantidade esperada para a distribuição balanceada dentre os cursos.

Uma outra situação que ocorre é que algumas disciplinas historicamente têm seus horários definidos em um mesmo horário ao longo dos anos. Caso essa alocação seja alterada, ocorre a possibilidade de reclamação por parte dos professores, mesmo que esta

alteração seja benéfica para os estudantes. Então por exemplo, os horários de 8h de uma segunda feira e de 16h de sexta feira, não são geralmente desejados pelos professores, mesmo que eles teoricamente tenham disponibilidade de 8 horas diárias.

Considerando a quantidade de laboratórios “concorrendo” simultaneamente às vagas, surge a dúvida: há ordem de precedência entre os laboratórios? A resposta para esta pergunta é “Não. As vagas são distribuídas com prioridade na ordem de chegada”.

Algumas outras informações que ele elenca:

- **As turmas de disciplinas básicas são grandes:** é esperado que uma grande quantidade de alunos se inscreva nas turmas de disciplinas essenciais e iniciais de seus cursos, sendo boa parte dela relacionada com o conceito das disciplinas de serviço e com o conceito de ciclo básico das engenharias;
- **As turmas de disciplinas de serviço devem ser alocadas primeiro:** visto a grande quantidade de conflitos possíveis dentre os diversos cursos, ao alocá-las primeiro, os conflitos passam a ocorrer em turmas com uma quantidade menor de pessoas e/ou que sejam de um mesmo curso;
- **As alterações vão até o final do período:** embora possa parecer que a alocação de turmas finalize após o encerramento do período de inscrição e desinscrição, na prática, a realocação ocorre durante todo o período;
- **Teoricamente matérias de um mesmo período não devem conflitar:** isso se dá segundo a percepção de que a maioria dos alunos está seguindo a mesma linha sequencial de disciplinas, o que muitas das vezes não é a realidade.

3.2.2 Desenvolvedor do Sistema Acadêmico

Considerando que a integração do sistema proposto seria certamente mais eficiente se integrada ao sistema acadêmico, viu-se como apropriado entrevistar o desenvolvedor do Sistema Acadêmico para se ponderar sobre o uso dos dados e a possível integração.

Durante a entrevista, foram listados alguns dados que seriam interessantes para a análise, sendo eles a demanda de disciplinas, a listagem dos professores, a listagem dos alunos aprovados e suas respectivas disciplinas e por fim os requisitos das disciplinas.

Outra questão analisada seria quanto a forma de integração. Boa parte das aplicações web se comunicam em forma de API, entretanto, devido à quantidade de alterações executadas ao longo do semestre no sistema acadêmico, o Desenvolvedor do Sistema Acadêmico utiliza o sistema de mensagerias através do RabbitMQ²¹.

²¹ <https://rabbitmq.com>

Foi citado sobre a abordagem do Coordenador de Computação para o cálculo das demandas, quanto a isso, o Desenvolvedor citou que poderia facilmente permitir o download de um CSV dos dados necessários.

Quanto à possibilidade de aprimoramentos no Sistema Acadêmico, ele disse que “eu faço o que me pedem”, se referindo ao repositório do Acadêmico disponível no [GitLab²²](#), onde alguns poucos usuários fazem solicitações de alterações e melhorias. Havendo então a possibilidade de que o Coordenador de Computação faça uma solicitação à SECACAD para que seja implementada uma funcionalidade que permita a exportação dos dados necessários para o cálculo das demandas.

Um outro problema apontado por ele é a falta de gente. Segundo ele, outras duas pessoas entraram junto com ele no mesmo concurso, mas foram realocadas para outras áreas da universidade. Ele cita também sobre a “cultura do trabalho opcional” existente na UENF, onde muitos servidores não se sentem obrigados a trabalhar.

Em relação a estrutura dos dados, o sistema acadêmico utiliza o SQL. Foi citado o uso de NOSQL e estrutura de Grafos como possibilidades de mudança, mas como a mesma não se mostrou necessária até o momento, não foi implementada.

Uma questão levantada pelo entrevistado diz respeito à manutenção do software desenvolvido neste trabalho. Não sabendo ele dizer se o mesmo seria mantido pela UENF.

Ele também sugere que, para evitar a complexidade de se trabalhar com dados reais de alunos, que sejam utilizados dados fictícios.

3.2.3 Chefia de Laboratório de Matemática

Considerando que um dos cargos relacionados com o processo de elaboração de grades horários é o de Chefe de Laboratório, foi entrevistada a atual Chefe de Laboratório de Matemática.

Assim como sugerido pelo Desenvolvedor do Sistema Acadêmico, a Chefe também sugeriu que dados fictícios fossem utilizados. Sugeriu ainda que fosse utilizado o schema do banco de dados do sistema acadêmico como sua criação. Outra sugestão foi a solicitação ao Desenvolvedor do Sistema Acadêmico uma listagem de possíveis valores recorrentes no banco de dados.

A entrevistada também relatou algumas problemáticas envolvendo a realocação dos horários das turmas. Segundo ela, qualquer alteração pode ser feita durante a semana anterior à matrícula, visto que, não havendo inscritos, não há problema na alteração. A partir do momento em que houver ao menos um aluno inscrito na disciplina, alterações só podem ser feitas caso não haja conflitos aparentes e preferencialmente com um documento

²² <https://about.gitlab.com>

assinado pelos alunos que estiverem inscritos.

3.2.4 Responsável pela Secretaria Acadêmica (SECACAD)

Inicialmente, alguns tópicos foram trazidos como ponto focal da entrevista, sendo alguns deles os seguintes:

- Dúvidas quanto as atribuições da SECACAD;
- Permissão de acesso aos dados que não são estritamente necessários, mas ajudariam;
- Definição dos períodos, demanda provisória e erros de estimativa;
- GitLab, tarefas (issues) e demandas;
- Automatização da burocracia;
- Ética VS Eficiência.

Logo de início, o entrevistado informou que ele não pode ceder dados de nenhum aluno, mesmo que anonimizados, mas sugeriu que poderia reencaminhar um formulário de pesquisa para os alunos, para que assim eles próprios pudessem fornecer os dados necessários.

Outra abordagem interessante informada por ele é quanto ao seu conhecimento técnico, onde sugeriu abordagens de análise multicritérios como forma de se auxiliar a criação das grades horárias.

Durante a conversa, foi citado de forma positiva quanto à demanda exata de cada disciplina. Reforçou-se a preferência pela alocação de disciplinas visando os estudantes mais próximos da conclusão do curso, estando em último na ordem de prioridade aqueles que decidem se adiantar com disciplinas de períodos mais avançados. Uma outra característica apontada é que a sequência de definições é a seguinte: Vagas → Professor → Sala → Horário.

Também se confirmou a não existência de um registro oficial das salas e suas capacidades. Essa informação é inserida como um campo de texto no sistema acadêmico, com isso, o sistema não impediria a alocação de duas turmas em uma mesma sala em um mesmo horário.

O responsável pela SECACAD também informou que cabe à Pró-Reitoria a mudança do início do primeiro semestre para expandir o período de preparação das grades horárias para o segundo período, sendo que este pedido deve partir da Câmara de Graduação.

Quanto ao tópico “ética VS eficiência”, ele citou que embora o sistema acadêmico impeça a realocação de turmas com alunos inscritos, é possível que o mesmo seja burlado

ao manualmente se excluir a inscrição do aluno. Sendo esta prática justificável em alguns casos.

Uma ferramenta que o beneficiaria seria a análise dos alunos que estão à beira de perder o vínculo com a universidade, para que a Secretaria Acadêmica possa tomar as medidas cabíveis.

3.2.5 Coordenação de Computação

Sendo o Coordenador de Computação o principal usuário do sistema, torna-se imprescindível a análise qualitativa de sua perspectiva.

Seguindo o conceito de Design Iterativo utilizado também por [Andre e Dinata \(2018\)](#), o Coordenador foi consultado em diversas etapas do desenvolvimento do sistema. Inicialmente, foi apresentado a ele o conceito do sistema, suas funcionalidades e possíveis benefícios. Em seguida, foi apresentado a ele um protótipo do sistema. Mas esta questão será melhor tratada em outro segmento deste mesmo trabalho, aqui será abordado apenas o conteúdo das entrevistas.

Assim como comentado pelo Diretor do CCT, o Coordenador também fala sobre a definição de matérias que se mostram fixas, porém, agora com outro olhar: enquanto o diretor vê as matérias fixas como uma forma de atribuição histórica seguindo a ideia de “já era assim quando eu cheguei”, o Coordenador por sua vez vê apenas como uma forma predefinida e imutável. Porém, olhando em um contexto mais amplo, essa definição de matérias não se mostra como obrigatória, visto que pode haver casos em que outra alocação de uma disciplina “fixa” apresente uma qualidade melhor do que seu horário usual. Segundo ele, “Houve uma resolução anos atrás para que os horários ficassem fixos”, ele porém, não soube definir qual resolução seria essa e nem quando ocorreu.

Outra questão levantada por ele é quanto a um problema já antigo no curso de Ciência da Computação na UENF, que há anos apresenta um corpo docente reduzido em comparação com outros cursos, sendo necessário um desdobramento maior para suprir a demanda de disciplinas dos alunos. Uma solução utilizada é a de solicitar a abertura de uma bolsa para docência complementar, onde um aluno de pós-graduação pode ser alocado como professor de uma disciplina. Solução que embora não seja a ideal, é a que se mostra mais viável, dada a diminuta inscrição de candidatos à docência.

Uma outra característica até então não citada pelos outros entrevistados é que existem salas que são vistas culturalmente como sendo de determinado curso, onde acaba sendo um certo tabu a alocação de uma disciplina de outro curso, mesmo que não se esteja infringindo regra alguma.

Quanto à priorização de veteranos já citada anteriormente, o Coordenador aponta uma outra forma de se enxergar a situação: em disciplinas dos períodos finais do curso, a

prioridade é dos veteranos, ficando os calouros que ocasionalmente possam ter se adiantado, em segundo plano. Já em disciplinas dos períodos iniciais, a prioridade é dos calouros, ficando os veteranos que por ventura tenham reprovado, em segundo plano.

Diferente de como foi respondido pelo Diretor do CCT, para o Coordenador de Computação a alocação de disciplinas em pares se mostra como “didática”, sendo ela então preferível, mas não necessariamente vista como obrigatória.

Considerando a recorrência de citação do conceito de estimativas de demanda, o Coordenador de Computação sugere que haja um campo no sistema para que seja inserida a demanda estimada de cada disciplina.

Considerando que no contexto atual do curso de Ciência da Computação na UENF é iminente a adoção de uma nova grade curricular, o Coordenador apresentou preocupação em relação à possibilidade de que o sistema não seja mais utilizado após a adoção da nova grade. Essa questão encontra-se atualmente fora do escopo do atual projeto, entretanto, não se mostra como um problema de difícil solução, visto que o sistema pode ser adaptado para a nova grade.

3.2.6 Entendimento geral das entrevistas

Podemos concluir após a análise qualitativa das entrevistas que há de fato um certo grau de insatisfação por parte dos usuários do sistema atual. Embora o sistema funcione, ele apresenta gargalos que poderiam ser resolvidos com a utilização de um sistema mais eficiente que envolvesse mais diferente as diferentes partes interessadas. Suas maiores insatisfações são quanto à burocracia e o curto período de tempo disposto para a elaboração das grades horárias.

Embora não sejam apontadas como insatisfação, algumas potenciais ferramentas e melhorias foram também citadas pelos entrevistados. Dentre elas, a demanda máxima possível, que passaria a evitar superestimações de demanda, a alocação de disciplinas de serviço como fixas, e em alguns casos, a alocação de disciplinas em pares, que embora não seja uma regra, é uma preferência de um dos entrevistados. Outra ferramenta que foi citada é a de análise de alunos à beira de perder o vínculo com a universidade, que poderia ser utilizada pela Secretaria Acadêmica para tomar as medidas cabíveis. Também se fazendo notória a necessidade de registro oficial das salas e suas capacidades, que atualmente é inserida como um campo de texto no sistema acadêmico.

Outros problemas encontrados, remetem à acomodação institucional de algumas práticas, como a alocação de disciplinas em horários fixos, em pares e/ou nas mesmas salas. Essas práticas, embora não sejam obrigatórias, são vistas como um costume e por isso são mantidas.

3.3 Sequência de criação das grades horárias

Ao somarmos o conhecimento presente no estatuto da UENF, com o conhecimento adquirido através das entrevistas, podemos ter uma visão geral de como se dá a criação das grades horárias na UENF. Assim, abaixo estão listados os passos que geralmente são seguidos para a criação das grades horárias.

1. Período ocorrendo normalmente;
2. Coordenadores enviam para Chefes de Laboratório uma demanda estimada de cada uma das disciplinas que serão ofertadas;
3. Chefes de Laboratório atrelam professores a disciplinas;
4. Chefes de Laboratório enviam para Diretores de Centro a demanda estimada;
5. Dependendo das disponibilidades dos professores, cabe solicitar a abertura de uma bolsa de apoio ao ensino;
6. O Diretor aloca provisoriamente as disciplinas em horários e salas;
7. O Coordenador de Curso analisa possíveis mudanças de horários que possam ser mais eficientes na distribuição dos alunos;
8. O período letivo acaba;
9. Estima-se mais precisamente a demanda de cada disciplina;
10. Turmas são abertas com a quantidade de vagas de acordo com as demandas estimadas;
11. Alunos se inscrevem;
12. Últimas mudanças são feitas;
13. Período de inclusão e exclusão;
14. Alguns possíveis ajustes finais;
15. Período ocorrendo normalmente.

Entrando em detalhes ainda maiores, podemos citar uma das etapas de criação das grades horárias que é a coleta de uma demanda esperada. Nela, cada Coordenador elabora de seu próprio modo. Uma possibilidade seria analisar quantos alunos costumam reprovar em determinada disciplina pela visualização estatística anterior, somado aos que possivelmente aprovarão na disciplina que é pré-requisito. Porém, toda essa pesquisa e estimativa é dispendiosa e pode desagradar a alguns coordenadores, ou então gerar estimativas incondizentes com a realidade.

Entendemos então que dentro do contexto da universidade, o problema de agendamento se torna mais complexo pois um dos recursos que está relacionado com o problema é a existência de prazos em cada uma das etapas, assim fazendo com que uma solução ideal seja aquela que é capaz de ser executada dentro do prazo estipulado, mesmo que não seja ótima.

3.4 Formulário quantitativo aos discentes

Já tendo sido expostas a forma como a UENF está [estruturada como instituição](#), as [percepções dos envolvidos no processo de criação de grades horárias](#) e a [sequência de criação das grades horárias](#), resta avaliar a percepção dos alunos quanto a este processo.

Como forma de analisar também a perspectiva dos discentes quanto à problemática abordada, foi elaborado um formulário de pesquisa com o intuito de se confirmar ou não a hipótese de que em sua maioria os alunos também se encontram insatisfeitos com a atual conjuntura de distribuição e alocação de turmas. Vale ressaltar que a **motivação** dessa pesquisa é a de servir como base comparativa para trabalhos posteriores que possam vir a ser realizados, servindo como um marco temporal para futuras análises. Uma segunda **motivação** para a realização de tal pesquisa é a de se obter a percepção também daqueles que se encontram no final da cadeia de decisões, os alunos, que são os principais afetados pela distribuição das turmas.

Para este fim, foi utilizado um formulário de pesquisa qualitativa dos alunos. O formulário foi divulgado através de um link disponibilizado no grupo de alunos do curso de Ciência da Computação no WhatsApp, e também através de um link distribuído via e-mail pela Secretaria Acadêmica a discentes da UENF.

O compilado das respostas dos alunos encontra-se no [Apêndice](#) junto com a sua análise.

3.5 Relações entre as variáveis

A criação de grades horárias é um problema complexo, que envolve uma grande quantidade de variáveis interconectadas. Dentre elas temos as disciplinas, professores, salas, laboratórios, blocos, centros, cursos, alunos, entre outros.

Esta seção visa a análise das relações entre estas variáveis, buscando entender como elas se relacionam e como estas relações podem ser utilizadas para a criação de uma grade horária mais eficiente.

- **Blocos/prédios:** são estruturas físicas que abrigam **salas** de aula. Cada bloco possui uma quantidade de salas, que são utilizadas para a realização das aulas;

- **Salas:** são estruturas físicas que abrigam as **turmas**. Cada sala possui uma capacidade máxima de **alunos**. Esse limite geralmente está atrelado à quantidade de carteiras disponíveis aos alunos, não sendo então necessariamente fixo, havendo geralmente a possibilidade de solicitar algumas carteiras de salas próximas. Elas estão teoricamente associadas aos **blocos**, visto que a compõem fisicamente, e são geridas pelos **centros**, que são responsáveis por distribuí-las. Na prática, algumas salas são preferencialmente utilizadas por determinados **cursos**, ficando então subgeridas pelos **laboratórios**;
- **Centros:** são estruturas organizacionais que gerem os **laboratórios** e diversas **salas**. Embora cada **centro** geralmente esteja associado a um **bloco** específico, os centros não estão limitados a gerir as salas apenas deste bloco;
- **Laboratórios:** são estruturas organizacionais que englobam **professores**, **cursos**, **disciplinas** e **turmas**. Em sua maioria, os **laboratórios** estão associados a um único curso, entretanto, existem casos em que um laboratório está associado a mais de um curso, como é o caso do LCMAT, que está associado aos cursos de Matemática e de Ciência da Computação. Existe também o caso contrário, onde um curso está associado a mais de um laboratório, como é o do Bacharelado em Biologia que está associado a seis laboratórios;
- **Professores:** são aqueles que ministram as **disciplinas**. Teoricamente professor está vinculado a um **laboratório**, entretanto, na prática, os professores geralmente são associados aos **cursos** que geralmente ministram **disciplinas**. Teoricamente cada professor está passível de ministrar qualquer disciplina de seu laboratório, entretanto, na prática, os professores geralmente têm um conjunto de disciplinas que mais usualmente ministram;
- **Disciplinas:** um dos pontos principais na criação de uma **grade horária**. As **disciplinas** são ofertadas pelos **laboratórios** em **turmas**, que são alocadas em **salas** pelos **Centros**. Cada **PPC** define quais são as disciplinas e pré-requisitos que os **alunos** precisarão cursar. Uma mesma disciplina pode constar no **PPC** de diversos **cursos**. Cada disciplina possui uma carga horária, uma **ementa**, um nome e um código. Este código é composto por três letras, geralmente indicando o laboratório que o oferta, mas, em casos específicos, está associado a centros;
- **Turmas:** são a base da grade horária, ela consiste na representação da oferta de uma **disciplina** em um determinado **horário** e **sala**, com um determinado **professor** e para um determinado grupo de **alunos**. Mesmo que em sua maioria as disciplinas sejam ofertadas em apenas uma **turma**, existem casos em que uma disciplina é ofertada em mais de uma turma, nesses casos, um professor é atribuído como o coordenador de todas as turmas dessa disciplina. Situação essa que ocorre também

com a atribuição de professores, havendo casos em que vários professores estão responsáveis por uma só turma; As turmas geralmente são direcionadas a **cursos** específicos, entretanto, existem casos em que uma turma é ofertada para mais de um curso simultaneamente;

- **Grade Horária:** é a representação de todas as **turmas** ofertadas em um determinado período letivo. A grade horária é composta por todas as **turmas** ofertadas, com seus respectivos **horários** e **salas**. A grade horária é a representação final do processo de criação de turmas. Esta representação geralmente é feita em formato de tabela e distribuída pelos **cursos** e **centros**;
- **Horário:** ao se criar a **grade horária** é necessário alojar as **turmas** em um determinado momento de determinado dia da semana. Cada horário consiste de uma hora de início e uma duração (ou hora de término). Geralmente as turmas que são oferecidas aos **cursos** seguem o turno respectivo do curso, mesmo que às vezes fuja deste padrão. As turmas em sua maioria são distribuídas em horários de 2 horas com um espaço de um dia entre elas, entretanto, existem casos em que as turmas possuem 3 horas de duração. Os horários disponíveis para alocação são das 7 às 22 horas;
- **Cursos:** são estruturas organizacionais nos quais os **alunos** estão matriculados. Cada curso possui um turno no qual as suas turmas são ofertadas, esses turnos podem ser matutino, vespertino ou noturno;
- **Projeto Pedagógico do Curso (PPC):** é o documento que define a estrutura curricular do **curso**, contendo as **disciplinas** obrigatórias, optativas e eletivas, bem como a quantidade de créditos necessários para a conclusão do curso;
- **Alunos:** são aqueles que frequentam as **turmas**. Cada **aluno** geralmente está vinculado ao PPC vigente para seu **curso** no ano de sua matrícula, havendo casos em que se é permitida a mudança de PPC. Segundo o PPC estão definidas quais são as **disciplinas** que o aluno deve cursar para integralizar o curso.

4 MODELAGEM GERAL DO SISTEMA

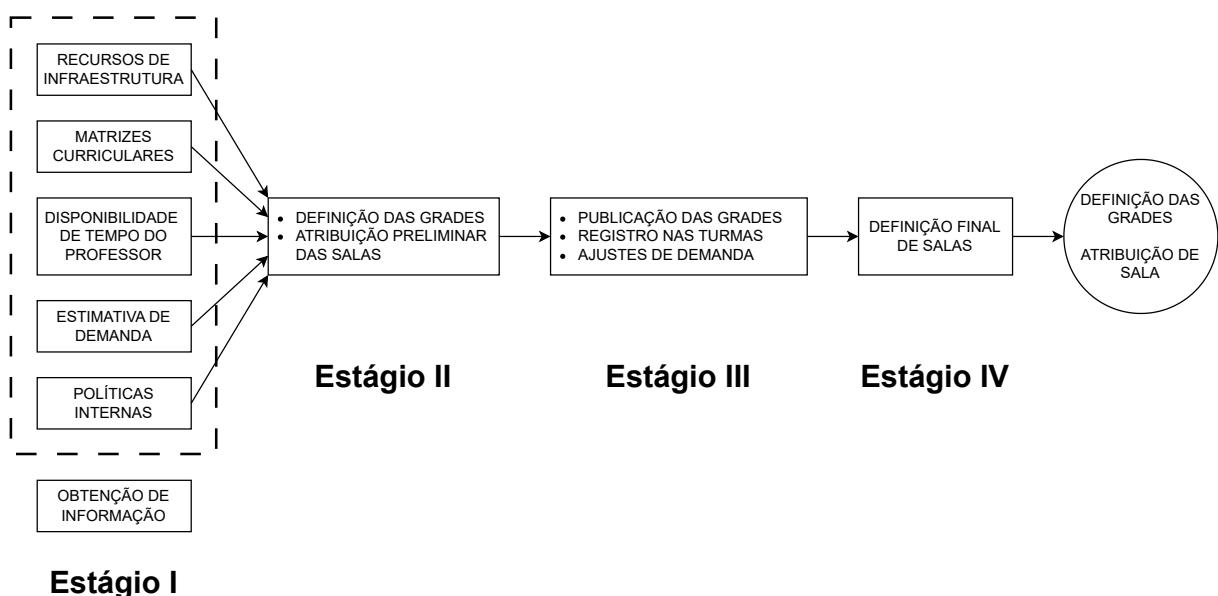
Tendo esclarecido sobre as questões gerais da área de estudo e de como se estrutura a instituição agora nos aprofundaremos na conceitualização do funcionamento geral do sistema e a forma como se dará a execução da metodologia proposta apresentando o funcionamento geral de como o sistema será estruturado e de como foi desenvolvido.

Ao longo deste capítulos serão abordados os estágios de execução sobre como se dão as etapas de criações de grade horária no geral, a iteração com os stakeholders, o funcionamento do sistema, o modelo de banco de dados e a API REST que será utilizada para a comunicação entre o *frontend* e o *backend* do sistema.

4.1 Estágios de execução

Em seu trabalho de aplicação prática, Miranda, Rey e Robles (2012) estruturaram estágios que compõem o processo necessário para que enfim se alcance a definição de tabelas horárias finais.

Figura 4.1 – Estágios para a obtenção de grade horária ótima



Fonte: Miranda, Rey e Robles (2012) - adaptado

Na Figura 4.1, estão dispostos 4 estágios principais. O primeiro dispõe da aquisição de informações, sendo elas a disponibilidade do professor, os recursos da infraestrutura, as grades dos cursos, as estimativas de demanda e as políticas internas. No segundo estágio são definidas grades horárias preliminares com atribuição preliminar das salas. No terceiro,

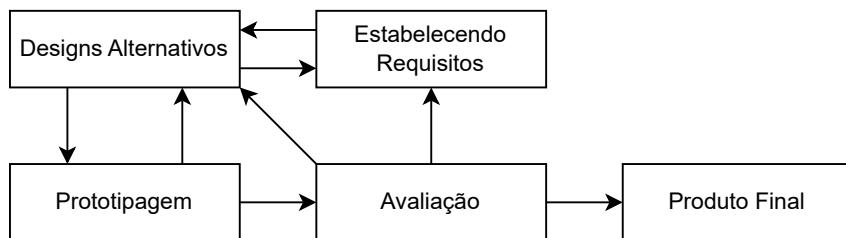
os alunos se inscrevem e a demanda é ajustada, por fim, no quarto estágio, ocorre a alocação final das salas. Com sua conclusão, são definidos as grades horárias finais junto com as respectivas salas.

A sequência geral condiz com o processo de criação das grades horárias, porém é necessário adaptar a metodologia para o contexto da UENF, e também ao escopo do trabalho. Sendo assim, no primeiro estágio a coleta de informações sobre a disponibilidade dos professores e as matrizes curriculares dos cursos não será necessária. Os recursos de infraestrutura serão coletados dos histórico de alocação das turmas, a estimativa de demandas será obtida através da média histórica de alunos inscritos nas turmas e as políticas internas serão compreendidas através da revisão documental e entrevistas com os *stakeholders*. O segundo tem caráter iterativo entre as coordenações e a diretoria do CCT. Os outros estágios não se distanciarão de forma significativa do que foi proposto.

4.2 Iteração

Para se alcançar uma alta satisfação por parte dos *stakeholders*, vê-se necessária a constante interação com os mesmos. Para isto, será seguida a estrutura utilizada por Andre e Dinata (2018).

Figura 4.2 – Etapas do Design de Interação



Fonte: Andre e Dinata (2018) - adaptado

Seguindo o conceito do Design de Interação, a Figura 4.2 ilustra o ciclo de ações a serem tomadas durante o desenvolvimento do sistema, caso este venha a ser necessário. Nesse modelo de pesquisa, os *stakeholders* serão consultados continuamente enquanto lhes for apresentados protótipos do sistema, para que assim informem quanto às suas percepções. Esta dinâmica tem como finalidade encontrar um design tal que seja adequado aos desejos e necessidades de seus usuários. Pois, considerando que para que o sistema seja efetivo, é necessário que ele seja aceito e utilizado pelos usuários finais.

4.3 Funcionamento

O sistema proposto funcionará de forma a auxiliar a coordenação do curso de Ciência da Computação da UENF criação de grades horárias para os semestres letivos.

Para isso, o sistema deverá ser capaz de gerenciar as informações referentes às disciplinas, professores, salas e horários disponíveis.

Para tanto, mesmo que disponha de informações pré-cadastradas, o sistema deverá permitir a inserção de novas informações, como disciplinas, professores e salas, pois, apesar de não ser o foco principal do sistema, é recorrente que haja alterações nessas informações ao longo do tempo, e caso o sistema não as comporte, não poderá cumprir seu propósito.

Como forma de agregar as informações cadastradas, o usuário será capaz de criar turmas para as disciplinas, informando o professor responsável, a sala e o horário em que a turma ocorrerá. Além disso, o sistema deverá permitir a visualização das turmas já cadastradas, bem como a edição e exclusão das mesmas, caso necessário. Ao criar uma turma, o sistema viabiliza a definição de qual ano e semestre a turma pertence.

As turmas contarão também com a informação de demanda estimada, que será utilizada como comparativo para a alocação das salas, de forma a evitar que uma sala seja alocada para uma turma cuja demanda seja maior do que a capacidade da sala. Outra informação contida nas turmas será um código descritor da turma em questão. Esse código será utilizado para auxiliar na compreensão das turmas, visto que ao longo do processo de criação das grades horárias é recorrente a criação de turmas criadas com um propósito específico, como por exemplo contemplar apenas repetentes ou alunos de um determinado conjunto de cursos.

Como ponto chave do funcionamento do sistema, o mesmo deverá ser capaz de ilustrar conflitos que vierem a surgir durante a alocação dos recursos. Esses conflitos podem ser de diversas naturezas, como por exemplo, a alocação de uma turma em um horário em que o professor responsável não está disponível, ou a alocação de duas turmas em uma mesma sala e horário. A identificação desses conflitos é essencial para que a coordenação possa corrigí-los de forma prática e direta, assim evitando que ocorram problemas durante a execução das turmas no semestre letivo.

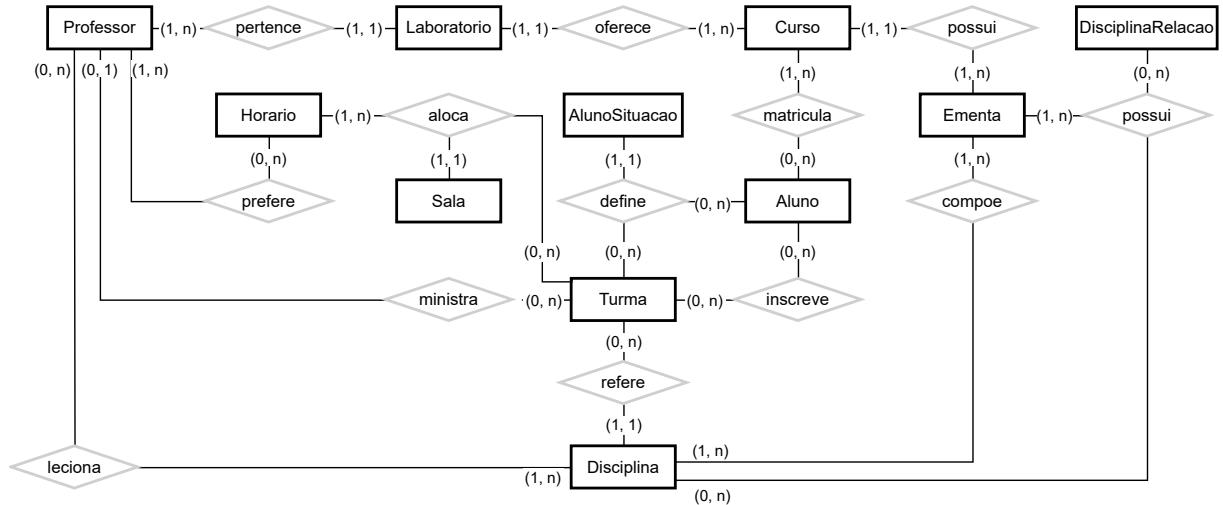
Por fim, o sistema viabiliza também um método de rápida criação de turmas para todas as disciplinas esperadas para determinado semestre letivo para o curso de Ciência da Computação. Esse método consiste em analisar as disciplinas esperadas para os estudantes de computação em determinado semestre letivo, e, através do cálculo das maiores recorrências de professor, sala e horário atribuídos às turmas dessa mesma disciplina em semestres anteriores, criar uma turma com essas informações. Esse método visa agilizar o processo de criação das turmas, visto que a coordenação não precisará criar cada turma manualmente, mas sim apenas revisar as informações geradas pelo sistema.

Dispondo de todas essas funcionalidades, o sistema deverá ser capaz de gerar uma grade horária final, que será utilizada como base para a criação da grade horária oficial do curso de Ciência da Computação da UENF.

4.4 Modelo de banco de dados

Considerando as informações necessárias para o presente trabalho, e também o preparo de campo para potenciais aplicações futuras, foi elaborado um diagrama conceitual parcial do banco de dados ([Figura 4.3](#)).

[Figura 4.3 – Diagrama Conceitual do banco de dados](#)



Fonte: autoria própria

O diagrama conceitual foi desenvolvido utilizando a ferramenta [draw.io](#)¹ citada na metodologia e ilustra as relações entre diversas entidades presentes na realidade da UENF. O emaranhamento presente no diagrama ilustra a complexidade envolvida na criação de uma grade horária, onde diversas entidades se relacionam entre si.

Como principais apontamentos, podemos citar a parte principal do modelo que é a alocação de turmas. Ela, como já descrito, envolve a correlação entre alunos de diferentes cursos, professores, disciplinas, salas e horários. Além disso, também é possível notar a presença de entidades que não são diretamente relacionadas à alocação de turmas, mas que podem se mostrar úteis no futuro, como a relação entre professores e laboratórios; e a de disciplinas e ementas.

Embora o diagrama apresente uma visão mais completa das interconexões possíveis, é importante ressaltar que o presente trabalho foca primordialmente na alocação das turmas para o curso de Ciência da Computação, e que a implementação do banco de dados será feita de forma a atender a essas necessidades, fazendo então uso de apenas uma parte do diagrama conceitual apresentado pela [Figura 4.3](#). Sendo posteriormente ilustrado ([Figura 5.16](#), [Figura 5.25](#) e [Figura 5.33](#)).

¹ <https://www.drawio.com>

4.5 API REST

Para exemplificar o funcionamento geral da permanência dos dados, consideremos o uso de uma *API REST* utilizando de 4 “camadas”: o **frontend**, os **endpoints**, as **funções** de execução e o **banco de dados**.

O **frontend** é a interface do sistema, onde o usuário interage com o sistema. Essa interface precisa se comunicar com o *backend* para realizar as quatro operações básicas no banco de dados (criação, leitura, atualização e deleção) por sobre as entidades existentes (turmas, professores, disciplinas, salas, etc.). Elas assim o fazer ao enviar requisições HTTP (*GET*, *POST*, *PUT* e *DELETE*), contendo pacotes de informações em formato JSON para os **endpoints**.

Os **endpoints** são as rotas que o *backend* disponibiliza para a recepção das requisições HTTP. Eles são responsáveis por encaminhar as requisições recebidas. Seu funcionamento consiste em rotear as requisições recebidas junto com sua carga útil. Para tanto, as rotas criadas refletem diretamente a qual entidade do banco de dados a requisição se refere, sendo então assim sabido qual **função** deve ser executada.

As **funções** são as responsáveis por executar as operações no banco de dados. Elas processam o pacote de informações recebido, e então realizam a operação desejada no **banco de dados**.

O **banco de dados** recebe a requisição, processa a operação, e então retorna o status da operação. Esse retorno é então repassado camada por camada, até chegar ao **frontend**, onde o usuário final pode visualizar o resultado da operação.

Resumidamente: O **frontend** envia uma requisição HTTP com uma carga de informações a um **endpoint**, que encaminha a requisição a uma **função** específica que executa uma operação no **banco de dados**, assim retornando o status da operação ao **frontend**.

4.6 Diferença dos trabalhos anteriores

Os trabalhos mais próximos ao presente, quanto a instituição são os trabalhos feitos por [Santos \(2013\)](#) e [Silveira \(2014\)](#), porém, o presente trabalho se distingue em algumas questões.

Talvez uma das mais marcantes distinções é a visualização presente nos dois trabalhos anteriores onde são considerados um limite de cinco disciplinas por horário, pois ambos consideram em seus trabalhos que o curso de computação está dividido entre as cinco turmas de cada grupo de períodos, sejam as do semestre ímpar (1, 3, 5, 7 e 9) ou as do semestre par (2, 4, 6, 8 e 10). No entanto, o presente trabalho avalia que, é

mais condizente com a realidade da instituição se for considerado que os alunos tendem a não seguir completamente essa subdivisão, tendendo a ficar com disciplinas de semestres diferentes geralmente aquém do esperado, mas também podendo estar além do que se espera.

Outra característica marcante é a dependência de salas disponíveis que comportem o número de alunos esperados para a turma, o que não é considerado pelos autores, visto que esta característica não é vista como pertencente ao escopo da Coordenação do Curso de Ciência da Computação. Já neste trabalho, considera-se que esta informação é essencial para que haja a confirmação da validade da alocação da turma. Além disso, como existem diversos outros cursos que utilizam as salas do CCT, é importante que a coordenação tenha conhecimento sobre a disponibilidade das salas para que possa alocar as turmas de forma eficiente, sendo então necessário o preenchimento de informações sobre as disciplinas necessárias para os outros cursos.

Existem também alguns critérios usados por [Santos \(2013\)](#) e [Silveira \(2014\)](#) que não são considerados no presente trabalho, como a preferência de professores associados e professores contratados, e o desejo por se evitar alocar disciplinas que possuam aulas em horários próximos ou que sobreponham o horário de almoço.

5 DESENVOLVIMENTO DO SISTEMA DE SUPORTE À DECISÃO

Este capítulo visa apresentar todo o processo realizado para o desenvolvimento do sistema de suporte à decisão. O propósito desta aplicação é auxiliar o coordenador do curso de Ciência da Computação da UENF na alocação interativa de turmas, professores e salas. O auxílio é dado através de duas formas: a primeira é a [criação automatizada da grade baseada no histórico das grades anteriores](#), a segunda é a [identificação de conflitos](#), como a alocação de uma turma grande demais para a capacidade da turma, com o objetivo de auxiliar no aprimoramento da grade inicialmente gerada.

Inicia-se com a apresentação de [projetos anteriores](#) que serviram de base para o desenvolvimento do presente trabalho. Em seguida, será apresentado o [acesso aos dados acadêmicos da instituição](#) e as dificuldades encontradas para a sua obtenção. Posteriormente, será apresentado o processo de [prototipagem](#) do sistema, onde foram criados os designs iniciais das telas do sistema. Em seguida, será apresentado o processo de [programação do sistema](#), que foi dividido em três grandes categorias que visavam entregar o sistema de forma gradual e funcional.

Por fim, será apresentado o cerne do sistema que é a [identificação e visualização de conflitos](#), ou seja, os casos em que a alocação dos recursos gera algum comportamento minimamente indesejável como por exemplo a alocação de uma sala a duas turmas ao mesmo tempo. Serão apresentados os tipos de conflitos que o sistema é capaz de identificar, como eles são visualizados e como o sistema lida com cada um deles. Conclui-se então com a apresentação do [preenchimento do banco de dados](#) com dados reais.

5.1 Projetos anteriores

Antes do desenvolvimento do presente trabalho, outros projetos já foram idealizados e até mesmo desenvolvidos. Dentre eles, dois se destacam por terem sido os precursores do atual projeto. O primeiro deles, que foi idealizado, mas não desenvolvido, almejava apresentar uma [visualização da progressão dos alunos na grade curricular](#). O segundo, que foi desenvolvido, realizava o [cálculo da demanda](#), ou seja, calculava quantos e quais eram os alunos que poderiam se inscrever nas disciplinas. Ambos os projetos se mostram como abordagens paralelas de auxiliar no planejamento do curso de Ciência da Computação da UENF, sendo eles também sistemas de suporte à decisão, mesmo que em menor escala.

Deve-se comentar também sobre os dois trabalhos anteriores desenvolvidos por [Santos \(2013\)](#) e [Silveira \(2014\)](#) que iniciaram os estudos sobre a alocação de turmas no

curso de Ciência da Computação na UENF, mas que, como já foram citados previamente, não se mostra necessária a repetição.

5.1.1 Andamento dos alunos

Como interesse próprio, cogitou-se o desenvolvimento de uma plataforma onde se pudesse ver em que ponto os alunos se encontram em relação ao andamento de seus cursos. Para isso, seria necessária a obtenção dos dados dos alunos, seja por parte dos mesmos, do coordenador ou por integração com o sistema acadêmico. Com estes dados, seria possível criar uma interface que mostrasse o andamento dos alunos, quais matérias já foram cursadas, quais estão sendo cursadas e quais ainda faltam. Além disso, seria possível mostrar quais matérias são pré-requisitos para outras. Assim, o aluno e a coordenação poderiam ter uma visão geral de seu andamento e de quais matérias ele precisará cursar para se formar. Esse projeto não saiu do mundo das ideias, entretanto, lá permaneceu sendo maturado.

Figura 5.1 – Andamento do aluno no Sistema Acadêmico

9º Período				
Disciplina	C.H.	Cumprida com		
INF01130 - Projeto de Monografia	136	INF01130	Inscrição	2023/1
10º Período				
Disciplina	C.H.	Cumprida com		
INF01127 - Estágio Supervisionado	204	INF01127	Inscrição	2023/2
INF01131 - Monografia	136	INF01131	Inscrição	EM CURSO
Disciplinas Optativas Eletivas				
<input checked="" type="checkbox"/> Somente aproveitadas				
Exigidas: 306 horas				
Obtidas: 272 horas				
Disciplina	C.H.	Cumprida com		
INF01220 - Fundamentos de Processamento de Imagens	68	INF01220	Inscrição	2023/1
INF01133 - Tópicos Especiais em Inteligência Artificial II - Sistemas Inteligentes	68	INF01133	Aproveitamento de atividade	2022/2E
INF01218 - Tópicos Especiais em Simulação Computacional II: Heurísticas e Complexidade	68	INF01218	Inscrição	2022/2
INF01226 - Tópicos Especiais em Computação II: Programação de Aplicações de Microcontrolador Digital	68	INF01226	Aproveitamento de atividade	2021/2

Fonte: autoria própria

Anos após a idealização dessa funcionalidade, o Sistema Acadêmico da UENF passou a disponibilizar uma funcionalidade semelhante, como pode ser visto na Figura 5.1. Nela, é possível ver o andamento do aluno por período, incluindo as disciplinas Eletivas Livres e Eletivas Optativas. Entretanto, essa visualização tabular não ilustra o andamento

de todos os alunos simultaneamente, sendo necessário navegar individualmente por cada um deles.

5.1.2 Cálculo de demanda

Ao longo dos semestres, foi percebido que durante o intervalo entre os semestres, os alunos precisam se inscrever nas matérias que desejam cursar no semestre seguinte. Para isso, é necessário que o coordenador saiba quantos alunos estão interessados em cada matéria para que ele possa definir quantas turmas serão abertas. Com este fim em mente, o coordenador dispõe de algumas alternativas como estimar quantos alunos se inscreverão em cada disciplina, checar manualmente no sistema acadêmico quais alunos podem fazer cada matéria, ou então obter diretamente dos alunos através de um formulário em quais disciplinas cada um dos alunos tem a intenção de cursar.

O método que o atual Coordenador de Ciência da Computação utiliza consiste em baixar o extrato de todos os alunos do curso e tabelar no Excel qual é o andamento de cada um dos alunos, para que assim, através da análise manual, pudesse ver qual é o andamento de cada um e de quantos alunos demandam quais disciplinas.

Entretanto, todas essas alternativas são trabalhosas e propensas a erros. Sendo assim, foi pensado em uma forma de automatizar esse processo. Foi então elaborado um código em [Python¹](#) que atualmente [se encontra no GitHub²](#). Este código tem como entrada os extratos de matrícula dos alunos e como saída a listagem das disciplinas demandadas e a listagem dos alunos que demandam cada disciplina.

Código 5.1 – Obter demanda por extratos em PDF

```

1 import code_1_set_working_directory      as swd
2 import code_2_get_pdf_list             as gpl
3 import code_3_get_string_from_pdf     as gsp
4 import code_4_structure_data_from_text as sdt
5 import code_5_filter_structured_data  as fsd
6 import code_6_get_demand_list         as gdl
7 import code_7_merge_demands          as mgd
8 import code_8_output_demand_as_txt   as odt
9
10 swd.set_cwd()
11 pdf_paths      = gpl.get_pdf_list()
12 texts          = gsp.get_pdf_text(pdf_paths)
13 structured_data = sdt.structure_data(texts)
14 approved_codes = fsd.get_approved_codes(structured_data)
15 demands        = gdl.get_demand_list(approved_codes)
16 demands_and_values = mgd.get_merged_demands(demands)

```

¹ <https://www.python.org>

² https://github.com/jvfd3/university_demand

```
17 odt.output_to_txt(demands_and_values)
```

Este código foi desenvolvido em 8 etapas, vistas no [Código 5.1](#), cada uma com um arquivo separado. Para alcançar a lista das demandas, é necessário primeiro obter a lista dos arquivos em formato PDF que serão processados, em seguida extrair seus dados com a biblioteca [PDFMiner³](#), estruturar os dados obtidos, filtrar os dados estruturados, obter a demanda de cada disciplina, juntar as demandas de cada disciplina e salvar os dados obtidos em um arquivo de texto.

Embora o código cumpra com seu objetivo, apresenta algumas características limitantes. A primeira é que os PDFs precisam ser obtidos manualmente, um por um, pelo coordenador, sendo ela por si só uma tarefa extenuante, o que não é desejado. Além disso, o seu uso não é muito intuitivo, sendo necessário que o usuário lide com o prompt de comando e instale as dependências necessárias, o que acaba trazendo uma dificuldade a mais ao usuário. O código também apresenta limitações por sistema operacional, não sendo garantido o seu funcionamento em sistemas operacionais diferentes do Windows.

5.2 Acesso aos dados acadêmicos da instituição

Em sua concepção original, o presente trabalho visaria integrar o sistema desenvolvido com o atual sistema acadêmico da UENF. Essa abordagem foi descartada devido às dificuldades encontradas por parte do setor administrativo da UENF que, devido à [Lei Geral de Proteção de Dados \(LGPD\)⁴](#), não podem divulgar dados dos alunos, mesmo anonimizados.

Para confirmação das informações recebidas, a LGPD foi estudada e talvez o presente estudo recaísse na alínea b do inciso 2º do artigo 4º do capítulo 1 da Lei Nº 13.709, de 14 de agosto de 2018. Informando este que esta lei, a LGPD, não se aplica ao tratamento de dados pessoais realizado para fins exclusivamente acadêmicos.

Segundo o [Estudo Técnico sobre o tratamento de dados pessoais para fins acadêmicos⁵](#), é reforçado que “o tratamento de dados pessoais para fins acadêmicos deve ser sempre lícito”.

Apesar das possibilidades de meios legalmente válidos para a aquisição dos dados, optou-se por abandonar a integração com o Sistema Acadêmico e o uso de dados reais dos alunos já existentes na plataforma. Rumando-se então para uma abordagem de inserção de dados manual por parte dos usuários do sistema.

³ <https://pypi.org/project/pdfminer>

⁴ https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm

⁵ https://www.gov.br/anpd/pt-br/assuntos/noticias/sei_00261-000810_2022_17.pdf

5.3 Prototipagem

A criação de protótipos, seguindo a abordagem tomada por Andre e Dinata (2018), se mostra como essencial para que se mantenha a constante satisfação por parte dos *stakeholders* e quais mudanças sugerem ao desenvolvimento do projeto, assim reduzindo a necessidade de retrabalho e de não alcançar as expectativas do projeto. Para este fim, foram feitos os designs iniciais das telas do sistema usando o software de design Figma⁶, designs esses que representam apenas o esboço, e não um sistema funcional. Esse esboço serviu como uma base visual de como o sistema estaria no final.

5.3.1 Protótipos de componentes

Antes de seguir com a descrição das telas, é válido descrever os componentes que foram utilizados na construção dos protótipos. Os componentes principais são os cartões de turma (Figura 5.2) e as caixas de seleção (Figura 5.3).

A Figura 5.2 mostra os cartões de turma que foram utilizados para representar as turmas criadas. O objetivo desses cartões é de apresentar de forma compacta o máximo possível de informações relevantes ao usuário. Considerando o caráter iterativo do desenvolvimento, foram criados diversas versões dos cartões, todos eles visando a apresentação de informações de forma clara, objetiva e intuitiva por parte do usuário.

Essas informações relevantes são as informações relacionadas à identificação da disciplina, do professor e da sala. Além dessas, são apresentadas informações adicionais como a quantidade de alunos que podem se inscrever na disciplina e alertas de possíveis conflitos.

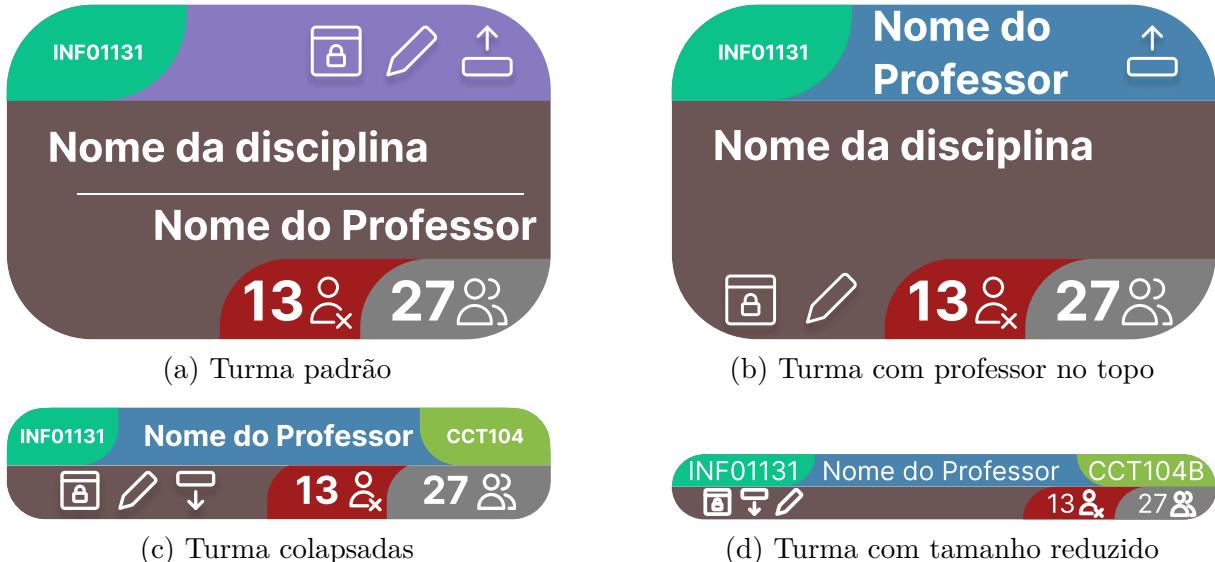
Em todas as versões dos cartões de turma, o código da disciplina está no canto superior esquerdo. No canto inferior direito está a quantidade de alunos que podem se inscrever na disciplina e, à sua esquerda, aqueles alunos que apresentam algum tipo de impedimento, como por exemplo ter alguma outra turma que ele possa se inscrever no mesmo horário. Figura 5.2c

Nestes cartões, estão presentes também três botões: **travar**, **editar** e **expandir/recolher**. O botão de **travar** tem como função fixar a turma no horário em que ela se encontra, impedindo que ela seja movida. O botão de **editar** tem como função abrir um painel onde é possível alterar as informações da turma. O botão de **expandir/recolher** tem como função expandir ou recolher o cartão, mostrando ou escondendo informações adicionais. Em cada uma das versões dos cartões, a posição dos botões foram alteradas.

Outras informações que também tiveram sua posição alterada foram o nome do professor, o nome da disciplina e o código da sala.

⁶ <https://www.figma.com>

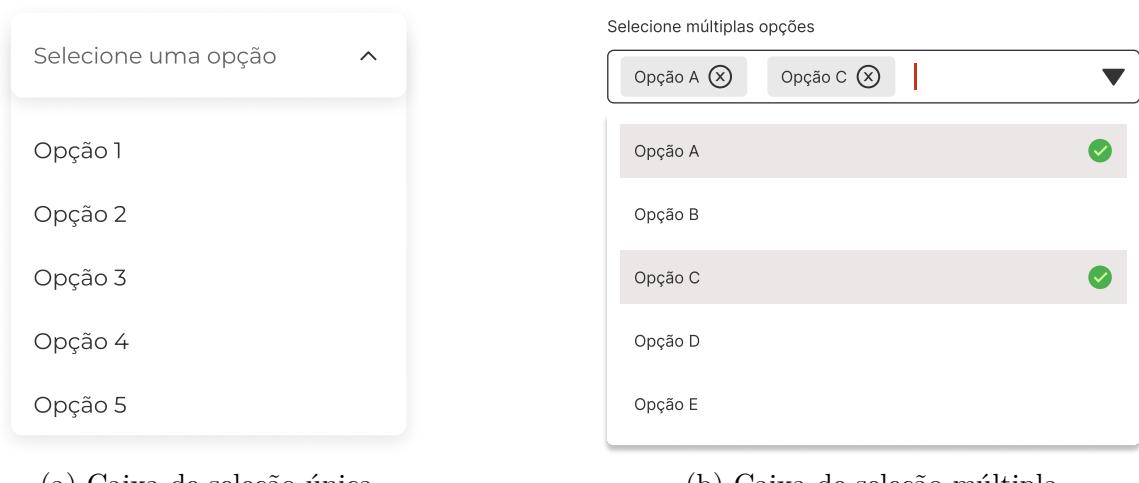
Figura 5.2 – Protótipos de cartões de turma



Fonte: autoria própria

Cada uma dessas versões dos cartões de turma se mostra com um propósito específico. O cartão padrão (Figura 5.2a) é um dos maiores cartões, o que facilita a visualização do usuário às informações importantes, o mesmo pode ser dito para o cartão com o professor no topo (Figura 5.2b), esse, por sua vez, distribui a posição dos botões. Já os botões colapsados (Figura 5.2c) e o cartão com tamanho reduzido (Figura 5.2d) prezam por uma maior economia de espaço, sendo úteis para quando o usuário deseja visualizar mais turmas ao mesmo tempo, como é o caso da visualização da grade horária, e por isso informam também o código da sala em que estão alocados.

Figura 5.3 – Protótipos de caixas de seleção



Fonte: autoria própria

A Figura 5.3 mostra os protótipos de caixas de seleção que foram utilizados para a seleção de dados. Na Figura 5.3a está a caixa de seleção onde o usuário pode selecionar apenas uma opção. Já na Figura 5.3b a caixa de seleção permite ao usuário selecionar várias opções.

5.3.2 Protótipos de páginas

Aqui serão elecadas as sete páginas esboçadas para o sistema. Sendo elas a página principal, a página de seleção, a página de salas, a página de alunos, a página de disciplinas, a página de professores e a página de turmas.

A primeira, e principal, é a ilustrada pela Figura 5.4 que permite que o usuário arraste todas as turmas listadas até o horário desejado. A listagem das turmas a serem distribuídas é disposta em um painel à esquerda. Este painel tem como funcionalidade, fixar as turmas ainda não alocadas, como se estivessem presas por fechos de gancho e laço⁷. Assim, dispondo de um local para que turmas que turmas em processo de mudança de horário sejam armazenadas temporariamente sem que sejam perdidas.

Figura 5.4 – Protótipo da página principal do sistema

Horários	SEG	TER	QUA	QUI	SEX
08:00 ~ 09:00					
09:00 ~ 10:00					
10:00 ~ 11:00					
11:00 ~ 12:00					
12:00 ~ 13:00					
13:00 ~ 14:00					
14:00 ~ 15:00					
15:00 ~ 16:00					
16:00 ~ 17:00					
17:00 ~ 18:00					
18:00 ~ 19:00					
19:00 ~ 20:00					
20:00 ~ 21:00					
21:00 ~ 22:00					

Fonte: autoria própria

⁷ https://en.wikipedia.org/wiki/Hook-and-loop_fastener

Em seguida, temos a tela de seleção da categoria dos dados que deseja-se modificar ([Figura 5.5](#)), podendo esses serem sobre as turmas, salas, disciplinas, professores ou alunos. Cada uma destas tendo a capacidade de criação, leitura, edição e deleção dos dados.

[Figura 5.5 – Protótipo da página de seleção](#)



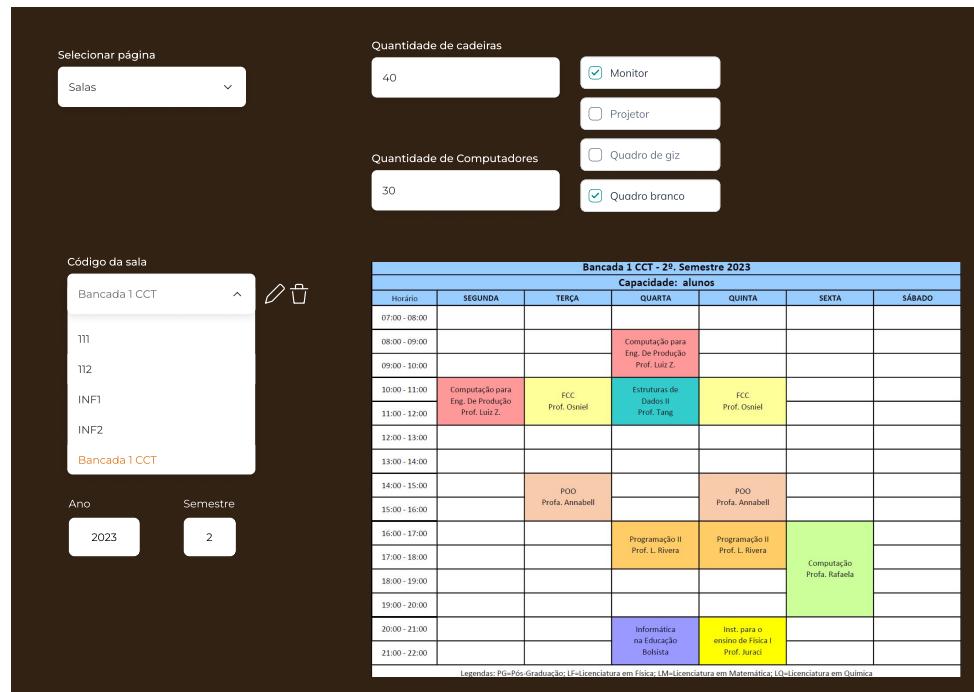
Fonte: autoria própria

Quanto à página das salas, temos primeiro a seleção da sala na qual deseja-se fazer alterações de cadastro. Abaixo desta caixa de seleção única há um filtro de visualização das alocações de determinado ano e semestre. Nessa página pode-se também registrar algumas características da sala, como a quantidade de cadeiras e computadores, e se possui monitor, projetor, quadro de giz e quadro branco. Um exemplo de sala ainda sem turmas alocadas é representado na [Figura 5.6](#). Essas últimas informações, embora não sejam essenciais para a alocação das turmas, podem ser úteis como forma de filtragem para a alocação de turmas, visto que certas disciplinas demandam salas com características específicas.

Na página dos alunos, pode-se cadastrar novos alunos informando o seu ano de entrada e a sua matrícula. Abaixo temos a visualização da grade, onde pode-se classificar cada uma das disciplinas como aprovada, reprovada e cursando. O exemplo da [Figura 5.7](#) mostra a grade de um aluno inscrito em 2019.1.

Podemos definir na página das disciplinas qual seu código, nome, e o seu período esperado segundo a ementa. Além dessas informações, pode-se cadastrar quais cursos a possuem em suas ementas, quais seus pré-requisitos, os professores que a ministram e quais requisitos a mesma possui em relação às características de sala. A [Figura 5.8](#) mostra a página de modificação de disciplinas.

Figura 5.6 – Protótipo da página de salas



Fonte: autoria própria

Figura 5.7 – Protótipo da página de alunos



Fonte: autoria própria

Figura 5.8 – Protótipo da página de disciplinas

O formulário para inserção de uma nova disciplina é dividido em seções:

- Selecionar página:** Disciplinas.
- Dados básicos:**
 - Código: MAT01203.
 - Nome da disciplina: Cálculo II.
 - Período esperado: 2.
- Quals cursos:** CC, Química, Ciência da Computação, Engenharia de Petróleo, Lic. Química, Licenciatura de Física, Licenciatura de Química.
- Pré-requisitos:** Cálculo I.
- Professores que ministram:** Oscar, Rigoberto, Oscar, Rigoberto, Elba, Annabell, Tang.
- Requisitos de sala:** Monitor, Monitor, Giz, Projetor, Piloto.

Fonte: autoria própria

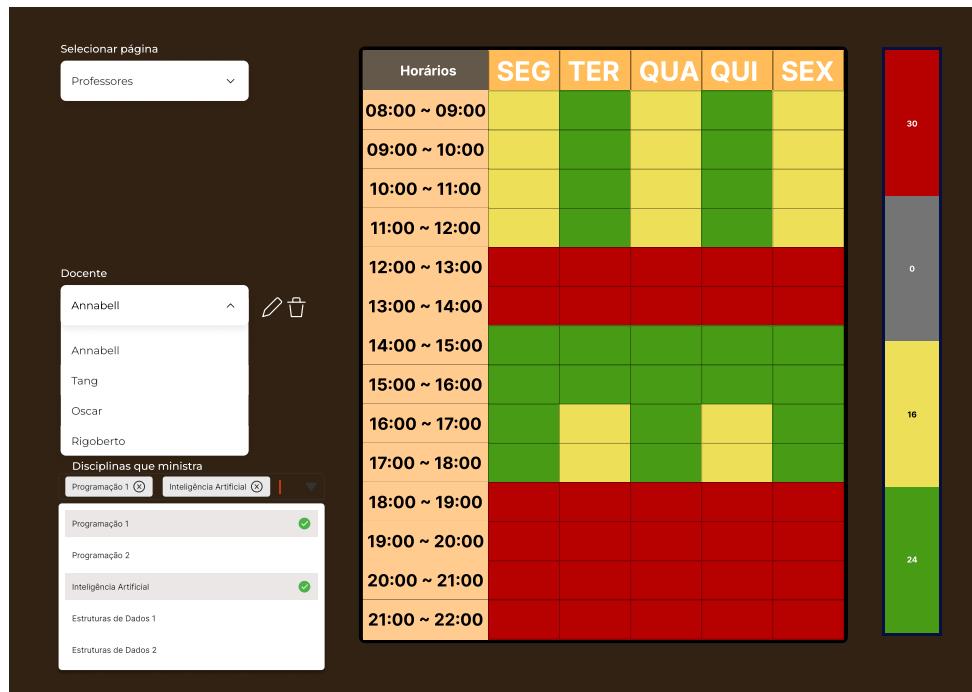
Na página de professores (Figura 5.9), temos a relação de disciplinas que os mesmos estão passíveis de ministrar, e também quais são suas preferências de horários ao longo da semana. Embora não seja essencial, essa informação pode ser útil para a alocação de turmas, pois alguns professores podem ter preferência, ou até mesmo não estarem disponíveis para ministrar aulas em determinados horários. Um exemplo deste caso seriam os bolsistas para docência complementar, dado que os mesmos podem estar também vinculados a outras instituições. E nesses casos, aquele que estiver desenvolvendo a grade horária pode alocar as turmas de acordo com a disponibilidade dos professores.

Por fim, na Figura 5.10, temos a junção de todas as informações registradas acima. Nela, podemos alocar os seus horários, definindo o ano, semestre, dia, hora de início e duração em que será ministrada. Também é necessário que seja definido em que sala cada um de seus horários estará alocada. Além de informar, também, qual professor a lecionará e a qual disciplina ela se refere.

Ainda na Figura 5.10 temos alguns exemplos de conflitos percebidos. O primeiro, com a cor amarelada, informado que o tempo de duração do segundo horário da turma não condiz com a preferência pessoal do professor selecionado. Este conflito não é impeditivo, entretanto, se possível, um outro horário poderia ser encontrado para atender melhor às preferências do professor.

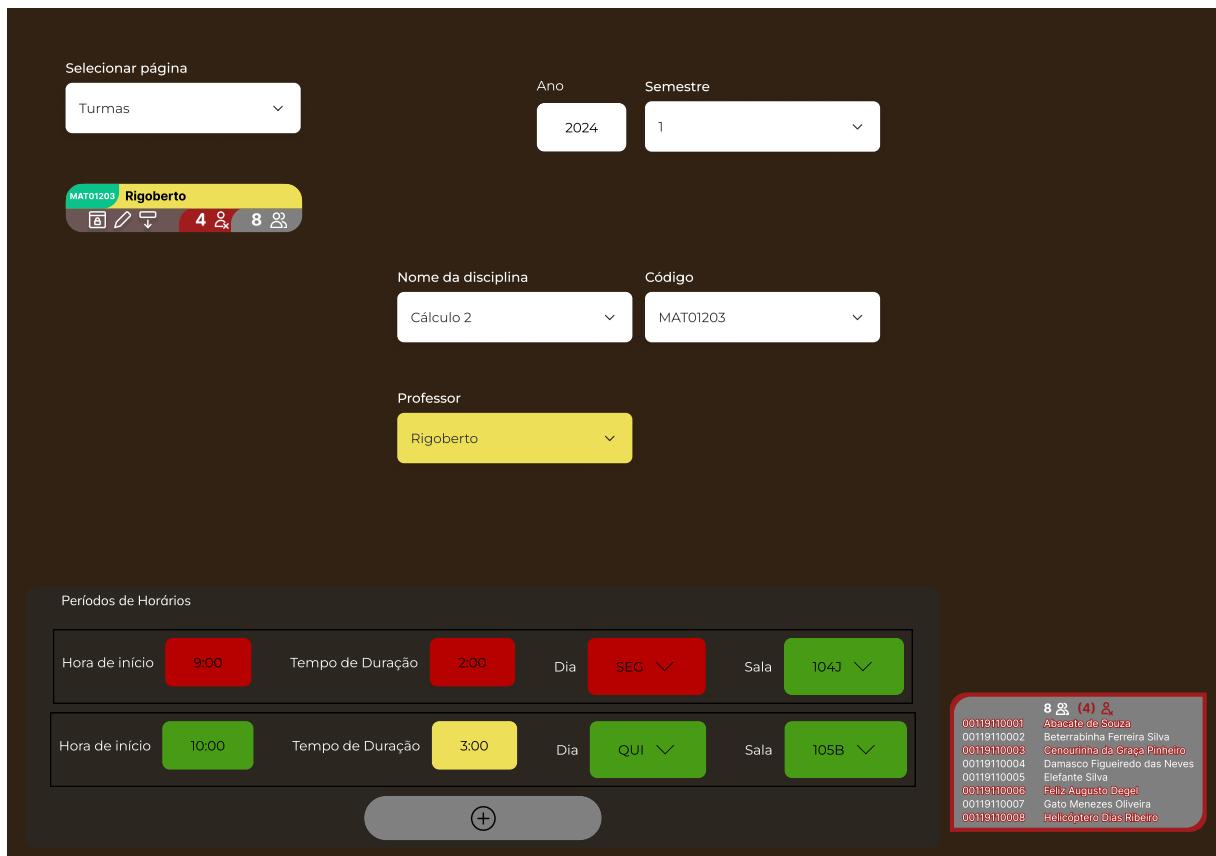
Outro conflito exibido é que a sala em que está alocado o primeiro horário da turma já está ocupada no mesmo horário por outra turma. Este conflito é impeditivo,

Figura 5.9 – Protótipo da página de professores



Fonte: autoria própria

Figura 5.10 – Protótipo da página de turmas



Fonte: autoria própria

sendo então representado na cor vermelha. Neste caso, a turma deve ser realocada para outro horário ou sala.

Por último, na listagem dos alunos que podem se inscrever, no canto inferior direito, há quatro alunos marcados em vermelho. Estes alunos poderiam se inscrever em outra turma no mesmo horário em que esta turma está alocada. Este conflito é impeditivo, mas apenas para os alunos, assim como no caso da preferência do professor, a turma pode ser realocada para outro horário para atender melhor às demandas dos alunos.

5.4 Programação do sistema

Após a conceitualização diagramática do banco de dados e a elaboração dos protótipos com o Figma, o desenvolvimento do sistema foi iniciado. Por maior familiaridade com a linguagem foi escolhida a linguagem **JavaScript**, utilizando a biblioteca **React** para a criação dos componentes visuais, ou seja, o *frontend*, e o **Node.js** para a criação do *backend* e a criação de um servidor local que permite visualizar as mudanças no código em tempo real. Suas logos estão ilustrados na [Figura 5.11](#).

Figura 5.11 – Recursos usados para o desenvolvimento do sistema

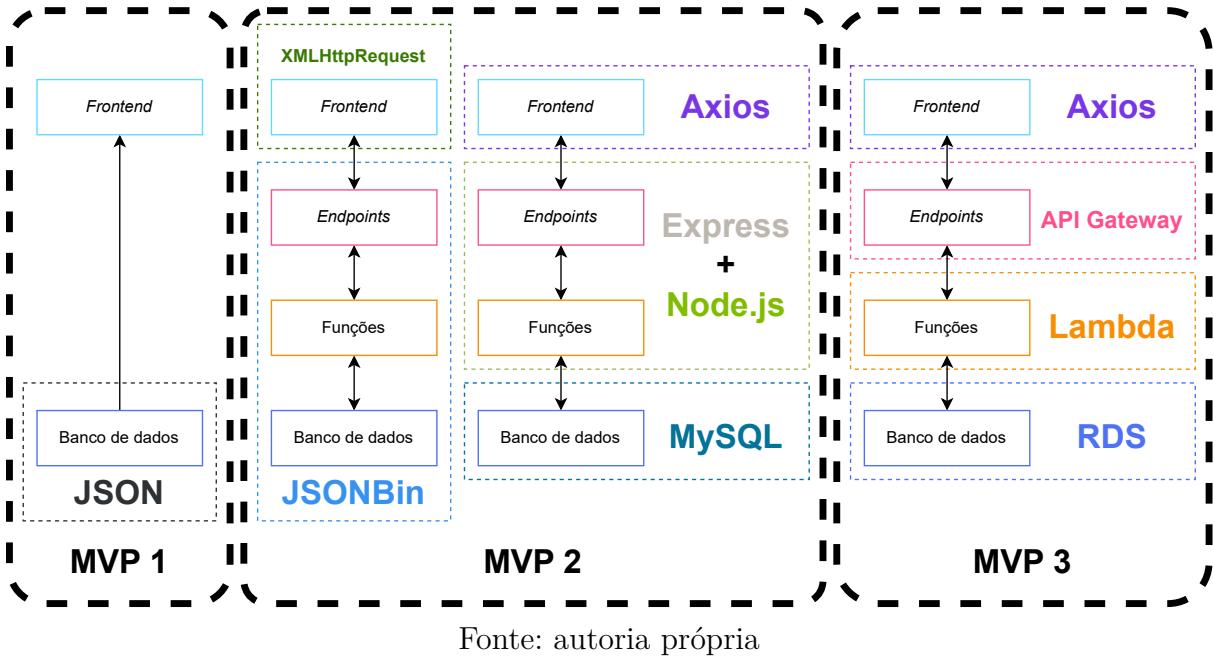
Fonte: autoria própria

Seguindo constantemente o conceito de iteratividade da apresentação do sistema, a programação foi marcada por dois conceitos: blocos de funcionalidade marcantes e blocos de funcionalidade apresentadas. O primeiro conceito se refere à grandes grupos de mudanças que estavam relacionadas a um mesmo tópico, ou que resultaram, quando em conjunto, num sistema consideravelmente distinto de como estava antes de as receber. O segundo conceito se refere à apresentação esporádica da situação atual do sistema para quem o iria utilizar, sendo essas versões chamadas de MVPs (Minimum Viable Product - Mínimo Produto Viável). Nesta seção serão apresentadas as versões do sistema repartidas de forma a apresentar os agrupamentos de mudanças notórias.

Sua [programação](#) foi repartida em três grandes categorias que visavam entregar o sistema de forma gradual e funcional. A [primeira versão do sistema](#) foi desenvolvida localmente com o objetivo de se aproximar ao máximo das páginas previstas no protótipo, sem a necessidade de um banco de dados que permitisse alterações. A [segunda versão do sistema](#) contou com a utilização de duas abordagens distintas de bancos de dados para se obter a permanência dos dados. Já a [terceira versão do sistema](#) foi desenvolvida de forma

a estar completamente hospedada na nuvem, incluindo o seu banco de dados, sendo então preenchido com mais dados.

Figura 5.12 – Diagrama da progressão funcionamento da permanência dos dados



Ao longo da implementação dessas versões, diversos métodos de manutenção dos dados foram utilizados, como a importação de arquivos JSON, a utilização de um banco de dados local e a utilização de um banco de dados hospedado na nuvem. A Figura 5.12 ilustra a progressão do funcionamento da permanência dos dados ao longo das versões do sistema. Os pormenores de cada versão serão descritos adiante.

5.4.1 Versão 1.0

A primeira versão do sistema foi desenvolvida em um ambiente local, com o objetivo de se aproximar ao máximo das páginas previstas no protótipo. Para isso, foi utilizada a biblioteca *React Router*⁸ para a navegação entre as páginas, e a biblioteca *React Select*⁹ para as caixas de seleção.

Banco de dados preliminar

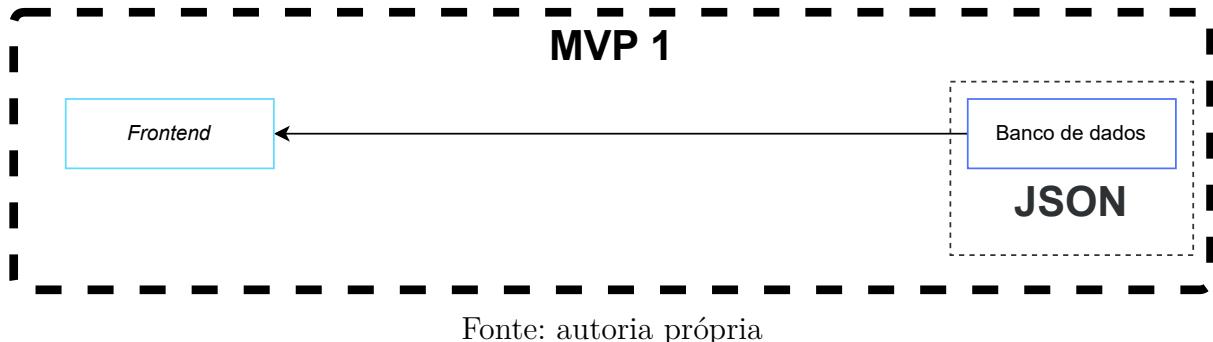
Os dados contidos no sistema foram inicialmente armazenados em arquivos JSON, que eram importados diretamente para o código (Figura 5.13). Isso foi feito para que fosse possível visualizar o funcionamento do sistema sem a necessidade de um banco de dados real. A partir disso, foi possível visualizar o funcionamento do sistema e realizar testes de usabilidade. Em contrapartida, os dados disponíveis não eram modificáveis, tendo apenas

⁸ <https://reactrouter.com>

⁹ <https://react-select.com>

a possibilidade de leitura e mutação temporária, visto que após recarregar ou mudar de página, as mudanças eram perdidas.

Figura 5.13 – Diagrama do armazenamento preliminar dos dados



Fonte: autoria própria

Nesse método, cada entidade era armazenada em um arquivo JSON separado, contendo esse um array de objetos, onde em cada objeto haviam as chaves, representando as propriedades da entidade, e os valores, representando os dados da entidade.

Como nesta dinâmica não havia uma forte correlação entre os dados, o *frontend* acabava sendo o responsável por unir todas as informações. Assim, por exemplo, para se obter a lista de professores de uma turma, era necessário importar todos os professores, todas as turmas, e então, a partir do nome do professor alocado àquela turma, buscar na listagem dos professores qual era o professor que correspondia àquele nome, para então agregar as informações.

Funcionalidades iniciais: CRUD e primeiros conflitos

Nessa primeira versão, algumas funcionalidades já começaram a ser esboçadas, principalmente as funcionalidades CRUD (*Create, Read, Update, Delete*) para as entidades principais do sistema. Embora, como já dito, os dados não fossem persistentes, foi possível visualizar o funcionamento das funcionalidades de criação e leitura de turmas, professores, disciplinas, salas e horários.

Nessa versão, também foi implementada uma checagem bruta de conflitos por alocação simultânea de professores em mais de uma turma e a checagem da quantidade de demanda de alunos em relação à capacidade das salas. Uma descrição mais detalhada das funcionalidades de conflitos está presente adiante na [seção 5.6](#) denominada [Detecção e alerta dos conflitos](#).

Preferências dos professores, progressão dos alunos e recursos da sala

Além das funcionalidades citadas anteriormente que se mantiveram até a conclusão do sistema, também foram desenvolvidas funcionalidades que não obtiveram o mesmo êxito e que foram deixadas de lado. Dentre elas, podemos citar a definição de níveis de

preferência de horários para professores, a progressão dos alunos em relação às disciplinas, e a definição das características especiais das salas. Houveram também outras que nem chegaram a ser desenvolvidas, como a realocação de turmas através de um sistema de arrastar e soltar e o uso de heurísticas para a realocação de turmas.

A tabela de preferências de horários para professores foi desenvolvida e consistia em permitir a definição de níveis de preferência para cada um dos horários da semana. A ideia era que, ao alocar as turmas, o sistema pudesse priorizar os horários que fossem mais bem avaliados pelos professores, assim aumentando sua satisfação sempre que possível. A funcionalidade foi descartada por ser considerada de menor prioridade em relação a outras funcionalidades, segundo a coordenação de computação, não sendo ela um dos fatores mais relevantes para a definição das grades horárias no curso de Ciência da Computação.

GitHub Pages

Após o desenvolvimento local foi feita a implementação da interface do sistema para um servidor online. Para isso, foi utilizado o serviço GitHub Pages que, por ser gratuito e de fácil utilização, foi a escolha mais adequada para o momento.

Para implantar no GitHub Pages o código desenvolvido, utilizou-se da [biblioteca gh-pages¹⁰](#), que viabiliza a publicação de um site diretamente do repositório do GitHub. A partir disso, o sistema foi disponibilizado para acesso público, permitindo que qualquer pessoa pudesse acessar o sistema e testar suas funcionalidades.

5.4.2 Versão 2.0

Utilizando do *feedback* quanto aos resultados entregues na [primeira versão](#), alguns pontos de melhoria foram identificados, sendo um deles, e o mais importante: o planejamento. Na primeira abordagem, o desenvolvimento foi feito seguindo notas e ideias soltas, sem um planejamento prévio, o que resultou em um sistema que, embora funcional, não atendia a todas as necessidades propostas. Também dispunha de funcionalidades que não eram de todo necessárias, ou, melhor dizendo, que tinham menor prioridade do que muitas outras. Como solução, foi utilizado o [GitHub Projects](#) para organizar as tarefas e priorizá-las.

Seguindo o planejamento feito, uma das primeiras tarefas foi a implementação do banco de dados que permitisse a [permanência dos dados](#), para isso, foram utilizados dois métodos de manutenção dos dados: o [JSONBin](#) e o [MySQL](#). Outro ponto foi a criação da [logomarca](#) do sistema, que foi feita para que o sistema tivesse uma identidade visual própria. Foram desenvolvidas também [algumas outras funcionalidades](#), como a análise de

¹⁰ [LinkBibliotecaGHPages](#)

mais conflitos, a possibilidade de se filtrar as turmas e a visualização de disciplinas que ainda não têm uma turma criada no presente semestre.

GitHub Projects

Utilizando o [GitHub Projects](#)¹¹, foi organizada uma tabela de tarefas, vista na Figura 5.14, onde foram unificadas as diversas anotações e ideias, antes soltas. A partir disso, foi possível visualizar o que era mais importante e o que poderia ser deixado de lado.

Figura 5.14 – Tabela de tarefas do GitHub Projects

Filter by keyword or by field							
	Title	Status	Milestone	Labels	Página	...	Assignees
20	Visualizar como está a possível grade horária semestral de cada um dos alunos #20	Pendente	Futuro: demanda	Nova função Qualidade de vida	Alunos		
21	Melhorar o efeito de hover da caixa de seleção #21	Pendente	Futuro	Nova função UX	Sistema		
22	Cada Multiselect deveria apenas mostrar um item selecionado por linha #22	Pendente	Futuro	Nova função Qualidade de vida	Sistema		
23	Finalizar o CRUD #23	Concluído	MVP 2	Nova função	Banco de Dados		jvfd3
24	Tabela que apresenta a preferência dos professores #24	Pendente	Futuro: conflitos	Nova função	Professores		
25	Fazer a grade curricular de CC clicável para definir como aprovado #25	Pendente	Futuro: demanda	Nova função	Alunos		
26	Mostrar a lista dos professores que ministram a disciplina #26	Pendente	Futuro	Nova função	Disciplinas		
27	Definir status dos alunos à turmas de anos anteriores #27	Pendente	Futuro: demanda	Nova função	Alunos		
28	Corrigir a remoção de horários da turma #28	Concluído	MVP 1	bug	Turmas		jvfd3
29	Corrigir seleção de professores na turma #29	Concluído	MVP 1	bug	Turmas		jvfd3
30	Visualizar conflitos impeditivos #30	Concluído	MVP 1	Nova função	Turmas		jvfd3
31	Conflito por capacidade da sala V2: duas turmas para a mesma disciplina. #31	Desenvolvendo	Futuro: conflitos	Nova função	MultiTurmas		jvfd3
32	Conflito por preferência do professor #32	Pendente	Futuro: conflitos	Nova função	Turmas		
33	Conflito por professor não podendo ministrar a disciplina definida #33	Pendente	Futuro: conflitos	Nova função	Turmas		
34	Mostrar quantos conflitos de cada nível estão ocorrendo #34	Pendente	Futuro: conflitos	Nova função UX	Turmas		
35	Conflito de aluno querendo fazer matéria que não tem requisito #35	Pendente	Futuro: conflitos	Nova função	Turmas		
36	Conflito de alunos que estão com matérias/turmas no mesmo horário #36	Pendente	Futuro: conflitos	Nova função	Turmas		
37	Impedir que a demanda estimada seja negativa #37	Concluído	MVP 2	Nova função UX	Componente		jvfd3

Fonte: autoria própria

Tendo este novo sistema de tarefas em prática, foi possível planejar melhor quais eram as funcionalidades que precisavam ser desenvolvidas, as que já estavam prontas, as que poderiam ter melhorias e quais se desejava implementar no futuro.

As tarefas foram inicialmente divididas em três principais categorias: *Status*, *Página* e *Sequência*. O *Status* reflete o andamento do código da tarefa, podendo ser este andamento **Pendente**, **Desenvolvendo**, ou **Concluído**. A *Página* reflete em qual página do sistema a tarefa se encontra, e a *Sequência* reflete a ordem de prioridade da tarefa.

Citando mais detalhes da Figura 5.14, temos à esquerda a numeração das tarefas, seguido do título da tarefa que descreve em poucas palavras sobre o que se trata. ao final do título há uma combinação do símbolo “#” e uma numeração. Essa numeração representa

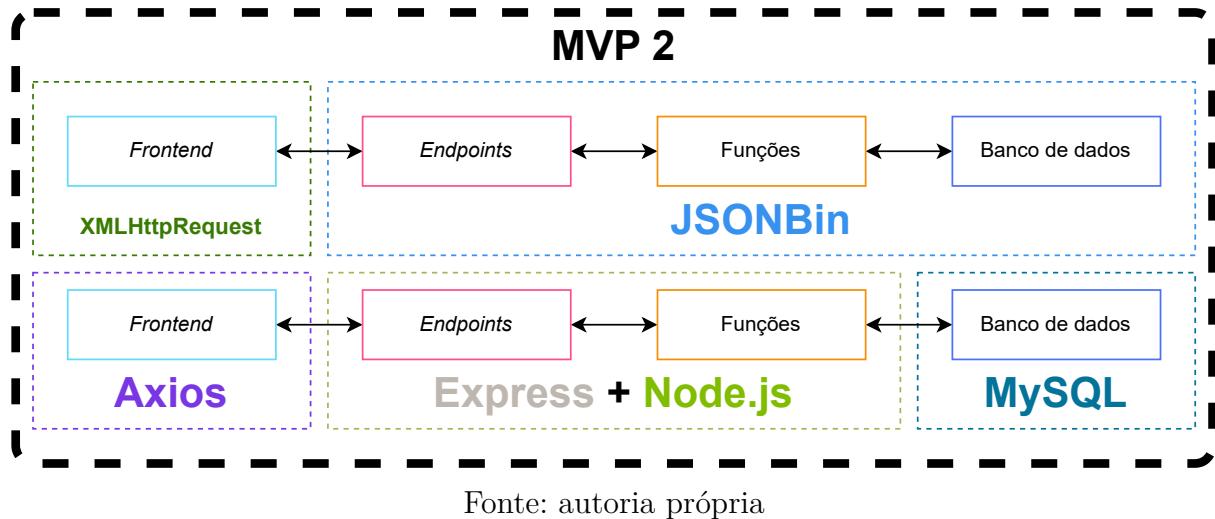
¹¹ <https://docs.github.com/pt/issues/planning-and-tracking-with-projects>

a sequência de criação das tarefas, código esse que pode ser usado como referência entre as tarefas, assim permitindo que tarefas correlacionadas tenham um link direto entre elas. Na próxima coluna está o *Status* da tarefa, seguido de duas colunas, **Milestone** e **Labels** que só foram criadas [posteriormente](#). Finalizando temos a coluna **Página** que primariamente distingue em qual página do sistema a tarefa se encontra, mas que também contém algumas categorias que fogem do conceito de página, como por exemplo é o caso das categorias “Componente”, “Sistema” e “Banco de Dados”. Por último há a coluna *Assignees* que indica quem é o responsável pela tarefa; nesse caso, todas as tarefas que estão em andamento ou que já se encontram concluídas foram atribuídas ao mesmo responsável.

Permanência na alteração dos dados

Tendo agora uma rota mais clara a ser seguida, o desenvolvimento foi retomado. Uma das características mais marcantes e ainda não atribuídas ao sistema era a manutenção das modificações feitas nos dados. Nesta etapa, foram utilizados dois métodos de se manter as alterações feitas aos dados. Com esse intuito foram utilizados dois métodos de manutenção dos dados o [JSONBin](#) que não atingiu às expectativas e o [MySQL](#) que serviu para a criação de um banco de dados local, já emulando o posterior uso de um [banco de dados hospedado na nuvem](#). Como nessa etapa houveram esses dois métodos de manutenção dos dados, a [Figura 5.15](#) ilustra o paralelo entre os dois métodos.

Figura 5.15 – Comparação entre bancos de dados da Versão 2.0



[JSONBin](#)¹²

Como até então os dados estavam armazenados em formato JSON, imaginou-se que a melhor forma de persistir os dados seria através de um banco de dados que lidasse

¹² <https://jsonbin.io>

com JSON, o escolhido para este fim foi o JSONBin por apresentar ser uma plataforma gratuita e de fácil utilização.

Esta plataforma permite a criação de *bins*, que são basicamente coleções de dados em formato JSON. A partir disso, é possível realizar requisições HTTP para a leitura, escrita, atualização e remoção dos dados. A utilização do JSONBin foi feita através de requisições HTTP usando o objeto *XMLHttpRequest* do JavaScript, e a comunicação entre o *frontend* e o JSONBin foi feita através de *tokens* de acesso. Este fluxo é representado pela seção superior da [Figura 5.15](#).

Com isso, se tornou possível ler e atualizar os dados de forma remota, e assim, manter os dados mesmo após a recarga da página. Embora cumprisse com o que promete e o que era desejado, o JSONBin não se mostrou a melhor escolha para o sistema, visto que a sua utilização não performou tão bem quanto se esperava. A utilização do JSONBin para a coleta dos dados, fazia com que a tela de carregamento do sistema demorasse alguns segundos para ser exibida, o que não é apropriado para a usabilidade do sistema proposto.

MySQL¹³

Embora houvesse o desejo do uso de informações em formato JSON, achou-se por bem utilizar um banco de dados mais usual, recorrendo então ao MySQL, sendo então necessário criar um banco de dados local que armazenasse os dados e que pudesse ser acessado pelo sistema. Essa configuração serviu para estabelecer a supracitada camada de banco de dados. E consistiu basicamente na instalação do MySQL Server.

Migração dos dados

Como os dados se encontravam em formato JSON, primariamente utilizou-se da ferramenta de importação de dados do próprio **MySQL Workbench**. Durante essa importação, o software automaticamente identifica os campos, criando a tabela e suas colunas. Porém, devido à quantidade dos dados, essa importação tendia a ser demorada, e por vezes, falhava .

Com isso, foi necessário recorrer a uma abordagem semimanual, sendo então desenvolvido um código em Python que lê os arquivos JSON e os converte em arquivos SQL para que as *queries* pudessem ser executadas no MySQL Workbench. A partir disso, foi possível importar os dados de forma mais rápida e eficiente.

Apesar da primeira tentativa de importação não ter sido completamente bem sucedida, foi desta forma que as tabelas, representadas pela [Figura 5.16](#), foram inicialmente criadas. Não seguindo objetivamente a modelagem anteriormente citada. Isso gerou

¹³ <https://www.mysql.com>

posteriormente a necessidade de ajustes manuais, como a adição de chaves primárias e estrangeiras, e a alteração de tipos de dados. Porém, como neste momento, o sistema visava apenas replicar o funcionamento do JSONBin, essas alterações não foram feitas de imediato.

Figura 5.16 – Diagrama inicial das tabelas de dados SQL

Disciplinas			Professores		
id	idDisciplina	PK	id	idProfessor	PK
int	periodoEsperado		string	apelidoProfessor	
string	apelidoDisciplina		string	curso	
string	codigoDisciplina		string	laboratorio	
string	nomeDisciplina		string	nomeProfessor	

Turmas			Salas		
id	idTurma	PK	id	idSala	PK
int	ano		int	capacidade	
int	demandEstimada		string	bloco	
int	semestre		string	blocoSala	
string	nomeDisciplina		string	codigoSala	
string	nomeProfessor		string	descricaoBloco	

Fonte: autoria própria

Acesso ao Banco de Dados

Seguindo a mesma sequência de camadas, o acesso ao banco de dados continua sendo feito através de requisições HTTP, porém, ao invés de serem enviadas ao JSONBin, são enviadas a um servidor local que executa as operações no banco de dados. Essa modificação é representada pela seção inferior da [Figura 5.15](#).

No *frontend*, enquanto que para acessar a API já pronta do JSONBin foi utilizado o objeto *XMLHttpRequest*, para a comunicação com o banco de dados local, foi utilizada a biblioteca [Axios¹⁴](#) para construir as requisições HTTP. E elas, ao invés de serem enviadas ao JSONBin, são enviadas ao **servidor local**.

Na criação deste **servidor local** utilizou-se a biblioteca [Express¹⁵](#) para desenvolver um *backend* local executado com o [Node.js¹⁶](#) paralelo ao *frontend*. Essa biblioteca é

¹⁴ <https://axios-http.com>

¹⁵ <https://expressjs.com>

¹⁶ <https://nodejs.org>

responsável por criar todas as rotas necessárias para a comunicação entre o *frontend* e o banco de dados. A partir disso, foi possível criar rotas para cada uma das entidades, e para cada uma das operações CRUD. Com isso, cada operação CRUD em cada uma das rotas é encaminhada para uma **função específica** que executa a operação no banco de dados.

Este uso, embora exemplifique a aplicação da permanência dos dados, está limitado por dois aspectos: em primeira instância, a permanência dos dados é limitada ao servidor local, não sendo este o desejo final do sistema. Em segunda instância, para haver o acesso aos dados, é necessário que, além do banco de dados, o *backend* também esteja em execução, entretanto, o GitHub Pages, onde o sistema está hospedado, não viabiliza essa execução. Com isso viu-se necessária a busca por um serviço de hospedagem que disponibilizasse a execução do *backend*.

Logomarca

Um dia após o Natal, foi criado um dos arquivos do sistema chamado “hourclassMagic.js”, este arquivo agrupava funções que consistiam basicamente em adicionar e remover horários (*hour*) e turmas (*class*) da listagem de turmas e horários.

Porém, considerando que a junção dos nomes *hour* e *class* era consideravelmente similar à palavra *hourglass* (“ampulheta” em inglês) e que a ampulheta é um símbolo que remete ao tempo, e que o sistema é um sistema de alocação de horários, optou-se por utilizar a ampulheta como símbolo do sistema.

Quanto ao nome, como o sistema visa a alocação de turmas, especificamente para o curso de Ciência da Computação na UENF, decidiu-se pela corruptela da palavra *hour* para que se tornasse *our*, que significa “nosso” em inglês, assim, remetendo ao sentido de individualidade do sistema. Com isso, o nome do sistema se tornou *OurClass*.

Figura 5.17 – Logomarcas do sistema



Fonte: autoria própria

Juntando o símbolo da ampulheta com o nome *OurClass* e outros elementos gráficos, como as tabelas horárias, foi criada a logomarca do sistema, que pode ser vista na [Figura 5.17](#). A logomarca foi criada utilizando de inteligências artificiais gerativas, principalmente o *Bing*. A partir disso, foi possível criar uma logomarca que refletisse o propósito do sistema, e que fosse agradável visualmente.

Figura 5.18 – Logomarca oficial



Fonte: autoria própria

Após analisar as possibilidades, foi escolhida a [Figura 5.18](#) como a logomarca oficial do sistema. A logomarca foi então adicionada ao sistema.

Funcionalidades adicionais: conflitos, filtragem e disciplinas não oferecidas

Acrescendo à visualização de conflitos desenvolvida na primeira versão, foi implementada a visualização de conflito por capacidade de salas, ao comparar com a quantidade de alunos estimados para a turma. Mais detalhes sobre os conflitos podem ser vistos mais adiante na [seção 5.6](#) denominada [Detecção e alerta dos conflitos](#).

Adicionou-se também diversas filtragens, principalmente na página de **Grade Horária**. Dessa forma, torna-se possível a visualização específica de turmas que atendam a certos critérios. Essa filtragem é feita através de caixas de seleção, onde é possível selecionar quais critérios se deseja filtrar, sendo eles: ano, semestre, categoria, disciplina, professor e sala. Essa coletânea de filtros viabiliza uma análise mais limpa das informações estruturadas, podendo então gerar *insights* quanto ao posicionamento histórico das turmas.

Outra utilidade adicionada, agora na página **MultiTurmas**, foi a seção de “Disciplinas ainda não oferecidas”. Sua funcionalidade consiste em dispor ao usuário uma lista de disciplinas que, segundo a ementa de Ciência da Computação, deveriam ser ofertadas

naquele semestre. A partir disso, o usuário pode então selecionar o botão correspondente àquela disciplina e, a partir disso, uma turma para esta disciplina é adicionada à lista de turmas ofertadas. Há também um botão no topo que permite a adição de todas as disciplinas de uma vez.

5.4.3 Versão 3.0

A primeira versão trouxe a possibilidade de se visualizar o sistema, a segunda trouxe a possibilidade de se manter os dados. A terceira e última versão do sistema foi desenvolvida de forma a estar completamente hospedada na nuvem, incluindo o seu banco de dados.

Assim como na versão anterior, alguns novos ajustes foram feitos na definição e distribuição de tarefas no [GitHub Projects](#). Mas os dois principais focos desta versão foram o uso da [Amazon Web Services](#) para a hospedagem do banco de dados e a [reimplementação da heurística para criação de uma grade horária inicial](#).

Mudanças no GitHub Projects

O uso do GitHub Projects se provou como uma excelente forma de organização das tarefas, e assim, foi mantido para a terceira versão. Alguns novos detalhes foram adicionados em sua organização, como a adição da categorização de tarefas por **Marcos** (*Milestones*, [Figura 5.19](#) e [Figura 5.20](#)) e **Etiquetas** (*Labels*, [Figura 5.21](#)).

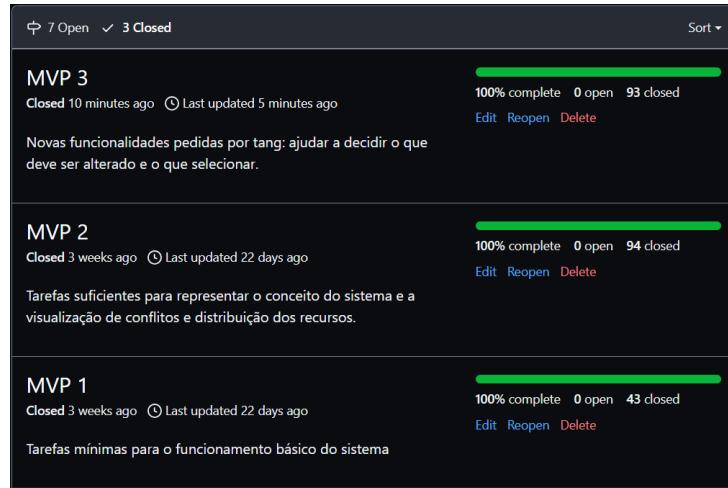
Marcos e Etiquetas

Os **Marcos** visam distinguir as tarefas por suas versões, e também por seus tipos de funcionalidades futuras, assim, a medida em que surgiam novas ideias, elas eram adicionadas ao GitHub Projects. Dessa forma, garantindo uma metrificação do andamento de cada categoria de funcionalidades, além de afunilar a quantidade de tarefas realmente prioritárias para o sistema.

Enquanto a [Figura 5.19](#) mostra os marcos que já foram concluídos, sendo eles as três versões desenvolvidas, a [Figura 5.20](#) mostra os marcos que ainda estão em aberto, podendo ser retomadas no futuro.

1. **MVP 1**: foram concluídas na primeira versão;
2. **MVP 2**: foram concluídas na segunda versão;
3. **MVP 3**: foram concluídas na terceira versão;
4. **MVP 4**: planejadas para a próxima versão a ser desenvolvida;

Figura 5.19 – Marcos concluídos do GitHub Projects



Fonte: autoria própria

5. **Futuro:** foram planejadas para o futuro do sistema;
 - a) **Heurísticas:** visam aprimorar a alocação de turmas através de heurísticas;
 - b) **Conflitos:** visam aprimorar a visualização, qualidade e/ou variedade de conflitos;
 - c) **Integração com o Sistema Acadêmico UENF:** visam a integração do atual sistema com o sistema acadêmico da UENF.
 - d) **Arrasta e solta:** visam aprimorar a alocação de turmas através de um sistema de arrastar e soltar;
 - e) **Demandas:** visam calcular a demanda dos alunos por disciplinas;

No Projects, também foram adicionadas as **etiquetas** que distinguem as tarefas por seu intuito.

1. **Nova função:** funcionalidades que ainda não foram implementadas;
2. **Qualidade de vida:** melhorias que não são necessárias, mas que aprimoram o processo de desenvolvimento;
3. **UX:** aprimoramento da experiência do usuário;
4. **Bug:** correção de bugs;
5. **Pergunta:** dúvidas sobre a validade da tarefa;
6. **Abandonada:** quando criada parecia interessante, mas que se decidiu por não implementar;
7. **Duplicata:** já foi criada antes e que foi descartada;

Figura 5.20 – Marcos abertos do GitHub Projects

Nome do Projeto	Última Atualização	Descrição	Progresso	Ações
Futuro	3 dias atrás	Tarefas idealizadas para que sejam feitas em algum momento futuro e...(more)	0% completo	Edit Close Delete
Futuro: heurísticas	4 dias atrás	Tarefas direcionadas para a implementação de heurísticas de criação e aprimoramento da grade	0% completo	Edit Close Delete
Futuro: conflitos	4 dias atrás	Tarefas relacionadas a novos códigos relacionados ao cálculo de conflitos	0% completo	Edit Close Delete
Future: Sistema Acadêmico UENF	9 dias atrás	Ideialmente, os dados dessa aplicação deveriam estar baseados nas tu...(more)	0% completo	Edit Close Delete
Futuro: arrasta e solta	16 dias atrás	Conceito inicial do sistema que permitiria ao usuário usar o sistema de arrastar e soltar para alterar os horários das turmas.	0% completo	Edit Close Delete
Futuro: demanda	16 dias atrás	Tarefas direcionadas para a implementação do cálculo de demandas dos alunos	0% completo	Edit Close Delete
MVP 4	cerca de 1 mês atrás	Restante das tarefas não feitas no MVP 3 e aprimoramentos futuros	0% completo	Edit Close Delete

Fonte: autoria própria

Gráficos

O GitHub Projects também oferece a possibilidade de visualização de gráficos na seção *Insights*, que mostram a quantidade de tarefas em cada uma das categorias. Esses gráficos são úteis para a visualização do andamento do projeto, e para a identificação de possíveis gargalos.

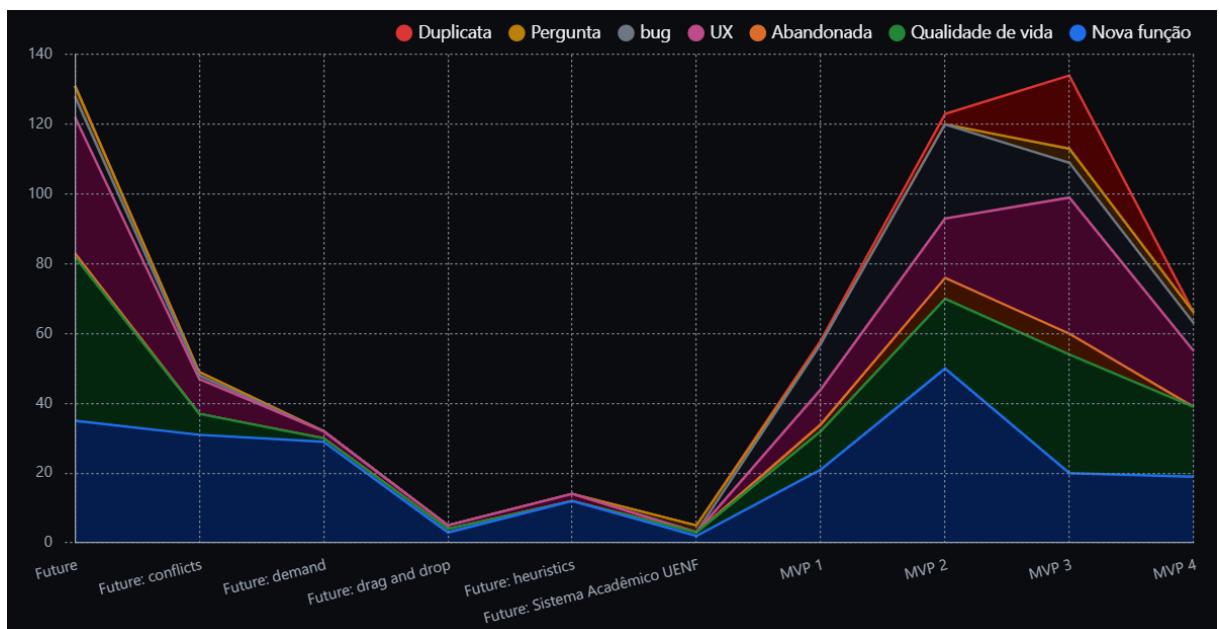
Com eles, pode-se ter uma noção das métricas do projeto, como por exemplo:

- Quantidade de tarefas de determinada etiqueta a cada marco, exemplificado na [Figura 5.22](#);
- Quantidade de tarefas por marco;

Figura 5.21 – Etiquetas do GitHub Projects

Labels		Milestones	Search all labels	New label
7 labels				
Nova função	Nova função do sistema a ser adicionada	131	<button>Edit</button> <button>Delete</button>	
Qualidade de vida	Voltado a boas práticas de programação, reutilização de código, eficiência e escalabilidade	76	<button>Edit</button> <button>Delete</button>	
UX	Foca em aprimorar a parte visual e a experiência do usuário no geral	70	<button>Edit</button> <button>Delete</button>	
bug	Algo não está funcionando	14	<button>Edit</button> <button>Delete</button>	
Pergunta	Alguma dúvida	9	<button>Edit</button> <button>Delete</button>	
Abandonada	Tarefa que não é mais tão necessária de se tocar em frente	1	<button>Edit</button> <button>Delete</button>	
Duplicata	Já criada antes		<button>Edit</button> <button>Delete</button>	

Fonte: autoria própria

Figura 5.22 – Gráfico de Marco *versus* quantidade de tarefas separadas por etiqueta

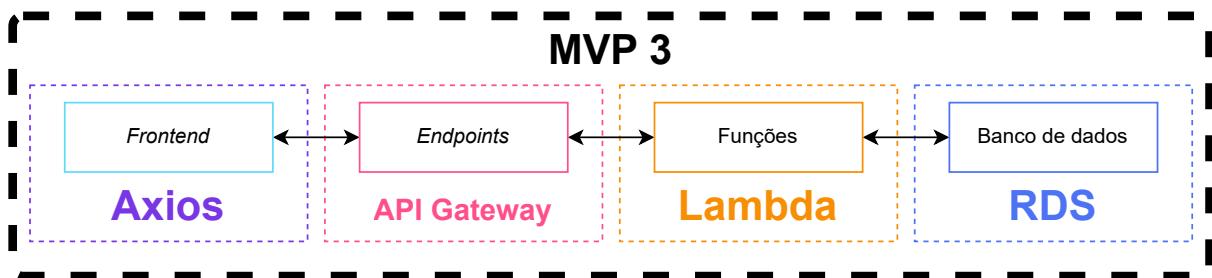
Fonte: autoria própria

- Quais páginas receberam tarefas de quais etiquetas;
- Quantas são as tarefas em cada um de seus estados (**completa**, **em progresso** ou **pendente**).

Amazon Web Services

Para suprir a necessidade de um servidor que pudesse executar o *backend* do sistema em conjunto com o banco de dados, foi escolhido a *Amazon Web Services* (AWS). A AWS é um serviço de computação em nuvem que oferece uma ampla gama de serviços, entretanto, apenas alguns deles foram necessários para o sistema.

Figura 5.23 – Diagrama da progressão funcionamento da permanência dos dados



Fonte: autoria própria

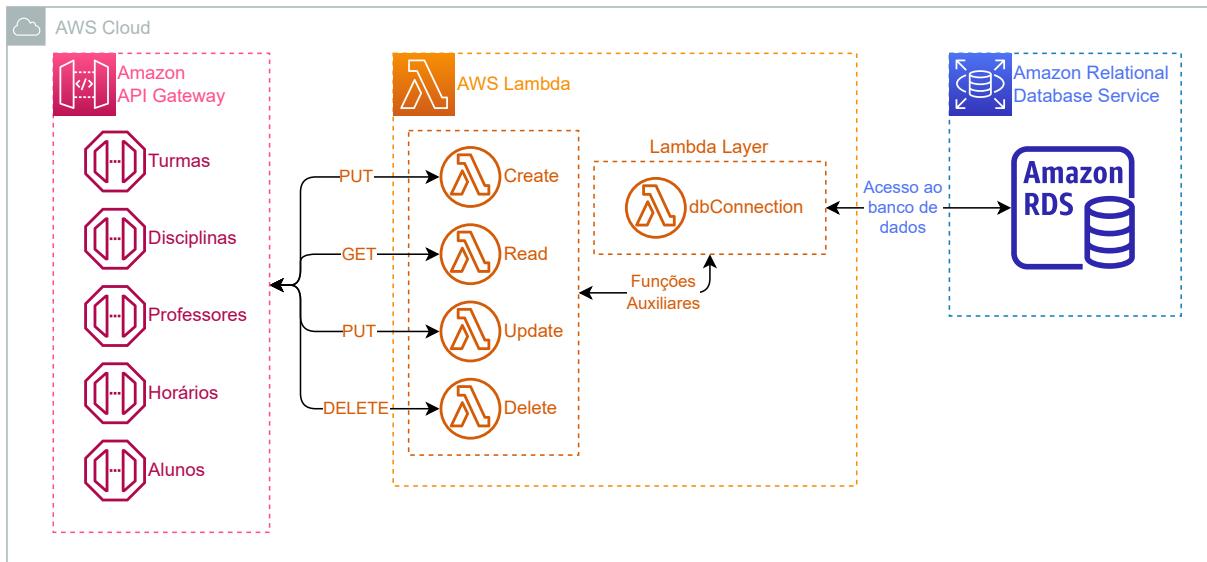
O uso da AWS segue a mesma lógica do servidor local, com a diferença de que o servidor está em nuvem, e não localmente, assim resolvendo o primeiro dos dois problemas citados. Neste contexto o uso da AWS, representado pela [Figura 5.24](#), foi feito através de três serviços principais: o *API Gateway* para a recepção das requisições HTTP, o *Lambda* para a execução das funções que acessam o banco de dados, e o *RDS* para o armazenamento dos dados; serviços estes que serão descritos mais detalhadamente a seguir.

O uso desses três serviços permitiu a execução do *backend* do sistema em nuvem, e assim, atingindo a permanência dos dados. Com isso, o sistema passou a ser capaz de manter os dados mesmo após a recarga da página, e assim, atender a uma das principais necessidades do sistema.

Implantação

O conjunto de funcionalidades da AWS envolve em grande parte o objetivo de manter um sistema constantemente acessível através da internet, ainda assim, durante o desenvolvimento, ou até mesmo durante o ciclo de vida do software, é esperado que ocorram manutenções periódicas nas quais é compreensível que o sistema fique fora do ar. Sendo assim, para manter-se visando ao máximo a acessibilidade do sistema, espera-se que o mesmo fique desconectado o mínimo possível.

Figura 5.24 – API REST no AWS



Fonte: autoria própria

Tanto o API Gateway quanto as funções Lambda precisaram sofrer diversas modificações ao longo do desenvolvimento. A aplicação dessas modificações é chamada de implantação (*deploy*), onde a AWS substitui a versão atual do sistema pela nova versão. Essa aplicação de modificações foi inicialmente feita através da interface *web* da AWS, porém, com o tempo, foi percebido que essa abordagem era ineficiente. No caso das implantações do API Gateway, visto que assim que as rotas estiverem configuradas, não há a necessidade de alterá-las, não havia grande impacto no fluxo de trabalho. Já no caso das funções Lambda, cada mínima mudança no código requisitava um novo *deploy* para cada uma das funções alteradas.

Outro detalhe percebido, foi que boa parte do código se repetia entre as funções que interagiam diretamente com o banco de dados. Sendo assim, passou-se a utilizar das *Lambda Layers*, que são camadas que podem ser compartilhadas entre diversas funções, e assim, diminuir a quantidade de código repetido. Essa nova abordagem permitiu que as funções fossem mais enxutas, e que as mudanças fossem aplicadas de forma mais rápida, visto que bibliotecas e funções comuns entre as funções eram compartilhadas entre elas.

AWS CLI

Nas primeiras tentativas de *deploy*, duas abordagens eram utilizadas: a primeira era a de copiar e colar o código diretamente na interface *web* da AWS, e a segunda era a de fazer o upload de um arquivo zip contendo o código.

Código 5.2 – Código de *deploy* de Lambda

```
1 aws lambda create-function \
```

```
2 --function-name createProfessor \
3 --runtime nodejs20.x \
4 --role arn:aws:iam::375423677214:role/LambdaRole \
5 --handler index.handler \
6 --zip-file fileb://Files/AWS/lambdas/createProfessor/createProfessor.zip
```

Como mostrado no [Código 5.2](#), usa-se o software AWS CLI para criar uma função lambda. O comando *create-function* é o comando que cria a função, e os argumentos que seguem são os parâmetros necessários para a criação da função. O *function-name* é o nome da função, o *runtime* é a versão do Node.js que a função utiliza, o *role* é o conjunto de permissões criadas na seção *AWS Identity and Access Management* (IAM), o *handler* é o nome do arquivo principal que contém a função, e o *zip-file* é o arquivo zip que contém o código da função.

Com isso, ao executar o comando, a função é criada, e então, a nova versão do código é aplicada. Com este fluxo de trabalho, embora permita o *deploy* sem a direta conexão ao sistema AWS, ainda assim é necessário executar comandos específicos para cada uma, tendo que manualmente compactar o código e fazer o upload de cada um dos arquivos das diversas funções coexistentes.

SAM

Como solução, foi utilizado o AWS SAM (*Serverless Application Model*), que é uma extensão do *AWS CloudFormation* que simplifica o desenvolvimento de aplicações sem servidor. O AWS SAM permite a definição de aplicações sem servidor de forma mais simples, e a partir disso, é possível fazer o *deploy* de toda a aplicação de uma vez só. O uso do AWS SAM foi feito através de um arquivo *template.yaml*, que contém a definição de todas as funções Lambda, e de recursos necessários para o funcionamento do sistema.

Como mostrado no [Código D.1](#) disposto no [Apêndice D](#), o arquivo *template.yaml* contém a definição de algumas das estruturas utilizadas para este sistema, principalmente as funções Lambda, visto que são quatro funções para cada uma das seis entidades, totalizando 24 funções. O arquivo contém a definição de cada uma das funções, e de cada uma das rotas que elas atendem. A partir disso, é possível fazer o *deploy* de todas as funções que foram alteradas de uma vez só, e assim, diminuir o tempo de *deploy* e a quantidade de comandos necessários.

Após o preparativo do arquivo *template.yaml*, o *deploy* é feito através do conjunto de comandos *sam build; sam deploy*, que primeiro combina o *CloudFormation Template* com o código da aplicação, e em seguida realiza o *deploy* de todas as funções definidas no arquivo. Com isso, o sistema passou a ser mais facilmente atualizável, e mais facilmente mantido.

Outros serviços da AWS

Os serviços listados até então foram os principais utilizados para a execução do *backend* do sistema, entretanto, diversos outros serviços foram utilizados para a manutenção do sistema como um todo e para a execução de tarefas secundárias. Abaixo estão listados alguns desses serviços.

- **S3:** serviço de armazenamento de objetos, utilizado para armazenar os objetos gerados pelo CloudFormation;
- **CloudWatch e CloudTrail:** serviço de monitoramento, utilizado para a visualização de métricas do sistema;
- **IAM:** serviço de gerenciamento de permissões, utilizado para a definição de permissões para as funções Lambda;
- **VPC:** serviço de rede privada virtual, utilizado para a definição de uma rede privada para o sistema;
- **Cost Explorer:** serviço de visualização de custos, utilizado para a visualização dos custos do sistema;

Melhorias no sistema

O sistema passou por diversas pequenas mudanças, e algumas maiores. Uma considerável parte delas foi relacionada à forma com que as informações eram estruturadas internamente, mudanças essas feitas com o intuito de tornar o sistema mais fácil de ser mantido posteriormente. Em seguida estão listadas algumas das várias melhorias feitas no sistema.

Filtros e ordenações

Levando em consideração a multidimensionalidade da estrutura dos dados, a possibilidade de realizar “curvas de nível” e as ordenar por diferentes critérios se mostrou uma funcionalidade essencial para a compreensão dos dados. Dessa forma, em diferentes páginas do sistema, foram adicionadas ordenações padrões e seletores de filtragem manual, assim permitindo ao usuário a visualização de dados específicos.

Alguns exemplos de casos de uso dessas funcionalidades seria: “Na página **Multi-Turmas** o Professor A está tendo conflitos em suas turmas, então o usuário pode filtrar as turmas do Professor A e visualizar apenas as suas turmas. Em seguida, o usuário percebe que o Professor A está em conflito com a Sala B123, então o usuário visualiza apenas

as turmas da Sala B123. Por fim, o usuário encontra um outro horário disponível para o Professor A e a Sala B123, e então, o conflito é resolvido.”

Uma das solicitações presentes ao final da versão anterior foi quanto a uma distinção mais clara entre disciplinas de Ciência da Computação e disciplinas de outros cursos. Para isso, embora não pareça apresentar grande robustez na forma como foi feito, apresenta suficiente clareza para o usuário final. A distinção foi feita ao utilizar o campo **Período Esperado** presente na entidade **Disciplina**, para definir que:

- **1 ≤ Período Esperado ≤ 10:** Disciplina obrigatória de Ciência da Computação;
- **Período Esperado = 11:** Disciplina eletiva optativa para Ciência da Computação;
- **Período Esperado = 12:** Disciplina eletiva livre para Ciência da Computação;
- **Período Esperado = 13:** Disciplina não ofertada para Ciência da Computação;

Com essa divisão, definiu-se o sistema para que, por padrão, apenas exibisse as turmas voltadas para Ciência da Computação, e que, caso o usuário desejasse, poderia visualizar as turmas de outros cursos.

MultiTurmas

Dentre as tarefas realizadas, uma das páginas que mais sofreu alterações foi a página de **MultiTurmas**. Nela, foram feitas diversas melhorias, como a adição de filtros, ordenações e aprimoramento dos textos contidos nas caixas de seleção.

Adição da propriedade **Descrição**

Disciplinas oferecidas em um mesmo semestre por vezes são oferecidas para alunos demais para que uma única turma os comporte, e assim, é necessário a criação de mais de uma turma. Para que os alunos e professores possam identificar facilmente a qual turma pertencem, foi adicionada a propriedade descrição, que é uma breve descrição da turma. Esse código descritor já se encontra no Sistema Acadêmico, porém com a limitação de apenas 3 caracteres. No presente trabalho, a descrição pode conter até 255 caracteres.

Adicionando esse campo, a visualização linear das informações da turma se tornou mais difícil de apresentar na tela. Considerando que em sua maioria as turmas possuem dois horários, dispõe-se então as informações em conjuntos de dois elementos, ao invés de uma lista única, tornando então a visualização mais densa.

Aprimoramento dos identificadores dos conflitos

Os conflitos ocorridos indicavam quais eram os identificadores (ids) das turmas que estavam em conflito, porém, esses ids eram referentes ao banco de dados, sendo ele um valor numérico, não continha valor semântico suficiente para ser facilmente identificado. Estes ids eram visualizados ao posicionar o ponteiro do mouse por sobre os componentes cujo conflito foi verificado. Com isso, foi feita a adição de um novo identificador, que é composto pelas informações contidas na turma, sendo elas o ano, semestre, nome da disciplina, nome do professor, e o código descritor da turma. A partir disso, tornou-se mais fácil identificar quais turmas estavam em conflito, e assim, corrigi-las. Essa identificação foi adicionada também aos horários, onde o identificador passou a ser composto pela sala, dia da semana, horário de início e fim.

Criação e deleção de turmas e horários

Embora seja uma funcionalidade básica e existente desde a primeira versão, a criação e deleção de turmas apresentou diversos problemas ao longo do desenvolvimento. Um dos mais cruciais era devido à assincronicidade intrínseca ao uso de um banco de dados remoto. O problema era que durante a criação sequencial de duas turmas, apenas a segunda era mostrada, mesmo que ambas tivessem sido criadas.

O que ocorria era que, ao começar com a lista de $Turmas = [A, B]$ tenta-se adicionar a turma C à lista de $Turmas$, mas para isso, a requisição enviada ao banco de dados deve retornar com o status de sucesso, e para que, só assim, fosse adicionada à listagem apresentada no sistema. Então, caso fosse feita a tentativa de se adicionar a turma D antes da confirmação anterior ser recebida, a adição seria realizada novamente na listagem inicial ($[A, B]$). Por fim, assim que a primeira requisição retornasse bem sucedida, por um breve instante a listagem seria $[A, B, C]$, e então, após a adição da turma D , a listagem seria $[A, B, D]$.

Para resolver esse problema, foi feita a adição de uma função de *callback* que passou a utilizar o estado mais atual da listagem de turmas, e não mais a listagem inicial.

Outra característica aprimorada, foi a velocidade de adição e deleção, principalmente a de deleção. Antes, a aprovação do banco de dados era necessária para que a lista de turmas fosse atualizada, e isso tornava o processo de deleção lento. Para resolver isso, foi feita a adição de uma função de deleção que remove a turma da listagem de turmas antes mesmo da confirmação do banco de dados. Essa não se mostra como a solução mais adequada, visto que em caso de falha na deleção, a turma poderá ser restaurada com a simples atualização da página, os pontos positivos na usabilidade superam os negativos.

Aprimoramento das disciplinas não oferecidas

A funcionalidade anteriormente denominada “Disciplinas ainda não oferecidas” foi aprimorada de tal forma que agora, além de criar uma turma para a disciplina selecionada, o sistema automaticamente analisa o histórico de criação de turmas, definindo previamente o professor, a demanda estimada, e os horários, incluindo seus dias, horas de início e sala em que é alocada. Então, após a criação de todas as turmas referentes ao curso de Ciência da Computação, uma solução inicial foi obtida. Sendo ela [mais detalhadamente descrita posteriormente](#).

Grade horária

A página que detinha o nome “CCTable” e que visava apresentar exclusivamente as disciplinas do curso de Ciência da Computação, foi renomeada para “Grade Horária”, e passou a ser possível de apresentar disciplinas de todos os cursos, embora ainda não seja possível distinguir as disciplinas de um curso para o outro.

As células das turmas sofreram um ligeiro aprimoramento visual e foram ordenadas primariamente por seu período esperado.

Banco de dados

Quanto ao banco de dados, os dados antes desconexos passaram a ser interligados adequadamente por chaves estrangeiras. Apresentando então restrições em casos de deleções inappropriadas no banco de dados. A API, por sua vez, deixou de retornar as listas de turmas, professores, disciplinas e salas, e passou a retornar os dados de forma mais estruturada em formato JSON.

Na [Figura 5.25](#) vemos as inter-relações entre as propriedades das entidades do banco de dados. Com o uso mais apropriado das chaves estrangeiras, o anterior uso dos nomes das disciplinas e professores como elementos de identificação foi substituído pelo uso de seus respectivos códigos identificadores. Houve também o surgimento da entidade **Horarios**, que interliga as turmas às salas.

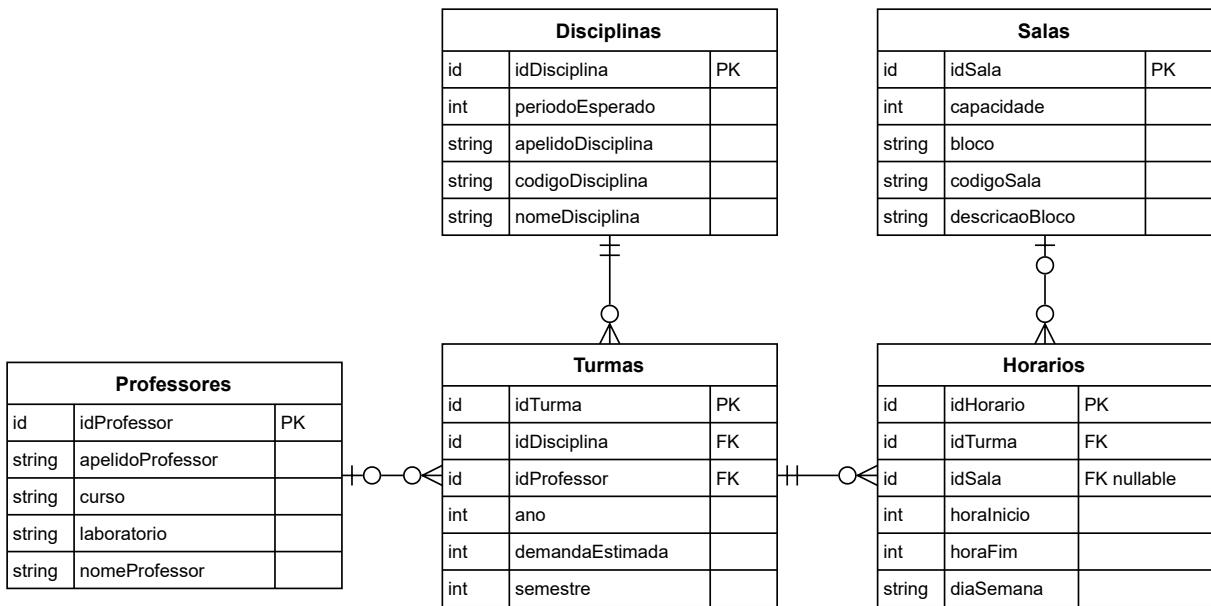
Gerais

Neste tópico estão descritas algumas melhorias gerais do sistema que não se enquadram nos tópicos anteriores.

Aprimoramento na forma de criação de itens

Antes, ao acessar a página de criação de entidades, uma entidade era previamente selecionada. E para a criação de uma nova entidade, os valores da entidade anterior

Figura 5.25 – Novo diagrama de banco de dados



Fonte: autoria própria

deveriam ser alterados, e, ao clicar em adicionar, esses valores alterados eram cadastrados no banco de dados. Essa sequência de ações apresentou intuitividade suficiente e, portanto, foi alterada.

A versão atual passou a não selecionar previamente a entidade. Assim, ao clicar em adicionar um novo item, uma nova entidade é criada. No caso das turmas, a entidade é criada com os valores de ano e semestre predefinidos baseado na filtragem selecionada.

Boas práticas

Além das funcionalidades voltadas para o usuário final, algumas mudanças classificadas como **Qualidade de vida** foram realizadas visando a manutenção do sistema. Dentre elas estão:

- **Repadronização de componentes**: como o conhecimento relacionado a boas práticas de programação foi adquirido ao longo do desenvolvimento, partes de componentes criados anteriormente foram reformulados para que estivessem estruturados de acordo com a estrutura recentemente desenvolvida no código.
- **Externalização de informações**: algumas informações constantes como cores de fundo e textos foram externalizadas para variáveis, assim, caso haja a necessidade de alteração, não será necessário a busca manual por todas as ocorrências. Um outro caso desses é referente aos textos dispostos nas caixas de seleção, que foram convertidos em funções que retornam o texto desejado.

- **Inglês:** como as linguagens de programação de modo geral se apresentam no idioma inglês, é considerado uma boa prática que as variáveis e funções também, o que não foi a abordagem inicialmente tomada. Assim, foi feita uma gradual migração para o inglês, e embora não tenha sido concluída, a maior parte do código já se encontra em inglês. Para facilitar essa migração, foi elaborado um sistema de “*getters*” como forma de obter a propriedade de determinado objeto, independente de qual língua ele esteja.
- **Remoção de estruturas obsoletas:** ao longo do desenvolvimento, algumas estruturas foram criadas e não utilizadas. Um exemplo dessas foi a propriedade “Ordem” que visava definir a ordem em que os horários da turma apareceriam, porém o resultado desta propriedade foi obtido com a ordenação dos horários por dia e em seguida por horário, não sendo então necessária. Ela então foi removida do sistema e do banco de dados.

5.5 Solução inicial

A funcionalidade anteriormente denominada “Disciplinas ainda não oferecidas”, que consistia na exibição de uma lista de disciplinas do curso de Ciência da Computação que espera-se que sejam oferecidas naquele semestre, foi aprimorada de tal forma que agora, além de criar uma turma para a disciplina selecionada, o sistema automaticamente analisa o histórico de criação de turmas, definindo previamente o professor, a demanda estimada, e os horários, incluindo seus dias, horas de início e sala em que é alocada.

O botão de adição de todas as disciplinas foi mantido, e agora, ao ser clicado, cria as turmas com suas características já preenchidas, não considerando, entretanto, a descrição, visto que esta é uma característica única de cada turma e deve ser adicionada manualmente em caso de necessidade. Então, após a criação de todas as turmas referentes ao curso de Ciência da Computação, uma solução inicial foi obtida.

5.5.1 Heurística de solução inicial

A solução inicial foi obtida através de uma heurística que analisa o histórico de criação de turmas, e a partir disso, define as características das turmas a serem criadas. Ela ocorre assim que o botão de adição de turmas a partir das disciplinas ainda não ofertadas é clicado.

Primariamente, a lista completa de turmas é filtrada para que **apenas as turmas daquela mesma disciplina sejam consideradas**. Em seguida, a heurística analisa o histórico de criação de turmas, e a partir disso, define as características das turmas a serem criadas.

Seu algoritmo é ilustrado no [Código 5.3](#) e consiste em 4 funções principais: *splitClasses*, *getMostFrequentClassTimeSizes*, *getMeanDemand* e *getMostFrequent*.

```

1 function getUsualInfo(classes) {
2   const classTimes = splitClasses(classes);
3   const quantity = getMostFrequentClassTimeSizes(classes);
4   const classUsualInfo = {
5     professor: getMostFrequent(classes, ["professor"]),
6     expectedDemand: getMeanDemand(classes),
7     // description: getDescription(classes, currentSemester),
8     description: null,
9     classTime: {
10       quantity,
11       day: getMostFrequent(classTimes, ["dia"], quantity),
12       room: getMostFrequent(classTimes, ["sala"]),
13       duration: getMostFrequent(classTimes, ["duracao"]),
14       startHour: getMostFrequent(classTimes, ["horaInicio"]),
15     },
16   };
17   return classUsualInfo;
18 }
```

Código 5.3 – Heurística de solução inicial

Na função *splitClasses* uma lista de objetos “turma” contendo múltiplos horários é passada por parâmetro e é dividido em uma lista de objetos “turma”, onde cada objeto contém as informações de um dos horários, então retornando uma lista de objetos “horário”. Essa função realiza esta conversão para que possa-se trabalhar com os dados referentes a cada horário de forma individual.

Na função *getMostFrequentClassTimeSizes* é passada uma lista de objetos “turma” contendo múltiplos horários e é retornado um número inteiro que representa a quantidade de horários que foi mais frequente dentre as turmas passadas por parâmetro. Essa função é utilizada para definir a quantidade de horários que a nova turma terá.

Na função *getMeanDemand* é passada uma lista de objetos “turma” e é retornado um número inteiro que representa a média da demanda de todas as turmas que possuam demanda estimada.

Na função *getMostFrequent* são passados três parâmetros usualmente: uma lista de objetos “turma” ou uma lista de objetos “horário”, uma lista de strings que representam o campo que se deseja analisar, e um número inteiro que representa a quantidade de valores que se deseja retornar, sendo este valor igual a 1 caso nenhum valor seja passado. A função então usa a lista de objetos passada por parâmetro para contar a frequência de cada valor registrado no campo desejado, e então, retorna os valores mais frequentes.

Com essas quatro funções a turma de cada disciplina ao ser criada é preenchida com as informações de professor, demanda estimada, quantidade de horários, dia, sala, duração e hora de início.

Como a heurística é baseada no histórico de criação de turmas, ela é capaz de prever com certa precisão as características das turmas a serem criadas. É esperado, entretanto, que esta solução apresente problemas em sua execução, visto que nem todas as alocações de turmas apresentam um padrão objetivo. Além disso, a heurística não checa se a turma a ser criada está alocada em um horário já ocupado pelo professor ou sala em questão.

As limitações da heurística encarregam então ao usuário a verificação e a correção dos possíveis conflitos gerados durante a criação da solução inicial. Essa correção se dá através da análise do sistema de [detecção e alerta dos conflitos](#).

5.6 Detecção e alerta dos conflitos

Uma das principais funcionalidades do sistema é a detecção de conflitos. Seu objetivo é auxiliar ao usuário a identificar possíveis problemas na alocação das turmas, e assim, permitir que ele possa corrigi-los antes de finalizar a grade horária. Diversas situações podem ser consideradas como conflitos, e cada uma delas é tratada de forma diferente.

Os conflitos aqui se colocam como uma forma de alerta ao usuário, e não como uma restrição, assim viabilizando ao usuário que uma ação seja tomada, ou não, a partir do alerta. O conceito da não restrição é importante, visto que embora idealmente espera-se que o processo de alocação disponha de todas as informações para que seja otimamente alocado, na prática, isso atualmente não se mostra uma realidade.

Essa flexibilização das restrições que poderiam ser tidas como rígidas em um problema de otimização, é uma característica do problema de alocação de turmas da UENF. Como na realidade da instituição as grades precisam ser criadas enquanto ainda se tem informações incompletas, certas decisões precisam ser tomadas manualmente, sendo então necessária esta flexibilidade para permitir que o usuário possa tomar essas decisões.

Além disso, diversos casos atípicos acabam por ocorrer na realidade da universidade, e que, embora possam não ser aconselháveis ou até mesmo tidos como conflituosos pelo sistema, não seriam de fato um problema para a execução prática das alocações.

5.6.1 Típicos conflitos atípicos

Para ilustração, abaixo estão descritos alguns exemplos de conflitos que poderiam ser alertadas pelo sistema, mas que não seriam realmente um restritor para a execução prática das alocações:

Considerando o corpo docente do curso de Ciência da Computação, que atualmente conta com seis professores doutores, é recorrente a solicitação de professores bolsistas para ministrar disciplinas. Devido aos prazos existentes ao longo do processo de criação da grade horária, é comum que ainda não se saibam quais e quantos professores bolsistas serão disponibilizados para quais turmas. Porém, como o Sistema Acadêmico requere a inserção de professores para a criação de turmas, uma solução encontrada foi a inserção de um desses professores permanentes como responsável pela turma. E, mesmo após se obter a informação quanto a quais e quantos bolsistas estarão disponíveis, ainda assim o sistema acadêmico não os permite serem inseridos, visto que eles não têm um vínculo permanente com a instituição. Com isso, seria possível ver, por exemplo, um conflito entre duas turmas que possuem o mesmo professor em um mesmo horário, mas que na prática, uma delas será ministrada por um professor bolsista.

Outras situações que podem ocorrer giram em torno da alocação das salas. Duas situações que podem ilustrar sua atipicidade são: a possibilidade de alocar uma turma a uma determinada sala, mesmo que se tenha a intenção de ministrá-la em outra, e também a possibilidade de se repartir a turma em duas salas de aula ocorrendo simultaneamente.

Esses e outros são exemplos de situações recorrentes ao longo do processo flexível da organização da tabela horária.

5.6.2 Conflitos tratados pelo sistema

Para a implementação, primeiro visou-se a detecção de conflitos que poderiam ser considerados restritores para a alocação das turmas. Sendo eles os de alocação simultânea de salas e professores, visto que um professor não pode ministrar duas turmas simultaneamente, nem uma sala deve comportar duas turmas simultaneamente (embora ambos sejam teoricamente possíveis).

Além disso, também foi implementada a detecção de conflitos de capacidade, onde a quantidade de alunos de uma turma é maior do que a capacidade da sala alocada, e alguns outros indicativos visuais que serão descritos abaixo.

Os conflitos calculados são representados de três formas diferentes. A primeira e mais perceptível é a mudança de cor de fundo das propriedades conflituosas. A segunda, visando evitar sobreposição de conflitos, é a adição de uma borda inferior que se estende por toda a largura da propriedade. E a terceira, mais descriptiva, é o uso do atributo *title* dos elementos HTML, que exibe uma mensagem de alerta flutuante ao passar o mouse sobre a propriedade conflituosa, assim dispondo de mais detalhes sobre os conflitos buscados e encontrados.

Embora o sistema seja projetado para ser permissivo quanto a inexistência de certas informações, é sempre esperado que a maior quantidade de informações possíveis seja

inserida, assim, caso algum campo não tenha sido preenchido a cor de fundo do elemento será alterada para um tom acinzentado.

Figura 5.26 – Paleta de cores do sistema

Período da disciplina	Semestre			Entidade	Conflito		
	Verão	Correto	Errado		Nulo	Sem	Com
1	#FFFF00	#00FF00	#FF0000	Disciplina	#E0E0E0		
2	#EFEF00	#00EF00	#EF0000	Professor	#D0D0D0	#47D902	#C72508
3	#DFDF00	#00DF00	#DF0000	Demanda	#C0C0C0	#3EC200	#DD3333
4	#CFCF00	#00CF00	#CF0000	Sala	#B0B0B0	#38B000	#E3580E
5	#BFBF00	#00BF00	#BF0000	Dia	#A0A0A0	#008000	
6	#AFAF00	#00AF00	#AF0000	Hora	#909090	#007200	
7	#9F9F00	#009F00	#9F0000	Duração	#808080	#006400	
8	#8F8F00	#008F00	#8F0000				
9	#7F7F00	#007F00	#7F0000				
10	#6F6F00	#006F00	#6F0000				

Categoria	Disciplina
Não Computação	#D66615
Eletiva Optativa	#0055FF
Eletiva Livre	#4800FF
Sem Categoria	#EEC0C0

Fonte: autoria própria

Os conflitos que são representados por cores, têm sua paleta de cores representada na [Figura 5.26](#). Nessa paleta, dispõe-se 3 conjuntos principais: a distribuição de cores para as disciplinas obrigatórias do curso de Ciência da Computação, que variam de acordo com o semestre em que são ofertadas; a categoria das disciplinas não obrigatórias para o curso de Ciência da Computação; e os conflitos das outras entidades, que são classificados amplamente entre “com conflito”, “sem conflito” e “conflito nulo”.

Cinco campos se encontram sem cores, sendo eles a **Disciplina sem conflito**, **Disciplina com conflito**, **Dia com conflito**, **Hora com conflito** e **Duração com conflito**. No caso dos dois primeiros, melhor explicado [a seguir](#), a representação dos conflitos é substituída pela representação de seu período esperado e de suas categorias. Já os três últimos também não possuem conflitos por si só, o que ocorre é que herdam a cor de fundo das entidades que têm conflito em determinado dia, hora e duração, como é o caso da alocação múltipla de turmas em uma mesma sala ou de professores em turmas simultâneas.

Figura 5.27 – Exemplo de conflito nulo

Fonte: autoria própria

A [Figura 5.27](#) ilustra os **conflitos nulos**. Esse tipo de conflito representa a incompletude de informações, e como durante parte do processo de criação da grade não se tem todas as informações sobre as alocações das turmas, ele acaba por ser um dos conflitos mais comuns. Esse conflito, representado pela cor acinzentada, é detectado quando um dos campos não se encontra preenchido.

Conflitos de professores

O sistema contempla a checagem de conflitos de alocação simultânea de professores em mais de uma turma. Ou seja, considerando todas as turmas ao qual o professor está atribuído no ano e semestre selecionados, o sistema compara todos os horários das turmas deste professor, e verifica se há alguma interseção entre horários que estão no mesmo dia, levando em conta a duração da aula.

[Figura 5.28 – Exemplo de conflito de alocação de professor](#)

Sala	Horário	Duração
Segunda	8	1 hora
Quarta	8	2 horas
Sexta	8	3 horas

Sala	Horário	Duração
Segunda	9	2 horas
Quarta	9	2 horas
Sexta	9	2 horas

Fonte: autoria própria

Caso haja algum conflito, o sistema destaca o professor em questão, tornando a sua cor de fundo avermelhada. Além disso, ao passar o mouse sobre o nome do professor, é exibido um alerta flutuante, informando que quais são as turmas e horários que estão em conflito. Esse comportamento é exemplificado na [Figura 5.28](#), onde o professor Tang é alocado em duas turmas que ocorrem simultaneamente durante algum intervalo de tempo durante os horários de quarta e sexta-feira, assim informando no alerta flutuante quais são as turmas e horários que estão alocados simultaneamente.

Conflitos de salas

As salas também apresentam a verificação do conflito de alocação simultânea. Porém, diferente dos professores, a checagem é feita conferindo todos os horários na qual a sala está alocada, e então é feita a mesma verificação de interseção citada anteriormente. Havendo o conflito, é exibida uma borda alaranjada na parte inferior das propriedades

referentes ao conflito, além de, assim como no caso dos professores, exibir o alerta flutuante. A [Figura 5.29](#) representa um caso de conflito de alocação de salas.

Figura 5.29 – Exemplo de conflito de alocação de sala

	Sala	Dia	Hora de início	Duração
	(24) P5 - inf1	Segunda	10	1 hora
	(24) P5 - inf1	Quarta	10	2 horas
	(24) P5 - inf1	Segunda	11	2 horas
	(24) P5 - inf1	Quarta	11	2 horas
	(24) P5 - inf1	Quarta	10	2 horas
	(24) P5 - inf1	Sexta	10	2 horas

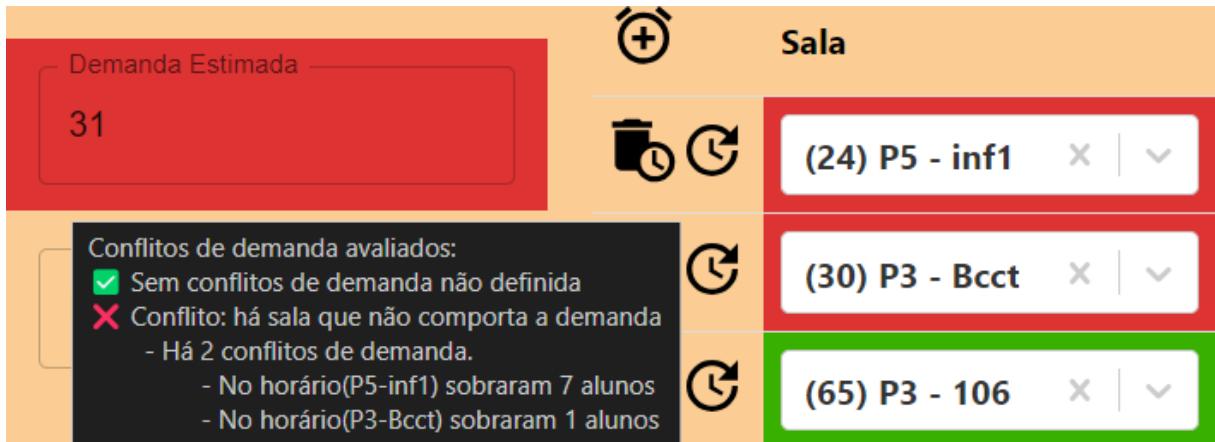
Conflitos de alocação de sala avaliados:
 Sem conflitos de sala não definida
 Conflito: Alocação simultânea de Salas
 - Sala sobreposta com 2 turmas
 - Turma: (2025.1, Prog 1, Annabell); horários: [(P5-inf1, QUA, 10~12)]
 - Turma: (2025.1, ED1, Tang); horários: [(P5-inf1, QUA, 10~12)]
 Sem conflitos de demanda de sala

Fonte: autoria própria

Além disso, também é feita a comparação entre a quantidade máxima de alunos comportados na sala e a quantidade de alunos estimados para a turma. Este conflito por sua vez é ilustrado tornando avermelhado o fundo da demanda estimada e da seleção de salas. Caso uma turma tenha mais de um horário, é calculada a quantidade remanescente dos alunos que demandam a disciplina com relação a cada uma das capacidades das salas destes horários, mostrando cada um deles no alerta flutuante.

Então, como pode-se perceber na [Figura 5.30](#), a turma ilustrada apresenta demanda estimada de 31 alunos não poderia ser adequadamente alocada às salas “P5 - inf1”, nem na “P3 - Bcct”, visto que a primeira apenas comporta 24 alunos e a segunda, 30 alunos. O alerta flutuante, ao ser acionado, informa ainda quantos são os alunos que não poderiam ser alocados em cada uma das salas.

Figura 5.30 – Exemplo de conflito de capacidade na sala



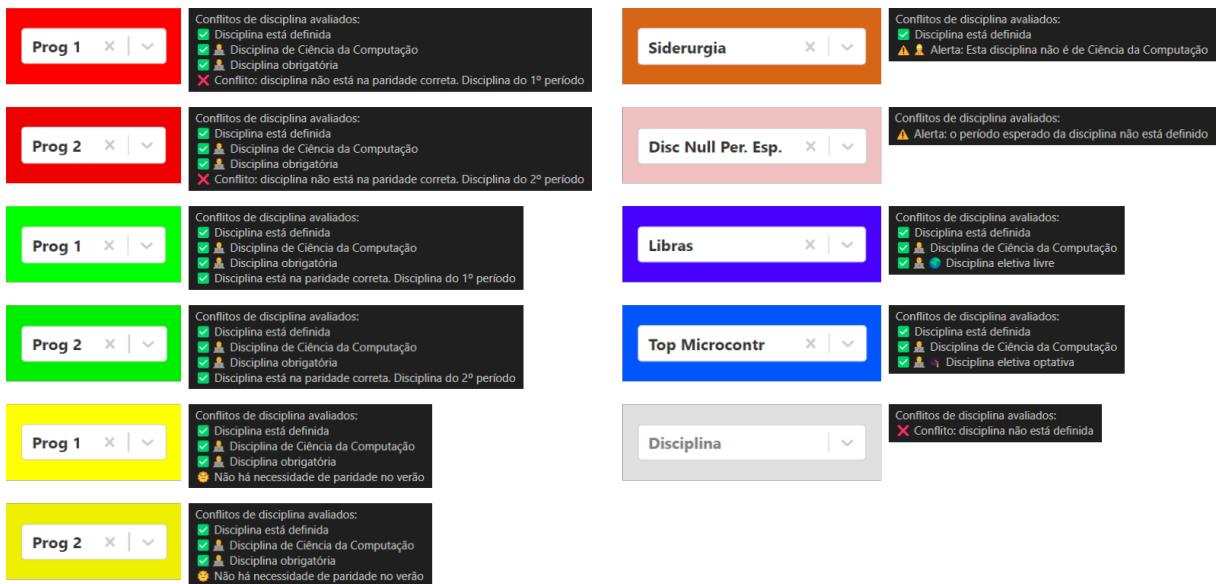
Fonte: autoria própria

Conflitos de disciplina

Além desses conflitos, outras características analisadas e representadas se referem às disciplinas atribuídas às turmas, que, embora não representem necessariamente um *conflito*, mas sim um indicativo, ainda assim serão tratados como conflitos por motivos de simplificação. Esse indicativo leva em consideração o semestre selecionado e o período esperado da disciplina de certa turma. Utilizando de lógica similar, também é indicado caso não tenha sido atribuído um período à disciplina, e se, para o curso de Ciência da Computação, a disciplina é considerada como **Eletiva Livre**, **Eletiva Optativa**, ambas em tons azulados, ou se não é uma disciplina para o curso de Ciência da Computação, sendo então representada em tons alaranjados. Estas características são ilustradas no lado direito da [Figura 5.31](#).

Já no lado esquerdo da [Figura 5.31](#), vemos os conflitos que correlacionam os períodos esperados das disciplinas obrigatórias do curso de Ciência da Computação com o semestre em que foram ofertadas. Os semestres possíveis são três: o primeiro semestre, o segundo semestre e o “período de verão”. No caso do período de verão, as disciplinas que têm o seu período esperado neste semestre são marcadas com um tom amarelado, visto que não há relevância da sua paridade em um período de férias. Já nos casos das disciplinas de paridade ímpar (disciplinas dos períodos 1, 3, 5, 7 e 9) no primeiro semestre, ou as disciplinas de paridade par (disciplinas dos períodos 2, 4, 6, 8 e 10) no segundo semestre, estas são marcadas com um tom esverdeado, sendo aquelas referentes aos períodos finais do curso marcadas com um tom mais escuro. Já as disciplinas pares em semestres ímpares, ou as disciplinas ímpares em semestre pares, são ilustradas com a cor avermelhada, seguindo a mesma lógica de gradiente escuro nos últimos períodos.

Figura 5.31 – Avisos flutuantes dos conflitos de disciplinas



Fonte: autoria própria

5.7 Preenchimento de dados

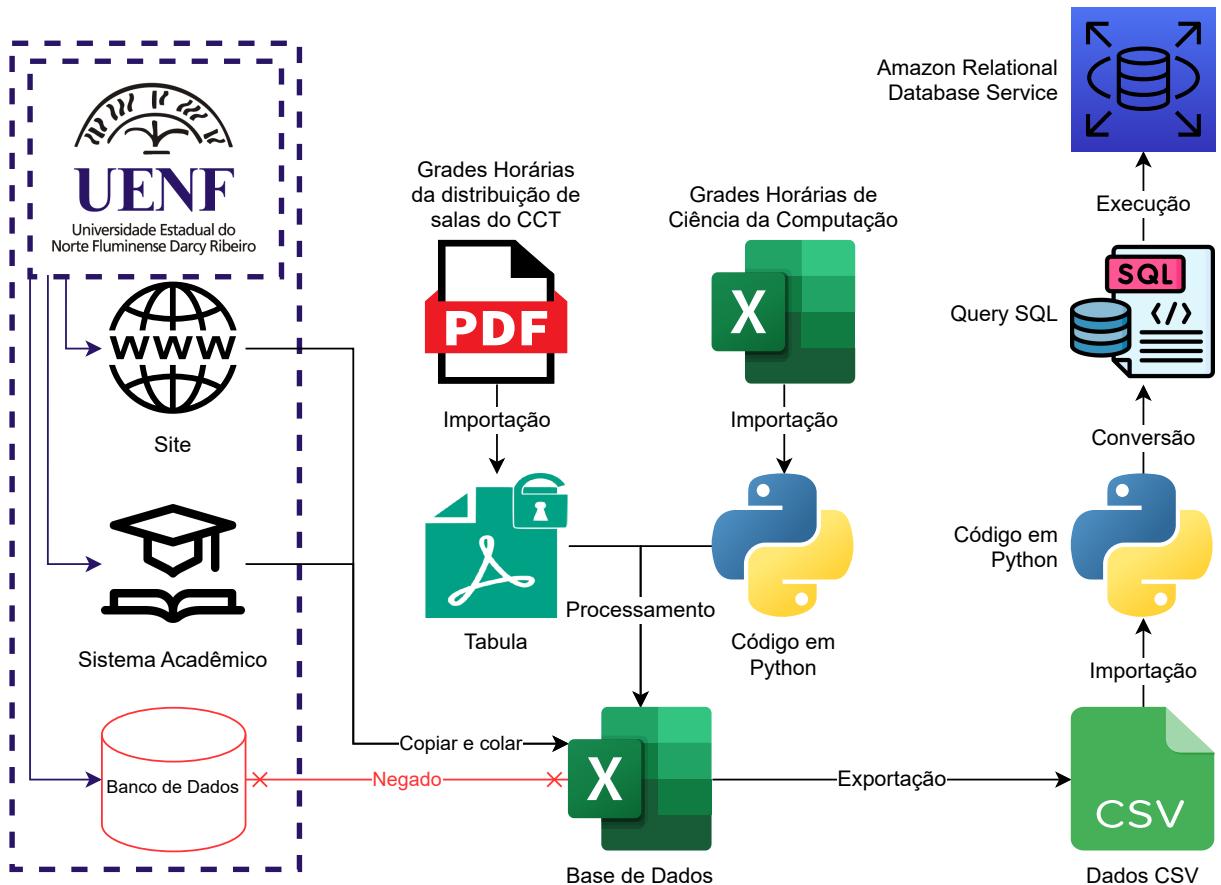
Inicialmente, os dados adicionados faziam jus diretamente às disciplinas, professores, salas e turmas do curso de Ciência da Computação. Porém, como para a análise completa dos conflitos é necessário que seja feita também a adição das turmas de outros cursos, foi feito o preenchimento de dados para as entidades de professores, disciplinas e salas.

Para acumular mais dados referentes às entidades do banco de dados, foram tomadas algumas abordagens: requisição dos dados diretamente do Sistema Acadêmico, processamento de tabelas, processamento de PDFs e *web scraping*, todos eles ilustrados pela Figura 5.32.

A forma teoricamente mais direta e eficiente para se obter os dados das entidades é a obtenção das informações contidas no banco de dados do sistema acadêmico. Para este fim, foi feita uma solicitação ao responsável pela Secretaria Acadêmica (SECACAD) da UENF, que direcionou a solicitação ao desenvolvedor do Sistema Acadêmico. A resposta obtida do desenvolvedor foi que a solicitação não poderia ser atendida, visto que não detinha a posse dos dados, e que para que pudesse fornecê-los, seria necessária uma solicitação formal à reitoria da UENF. Essa solicitação foi então passada à Coordenação do curso de Ciência da Computação, com o qual ficou decidido abandonar a ideia e buscar outras formas de obtenção dos dados.

Paralelamente à abordagem anterior foram feitas tentativas individuais de obtenção dos dados. O processamento de tabelas e PDFs e o *web scraping* foram as abordagens utilizadas. Inicialmente os dados foram coletados e armazenados em tabelas, e então, convertidos para o formato CSV. Os arquivos CSV, por sua vez, foram utilizados em

Figura 5.32 – Diagrama do fluxo de obtenção de dados



Fonte: autoria própria

scripts Python que os convertiam em *queries* SQL, para que assim então fossem adicionados ao banco de dados.

Os PDFs analisados dispunham de tabelas referentes à oferta de turmas para o CCT, mas os dados advindos do processamento de PDF não são tão estruturados quanto os de uma tabela Excel, por este motivo, a primeira abordagem foi a solicitação dos arquivos tabulares para aquele que os produziu. Não havendo resposta favorável quanto a isso, diversos *softwares* de conversão de tabelas em PDF para Excel foram testados, porém, nenhum deles foi capaz de converter as tabelas de forma satisfatória. Um dos agravantes é a existência de células mescladas, o que torna mais complicada a conversão direta. Outra abordagem testada foi a de importação direta dos PDFs através do Excel e também o simples copiar e colar. Nenhum desses métodos foi eficiente, então com isso alguns dados foram coletados, mas sem certeza quanto à sua precisão. Deste método foram coletadas as informações referentes à nomes de professores e disciplinas, capacidades das salas, e horários de aulas.

Além dos PDFs anteriores, haviam também os arquivos referentes à oferta de turmas para o curso de Ciência da Computação. Estes sim dispunham também de sua

versão em Excel, e assim, foram processados utilizando *scripts* Python. A abordagem apresentou falhas, visto que a notação das informações não apresentava o mesmo padrão ao longo dos anos, então foi necessário fazer ajustes manuais, resultando em uma tabela normalizada com diversas pastas de trabalhos referentes a cada um dos semestres desde 2019, não considerando os semestres de verão. Deste método foram coletadas as informações referentes à nomes e apelidos de professores e disciplinas, demandas estimadas dos alunos pela turma, descrição da turma, e horários de aulas.

Por fim, foi feita o *web scraping* que consistiu na busca por informações em diversos sites, principalmente o Sistema Acadêmico¹⁷, o site da UENF¹⁸ e outros sites. Nessa etapa, foi possível encontrar lotes de informações estruturadas. Um dos lotes foi a listagem de disciplinas e suas características que se encontram disponíveis no Sistema Acadêmico da UENF, essas informações foram copiadas e coladas no arquivo Excel unificado. Outros lotes foram encontrados dispersos ao longo do site da UENF e consistiam basicamente em listagem de professores e seus respectivos laboratórios. Essas informações estavam dispersas dos sites dos diversos centros e cursos, alguns disponíveis no próprio site, outros em formato de arquivo. Deste método foram coletadas as informações referentes à nomes de professores, seus laboratórios e centros; e disciplinas e seus nomes, códigos e períodos de vigência. Além disso, foram encontrados documentos oficiais que referenciam a capacidade de ocupantes das “salas” disponíveis do Centro de Convenções, popularmente conhecido como “Apitão” que mesmo não sendo propriamente uma sala de aula, já foi utilizado previamente para tal fim. Outras salas já obtiveram alocações similares, assim como a Sala dos Professores.

5.8 Próximos desenvolvimentos

Ao longo de todo o desenvolvimento, podemos dividir as funcionalidades em quatro categorias: as que foram **implementadas e se mantiveram**, as que foram **implementadas e descartadas**, as que **não foram implementadas** e as que foram **preparadas** para próximos desenvolvimentos.

As funcionalidades que foram implementadas e se mantiveram são aquelas principais e essenciais para o funcionamento do sistema, como a adição de turmas, professores, disciplinas e salas, a visualização da grade horária, a detecção de conflitos.

As funcionalidades que foram implementadas e descartadas são aquelas que por diversos motivos, não se mantiveram no sistema. Um exemplo disso é a funcionalidade de travamento de turmas, que foi implementada, mas descartada por não ser considerada essencial.

¹⁷ <https://academico.uenf.br>

¹⁸ <https://uenf.br/portal>

As funcionalidades que não foram implementadas são aquelas que, devido a priorização das tarefas, não detiveram importância o bastante para que chegassem à etapa de implementação. Um exemplo disso é a funcionalidade de alterar a grade horária através do sistema de arrastar e soltar.

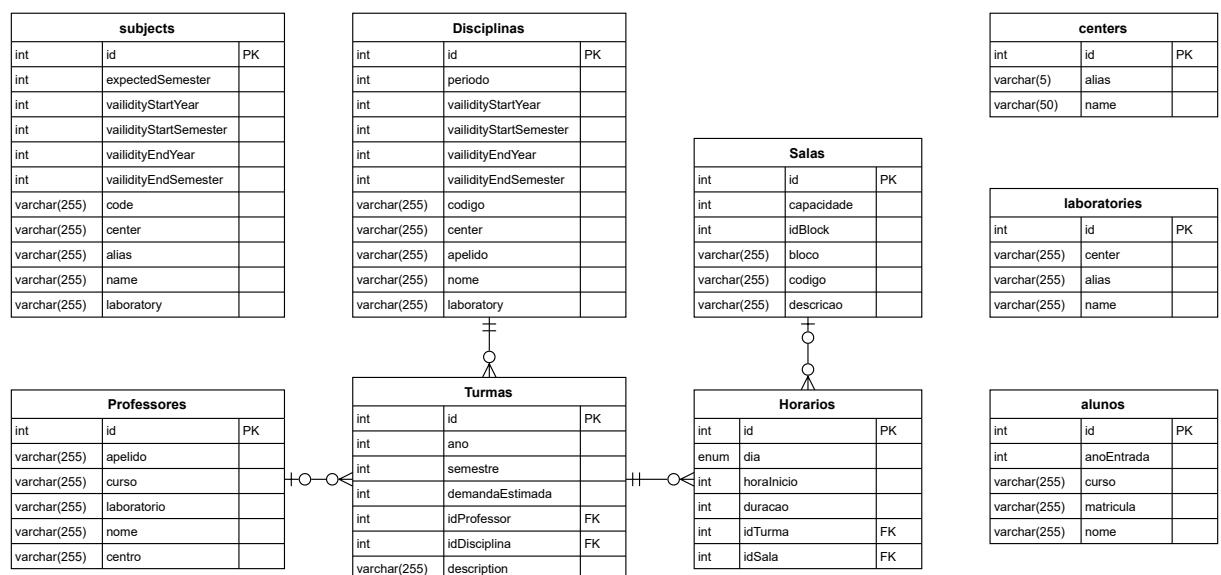
As funcionalidades que foram preparadas mas não implementadas são aquelas que, por conveniência ou por praticidade do momento, foram parcialmente desenvolvidas, mesmo que não tenham grande participação no sistema final. Dentre elas, podemos citar a funcionalidade de adição de alunos, e diversas tabelas e colunas no banco de dados que foram criadas e preenchidas mas não utilizadas.

5.8.1 Funcionalidades preparadas

Dentre as funcionalidades preparadas temos o sistema de cadastro de alunos, que consiste basicamente na adição das informações básicas dos alunos, como ano de entrada, curso, matrícula e nome como é visto em sua tabela no canto inferior direito na [Figura 5.33](#). Seus dados podem ser armazenados no banco de dados junto com os outros dados das entidades que são mais diretamente relacionadas à grade horária, como as turmas, professores, professores, disciplinas e salas.

Além dos alunos, outras entidades também foram estruturadas, sendo elas os centros e laboratórios. Isso se deu com o objetivo de estruturar mais objetivamente as informações referentes aos professores e disciplinas, visto que os professores são vinculados a um centro e a um laboratório, assim tendo a tendência a reduzir a necessidade de repetição de informações.

Figura 5.33 – Diagrama entidade relacionamento final



Fonte: autoria própria

Nota-se que na [Figura 5.33](#), alguns atributos encontram-se em inglês e outros em português. Isso se dá pelo fato de que a transição para o inglês foi feita de forma gradual, e embora a maior parte do sistema já esteja em inglês, ainda há partes que se encontram em português. A decisão pela migração para o inglês se deu pelo fato de que na programação de modo geral, as variáveis e funções são escritas em inglês, e embora não seja uma regra, torna o código mais consistente em sua linguagem.

A interface para definição da preferência dos professores também se encontra pronta, e consiste em uma tabela onde cada linha representa um horário e cada coluna representa um dia da semana. Nela, o professor pode marcar os horários em que não pode ministrar aulas, e também indicar os horários em que prefere ministrar aulas. Essa funcionalidade pode vir a ser retomada posteriormente por auxiliar na tomada de decisão voltada a alocação dos professores bolsistas, visto que geralmente são os que menor se sabe sobre a disponibilidade, então, tendo a possibilidade de marcar os horários em que não podem ministrar aulas, pode-se evitar a alocação de turmas nesses horários.

Outro sistema também preparado envolve a visualização da progressão de disciplinas dos alunos que foi dividida basicamente em três categorias: **aprovadas**, **cursando** e **reprovadas**. A visualização era feita em uma caixa de seleção de múltipla escolha, onde as opções eram todas as disciplinas disponíveis. Com esse registro, torna-se possível ter uma estimativa calculada da demanda de alunos para as turmas, pois considera-se a quantidade de alunos que ainda não a concluíram, seja por reprovação ou por escolherem não se inscrever. Tendo também o histórico de reprovações médias, pode-se calcular mais precisamente a demanda de alunos para as turmas.

Cada uma dessas funcionalidades foi parcialmente implementada, podendo ser então retomadas e finalizadas em futuros desenvolvimentos.

6 RESULTADOS

Neste capítulos serão apresentados os resultados obtidos com a implementação do sistema de gerenciamento de suporte à decisão discutido ao longo deste trabalho. Serão apresentadas as páginas desenvolvidas, bem como as funcionalidades implementadas.

6.1 Alternativas burocráticas

Além da busca pela solução ótima, o presente trabalho também se propõe a buscar métodos ainda mais alternativos para se amenizar a problemática abordada. Sendo, de forma simples, o uso de meios burocráticos disponíveis na instituição que abre alguns caminhos para uma maior praticidade no processo de resolução do problema. Entretanto, é necessário que se tenha em mente que a burocacia é um processo lento e que pode ser desgastante, sendo até mesmo esperado que não seja desejado por parte dos construtores da grade horária.

Essas alternativas não geram por si só uma solução para o problema, em termos metafóricos, se o sistema é a engrenagem que faz a máquina funcionar, as alternativas burocráticas são o óleo que pode fazer a engrenagem funcionar de forma mais suave, mesmo que não seja estritamente necessário.

6.1.1 Tempo de elaboração das grades

Durante as entrevistas do Capítulo 3 da seção 3.2, uma alternativa válida para a amenização da problemática abordada é a alteração do calendário anual da UENF que define férias de duas semanas entre os semestres. Caso seu calendário seja alterado para que as férias sejam de três semanas, o problema de agendamento teria maior tempo para ser resolvido, assim fazendo com que a solução ótima seja provável de ser alcançada.

Segundo o Artigo 28 do (UENF, 2002, Estatuto da UENF), compete à secretaria acadêmica a elaboração da proposta de calendário escolar para que seja aprovado pelo Colegiado Acadêmico. Enquanto que o Artigo 63 da seção 2 do capítulo 1, informa que os calendários do curso de graduação devem ser aprovados pelas correspondentes câmaras, com observância do calendário da universidade.

Logo, quanto à alteração do calendário acadêmico, a alteração mostra-se como possível, sendo necessário apenas que o processo burocrático necessário seja enfrentado, o que pode acabar não sendo do desejo daqueles que constroem a grade horária.

6.1.2 Alteração forçada de horários

Segundo o parágrafo primeiro do artigo 36 das Normas de Graduação, “qualquer alteração de horário/turno após o período de matrícula deverá ter a anuência por escrito de todos os discentes matriculados na turma”. Seguindo ao segundo parágrafo do mesmo artigo, temos que “a alteração de horário das aulas da turma deverá ter a anuência da Coordenação de Curso e a ciência do Chefe do Laboratório responsável pela disciplina”.

Outra alternativa que aproveita de uma brecha nas normas supracitadas é a possibilidade de se criar novas turmas para as disciplinas que possuem horários conflitantes, assim direcionando os alunos para que se desinscrevam da turma anterior.

Ambas as alternativas supracitadas visam a alteração dos horários das turmas mesmo após o estimado período de construção da grade horária, assim efetivamente aumentando novamente o tempo disponível para a resolução do problema.

6.1.3 Aplicação prática dos métodos burocráticos

Consideremos o ano de 2023, os seus semestres e seus respectivos calendário acadêmicos ([UENF; SECACAD; Reitoria, 2023a](#); [UENF; SECACAD; Reitoria, 2023b](#)) para a graduação, onde a [Tabela 6.1](#) mostra o calendário do primeiro semestre e a [Tabela 6.2](#) mostra o calendário do segundo semestre.

Tabela 6.1 – Calendário Acadêmico da SECACAD de 2023.1 (simplificado)

Atividades	Data
Prazo limite de cad. de nov. discip. a serem ofer. em 2023.1	até 20/01
Prazo limite para criação de turmas a serem oferecidas em 2023.1	20/01 a 15/02
Renovação de matrícula de 2023.1	28/02 a 03/03
Início do período letivo de 2023.1	06/03
Inclusão e exclusão de disciplinas	06/03 à 20/03
Encerramento do período letivo de 2023.1	07/07
Prazo limite: Entrega dos resultados à SECACAD	14/07

Tabela 6.2 – Calendário Acadêmico da SECACAD de 2023.2 (simplificado)

Atividades	Data
Prazo limite de cad. de nov. discip. a serem ofer. em 2023.2	até 14/07
Prazo limite para criação de turmas a serem oferecidas em 2023.2	17 a 28/07
Renovação de matrícula de 2023.2	01/08 a 04/08
Início do período letivo de 2023.2	07/08
Inclusão e exclusão de disciplinas	14 a 21/08
Encerramento do período letivo de 2023.2	08/12
Prazo limite: Entrega dos resultados à SECACAD	15/12

Com objetivo de tornar a visualização das informações mais coesa, a [Tabela 6.3](#) mostra um calendário acadêmico normalizado, onde os prazos, períodos e marcações de início e fim foram convertidos em termos de “início” e “fim”, assim distinguindo também o seu período de vigência e a data específica.

Tabela 6.3 – Calendário Acadêmico de 2023 (simplificado)

Vigência	Fase	Atividades	Data
2023.1	Fim	cadastro de novas disciplinas a serem oferecidas	20/01/23
2023.1	Início	criação de turmas a serem oferecidas	20/01/23
2023.1	Fim	criação de turmas a serem oferecidas	15/02/23
2023.1	Início	renovação de matrícula	28/02/23
2023.1	Fim	renovação de matrícula	03/03/23
2023.1	Início	período letivo	06/03/23
2023.1	Início	inclusão e exclusão de disciplinas	06/03/23
2023.1	Fim	inclusão e exclusão de disciplinas	20/03/23
2023.1	Fim	período letivo	07/07/23
2023.1	Fim	entrega dos resultados à SECACAD	14/07/23
2023.2	Fim	cadastro de novas disciplinas a serem oferecidas	14/07/23
2023.2	Início	criação de turmas a serem oferecidas	17/07/23
2023.2	Fim	criação de turmas a serem oferecidas	28/07/23
2023.2	Início	renovação de matrícula	01/08/23
2023.2	Fim	renovação de matrícula	04/08/23
2023.2	Início	período letivo	07/08/23
2023.2	Início	inclusão e exclusão de disciplinas	14/08/23
2023.2	Fim	inclusão e exclusão de disciplinas	21/08/23
2023.2	Fim	período letivo	08/12/23
2023.2	Fim	entrega dos resultados à SECACAD	15/12/23

Levando em consideração as ações sugeridas na [subseção 6.1.1](#) e na [subseção 6.1.2](#), um cronograma alterado seria o disposto na [Tabela 6.4](#), onde as ações sugeridas foram aplicadas. Essas ações incluem a alteração do calendário acadêmico para que todo o primeiro semestre comece suas atividades uma semana antes. Além disso, define-se que o período final de criação de turmas não é efetivamente finalizado no antes da renovação de matrícula, mas sim após a finalização da inclusão e exclusão de disciplinas.

Nota-se que o único caso em que não foi feita a migração exata de uma semana foi na renovação de matrícula, visto que caso fosse alterada para uma semana exata, a renovação de matrícula seria feita no período de carnaval, o que não é desejado. Assim, foi mantido o mesmo intervalo de tempo, porém, começando após o feriado nacional “Quarta-feira de Cinzas”.

Tabela 6.4 – Calendário Acadêmico de 2023 - Alterado

Vigência	Fase	Atividades	Data
2023.1	Fim	cadastro de novas disciplinas	20/01 13/01/23
2023.1	Início	criação de turmas	20/01 13/01/23
2023.1	Fim	criação de turmas	15/02/23
2023.1	Início	renovação de matrícula	28/02 23/02/23
2023.1	Fim	renovação de matrícula	03/03 26/02/23
2023.1	Início	período letivo	06/03 27/02/23
2023.1	Início	inclusão e exclusão de disciplinas	06/03 27/02/23
2023.1	Fim	inclusão e exclusão de disciplinas	20/03 13/03/23
2023.1	Fim	criação de turmas	20/03/23
2023.1	Fim	período letivo	07/07 30/06/23
2023.1	Fim	entrega dos resultados à SECACAD	14/07 07/07/23
2023.2	Fim	cadastro de novas disciplinas	14/07/23
2023.2	Início	criação de turmas	17/07/23
2023.2	Fim	criação de turmas	28/07/23
2023.2	Início	renovação de matrícula	01/08/23
2023.2	Fim	renovação de matrícula	04/08/23
2023.2	Início	período letivo	07/08/23
2023.2	Início	inclusão e exclusão de disciplinas	14/08/23
2023.2	Fim	inclusão e exclusão de disciplinas	21/08/23
2023.2	Fim	criação de turmas	21/08/23
2023.2	Fim	período letivo	08/12/23
2023.2	Fim	entrega dos resultados à SECACAD	15/12/23

Com a utilização desses dois métodos burocráticos, aqueles encarregados de realizar a criação e alocação das turmas nos devido horários disporá de uma semana extra no segmento entre os semestres, assim abrindo uma janela de tempo maior para a manipulação dos horários, e consequentemente, ampliando possibilidade de se alcançar uma solução ótima.

6.2 Sistema desenvolvido

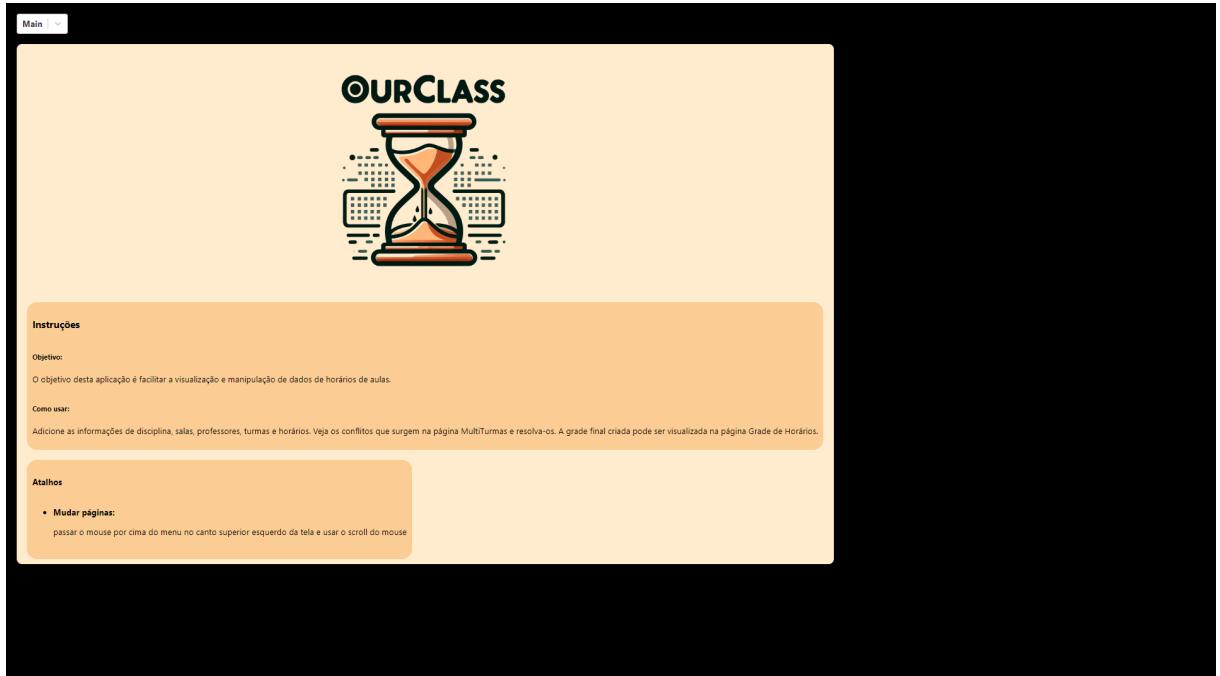
O sistema foi desenvolvido utilizando a linguagem de programação JavaScript em conjunto com o React, um framework de desenvolvimento de interfaces de usuário. O sistema consiste de um conjunto de [oito páginas](#) principais, das quais emergem diversas [funcionalidades](#).

6.2.1 Páginas

A página inicial ([Figura 6.1](#)) apresenta um resumo do objetivo e seu funcionamento básico, informa também sobre atalhos possíveis para uso mais dinâmico do sistema. básica

de dados, se mostra como uma página importante para futuros desenvolvimentos no campo de alocação de alunos em turmas.

Figura 6.1 – Página inicial do sistema



Fonte: autoria própria

A página **MultiTurmas** é repartida em três partes principais: a primeira, com filtros para a seleção das turmas mostradas ([Figura 6.2](#)); a segunda, com a visualização dos conflitos de horários entre as turmas selecionadas ([Figura 6.3](#)); e a terceira, com a visualização das disciplinas pendentes de alocação nas turmas selecionadas ([Figura 6.4](#)).

A página **Grade de Horários** ([Figura 6.5](#)) apresenta a grade de horários de todas as turmas, podendo também haver a filtragem de quais turmas serão mostradas.

A página **Turmas** ([Figura 6.6](#)) apresenta todas as turmas cadastradas, podendo ser feita a edição ou exclusão individual de cada uma delas. Embora ela apresente um funcionamento similar à página MultiTurmas, tem potencial para trabalhar posteriormente com um ajuste fino específico de cada turma.

As páginas **Professores** ([Figura 6.7](#)), **Salas** ([Figura 6.8](#)) e **Disciplinas** ([Figura 6.9](#)) apresentam um funcionamento muito similar entre si. Todas elas permitem as operações de criação, leitura, atualização e exclusão de seus respectivos itens. Além de apresentar também uma tabela inferior onde são listadas os horários históricos em que cada um desses itens foi alocado, podendo haver também a filtragem por ano, semestre, dia e hora.

A página **Alunos** ([Figura 6.10](#)), por fim, apresenta a possibilidade de visualização de todos os alunos cadastrados. Mesmo não possuindo funcionalidades além da manipulação

Figura 6.2 – Página de multiturmas com filtros

The screenshot shows the 'MultiTurmas' application interface. At the top, there are filters for 'Ano: 2024', 'Semestre: 1', 'Categoria: Disciplinas CC', 'Disciplina: Disciplina', 'Professor: Professor', and 'Sala: Sala'. Below the filters, there are four rows of course information. Each row contains a 'Disciplina/Professor' section and a 'Demandas Estimadas/Descrição' section. The 'Disciplina/Professor' section includes fields for 'Disciplina' (Calc 1), 'Professor' (Ausberto), and 'Descrição'. The 'Demandas Estimadas/Descrição' section includes a 'Demandas Estimadas' field (e.g., 35) and a 'Descrição' field. To the right of these sections are four tables, each with columns 'Sala', 'Dia', 'Hora de inicio', and 'Duração'. The first three rows show data for 'Calc 1' with different room assignments (65 P3 - 106, 65 P3 - 106, 35 P3 - 109A). The fourth row shows data for 'Calc 1' with room assignments (Sala, Sala).

Fonte: autoria própria

Figura 6.3 – Página de multiturmas com conflitos

The screenshot shows the 'MultiTurmas' application interface. The layout is similar to Figure 6.2, with a header for filters and four rows of course information. The 'Disciplina/Professor' and 'Demandas Estimadas/Descrição' sections are present. The 'Demandas Estimadas' fields are highlighted in red, indicating conflicts. The 'Descrição' fields are also highlighted in red. To the right of these sections are four tables showing room assignments and times. Conflicts are visible in the second and third rows, where multiple courses are assigned to the same room at the same time.

Fonte: autoria própria

Figura 6.4 – Página de multiturmas com disciplinas pendentes

C C C	C C C	C C C
C C C	C C C	C C C
C C C	C C C	C C C
C C C	C C C	C C C

Fonte: autoria própria

Figura 6.5 – Página de grade de horários

Grade de Horários											
Ano:	2024	Semestre:	1	Categoria:	Disciplinas CC	Disciplina:	Disciplina	Professor:	Professor	Sala:	Sala
		Segunda	Terça	Quarta	Quinta	Sexta					
8	4 - Estatística (Tang / P3-109A)	1 - Calc 1 (Prof. ? / P3-106) 1 - Calc 1 (Prof. ? / P3-109A) 1 - Calc 1 (Prof. ? / Sala ?) 1 - Calc 1 - CC (Elba / P3-108) 3 - Calc 3 (Prof. ? / P3-206) 3 - Calc 3 (Prof. ? / P3-104) 5 - Estocásticos (Guillermo / P5-inf2)	4 - Estatística (Tang / P3-109A)	1 - Intro CC (Rivera / Sala ?)	1 - Calc 1 (Prof. ? / P3-106) 1 - Calc 1 (Prof. ? / P3-109A) 1 - Calc 1 (Prof. ? / Sala ?) 1 - Calc 1 - CC (Elba / P3-108) 3 - Calc 3 (Prof. ? / P3-206) 3 - Calc 3 (Prof. ? / P3-104) 5 - Estocásticos (Guillermo / P5-inf2)	1 - Calc 1 (Prof. ? / P3-106) 1 - Calc 1 (Prof. ? / P3-109A) 1 - Calc 1 (Prof. ? / Sala ?) 1 - Calc 1 - CC (Elba / P3-108)					
10	3 - ED1 (Tang / P3-Bcct) 5 - Redes (João Luiz / P3-206) OPT - Grafos (Luis Mariano / P5-inf1)	1 - FCC (Prof. ? / P3-Bcct) 3 - Parad Ling Prog (Ausberto / P3-206) 4 - IPE Intro Prob Est (Bolsista LCMAT 1 / P3-203A) 4 - IPE Intro Prob Est (Gudelia / P3-106) 5 - Ling Form (Annabell / P3-109B) 7 - Interface HM (Rivera / P5-inf1)	3 - ED1 (Tang / P3-Bcct) 5 - Redes (João Luiz / P3-206) OPT - Grafos (Luis Mariano / P5-inf1)	1 - FCC (Prof. ? / P3-Bcct) 3 - Parad Ling Prog (Ausberto / P3-206) 4 - IPE Intro Prob Est (Bolsista LCMAT 1 / P3-203A) 5 - Ling Form (Annabell / P3-109B) 7 - Interface HM (Rivera / P5-inf1)	1 - FCC (Prof. ? / P3-Bcct) 3 - Parad Ling Prog (Ausberto / P3-206) 4 - IPE Intro Prob Est (Bolsista LCMAT 1 / P3-203A) 5 - Ling Form (Annabell / P3-109B) 7 - Interface HM (Rivera / P5-inf1)	4 - IPE Intro Prob Est (Gudelia / P3-106) 5 - Comp Soc (Prof. ? / P3-208) 9 - Proj Monog (Rivera / P5-inf2)					
12	1 - Lógic Mat (Tang / P3-206) 3 - Fis 2 (Prof. ? / P3-208) 3 - Fis 2 (Max / P3-108)	1 - FCC (Prof. ? / P3-Bcct) 1 - Geom Analit (Elba / P3-108) 3 - Arq Comp (Prof. ? / P3-207)	1 - Logic Mat (Tang / P3-206) 3 - Fis 2 (Prof. ? / P3-208) 3 - Fis 2 - A (Roberto Franco / P3-107)	1 - FCC (Prof. ? / P3-Bcct) 1 - Geom Analit (Elba / P3-108) 3 - Arq Comp (Prof. ? / P3-207)	1 - FCC (Prof. ? / P3-Bcct) 1 - Geom Analit (Elba / P3-108) 3 - Arq Comp (Prof. ? / P3-207)	1 - Inglês 1 (Bolsista CCH 1 / Sala ?) (3h) 10 - Estágio (Tang / P5-inf1) (0h)					
14	3 - Fis 2 - A (Roberto Franco / P3-107) 3 - Met Mat (Prof. ? / P3-207) 7 - Sist Dist (João Luiz / P5-inf1)	5 - Eng Soft (Ausberto / P3-109B) OPT - Fund Proces Img (Rivera / P5-inf1)	5 - Eng Soft (Ausberto / P3-109B) 3 - Fis 2 - A (Roberto Franco / P3-107) 3 - Met Mat (Prof. ? / P3-207) 7 - Sist Dist (João Luiz / P5-inf1) 8 - Emprend (Prof. ? / E1-220)	5 - Eng Soft (Ausberto / E1-206) OPT - Fund Proces Img (Rivera / P5-inf1)	5 - Eng Soft (Ausberto / E1-206) 3 - Met Mat (Prof. ? / P3-207) 7 - Sist Dist (João Luiz / P5-inf1) 8 - Emprend (Prof. ? / E1-220)						
16	1 - Org Comp (Bolsista CC 1 / P3-109A) 2 - Fis 1 (André Oliveira Guimarães / P3-104) 2 - Fis 1 (Denise Ribeiro dos Santos / P3-208) 3 - Fis Exp 2 - A (Delson / P3-203A) 3 - Met Mat (Prof. ? / P3-106) 5 - BD1 (Luis Mariano / P3-Bcct) 7 - Test Soft (Bolsista CC 3 / P5-inf1)	1 - Prog 1 (Annabell / P3-Bcct) 7 - Metod Trab Cien (Molina / P3-109A)	1 - Org Comp (Bolsista CC 1 / P3-109A) 2 - Fis 1 (André Oliveira Guimarães / P3-104) 2 - Fis 1 (Denise Ribeiro dos Santos / P3-208) 3 - Met Mat (Prof. ? / P3-106) 5 - BD1 (Luis Mariano / P3-Bcct) 7 - Test Soft (Bolsista CC 3 / P5-inf1)	1 - Org Comp (Bolsista CC 1 / P3-109A) 2 - Fis 1 (André Oliveira Guimarães / P3-104) 2 - Fis 1 (Denise Ribeiro dos Santos / P3-208) 3 - Met Mat (Prof. ? / P3-106) 5 - BD1 (Luis Mariano / P3-Bcct) 7 - Test Soft (Bolsista CC 3 / P5-inf1)	1 - Prog 1 (Annabell / P3-Bcct) 7 - Metod Trab Cien (Molina / P3-109A)	10 - Monog (Tang / P5-inf1) (0h)					
18	1 - Calc 1 (Prof. ? / P3-106) 3 - Fis Exp 2 - B (Delson / P3-205) 4 - Calc Num (Prof. ? / P3-203A)	1 - Calc 1 (Prof. ? / P3-106) 2 - Álgebra Lin (Oscar / P3-203A) 3 - Calc 3 (Prof. ? / P3-108) 3 - Calc 3 (Prof. ? / P3-207)	4 - Calc Num (Prof. ? / P3-203A)	1 - Calc 1 (Prof. ? / P3-106) 2 - Álgebra Lin (Oscar / P3-203A) 3 - Calc 3 (Prof. ? / P3-108) 3 - Calc 3 (Prof. ? / P3-207)	1 - Calc 1 (Prof. ? / P3-106) 2 - Álgebra Lin (Oscar / P3-203A) 3 - Calc 3 (Prof. ? / P3-108) 3 - Calc 3 (Prof. ? / P3-207)						

Fonte: autoria própria

Figura 6.6 – Página de turmas

INFORMAÇÕES DA TURMA

Ano/Semestre: 2024 | 1

Disciplina: Interface HM

Professor: (LCMAT) Rivera

Demanda Estimada: 10

Descrição: Descrição

ID: 20240121

Horários

Sala	Dia	Hora de início	Duração
(24) P5 - inf1	Terça	10	2 horas
(24) P5 - inf1	Quinta	10	2 horas

Fonte: autoria própria

Figura 6.7 – Página de professores

INFORMAÇÕES DO PROFESSOR

laboratório: (CCT) LCMAT

curso: Ciência da Computação

nome: Fermín Alfredo Tang Montañé

apelido: Tang

ID: 3

Turmas desse professor:

idTurma	Ano.Semestre	Disciplina	Professor	Demanda Estimada	Descrição	idHorario	Sala	Dia	Hora Início	Duração
20292008	2024.1	MAT01201 - Estatística	Tang	43		2028033038	(35) P3 - 109A	SEG	8	2
20240114	2024.1	INFO1114 - ED1	Tang	25		2024011425	(30) P3 - Bct	SEG	10	2
20240101	2024.1	MAT01104 - Lógic Mat	Tang	30		2024010101	(40) P3 - 206	SEG	14	2
20292008	2024.1	MAT01201 - Estatística	Tang	43		2028033039	(35) P3 - 109A	QUA	8	2
20240114	2024.1	INFO1114 - ED1	Tang	25		2024011426	(30) P3 - Bct	QUA	10	2
20240101	2024.1	MAT01104 - Lógic Mat	Tang	30		2024010102	(40) P3 - 206	QUA	14	2
20240128	2024.1	INFO1127 - Estágio	Tang	5		2024012851	(24) P5 - inf1	SEX	14	Indef.
20240129	2024.1	INFO131 - Monog	Tang	4		2024012952	(24) P5 - inf1	SEX	16	Indef.

Fonte: autoria própria

Figura 6.8 – Página de salas

The screenshot shows a software interface for managing rooms. At the top, there's a header with a search bar and a dropdown menu '(24) P5 - inf1'. Below it is a section titled 'INFORMAÇÕES DA SALA' containing fields for 'Bloco' (P5), 'Descrição' (Descrição), 'Código' (inf1), 'Capacidade' (24), and 'ID' (2). A large table below is titled 'Turmas nesta sala: 12' and lists 12 classes scheduled for room P5. The columns include: idTurma, Ano.Semestre, Disciplina, Professor, Demanda Estimada, Descrição, idHorario, Sala, Dia, Hora Inicio, and Duração. The data includes various professors like Luís Mariano, João Luiz, and Bolsista CC, and disciplines like INF0129, INF0122, INF0121, INF0123, INF0120, INF01219, INF0122, INF0121, INF01221, INF0123, INF0120, INF01127, and INF01131.

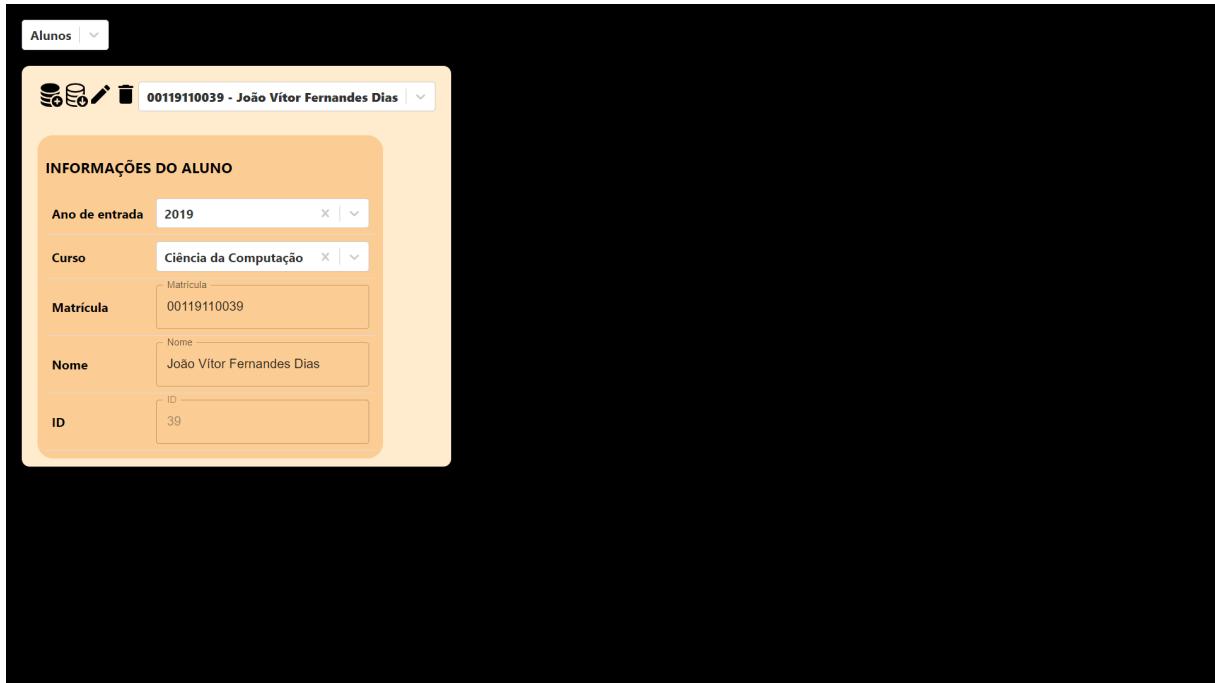
Fonte: autoria própria

Figura 6.9 – Página de disciplinas

The screenshot shows a software interface for managing courses. At the top, there's a header with a search bar and a dropdown menu 'Calc 1'. Below it is a section titled 'INFORMAÇÕES DA DISCIPLINA' containing fields for 'Nome' (Cálculo Diferencial e Integral), 'Apelido' (Calc 1), and 'Período Esperado' (1º Período). A large table below is titled 'Turmas desta disciplina: 15' and lists 15 classes for the 'Calc 1' course. The columns include: idTurma, Ano.Semestre, Disciplina, Professor, Demanda Estimada, Descrição, idHorario, Sala, Dia, Hora Inicio, and Duração. The data includes various professors like Instr. MAT01101, and Instr. MAT01101.

Fonte: autoria própria

Figura 6.10 – Página de alunos



Fonte: autoria própria

6.2.2 Funcionalidades

Nas páginas apresentadas, foram implementadas diversas funcionalidades que permitem a interação do usuário com o sistema. Dentre elas, destacam-se as [quatro operações básicas de armazenamento persistente](#), a [análise histórica](#), a [criação de grade inicial](#), a [Visualização de conflitos](#) e a [visualização das tabelas horárias](#).

CRUD

Com exceção da página inicial e da página de grade de horários, todas as páginas apresentam as operações de criação, leitura, atualização e exclusão de seus respectivos itens. Isso se dá pois esta é a função básica de um sistema de gerenciamento de banco de dados, sendo necessária para que todas as outras funcionalidades possam ser implementadas e utilizáveis.

Análise histórica

O conceito da **Análise Histórica** é a capacidade de visualizar quais foram todos os horários em que um determinado item foi alocado. Isso é útil para avaliar os padrões emergentes nas alocações dos recursos.

Criação de grade inicial

A **Criação de Grade Inicial**, já descritas no [segmento sobre heurística de criação de grade](#), é uma funcionalidade que permite a criação de uma grade de horários inicial através da avaliação das alocações anteriores para determinada disciplina. Através dela é possível criar uma grade de horários inicial para o curso de Ciência da Computação em poucos segundos.

Visualização de conflitos

Na página **Turmas** e principalmente na página **MultiTurmas**, é possível visualizar os conflitos existentes entre as turmas. Os [conflitos](#) são apresentados de forma clara e objetiva, permitindo a rápida identificação e posterior resolução.

Visualização de tabelas horárias

Por fim, especificamente na página **Grade de Horários**, é possível visualizar as tabelas horárias de todas as turmas cadastradas, podendo haver também a filtragem por ano, semestre, categoria, disciplina, professor e sala. Com esta visualização tabular, pode-se gerar as grades de horários específicas das turmas de disciplinas de cada um dos períodos letivos, bem como distingue as turmas entre as que pertencem à ementa do Curso de Ciência da Computação e as que não pertencem.

Assim como no caso da [análise histórica](#), a visualização das tabelas horárias é uma funcionalidade que permite a visualização de padrões emergentes nas alocações dos recursos.

7 CONCLUSÕES

Estando no início do fim, seguiremos agora numa jornada retrospectiva ao que foi abordado no corrente trabalho.

Pudemos ver através da [pesquisa acadêmica](#) aos artigos e trabalhos relacionados com a área da construção de grades horárias em específico, o *university course timetabling*, que a área é vasta, tanto em quantidade de artigos publicados quanto às suas possibilidades de pesquisa e desenvolvimento. Uma das principais questões que mantêm a área em constante inconclusão é o nível de especificidade que cada instituição de ensino possui, o desafio deixa de ser a implementação do método resolutivo, e passa a ser modelar o problema específico e lidar com as preocupações dos usuários. A UENF não é diferente, já tendo sido alvo de pesquisas e desenvolvimentos de sistemas de otimização de grade horária em anos anteriores.

Visando enfrentar diretamente o problema da especificidade e modelagem do problema na UENF [foi feito um estudo sobre como os diversos setores relacionados à construção da grade horária interagem entre si](#), quais são os seus responsáveis e qual é a sequência de ações que cada setor realiza para a construção da grade horária. Essa pesquisa exploratória foi realizada através de buscas por documentos oficiais da UENF e entrevistas com os responsáveis de cada setor.

Com base nos [documentos oficiais da UENF](#), foi possível identificar os setores responsáveis pela construção da grade horária, que são a Secretaria Acadêmica, a Direção de Centro, a Chefia do Laboratório e a Coordenação do Curso. Também relacionado com o processo de oferecimento das turmas está o Sistema Acadêmico da UENF, não sendo ele recorrentemente citado nas documentações, mesmo que no presente momento seja um dos elementos de interligação entre os setores.

Como o sistema pretendido é voltado para se enquadrar no contexto prático e não apenas no teórico visto previamente, viu-se necessário a realização de pesquisas qualitativas em forma de [entrevista](#) com os responsáveis de cada setor para entender suas percepções pessoais à realidade prática recorrente na instituição. Com isto, pode-se obter informações mais detalhadas da realidade prática da UENF, que muitas vezes se encontra além daquilo que é descrito nos documentos oficiais.

Para avaliar a percepção dos alunos quanto à construção da grade horária, foi elaborado um [formulário](#) com perguntas abertas e fechadas, respondido por mais de 200 alunos da UENF. Com isso, constatando considerável insatisfação dos alunos com o processo de construção da grade horária, cabendo assim um aprimoramento no processo.

Uma alternativa encontrada para contornar a dificuldade encontrada pelos trabalhos anteriores na UENF no campo da **modelagem do problema** foi utilizar uma abordagem diferente, onde se busca uma solução boa o suficiente para que seja utilizada na prática, mesmo que não seja ótima, isso através do método de manipulação manual dos dados, assim permitindo a participação ativa dos usuários na construção da grade horária, o que poderia facilitar a aceitação do sistema pela instituição.

Para este fim foi **desenvolvido um sistema de suporte à decisão** para auxiliar os setores da Universidade Estadual do Norte Fluminense Darcy Ribeiro (UENF) responsáveis pela criação de grades horárias. O sistema foi desenvolvido com o intuito de ser utilizado como uma ferramenta auxiliar, onde os usuários possam manipular os dados de forma mais intuitiva e visual, assim reduzindo a necessidade de retrabalho e aumentando a produtividade.

O sistema permite que as **quatro operações básicas de armazenamento persistente**, com isso, os usuários podem adicionar manualmente as informações referentes ao trabalho de criação de grade horária de forma centralizada, assim reduzindo a necessidade de se lidar com diversos arquivos e planilhas. Facilitando também a visualização de informações, como a alocação de turmas, que pode ser **visualizada de forma gráfica**, assim facilitando a **identificação de conflitos**. O que consequentemente tende a agilizar o processo de busca por novas soluções e a redução dos conflitos.

O código-fonte para o sistema desenvolvido está disponível no **apêndice**.

7.1 Trabalhos futuros

Como trabalhos futuros, vê-se uma ampla gama de pesquisa e aprimoramento ao presente trabalho, visto que este busca um método alternativo de solução ao mesmo problema abordado por outros dois pesquisadores em tempos anteriores. Pode-se então elaborar uma conexão entre o atual sistema e modelos aos métodos heurísticos propostos, permitindo então uma abordagem híbrida humano-computador na busca da grade horária ótima. Sugere-se inclusive o estudo sobre a aplicação de métodos de programação inteira, visto que através da revisão bibliográfica este método apresentou consideráveis resultados.

Quanto ao *software*, mesmo que o prioritário seja a sua funcionalidade, é esperado que o seu design seja o mais intuitivo, fluido e prático quanto for possível. Sendo esta tarefa direcionada mais à experiência do usuário, possivelmente tangenciando o problema central de construção de grades horárias. Quanto às tarefas indicadas para este processo, atualmente seu projeto conta com 261 *issues* abertas¹ no projeto² do repositório do *GitHub*³,

¹ <https://github.com/jvfd3/timetabling-UENF/issues>

² <https://github.com/users/jvfd3/projects/3>

³ <https://github.com/jvfd3/timetabling-UENF>

podendo ser um bom ponto de partida para futuros desenvolvedores.

Como um dos principais desafios na área de construção de grade horária é a modelagem do problema, assim como ocorreu com os modelos anteriores, é esperado que este trabalho acabe por trilhar o mesmo caminho, visto que o problema em questão realmente apresenta grande parte de sua complexidade no entendimento e modelagem de como as diversas partes da instituição interagem entre si, porém, espera-se que este documento possa servir como uma boa base para o entendimento de sua estrutura.

Assim como ocorrido anteriormente, é esperado que por diversos motivos este trabalho também não implementados na prática. Tendo isso em vista, se o indesejado ocorrer, espera-se que a principal linha de prosseguimento deste trabalho seja pesquisar, descobrir, analise e corrigir os motivos de falha do uso prático do atual sistema.

Apelo

Eu gostaria de deixar aqui um alerta para quem for utilizar este documento como base para futuros trabalhos: a maior dificuldade a ser superada é o fator organizacional. A minha percepção é de que a UENF atualmente se encontra tal qual um osso quebrado que se regenerou sem o uso de gesso para o fixar no local certo: funciona, mas não tão bem quanto seria capaz. E, assim como no caso ósseo, para que você atinja um resultado ótimo, certamente terá que quebrar algumas estruturas já consolidadas para que possa reorganizá-las de forma mais eficiente.

Neste trabalho tentei pavimentar o caminho na direção que acredito ser a mais adequada para a adoção do sistema. Nesse caminho, acabei abrindo mão de meus desejos pessoais que envolviam o sistema direcionado às demandas dos alunos, visto que, mesmo que atingisse um resultado ótimo aos alunos, nada adiantaria se o sistema não fosse adequado àqueles que o usarão. Eu espero que este trabalho não se torne apenas mais uma monografia que será esquecida em uma prateleira, mas sim que ele possa ser utilizado como um guia para a construção do sistema que um dia sonhei em desenvolver. Por isso se lembre: o teu trabalho é apenas um passo, não o fim. Não queira fazer tudo, foque em fazer o máximo que você puder para que o trabalho atinja seu objetivo e se torne o melhor possível para que seja desenvolvido por outros. Afinal, *everything is a draft* (STARK; PETTIS, 2016).

Se você chegou até aqui, eu agradeço por ter lido este trabalho. E, se você for um estudante da UENF, eu peço que você não desista de lutar por um ensino melhor. A UENF é uma instituição que tem um grande potencial, e eu acredito que ela pode ser muito mais do que é hoje. Eu espero que este trabalho possa ser um pequeno passo na direção de um futuro melhor para a nossa universidade.

Caso o sistema ainda esteja em funcionamento, excelente, isso significa que consegui

atingir um de meus objetivos, então, continue a aprimorá-lo. Caso contrário, torne como seu objetivo consertar os meus erros. Descubra o motivo da não adoção do sistema e corrija-o. E, se possível, me avise, eu adoraria saber que o meu trabalho não foi em vão.

Além do desenvolvimento da monografia como objetivo para a conclusão do curso, o que desejo é conseguir auxiliar as pessoas em suas atividades diárias. Ainda mais se considerarmos que este sistema, se bem executado, tende a ajudar semestralmente centenas, senão milhares, de alunos e professores. Por isso, se não sendo adotado atualmente, espera-se que o sistema possa ser aprimorado e posto em prática no futuro.

REFERÊNCIAS

ALENCAR, W. de S. et al. Visualization methods for educational timetabling problems: A systematic review of literature. In: Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. Prague, Czech Republic: SCITEPRESS - Science and Technology Publications, 2019. p. 275–281. ISBN 978-989-758-354-4. Disponível em: <https://doi.org/10.5220/0007375802750281>. Acesso em: 25/04/2024. Citado 5 vezes nas páginas 18, 19, 20, 21 e 22.

ANDRE, A.; DINATA, H. Interaction design to enhance ux of university timetable plotting system on mobile version. IOP Conference Series: Materials Science and Engineering, IOP Publishing, v. 407, n. 1, p. 8, set. 2018. ISSN 1757-899X. Disponível em: <https://doi.org/10.1088/1757-899X/407/1/012174>. Acesso em: 25/04/2024. Citado 5 vezes nas páginas 16, 22, 35, 42 e 51.

ARRATIA-MARTINEZ, N. M.; PADRÓN, C. M.; TORRES, P. A. Ávila. University course timetabling problem with professor assignment. Mathematical Problems in Engineering, v. 2021, p. 1–9, jan. 2021. ISSN 1563-5147, 1024-123X. Disponível em: <https://doi.org/10.1155/2021/6617177>. Acesso em: 25/04/2024. Citado 3 vezes nas páginas 18, 20 e 21.

BARHAM, A. M.; WESTWOOD, J. B. A simple heuristic to facilitate course timetabling. The Journal of the Operational Research Society, Palgrave Macmillan Journals, v. 29, n. 11, p. 1055–1060, nov. 1978. ISSN 01605682, 14769360. Disponível em: <https://doi.org/10.2307/3009353>. Acesso em: 25/04/2024. Citado na página 14.

HADDAD, N. Horário - Engenharia Civil - 2017-02 e 2018-02. 2018. Página na internet. Disponível em: <https://uenf.br/cct/leciv/graduacao/horario-2017-01>. Acesso em: 07/05/2024. Citado na página 24.

MIRANDA, J.; REY, P. A.; ROBLES, J. M. udpskeduler: A web architecture based decision support system for course and classroom scheduling. Decision Support Systems, v. 52, n. 2, p. 505–513, jan. 2012. ISSN 01679236. Disponível em: <https://doi.org/10.1016/j.dss.2011.10.011>. Acesso em: 25/04/2024. Citado 4 vezes nas páginas 15, 19, 24 e 41.

MURRAY, K.; MÜLLER, T.; RUDOVÁ, H. Modeling and solution of a complex university course timetabling problem. In: BURKE, E. K.; RUDOVÁ, H. (Ed.). Practice and Theory of Automated Timetabling VI. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. v. 3867, p. 189–209. ISBN 978-3-540-77344-3. Series Title: Lecture Notes in Computer Science. Disponível em: https://doi.org/10.1007/978-3-540-77345-0_13. Acesso em: 25/04/2024. Citado na página 22.

PIERRE, B.; FAIRLEY, R. E.; SOCIETY, I. C. SWEBOK: Guide to the Software Engineering Body of Knowledge. Version 3.0. Washington, DC, USA: IEEE Computer Society Press, 2014. OCLC: 880350861. ISBN 978-0769551661. Disponível em: <https://ieeecs-media.computer.org/media/education/swebok/swebok-v3.pdf>. Acesso em: 25/04/2024. Citado na página 30.

- POULSEN, C. J. B.; BANDEIRA, D. L. Desenvolvimento de um modelo para o school timetabling problem baseado na meta-heurística simulated annealing. Simulated Annealing, fev. 2012. Disponível em: <https://lume.ufrgs.br/handle/10183/39522>. Acesso em: 25/04/2024. Citado 2 vezes nas páginas 19 e 20.
- SANTOS, S. C. dos. Heurística para o problema da programação de horário: estudo de caso do curso de ciência da computação da Universidade Estadual do Norte Fluminense. Tese de Bacharelado — Universidade Estadual do Norte Fluminense, Campos dos Goytacazes, RJ, set. 2013. Citado 4 vezes nas páginas 22, 45, 46 e 47.
- SILVEIRA, R. C. da. Métodos heurísticos na resolução do Problema de Programação de Horários. Tese de Bacharelado — Universidade Estadual do Norte Fluminense, Campos dos Goytacazes, RJ, dez. 2014. Citado 5 vezes nas páginas 22, 23, 45, 46 e 47.
- STARK, K.; PETTIS, B. The Cult of Done Manifesto. 2016. Página na internet. Disponível em: <https://medium.com/@bre/the-cult-of-done-manifesto-724ca1c2ff13>. Acesso em: 09/05/2024. Citado na página 106.
- THOMAS, J. J.; KHADER, A. T.; BELATON, B. Visualization techniques on the examination timetabling pre-processing data. In: 2009 Sixth International Conference on Computer Graphics, Imaging and Visualization. Tianjin, China: IEEE, 2009. p. 454–458. ISBN 978-0-7695-3789-4. Disponível em: <https://doi.org/10.1109/CGIV.2009.23>. Acesso em: 25/04/2024. Citado na página 14.
- UENF, U. Estatuto. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2002. 36 p. Disponível em: https://uenf.br/UENF_ARQUIVOS/Downloads/REITORIA_1360_1101117875.pdf. Acesso em: 04/05/2024. Citado 2 vezes nas páginas 27 e 93.
- UENF, U. Projeto Pedagógico do Curso de Ciência da Computação (Modalidade Bacharelado) - 2005. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2005. 37 p. Comissão para Elaboração da Proposta de Criação do Curso de Graduação em Computação e Informática do CCT. Também disponível em: <https://www.academia.edu/2584459>. Disponível em: <https://doi.org/10.13140/RG.2.2.28825.01127>. Acesso em: 04/05/2024. Citado na página 27.
- UENF, U. Regimento Geral da Graduação. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2012. 59 p. Disponível em: <https://uenf.br/reitoria/secacad/files/2012/08/Regimento-atualizado-2016-2.pdf>. Acesso em: 04/05/2024. Citado na página 27.
- UENF, U. Matriz Curricular - Curso de Ciência da Computação. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2015. 1 p. Coordenação de Ciência da Computação. Disponível em: <https://cc.uenf.br/arquivos/2014-gradeCurricularComputacao.pdf>. Acesso em: 04/05/2024. Citado na página 27.
- UENF, U. Projeto Pedagógico do Curso de Ciência da Computação (Modalidade Bacharelado) - 2015. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2015. Direção Administrativa e Acadêmica da UENF/CCT. Citado na página 27.
- UENF, U. Projeto Pedagógico do Curso de Ciência da Computação (Modalidade Bacharelado) - 2022. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2022. 128 p. Direção Administrativa e Acadêmica da UENF/CCT. Disponível em: https://uenf.br/graduacao/wp-content/uploads/2024/01/PPC_Ciencia_da_Computacao.pdf. Acesso em: 04/05/2024. Citado na página 27.

- UENF, U. et al. Regimento da Câmara de Graduação da UENF. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2019. 17 p. Disponível em: <https://uenf.br/graduacao/wp-content/uploads/2019/10/RegimentoGraduacao2019.pdf>. Acesso em: 04/05/2024. Citado na página 27.
- UENF, U.; CONSUNI, C. Regimento Geral da UENF. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2006. 25 p. Disponível em: <https://uenf.br/dga/grh/files/2016/06/Pessoal.Regimento-interno.pdf>. Acesso em: 04/05/2024. Citado na página 27.
- UENF, U.; CONSUNI, C. Plano de Desenvolvimento Institucional 2023-2027. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2023. 144 p. Disponível em: <https://uenf.br/portal/wp-content/uploads/2023/12/Minuta-PDI-2023-2027-FinalRev-08-12-1.pdf>. Acesso em: 04/05/2024. Citado na página 27.
- UENF, U.; PROGRAD, P. Normas da Graduação. Universidade Estadual do Norte Fluminense Darcy Ribeiro, 2019. 107 p. Disponível em: https://uenf.br/graduacao/wp-content/uploads/2019/12/NormasGraduacaoRESCOLAC07_2019.pdf. Acesso em: 04/05/2024. Citado na página 27.
- UENF, U.; SECACAD, S.; Reitoria. Calendário da Graduação - 2023.1. [S.l.], 2023. Disponível em: https://uenf.br/portal/wp-content/uploads/2022/11/Calendario_Academico_Graduacao_2023_11.pdf. Acesso em: 04/05/2024. Citado na página 94.
- UENF, U.; SECACAD, S.; Reitoria. Calendário da Graduação - 2023.2. [S.l.], 2023. Disponível em: <https://uenf.br/portal/wp-content/uploads/2023/12/Novo-Calendario-2023-02.pdf>. Acesso em: 04/05/2024. Citado na página 94.
- WREN, A. Scheduling, timetabling and rostering — a special relationship? In: GOOS, G. et al. (Ed.). Practice and Theory of Automated Timetabling. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, (Lecture Notes in Computer Science, v. 1153). p. 46–75. ISBN 978-3-540-61794-5 978-3-540-70682-3. Series Title: Lecture Notes in Computer Science. Disponível em: https://doi.org/10.1007/3-540-61794-9_51. Acesso em: 25/04/2024. Citado 2 vezes nas páginas 18 e 19.

APÊNDICE A – RESULTADOS DO FORMULÁRIO

Como forma de analisar também a perspectiva dos discentes quanto à problemática abordada, foi elaborado um formulário de pesquisa com o intuito de se confirmar ou não a hipótese de que em sua maioria os alunos também se encontram insatisfeitos com a atual conjuntura de distribuição e alocação de turmas.

Para este fim, foi utilizado um formulário de pesquisa qualitativa dos alunos. O formulário foi divulgado através de um [link¹](#) disponibilizado no grupo de alunos do curso de Ciência da Computação no WhatsApp, e também distribuído pela Secretaria Acadêmica a discentes da UENF por e-mail.

Além das análises realizadas, disponho também como anexo do arquivo deste trabalho o [arquivo com os dados brutos coletados.xlsx](#) que podem ser utilizados para análises futuras.

Respondentes

Inicialmente foram solicitadas algumas informações dos alunos, como seu curso, ano de ingresso ([Tabela A.1](#)).

O formulário foi respondido por 207 alunos, sendo os cursos de maior incidência o bacharelado de Ciências Biológicas com 40 respondentes, Ciência da Computação com 29 e Medicina Veterinária com 19. Vale ressaltar que 96 dos respondentes são alunos de cursos que envolvem diretamente o CCT. Entretanto, a análise dos gráficos abordará a percepção de todos igualmente.

Vemos também a distribuição dos anos de ingresso dos alunos que responderam o formulário ([Tabela A.2](#)), sendo seu quantitativo bem distribuído entre os anos de 2019 e 2023, tendo os anos de 2017 e 2018 uma quantidade menor de respostas, os outros anos tendo em conjunto um total de 4 respostas.

Avaliação de experiência acadêmica

Considerando que o escopo deste trabalho gira em torno da alocação de recursos físicos e humanos, como salas, professores e alunos, foi elaborada uma seção do formulário

¹ <https://forms.gle/zFfbvVtmKTVPX3cg9>

Tabela A.1 – Número de respondentes por curso

Quantidade	%	Curso
40	19,3	Ciências Biológicas (bacharelado)
29	14,0	Ciência da Computação
19	9,2	Medicina Veterinária
17	8,2	Engenharia Civil
13	6,3	Química (Licenciatura)
12	5,8	Engenharia Metalúrgica
11	5,3	Engenharia de Produção
11	5,3	Administração Pública
9	4,3	Ciências Sociais
8	3,9	Agronomia
8	3,9	Biologia (Licenciatura)
8	3,9	Pedagogia (Licenciatura)
8	3,9	Zootecnia
7	3,4	Engenharia de Exploração e Produção de Petróleo
6	2,9	Física (licenciatura)
1	0,5	Matemática (Licenciatura)
0	0,0	Engenharia Meteorológica
0	0,0	Outro

Tabela A.2 – Número de respondentes por ano

Quantidade	%	Ano
36	17,4	2023
35	16,9	2022
39	18,8	2021
34	16,4	2020
35	16,9	2019
11	5,3	2018
13	6,3	2017
2	1,0	2016
1	0,5	2015
0	0,0	2014
0	0,0	2013
1	0,5	Outro

de pesquisa com o intuito de se analisar a frequência de ocorrência de certas situações no contexto universitário. Para isto, foram feitas as sete perguntas listadas a seguir.

1. **Salas:** você já teve que mudar de sala por falta de algum acessório como quadro, projetor ou monitor?
2. **Salas:** você já teve aula cuja sala não dispunha de carteiras o suficiente?
3. **Vagas:** você já quis entrar em uma disciplina, mas ela não tinha vaga?
4. **Vagas:** você já ficou acordado após meia-noite por medo de não ter vaga para as disciplinas que deseja cursar?
5. **Conflitos:** você já deixou de se inscrever em uma disciplina por causa de conflito de horário?
6. **Preferências:** você já preferiu não se inscrever em uma disciplina para cursá-la em outro momento mais oportuno?
7. **Opiniões:** você acha que a universidade deveria oferecer horários diferentes para as disciplinas mais demandadas para evitar conflitos com outras?

A enumeração das perguntas feitas se encontra representada com suas respectivas respostas na [Tabela A.3](#).

Tabela A.3 – Ocorrência de experiências acadêmicas

Pergunta	Respostas			Respostas (%)		
	Sim	Não	Outro	Sim (%)	Não (%)	Outro (%)
1	110	87	10	53,1	42,0	4,8
2	109	95	3	52,7	45,9	1,4
3	176	28	3	85,0	13,5	1,4
4	189	17	1	91,3	8,2	0,5
5	188	15	4	90,8	7,2	1,9
6	178	27	2	86,0	13,0	1,0
7	207	0	0	100,0	0,0	0,0

Quanto à distribuição dos recursos físicos, vemos uma taxa de 53,1% de alunos que já tiveram que mudar de sala por falta de algum acessório disposto necessário para a aula. Já a necessidade de mudança de sala devido à ausência de carteiras suficientes obteve taxa de 52,7%.

É notório o receio dos alunos quanto à possibilidade de não conseguir se inscrever nas disciplinas que desejam cursar, tendo sido confirmado por 85,0% dos respondentes que se viram em situação em que a disciplina na qual desejavam se matricular não dispunha de vagas o bastante. Essa realidade resulta no temor por semestralmente não conseguir

se inscrever na disciplina desejada, fazendo com que 91,3% dos respondentes tenham se mantido acordados após a meia-noite por causa deste medo.

O temor de não conseguir se inscrever nas disciplinas desejadas é ainda agravado pelo fato de que 90,8% dos alunos que já deixaram de se inscrever em disciplinas devido a conflitos de horário.

O que se apresenta como um agravante ainda maior na percepção da progressão não sequencial dos alunos é a quantidade de alunos que já preferiram não se inscrever em uma disciplina para cursá-la em outro momento mais oportuno, mesmo que isto signifique um atraso na progressão do curso, sendo seu percentual 86,0%.

Embora seja uma prática recorrente a oferta de diversas turmas para uma mesma disciplina, o que usualmente é feito de forma que as turmas sejam ofertadas no mesmo horário. Entretanto, os alunos, unanimemente, não se mostram satisfeitos com esta prática, visto que 100% dos respondentes consideram que a universidade deveria dispor de outros horários para as disciplinas mais demandadas com o intuito de evitar conflitos de horários.

Este resultado é curioso, visto que o temor de não se atrasar em seu progresso e conseguir se inscrever nas disciplinas desejadas, contrasta diretamente com a preferência pessoal de não se inscrever em disciplinas e cursá-las posteriormente, mesmo que isso possa atrasar seu progresso. Entende-se que nem todas as disciplinas, caso não cursadas em seu período esperado, resultarão no atraso da grade, mas ainda assim, a antítese é evidente.

Preferências pessoais

Neste segmento, visa-se entender um pouco melhor o processo decisório dos alunos quanto à escolha das disciplinas que desejam cursar. Primeiro, lhes é indagado quanto à disposição das disciplinas, variando entre disciplinas concentradas em poucos dias ou espalhadas durante a semana e quanto à preferência de horários, variando entre horários matutinos e vespertino.

Embora não lide com conflitos, a análise de seus resultados pode auxiliar na escolha de distribuição futura dos usuários do sistema, ao desenvolverem a grade horária, caso desejem considerar as preferências dos estudantes.

Podemos ver na [Tabela A.4](#) que há uma grande distribuição entre as preferências dos alunos, tendendo às extremidades, onde alguns preferem bastante as disciplinas distribuídas ao longo da semana enquanto outros preferem disciplinas acumuladas em poucos dias, vindo como terceira opção mais votada a neutralidade. Observação similar se mostra presente também na [Tabela A.5](#).

Em seguida, é questionado sobre qual é o critério de seleção de disciplinas que se apresentam conflituosas ([Tabela A.6](#)). Nesta vertente vemos uma maior propensão à

Tabela A.4 – Preferências por distribuição de disciplinas em um mesmo dia

Quantidade	%	Distribuição na semana
58	28,0	Distribuídas ao longo da semana
34	16,4	Preferencialmente ao longo da semana
38	18,4	Não tenho preferência
35	16,9	Preferencialmente acumulada em poucos dias
42	20,3	Acumuladas em poucos dias

Tabela A.5 – Preferências por distribuição de disciplinas em um mesmo dia

Quantidade	%	Distribuição no dia
65	31,4	Na parte da manhã
19	9,2	Preferencialmente na parte da manhã
40	19,3	Não tenho preferência
28	13,5	Preferencialmente na parte da tarde
55	26,6	Na parte da tarde

escolha por disciplinas que são pré-requisito de uma grande quantidade de disciplinas, ou seja, disciplinas que, caso se tenham reprovação ou não sejam cursadas, resultam no que é coloquialmente chamado de “prender disciplinas”, assim atrasando mais a progressão do aluno. Sendo este o critério adotado por 81,6% dos respondentes. A segunda maior opção selecionada foi a de escolher a disciplina mais concorrida, ou seja, a disciplina que possui uma maior demanda de alunos, sendo um critério utilizado por 25,1% dos alunos.

Tabela A.6 – Critérios para a escolha de disciplinas conflituosas

Quantidade	%	Forma de escolher disciplina conflituosa
169	81,6	Escolho a que “prende” mais matérias
52	25,1	Escolho a disciplina mais concorrida
41	19,8	Escolho a que prefiro
23	11,1	Escolho a que tem mais créditos
18	8,7	Escolho a mais difícil
12	5,8	Outro
9	4,3	Escolho a mais fácil

Vale ressaltar que as respostas ilustradas pela Tabela A.6 permite a seleção de múltiplas escolhas, inclusive permitindo que adicionassem outros critérios próprios. Dentre eles, um se mostrou ligeiramente recorrente que seria escolher a disciplina mais atrasada segundo a grade curricular. Outra possibilidade citada foi selecionar a que se encaixam na disponibilidade de horário pessoal, para que não conflite, por exemplo, com o horário do estágio obrigatório.

Experiências com atrasos e disciplinas

Quanto aos atrasos para a realização de disciplinas, o ideal desejado é que não haja nenhum atraso. Nessa situação, todos os alunos que entram na universidade poderão seguir a disponibilidade usual das disciplinas dispostas em suas grades curriculares, que apresentam o período esperado para que cada disciplina seja realizada. Sendo elas, usualmente divididas como disciplinas pares e ímpares. As ímpares se referem às disciplinas em que se espera que sejam cursadas nos períodos 1, 3, 5, 7 e 9, e que são ofertadas no primeiro semestre letivo. Enquanto que as pares se referem às disciplinas oferecidas no segundo período letivo, onde geralmente se alocam as disciplinas dos períodos 2, 4, 6, 8 e 10.

Entretanto, a realidade dos alunos é outra. Isso se dá por diversos motivos, seja por reprovação, por não conseguir se inscrever na disciplina desejada ou por simplesmente não ter interesse em cursar a disciplina naquele momento, como já ilustrado na [Tabela A.3](#). Esta característica se confirma na percepção da frequência e distância que percebemos dos atrasos, representado pela [Tabela A.7](#). Períodos de atraso

1. Quanto tempo (em períodos) você já teve que esperar para fazer uma disciplina da sua grade?
2. Qual foi a quantidade máxima de períodos que você se distanciou de uma disciplina de determinado período?

Tabela A.7 – Tempo de atraso em disciplinas

Pergunta	Períodos de atraso										
	0	1	2	3	4	5	6	7	8	9	10
1	53	38	79	19	11	1	1	3	1	1	0
2	33	27	60	31	28	15	9	1	2	1	0
1 (%)	25,6	18,4	38,2	9,2	5,3	0,5	0,5	1,4	0,5	0,5	0,0
2 (%)	15,9	13,0	29,0	15,0	13,5	7,2	4,3	0,5	1,0	0,5	0,0

Apresenta-se notável que é minoria a quantidade de alunos que nunca tiveram que esperar para cursar uma disciplina, sendo estes apenas 25,6% dos respondentes. É ainda mais notável o fato de que o tempo de espera médio é de mais de um semestre. Quanto ao distanciamento de disciplinas, seja por reprovações ou por escolha própria se mostra ainda mais presente, sendo que apenas 15,9% dos respondentes não se distanciaram das disciplinas esperadas para o período, sendo o tempo médio de distanciamento maior que dois semestres.

Pesquisa de opinião

Aqui, busca-se uma análise mais bruta e direta à concordância dos respondentes quanto às características atribuídas à distribuição de disciplinas semestrais, ondeles eles avaliam com notas de 1 a 5 o quanto concordam com cada uma das características dadas à distribuição de disciplinas, sendo eles “Justa” (feita de acordo a atender os desejos da maioria), “Variada” (bem diversa e abrange diversos interesses), “Contínua” (oferecida de forma a ter aulas sequenciais), “Eficiente” (bem sucedida em atender aos desejos dos alunos), “Distribuída” (bem espaçada ao longo da semana) e “Satisfatória” (agradável aos meus desejos pessoais). Estando as opiniões dos alunos refletidas nos resultados expostos pela [Tabela A.8](#), seguidas das médias por característica.

Tabela A.8 – Notas dadas às características da distribuição de disciplinas

Opções	Notas					Médias
	1	2	3	4	5	
Justa	59	69	52	17	10	2,3
Variada	40	65	64	26	12	2,5
Contínua	34	44	78	41	10	2,8
Eficiente	61	76	45	17	8	2,2
Distribuída	32	40	61	60	14	2,9
Satisfatória	40	61	68	30	8	2,5

De uma forma geral, conseguimos ver todos os gráficos com menos de 7% dos respondentes dando nota 5 em cada uma das características. A maioria das respostas tende a estar entre 1 e 3, sendo exceção apenas no caso da característica “distribuída”, que por sua vez apresenta uma elevada quantidade de notas 4, sendo referente a 29,3% dos respondentes.

Ao analisarmos a média de cada uma, podemos dizer que, em suma, há o visível desagrado do corpo discente quanto à distribuição de disciplinas semestrais, com ênfase nas duas piores notas que são 2,2 para “eficiente” e 2,3 para “justa” o que reforça a necessidade de aprimoramento do sistema atual.

Respostas qualitativas

Por fim, havia um espaço livre no formulário para que os alunos pudessem expressar suas opiniões de forma mais livre. Após a leitura de todas e a filtragem das opiniões expressas, resumem-se em 4 parabenizações pelo desenvolvimento do presente projeto, 18 reclamações e 16 sugestões. Dentre elas, algumas apresentaram maior recorrência, sendo elas:

- 5 reclamações sobre a usual oferta de disciplinas separadas entre pares e ímpares;

- 4 reclamações sobre o Sistema Acadêmico, principalmente sobre não ser capaz de suportar a carga nos momentos iniciais de inscrição de disciplinas;
- 3 sugestões de ofertas de disciplinas recorrentemente, com ênfase nas disciplinas de matemática/que contemplam diversos cursos/com alta taxa de reprovação;
- 3 sugestões de mais oferta de disciplinas no período de verão;
- 2 sugestões de que inscrições em matérias do semestre atual esperado do aluno fossem feitas automaticamente, mas ainda permitindo a sua exclusão caso desejado;
- 1 sugestão de criação de um formulário de demanda no acadêmico que computasse a intenção de matrícula dos alunos.

Abaixo estão dispostas algumas das respostas obtidas:

- “A estrutura curricular deveria ser predefinida, automatizando a matrícula dos alunos nas disciplinas correspondentes aos seus períodos acadêmicos. No entanto, permitir-se-ia a edição do cronograma por parte dos alunos, caso desejem quitar pendências de períodos anteriores ou disciplinas antecipadas. Além disso, disciplinas que abrangem múltiplos cursos devem ser oferecidas em ambos os semestres.

Para melhorar a oferta de disciplinas, seria aconselhável ampliar a disponibilidade de disciplinas durante o período de verão. Isso facilitaria o acesso dos alunos ao estágio obrigatório durante as férias, viabilizando a conclusão dessa etapa essencial do curso. [...] Essas melhorias no sistema acadêmico [...] agilizariam a trajetória do estudante, permitindo maior flexibilidade na escolha e realização de disciplinas [...].”

- “Existe muita desorganização em relação a grade, disciplinas e sistema acadêmico por parte dos coordenadores dos cursos. Me inscrevi numa matéria que tinha requisito de acordo com o plano pedagógico, mas tanto na grade como no sistema acadêmico não tinha pré-requisito nenhum. Tive que desistir da matéria pela dificuldade, até porque a matéria que era pré-requisito eu perdi. Detalhe: Outras pessoas perderam na matéria pré-requisito e continuaram fazendo a matéria desse período.”
- “Acho que as coordenações precisam estabelecer um melhor diálogo com os alunos. Esse sistema de período par e ímpar na UENF é antigo e perpetua um comodismo dos professores que acabam não ofertando disciplinas todos os semestres e prejudicando os alunos nas escolhas de matérias optativas, eletivas, instrumentais e obrigatórias tendo que esperar um ano para realizar a disciplina caso você não consiga fazer por choque de horário e/ou reprovação.”

Conclusões

Por fim, entendemos que, além das insatisfações dormentes por parte dos gestores e criadores de grades horárias, os alunos também se mostram insatisfeitos com a atual estrutura de distribuição de disciplinas semestrais. Os interesses dos alunos se mostram em sua maioria alinhados com os interesses dos gestores, onde ambos visam reduzir a quantidade de atrasos na progressão do curso.

APÊNDICE B – FORMULÁRIO DE PESQUISA

Abaixo está o formulário de pesquisa quantitativa de alunos da UENF sobre distribuição e oferta de disciplinas. Este formulário foi feito em L^AT_EXe é editável, podendo ser preenchido e enviado por e-mail. Para preenchê-lo, basta clicar no campo desejado e preencher com as informações solicitadas. Para enviar o formulário, clique no botão “Enviar”.

Pesquisa quantitativa de alunos da UENF sobre distribuição e oferta de disciplinas

Olá! Desde já agradeço por ceder em torno de 4 minutos do seu tempo para responder a este formulário usando o seu e-mail institucional. Considerando que nosso tempo é valioso, vamos direto ao objetivo:

Me chamo João Vítor Fernandes Dias, estudante de Ciência da Computação na UENF, e estou fazendo minha Monografia. Ela trata da elaboração de um sistema para a coordenação de curso poder analisar mais facilmente quais são as disciplinas que serão disponibilizadas a cada semestre e a quais salas e professores serão atribuídas.

O objetivo da minha monografia é conseguir tornar mais eficiente a distribuição das disciplinas, para que se resulte em um conjunto de disciplinas ofertadas com melhor qualidade. Espera-se com isso que as demandas de disciplina dos alunos sejam melhor atendidas, assim como as preferências de horários dos professores.

Este formulário tem como objetivo avaliar a sua satisfação em relação ao processo de inscrição semestral nas disciplinas.

Sobre você

Nesta seção, peço que informe algumas características suas para que a análise estatística se torne mais rica.

1. Qual o seu curso?

2. Em que ano você ingressou na UENF?

Pesquisa de satisfação

Agora serão feitas algumas perguntas em relação à sua satisfação com algumas características da Universidade.

Abaixo, estão algumas perguntas gerais em relação à sua satisfação com a distribuição de disciplinas semestralmente.

1. **Salas:** Você já teve que mudar de sala por falta de algum acessório como quadro, projetor ou monitor?

2. **Salas:** Você já teve aula cuja sala não dispunha de carteiras o suficiente?

3. **Vagas:** Você já quis entrar em uma disciplina, mas ela não tinha vaga?

4. **Vagas:** Você já ficou acordado após meia-noite por medo de não ter vaga para as disciplinas que deseja cursar?

5. **Conflitos:** Você já deixou de se inscrever em uma disciplina por causa de conflito de horário?

6. **Preferências:** Você já preferiu não se inscrever em uma disciplina para cursá-la em outro momento mais oportuno?

7. **Opiniões:** Você acha que a universidade deveria oferecer horários diferentes para as disciplinas mais demandadas para evitar conflitos com outras disciplinas?

Preferências pessoais

Esta seção visa saber um pouco mais sobre as suas preferências pessoais quanto a escolha das disciplinas ofertadas.

1. Você prefere disciplinas distribuídas ao longo da semana ou acumuladas em poucos dias?

2. Você prefere disciplinas na parte da manhã ou na parte da tarde?

3. Como você lida com conflitos de horário entre as disciplinas que deseja cursar?
 - Escolho a mais difícil
 - Escolho a mais fácil
 - Escolho a que tem mais créditos
 - Escolho a que prefiro
 - Escolho a que “prende” mais matérias
 - Escolho a disciplina mais concorrida
 - Outro...

Experiências passadas com atrasos e disciplinas

Aqui estão algumas perguntas relacionadas à divergência entre o período esperado de conclusão das disciplinas VS o período em que elas de fato são realizadas.

1. Quanto tempo (em períodos) você já teve que esperar para fazer uma disciplina da sua grade?

2. Qual foi a quantidade máxima de períodos que você se distanciou de uma disciplina de determinado período?

Você acha que a distribuição de disciplinas semestrais é...

1. Justa (feita de acordo a atender os desejos da maioria)

2. Variada (bem diversa e abrange diversos interesses)

3. Contínua (oferecida de forma a ter aulas sequenciais)

4. Eficiente (bem sucedida em atender aos desejos dos alunos)

5. Distribuída (bem espaçada ao longo da semana)

6. Satisfatória (agradável aos meus desejos pessoais)

Optional

Por fim, deixo aqui um espaço caso deseje compartilhar algum comentário, opinião ou sugestão quanto ao meu trabalho ou formulário.

Escreva aqui caso haja algo que gostaria de comentar, opinar ou sugerir. Tudo bem deixar em branco, suas informações já foram de grande ajuda.

[Resetar](#) [Enviar](#)

APÊNDICE C – CÓDIGO-FONTE DA MONOGRAFIA

Como forma mais prática é sugerida de acesso ao código-fonte da monografia, recomenda-se acessar o [repositório do GitHub¹](https://github.com/jvfd3/timetabling-UENF), nele constará o código-fonte mais atualizado do sistema. Caso por algum motivo o link não esteja funcionando apropriadamente, está disponibilizado também na guia de Anexos do documento PDF deste trabalho o código-fonte do sistema. Para acessá-lo, basta procurar a aba de anexos em algum leitor de PDF que suporte anexos, como por exemplo o [Adobe Reader²](#), ou clique [aqui](#).

Além do código do sistema, também está disponível o código-fonte L^AT_EXdeste documento, também num [repositório do GitHub³](https://github.com/jvfd3/timetabling-UENF), porém nesse caso, talvez seja necessário passar por uma burocracia antes, visto que o primeiro link é de um repositório contido em uma organização privada, sendo necessário primeiro obter acesso a ela através de um procedimento descrito [aqui⁴](#). Caso prefira, assim como no caso anterior, dispõe-se também o código-fonte L^AT_EXdeste documento na guia de Anexos do documento PDF, ou clique [aqui](#).

Como em alguns casos o leitor de PDF pode não ter suporte para abrir arquivos compactados, a extensão do arquivo foi alterada. Para acessar o arquivo, basta salvá-lo em seu computador, remover a parte final do nome do arquivo e descompactá-lo utilizando um programa de sua preferência. Caso prefira, também é possível abrir o arquivo diretamente com o programa de compactação, sem a necessidade de alterar o nome.

¹ <https://github.com/jvfd3/timetabling-UENF>

² <https://get.adobe.com/br/reader>

³ <https://github.com/UENF-Organizacao-de-Disciplinas/INF01131-Monografia>

⁴ <https://github.com/UENF-Organizacao-de-Disciplinas>

APÊNDICE D – EXEMPLO DE *TEMPLATE.YAML*

O arquivo *template.yaml* é um arquivo de configuração do *AWS SAM* que define como a aplicação será construída e como será feito o deploy dela. Abaixo está um exemplo deste arquivo.

Código D.1 – Exemplo de *template.yaml*

```

1 AWSTemplateFormatVersion: "2010-09-09"
2 Transform: AWS::Serverless-2016-10-31
3 Description: "OurClass>SAM>General>"
4
5 Resources:
6   # LAYERS
7     dbFunctionLayer:
8       Type: AWS::Serverless::LayerVersion
9       Properties:
10      LayerName: dbFunctionSAM
11      Description: connect to db
12      ContentUri: ./layers/dbConnection
13      CompatibleRuntimes:
14        - nodejs14.x
15        - nodejs16.x
16        - nodejs18.x
17   # API GATEWAY
18   TimetablingAPI:
19     Type: AWS::Serverless::Api
20     Properties:
21       StageName: timetablingStage
22       EndpointConfiguration: REGIONAL
23       Cors:
24         AllowMethods: "'GET,POST,PUT,DELETE,OPTIONS'"
25         AllowHeaders: "'Content-Type,X-Amz-Date,
26           Authorization,X-Api-Key,X-Amz-Security-Token'"
27         AllowOrigin: "'*'"
28   # LAMBDA FUNCTIONS

```

```
29 createProfessor:
30   Type: AWS::Serverless::Function
31   Properties:
32     FunctionName: createProfessor
33     CodeUri: ./lambdas/professor/createProfessor
34   Events:
35     ApiEvent:
36       Type: Api
37       Properties:
38         Path: /professor
39         Method: post
40         RestApiId: !Ref TimetablingAPI
41     Runtime: nodejs14.x
42     Handler: index.handler
43     Role:
44       Fn::Sub: arn:aws:iam::${AWS::AccountId}:role/
LambdaRole
45   Layers:
46     - !Ref dbFunctionLayer
47   Environment:
48     Variables:
49       DB_HOST: ourclassuirds.cgsgwtemx5r8.us-east-2.
rds.amazonaws.com
50       DB_NAME: OurClassDB
51       DB_USER: tang
52       DB_PSWD: annabell
53   readProfessor:
54   # [...]
55   updateProfessor:
56   # [...]
57   deleteClassTime:
58   Type: AWS::Serverless::Function
59   Properties:
60     FunctionName: deleteClassTime
61     CodeUri: ./lambdas/classTime/deleteClassTime
62   Events:
63     ApiEvent:
64       Type: Api
65       Properties:
```

```
66      Path: / classTime/{ id }
67      Method: delete
68      RestApiId: ! Ref TimetablingAPI
69      Runtime: nodejs14.x
70      Handler: index.handler
71      Role:
72          Fn::Sub: arn:aws:iam::${AWS::AccountId}:role/LambdaRole
73      Layers:
74          - ! Ref dbFunctionLayer
75      Environment:
76      Variables:
77          DB_HOST: ourclassuirds.cgsgwtemx5r8.us-east-2.
78          rds.amazonaws.com
79          DB_NAME: OurClassDB
80          DB_USER: tang
81          DB_PSWD: annabell
```