

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

13 de abril 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

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Ciência da Com-
putação 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

13 de abril de 2024

Banca

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 ao professor Dr. Manuel Antonio Molina Palma pela dedicação e paciência durante o lecionamento desta disciplina, e obrigado pela ajuda e por estar disponível nos momentos de necessidade.

Por último, mas não menos importante, agradeço a toda minha família e amigos que estiveram comigo em todos os momentos da minha vida.

Resumo

Este artigo visa apresentar uma análise da situação em que atualmente se encontra a criação de grades horárias na Universidade Estadual do Norte Fluminense, bem como apontar seus maiores problemas e desenvolver um sistema de decisão capaz de auxiliar diversos centros e laboratórios no desenvolvimento de suas grades horárias, buscando a otimização do uso de salas e redução dos conflitos existentes entre demandas de diferentes alunos por matérias.

Palavras-chave: tabela de horários. agendamento de aulas universitárias. heurísticas. representação do conhecimento. interação humano-computador.

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

Vovô Nando, 01/12/2023

Abstract

This paper aims to present an analysis of the current situation of the schedule creation at the Universidade Estadual do Norte Fluminense, as well as to point out its major problems and to develop a decision system capable of helping several centers and laboratories in the development of their schedules, seeking to optimize the use of rooms and to reduce the conflicts between different students' demands for subjects.

Keywords: Timetabling. University class scheduling. Heuristics. Integer Programming. Knowledge Representation. Human Computer Interaction.

Sumário

1	INTRODUÇÃO	14
1.1	Problemáticas	15
1.2	Hipótese	15
1.3	Objetivos	16
1.3.1	Gerais	16
1.3.2	Específicos	16
1.4	Justificativas	16
1.5	Metodologia	17
1.6	Organização	19
2	CONTEXTO ACADÊMICO <i>TIMETABLING PROBLEM</i>	20
2.1	Definição de termos	20
2.2	Métodos de resolução	21
2.3	Desafios recorrentes	23
2.4	Trabalhos anteriores	24
2.4.1	Sânya	24
2.4.2	Ricardo	25
2.4.3	Divergências	25
2.4.3.1	Sânya	25
2.4.3.2	Ricardo	27
2.4.3.3	Parecer geral das divergências	28
3	ESTRUTURA	29
3.1	A UENF e seu estatuto	29
3.2	Entrevistas	31
3.2.1	Diretor do CCT	31
3.2.2	Desenvolvedor do Sistema Acadêmico	33
3.2.3	Chefe de Laboratório de Matemática	34
3.2.4	Responsável pela Secretaria Acadêmica (SECACAD)	34
3.2.5	Coordenador de Computação	35
3.2.6	Entendimento geral das entrevistas	37
3.3	Sequência de criação das grades horárias	37
3.4	Formulário de pesquisa	38
3.4.1	Respondentes	39
3.4.2	Avaliação de experiência acadêmica	41
3.4.3	Preferências pessoais	43

3.4.4	Experiências com atrasos e disciplinas	45
3.4.5	Pesquisa de opinião	47
3.4.6	Respostas qualitativas	48
3.4.7	Conclusões	51
3.5	Exemplo de erros humanos	52
3.6	Relações entre as variáveis	52
3.6.1	Turmas	52
3.6.2	Laboratórios	54
3.6.3	Professores	55
3.6.4	Centros	55
3.6.5	Salas	55
3.6.6	Cursos	55
3.6.7	Alunos	55
3.6.8	Disciplinas	55
3.6.9	Blocos	55
3.6.10	Pactos Sociais	55
4	MODELAGEM GERAL DO SISTEMA	56
4.1	Estágios de execução	56
4.2	Iteração	57
4.3	Funcionamento	57
4.4	Modelo de Banco de Dados	58
4.4.1	Diagrama de Entidade e relacionamento	59
5	DESENVOLVIMENTO	60
5.1	Projetos anteriores	60
5.1.1	Andamento dos alunos	60
5.1.2	Cálculo de demanda	60
5.2	Dados pessoais e a LGPD	62
5.3	Prototipagem	62
5.4	Programação do sistema	70
5.4.1	Primeira Versão	70
5.4.1.1	Banco de Dados Primitivo	70
5.4.1.2	Funcionalidades Iniciais	71
5.4.1.3	GitHub Pages	71
5.4.2	Segunda Versão	71
5.4.2.1	GitHub Projects	72
5.4.2.2	Permanência dos Dados	72
5.4.2.3	<i>Amazon Web Services</i>	76
5.4.2.4	Funcionalidades Adicionais	81

5.4.3	Terceira Versão	81
5.4.3.1	Mudanças no <i>GithHub Projects</i>	81
5.4.3.2	Melhorias no Sistema	84
5.5	Conflitos	91
5.5.1	Típicos Conflitos Atípicos	91
5.5.2	Conflitos Tratados pelo Sistema	92
5.5.2.1	Professores	93
5.5.2.2	Salas	94
5.5.2.3	Disciplina	95
6	EXPERIMENTOS	97
6.1	Aquisição dos dados	97
6.2	Cenários	97
7	RESULTADOS	99
7.1	Soluções Burocráticas	99
7.1.1	Tempo de elaboração das grades	99
7.1.2	Alteração forçada de horários	100
8	CONCLUSÕES	101
8.1	Trabalhos futuros	101
	REFERÊNCIAS	104
	APÊNDICE A – FORMULÁRIO DE PESQUISA QUANTITATIVA	106

Listas de ilustrações

Figura 1 – Resumo de trabalhos, parâmetros, dimensões, tempo e técnicas	22
Figura 2 – Comparação entre artigos que solucionam o problema de grade horária	23
Figura 3 – Análise de publicações aceitas	23
Figura 4 – Perguntas sobre o curso e ano de ingresso dos estudantes	39
Figura 5 – Respostas sobre a satisfação dos estudantes	42
Figura 6 – Preferências por distribuição de disciplinas ao longo da semana	43
Figura 7 – Preferências por distribuição de disciplinas em um mesmo dia	44
Figura 8 – Critérios para a escolha de disciplinas conflituosas	45
Figura 9 – Períodos de atraso por espera	46
Figura 10 – Distância de atraso	47
Figura 11 – Notas dadas para as características “Justa”, “Variada” e “Contínua” . .	49
Figura 12 – Notas dadas para as características “Eficiente”, “Distribuída” e “Satisfatória”	50
Figura 13 – Disciplina atribuída no sistema acadêmico à determinada hora e local .	53
Figura 14 – Disciplina não atribuída à determinada hora e local na grade de horários do CCT	54
Figura 15 – Estágios para a obtenção de grade horária ótima	56
Figura 16 – Etapas do Design de Interação	57
Figura 17 – Funcionamento geral do sistema	58
Figura 18 – Diagrama Conceitual do banco de dados	59
Figura 19 – Página principal do sistema	63
Figura 20 – Página principal de modificação dos dados	64
Figura 21 – Página de modificação das informações de salas	65
Figura 22 – Página de modificação das informações de alunos	66
Figura 23 – Página de modificação das informações de disciplinas	67
Figura 24 – Página de modificação	68
Figura 25 – Página de modificação das informações de turmas	69
Figura 26 – Recursos usados para o desenvolvimento do <i>Frontend</i>	70
Figura 27 – Tabela de tarefas do GitHub Projects	72
Figura 28 – Diagrama da progressão funcionamento da permanência dos dados . .	73
Figura 29 – Diagrama primitivo das tabelas de dados SQL	75
Figura 30 – API REST no AWS	76
Figura 31 – Marcos do GitHub Projects	82
Figura 32 – Etiquetas do GitHub Projects	83
Figura 33 – Gráfico de Marco <i>versus</i> quantidade de tarefas separadas por etiqueta .	84
Figura 34 – Diagrama do fluxo de obtenção de dados	88

Figura 35 – Paleta de cores do sistema	93
Figura 36 – Exemplo de conflito de alocação de professor	93
Figura 37 – Exemplo de conflito de alocação de sala	94
Figura 38 – Exemplo de conflito de capacidade na sala	95
Figura 39 – Avisos flutuantes dos conflitos de disciplinas	96

Lista de tabelas

Tabela 1 – Número de respondentes por curso	40
Tabela 2 – Número de respondentes por ano	40
Tabela 3 – Ocorrência de experiências acadêmicas	41
Tabela 4 – Preferências por distribuição de disciplinas em um mesmo dia	44
Tabela 5 – Preferências por distribuição de disciplinas em um mesmo dia	44
Tabela 6 – Critérios para a escolha de disciplinas conflituosas	46
Tabela 7 – Tempo de atraso em disciplinas	47
Tabela 8 – Notas dadas às características da distribuição de disciplinas	48

Lista de códigos

5.1	Obter demanda por extratos em PDF	61
5.2	Código de <i>deploy</i> de Lambda	77
5.3	<i>Exemplo de template.yaml</i>	78

*

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. Por exemplo, é esperado que o discente apenas curse a disciplina Cálculo 2 após haver obtido a aprovação prévia na disciplina Cálculo 1.

Embora haja este planejamento de duração do curso, diversos fatores podem influenciar esta previsão, dentre eles podemos citar eventos como:

- **Quebra de pré-requisitos:** onde o discente solicita permissão para inscrição em uma disciplina cujos pré-requisitos não são completamente cumpridos por si;
- **Trancamento de matrícula:** onde o discente suspende temporariamente seus estudos na instituição;
- **Transferência interna:** onde o discente migra entre cursos dentro da mesma instituição;
- **Transferência externa:** onde o discente migra entre cursos entre diferentes instituições;
- **Reprovações:** onde o discente não cumpre com o mínimo desempenho esperado na disciplina, geralmente está associado a ausência nas aulas e/ou desempenho inferior ao mínimo esperado nas avaliações;
- **Disponibilidade de professores:** onde os docentes não são suficientes para ministrar todas as disciplinas demandadas pelos discentes em um mesmo semestre.

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 a 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 as reprovações nas disciplinas causam nos semestres consecutivos. Para isso sendo então necessária uma análise das disciplinas que devem ser ministradas 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 Joshua (THOMAS; KHADER; 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 J. Miranda, 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 deste documento 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.

1.3.1 Gerais

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.

1.3.2 Específicos

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 as demandas de aprimoramentos desejadas pelos diferentes centros e laboratórios;
- Modelar o sistema de resolução de *timetabling* de acordo com os requisitos demandados;
- Encontrar o que é necessário para a adoção da aplicação de tabelamento de horário;
- 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 este 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; DINATA, 2018).

Nesta pesquisa exploratória em formato de entrevista, algumas informações esperadas revolvem em torno das percepções dos *stakeholders* do sistema proposto, sendo esses principalmente os professores, coordenadores de cursos, chefes de laboratório e diretores de centro. Estas percepções incluem o entendimento deles quanto ao método atual e às alternativas existentes, nível de insatisfação com o método atual, nível de desejo quanto à um novo método. Além disso, espera-se aproveitar o ensejo para eliciar as características e funcionalidades que gostariam de ter em um sistema de suporte à decisão, solicitando também que deem informações adicionais que gostariam de acrescentar.

Essas informações serão relevantes para se atingir a satisfação e uso futuro do sistema proposto. Pois, como é informado no (BOURQUE; FAIRLEY, 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.

Questionamentos similares também serão realizados com alunos, porém em formato de formulário online para facilitar o processamento dos dados coletados.

Tendo obtido as informações dos *stakeholders* primários, será então necessário modelar quais são as regras que ditam a estrutura organizacional em foco. Para este fim, serão utilizados diagramas conceituais utilizando *softwares* de suporte como o [draw.io](#) e o [Mermaid](#).

Esta etapa será de grande importância pois guiará a pesquisa para quais serão os detalhes dos módulos existentes durante o desenvolvimento do projeto, bem como esclarecerá visualmente quais são as informações sobre os recursos que são necessárias para se calcular a grade ótima. Como por exemplo:

1. Salas

- a) Quais são as salas disponíveis?
- b) Quais as capacidades de cada um?
- c) Em quais horários estão disponíveis?
- d) Quais são suas peculiaridades?
 - i. Têm computadores?
 - ii. Têm quadro?
 - iii. Têm televisão?
 - iv. Têm projetor?

2. Alunos

- a) Quantos são?
- b) Quais matérias demandam?

3. Professores

- a) Quais disciplinas ministram?
- b) Quantas disciplinas podem ministrar?
- c) Quais seus horários de preferência?

Com as regras organizacionais e variáveis bem definidas, serão testados alguns *softwares* que visam a criação de grades horárias para confirmar se há a real necessidade de se desenvolver um *software* específico para a instituição. Após realizados os testes, caso os *softwares* existentes supram as necessidades, este será utilizado nos passos seguintes. De outro modo, haverá a necessidade de desenvolvimento de um sistema de suporte à decisão como ferramenta centralizada para este fim.

Independente de qual dos *softwares* será testada a aplicabilidade do mesmo no contexto universitário e será mensurada a satisfação dos *stakeholders* durante o seu uso, assim buscando assegurar o seu uso na criação de grades horárias ótimas futuras.

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 20 de revisão literária informa mais detalhadamente sobre os problemas de agendamento, suas categorias, soluções, desafios e definições de termos;
- O Capítulo 3 na página 29...
- O Capítulo 4 na página 56...
- O Capítulo 5 na página 60 de desenvolvimento apresenta as informações coletadas durante as entrevistas. Apresenta também a estrutura geral dos códigos feitos, principalmente ilustrando quais os comportamentos esperados em cada um dos módulos, bem como quais foram as ferramentas utilizadas e as práticas seguidas;
- O Capítulo 6 na página 97...
- O Capítulo 7 na página 99 de resultados e discussões demonstra o *software* final utilizado, apresenta comparações das qualidades entre grades horárias geradas pelo *software* e as que foram utilizadas nos últimos semestres. Apresenta também a pesquisa de satisfação realizada com os *stakeholders* entrevistados no início do desenvolvimento;
- O Capítulo 8 na página 101 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 Contexto acadêmico *Timetabling Problem*

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 tempo pode se referir a vários conceitos diferentes de acordo com o autor.

Assim como feito por ([WREN, 1996](#)), definiremos os conceitos dos termos que serão usados ao longo deste trabalho.

O termo *timetable* tem o mesmo valor que “grade horária” e serão usadas como se fossem sinônimos mesmo sendo de línguas diferentes. 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; MAYA-PADRON; AVILA-TORRES, 2021](#)). Entretanto, neste trabalho, abordaremos principalmente os termos relacionados ao que pode ser chamado de *Educational Timetabling* (Ed-TT) ([ALENCAR et al., 2019a](#)), que é o que tende a envolver um conjunto específico de recursos relacionados à educação.

Wren 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.

Trazendo para a realidade do Curso de Ciência da Computação na UENF podemos exemplificar que enquanto que disciplina **laboratório de física** exige que a aula seja ministrada em um tipo de sala específica com os equipamentos necessários, já a disciplina **computação e sociedade** não apresenta esta restrição, ficando limitada apenas à quantidade de pessoas na turma.

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., 2019a) 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., 2019a) 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; ROBLES, 2012), J. Miranda informa sobre diversos sistemas baseados em computador para auxiliar na tarefa de agendamento. J. Miranda também cita 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 1, Alegre \(2012\)](#) 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 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	Tempo	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 Colomni <i>et al.</i>	1998	Itália	✓	✓	✓		10	24		300	8 h		✓	✓	✓				
10 Schaeff	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: ([ALEGRE, 2012](#))

Na [Figura 2, \(ARRATIA-MARTINEZ; MAYA-PADRON; AVILA-TORRES, 2021\)](#), apresenta 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 3, \(ALENCAR et al., 2019a\)](#) explora uma categoria mais específica do problema, que é a característica da interatividade das interfaces desenvolvidas. Este

Figura 2 – Comparaçāo 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]	Oladjejo 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; MAYA-PADRON; AVILA-TORRES, 2021](#)) - editado

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 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	Applc.	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	Y	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 constraints network, with back-tracking) 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 constraints 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., 2019a](#)) - 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.

Tomáš Müller ([MURRAY; MÜLLER; RUDOVÁ, 2007](#)) traz 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, Tomáš cita 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 documentados.

Com isso, entramos também no ramo da Interação Homem-Máquina, ramo abordado por Dinata ([ANDRE; DINATA, 2018](#)) que visou 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., 2019a](#)) em sua revisão literária

2.4 Trabalhos anteriores

Este trabalho não se mostra desprovido de histórico na tentativa de resolução do mesmo problema. Sânya e Ricardo, ambos estudantes de Ciência da Computação da UENF, já realizaram trabalhos com o mesmo fim, porém com abordagens diferentes da atual proposta.

Tendo vista que atualmente o problema de Programação Horária da UENF ainda perdura, podemos considerar que embora os trabalhos anteriores tenham se mostrado importantes ao pavimentar o caminho em direção à resolução da problemática disposta, as soluções ótimas encontradas por ambos, embora ótimas para a modelagem proposta, não se mostraram ótimas para a realidade da universidade.

Abaixo são listados os trabalhos anteriores e suas respectivas abordagens, bem como os apontamentos do que se mostrou inviável para a realidade da universidade.

2.4.1 Sânya

Em seu trabalho, Sânya 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

Em seu trabalho, Ricardo 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 (CG) e Busca Local (BL) e os métodos metaheurísticos *Simulated Annealing* (SA) e Busca Tabu (BT). 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 Divergências

Embora ambos os trabalhos tenham sido de grande valia ao rumarem na direção de uma solução para este amplo e complexo problema, percebe-se que o problema que buscaram solucionar ainda se encontra em aberto. Não tendo suas ferramentas alçado voos altos o suficiente para que se tornassem soluções definitivas para o problema no contexto da UENF.

2.4.3.1 Sânya

É dito por Sânya que “[...] Como na UENF a tarefa de distribuição de sala não varia muito a cada período, sendo feito separadamente por cada centro [...]”. Embora possamos entender o conceito de “variar muito” como subjetivo, considerando que mesmo ao longo de um mesmo semestre existem realocações de salas e professores dentro do contexto de um mesmo Centro, podemos entender que a realidade da UENF é de fato

muito dinâmica, não se encaixando completamente na solução de alocação única inicial de salas e professores.

Pode-se alegar que tratar da variabilidade de alocações de salas de um mesmo Centro foge do escopo do trabalho, porém, para que o coordenador da Computação tenha fácil acesso aos dados de alocação de salas disponíveis, faz-se necessário que seu uso esteja compartilhado com o Diretor do Centro de Ciência e Tecnologia (CCT), visto que este é o responsável pela alocação de salas de todos os cursos do CCT.

Em outro segmento ela diz que “[...] as aulas que necessitam de salas com recursos especiais são geralmente já preestabelecidas, não há necessidade de automatizar esta tarefa de distribuição de salas”, mas a realidade é que embora algumas disciplinas tenham suas salas pré-estabelecidas, não são todas as disciplinas que possuem esta característica, isso não significa necessariamente que esta alocação é a mais adequada para a mesma. Então, todas as salas, mesmo que inicialmente pré-estabelecidas, devem estar passíveis de mudanças, mas com possibilidade de se fixar.

Ela diz também que “Outra tarefa que no presente cenário do curso de Ciência da Computação não viabiliza algum tipo de automatização é a distribuição de professores, pois além de um número muito pequeno destes, não há muitas alternativas de mudanças de suas respectivas disciplinas.”

Quanto à distribuição de professores, a realidade do curso de Ciência da Computação segue a mesma da que foi apontada em 2013 por Sânya. Entretanto, cada professor tem sua própria gama de disciplinas que se dispõe a ministrar, e a coordenação tende a distribuí-los de acordo com sua preferência. Entretanto, como a demanda dos alunos não se mostra linear como foi estudado, é possível que a distribuição de professores seja feita de forma mais eficiente, considerando a demanda dos alunos, ainda que não se descartem suas preferências pessoais.

Sânya apresenta em seu trabalho uma série de requisitos essenciais e não essenciais para a resolução do problema. Porém, alguns deles não se mostram condizentes com a realidade da universidade. São eles:

- Requisitos essenciais, ou seja, obrigatórios:
 - **RE1** - Um professor não pode lecionar aula em duas turmas diferentes no mesmo horário.
 - **RE2** - Uma turma não pode ter aula em duas disciplinas no mesmo horário.
- Requisitos não essenciais, de qualidade:
 - **RNE1** - O ideal é que existam no máximo duas aulas consecutivas da mesma disciplina.
 - **RNE2** - Não devem haver mais de duas aulas da mesma disciplina em um dia.
 - **RNE3** - Não preencher os horários de 12h às 14h, pois se trata de horário de almoço.

- **RNE4** - Os professores associados, por terem exclusividade com a instituição, preferem espalhar os horários das aulas dadas, e não acumular todas no mesmo dia.
- **RNE5** - Os professores contratados, por outro lado, preferem que suas aulas sejam alocadas num mesmo dia, ou no menor número de dias possíveis.

Quanto à citada RE2, a limitação deveria ser mais criteriosa, e se tratando de um requisito não essencial, pois, o conceito de turma é dado pela junção de estudantes que cursam a mesma disciplina, ministrada por um mesmo professor, em um mesmo semestre. Mas em seu trabalho, Sânya considera o conceito de turma como sendo o conjunto de estudantes que ingressaram em um mesmo ano, independente da consideração da existência de repetentes e de suas escolhas pessoais de inscrição.

RNE1, RNE2 e RNE3: todas elas não consideram a existência de disciplinas que necessitam de um total de cinco tempos de aula semanais, sendo elas regularmente divididas em dois períodos, um de duas horas e outro de três horas. Que, em situações de necessidades, como é visto na entrevista com o diretor do CCT, acaba sim sendo necessário que se aloque em período de almoço.

RNE4 e RNE5: embora estejam direcionadas corretamente, ainda assim não engloba casos de preferência pessoal de cada um dos professores citados. Como por exemplo a possibilidade de não se ministrar aulas em determinados dias da semana por motivos religiosos, seja por parte do quadro permanente, quanto de professores associados.

Outra considerável divergência entre o modelo e a realidade é a definição de que a cada semestre contém apenas 5 turmas de computação. Sendo estas compostas pelos estudantes ingressantes de 5 anos consecutivos, caso este que não se aplica à realidade da universidade, visto que a quantidade de turmas varia de acordo com a demanda semestral, que não necessariamente condiz com todos os estudantes ingressantes de um mesmo ano.

Por fim, Sânya não considera a possibilidade de alocação de professores em mais de uma disciplina, o que é uma realidade na universidade, visto que alguns professores ministram aulas em mais de um curso, e em mais de uma disciplina.

2.4.3.2 Ricardo

Ricardo, em seu trabalho, apresenta alguns conceitos que se mostraram limitantes em sua modelagem conceitual do problema na UENF. Em sua monografia, ele aborda as turmas como sendo compostas por alunos que ingressaram no mesmo ano, independente de suas escolhas, da demanda, e de possíveis defasagens na progressão do curso. O que não se mostra condizente com a realidade da universidade atualmente, então, mesmo que em 2014, quando o trabalho foi realizado, a realidade fosse esta, não se mostra condizente com a realidade atual, havendo então a necessidade de uma reformulação conceitual.

Ele, assim com Sânya, também não considera a possibilidade de alocação de professores em mais de uma disciplina, múltiplas turmas de uma mesma disciplina, assim como não lida com a distribuição das salas. Ele também considera algumas disciplinas estão fixas, mesmo que pudessem sim ser modificadas por seu responsável, o Diretor do CCT.

Por fim, podemos citar um outro ponto que é a busca pela eficiência e baixo custo de tempo dos métodos utilizados. O que se mostra como ponto positivo, entretanto, mesmo com tamanha eficiência ainda peca em atingir a praticabilidade da solução.

2.4.3.3 Parecer geral das divergências

O entendimento das falhas passadas em se encontrar uma solução prática ao problema de Programação de Horários da UENF é de suma importância para que se possa evitar que os mesmos erros sejam cometidos novamente. Assim, é necessário que se tenha em mente que a realidade da universidade é dinâmica, e que a solução deve ser capaz de se adaptar a esta dinamicidade.

Vale-se também relembrar o que é dito por Tomáš Müller ([MURRAY; MüLLER; RUDOVÁ, 2007](#)) que cita em seu trabalho justamente sobre a complexidade presente na modelagem do problema.

3 Estrutura organizacional da instituição estudada

Para que se possa entender melhor o problema, é necessário que se entenda a estrutura organizacional da UENF disposta no [Estatuto da UENF](#). A Universidade Estadual do Norte Fluminense Darcy Ribeiro (UENF), ainda que limitando ao que convém neste trabalho.

3.1 A UENF e seu estatuto

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.

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 - [CCH](#);
2. Centro de Ciência e Tecnologia - [CCT](#);
3. Centro de Biociências e Biotecnologia - [CBB](#);
4. Centro de Ciências e Tecnologias Agropecuárias - [CCTA](#).

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

1. Laboratório de Meteorologia – [LAMET](#);
2. Laboratório de Ciências Físicas – [LCFIS](#);
3. Laboratório de Engenharia Civil – [LECIV](#);
4. Laboratório de Ciências Químicas – [LCQUI](#);
5. Laboratório de Materiais Avançados – [LAMAV](#);
6. Laboratório de Ciências Matemáticas – [LCMAT](#);
7. Laboratório de Engenharia de Produção – [LEPROD](#);
8. Laboratório de Engenharia e Exploração de Petróleo – [LENEP](#).

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 Matemática](#);
2. [Bacharelado em Ciência da Computação](#);
3. [Mestrado Profissional em Matemática – PROFMAT / SBM](#).

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.

3.2.1 Diretor 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 CCT](#). 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:

- **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:

- Os períodos ímpares são os piores;
- Essa opinião pode ser resultado do fato de que os períodos ímpares apresentam um intervalo de tempo para preparo das grades menor do que os períodos pares;
- As disciplinas básicas são grandes;
- É esperado que uma grande quantidade de alunos se inscreva nas 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 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 [RabbitMQ](#).

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 [GitLab](#), 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 Chefe 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 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 Coordenador 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; 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.

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 de pesquisa

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.

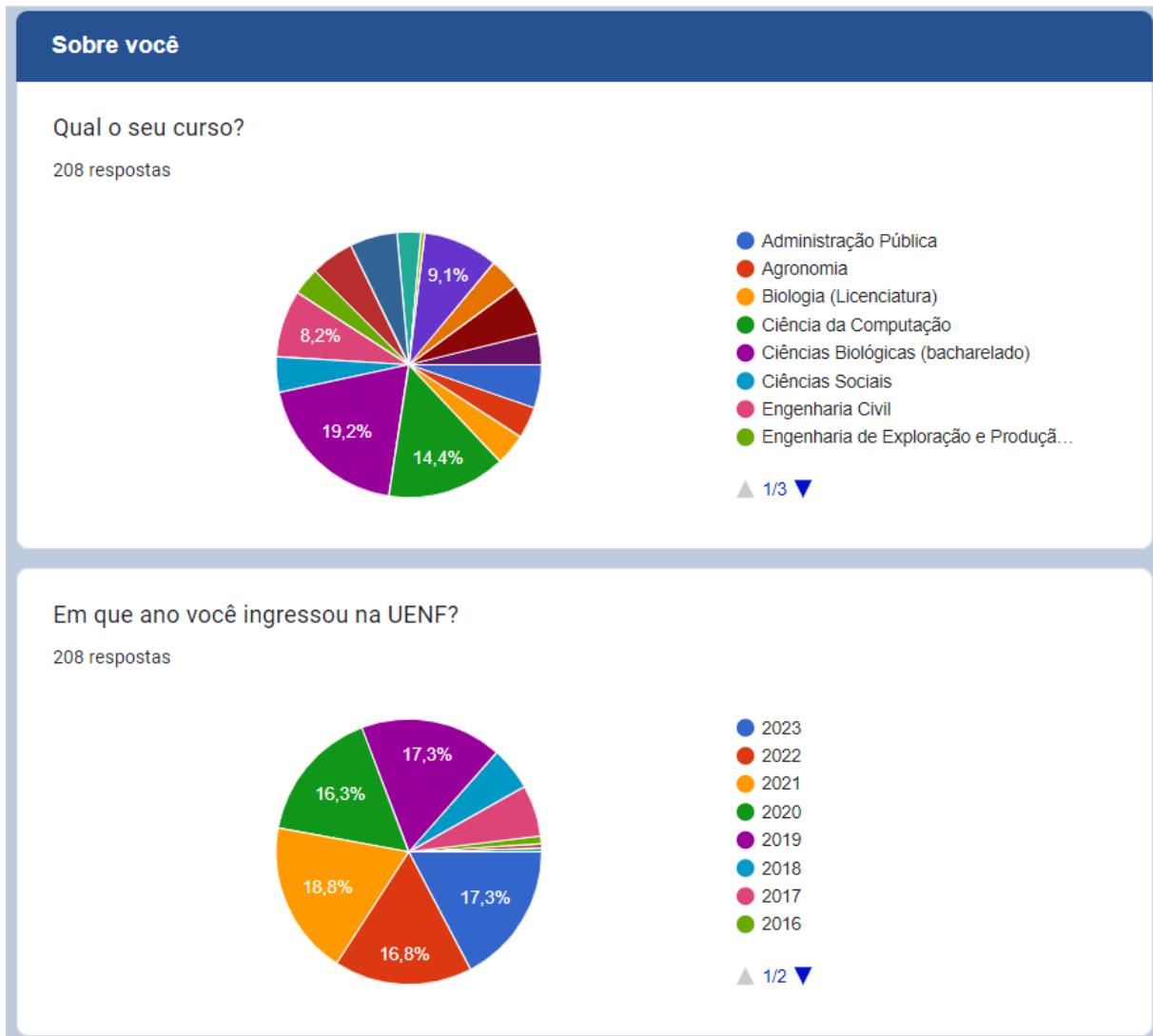
Para este fim, foi utilizado um formulário de pesquisa qualitativa dos alunos disponível no Apêndice 1. O formulário foi divulgado através de um link disponibilizado no grupo de alunos do curso de Ciência da Computação no WhatsApp, e também através de um link distribuído pela Secretaria Acadêmica a discentes da UENF.

A seguir, estão dispostos alguns resultados obtidos pelo formulário.

3.4.1 Respondentes

Inicialmente foi solicitadas algumas informações dos alunos, como seu curso, ano de ingresso. As respostas estão dispostas na [Figura 4](#).

Figura 4 – Perguntas sobre o curso e ano de ingresso dos estudantes



Fonte: autoria própria

O formulário foi respondido por 208 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 ([Tabela 1](#)). 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, 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 ([Tabela 2](#)).

Tabela 1 – Número de respondentes por curso

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

Tabela 2 – Número de respondentes por ano

Quantidade	Ano
36	2023
35	2022
39	2021
34	2020
35	2019
11	2018
13	2017
2	2016
1	2015
0	2014
0	2013
1	Outro

3.4.2 Avaliação de experiência acadêmica

Considerando que o escopo deste trabalho revolve em torno da alocação de recursos físicos e humanos, como salas, professores e alunos, foi elaborada uma seção do formulário de pesquisa com o intuito de se analisar a frequência de ocorrência de certas situações no contexto universitário. Para isto, foram feitas as seguintes perguntas:

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 3](#).

Tabela 3 – Ocorrência de experiências acadêmicas

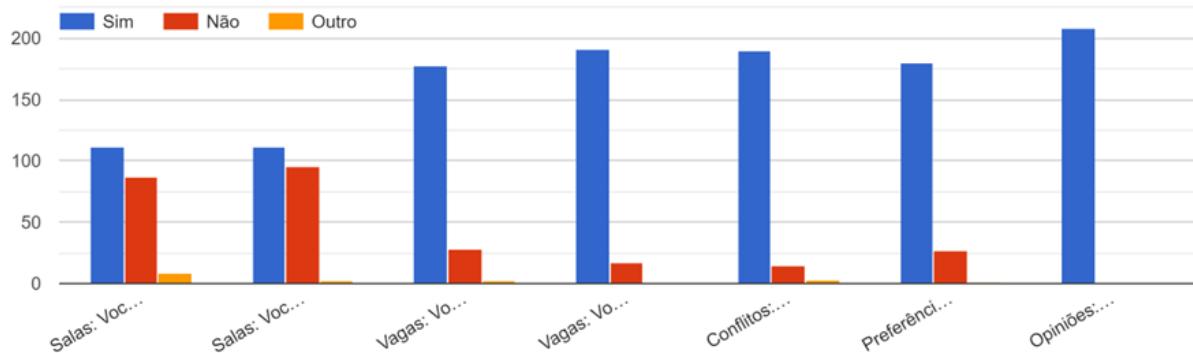
Pergunta	Respostas		
	Sim	Não	Outro
1	110	87	10
2	109	95	3
3	176	28	3
4	189	17	1
5	188	15	4
6	178	27	2
7	207	0	0

As respostas respectivas às perguntas se encontram na [Figura 5](#) onde estão dispostos os resultados encontrados nesta seção.

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.

Figura 5 – Respostas sobre a satisfação dos estudantes

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



Fonte: autoria própria

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.

3.4.3 Preferências pessoais

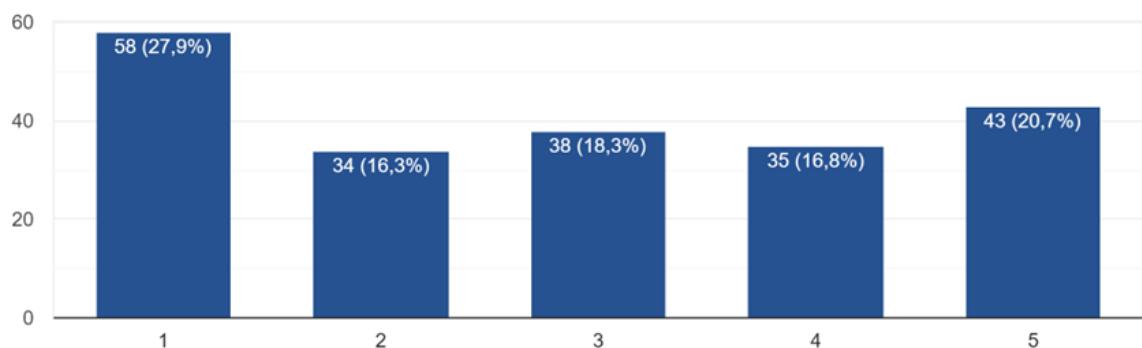
Neste segmento, visa-se entender um pouco melhor o processo decisório dos alunos quanto à escolha das disciplinas que desejam cursar. Primeiro, lhes é indagado quanto à disposição das disciplinas, variando entre disciplinas concentradas em poucos dias ou espalhadas durante a semana ([Figura 6](#)) e quanto à preferência de horários, variando entre horários matutinos e vespertinos ([Figura 7](#)).

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.

Figura 6 – Preferências por distribuição de disciplinas ao longo da semana

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

208 respostas



Fonte: autoria própria

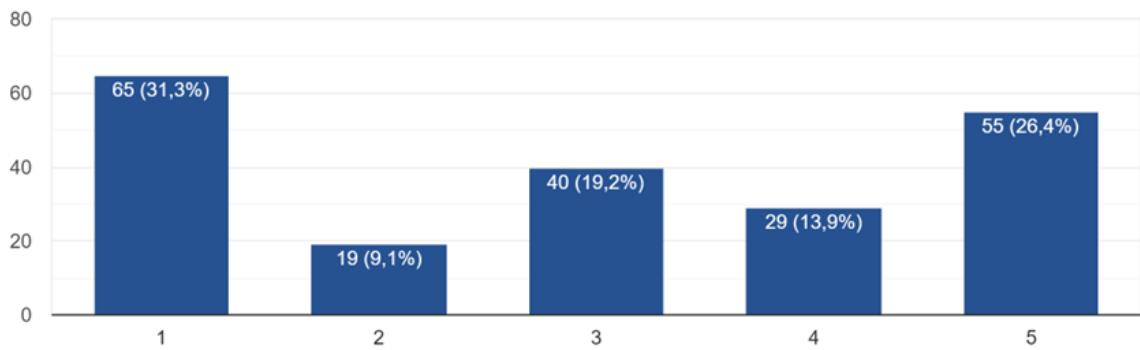
Podemos ver na [Figura 6](#) que há uma grande distribuição entre as preferências dos alunos, tendendo às extremidades, onde alguns preferem bastante as disciplinas distribuídas ao longo da semana enquanto outros preferem disciplinas acumuladas em poucos dias, vindo como terceira opção mais votada a neutralidade. Observação similar se mostra presente também na [Figura 7](#).

Como alternativa de visualização, dispomos aqui da [Tabela 4](#) e da [Tabela 5](#) com os resultados numéricos.

Em seguida, é questionado sobre qual é o critério de seleção de disciplinas que se apresentam conflituosas ([Figura 8](#)). Nesta vertente vemos uma maior propensão às

Figura 7 – Preferências por distribuição de disciplinas em um mesmo dia

Você prefere disciplinas na parte da manhã ou na parte da tarde?
208 respostas



Fonte: autoria própria

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

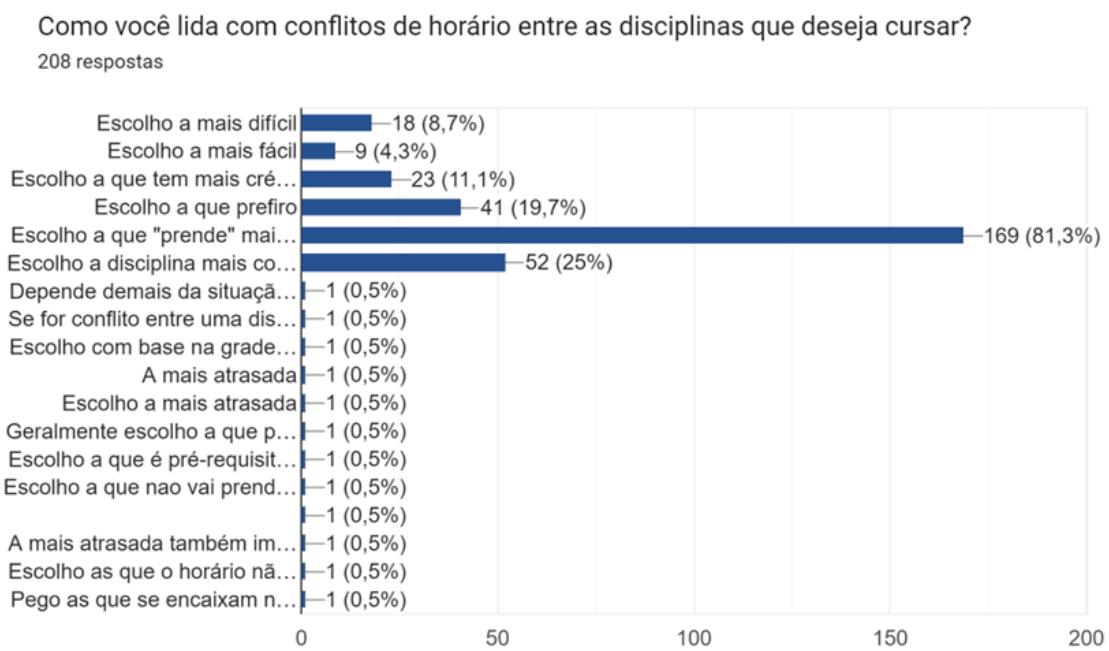
Quantidade	Distribuição na semana
58	Distribuídas ao longo da semana
34	~
38	Não tenho preferência
35	~
42	Acumuladas em poucos dias

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

Quantidade	Distribuição no dia
65	Na parte da manhã
19	~
40	Não tenho preferência
28	~
55	Na parte da tarde

disciplinas que é 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 81,3% 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% dos alunos. Os valores exatos podem ser vistos na [Tabela 6](#).

Figura 8 – Critérios para a escolha de disciplinas conflituosas



Fonte: autoria própria

Vale ressaltar que as respostas ilustradas pela [Figura 8](#) 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.

3.4.4 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

Tabela 6 – Critérios para a escolha de disciplinas conflituosas

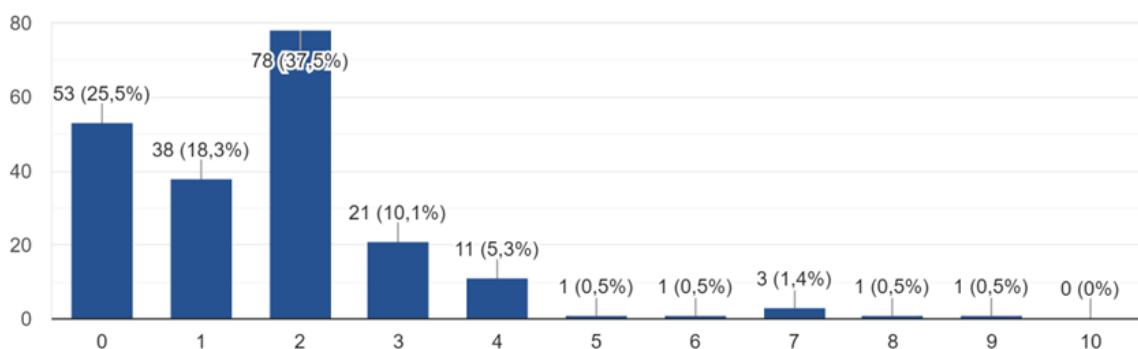
Quantidade	Forma de escolher disciplina conflituosa
169	Escolho a que “prende” mais matérias
52	Escolho a disciplina mais concorrida
41	Escolho a que prefiro
23	Escolho a que tem mais créditos
18	Escolho a mais difícil
11	Outro
9	Escolho a mais fácil

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 [Figura 5](#). Esta característica se confirma na percepção da frequência e distância que percebemos dos atrasos, representado pela [Figura 9](#) e pela [Figura 10](#).

Figura 9 – Períodos de atraso por espera

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



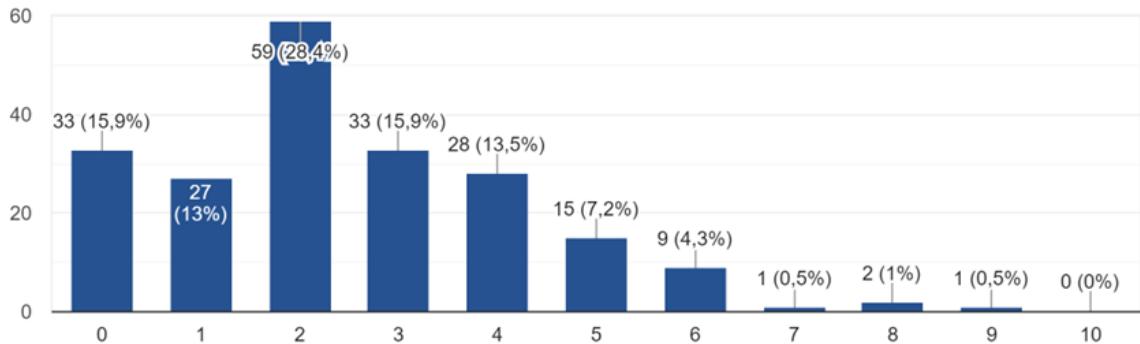
Fonte: autoria própria

Apresenta-se notável que é minoria a quantidade de alunos que nunca tiveram que esperar para cursar uma disciplina, sendo estes apenas 25,5% dos respondentes. É ainda mais notável o fato de que o tempo de espera médio é de mais de um ano e meio. 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

Figura 10 – Distância de atraso

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

208 respostas



Fonte: autoria própria

disciplinas esperadas para o período, sendo 2.8 anos o tempo médio de distanciamento.

Abaixo, encontra-se disposto na [Tabela 7](#) os resultados obtidos através desta seção do formulário.

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 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

3.4.5 Pesquisa de opinião

Aqui, buscamos 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”, “Variada”, “Contínua”, “Eficiente”, “Distribuída” e “Satisfatória”. Estando as opiniões dos alunos quanto às três primeiras

refletidas nos resultados expostos pela [Figura 11](#) enquanto que as três últimas são ilustradas pela [Figura 12](#).

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.

Como forma tabular, temos, na [Tabela 8](#), as respostas dos alunos, seguidas das médias por característica.

Tabela 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

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.

3.4.6 Respostas qualitativas

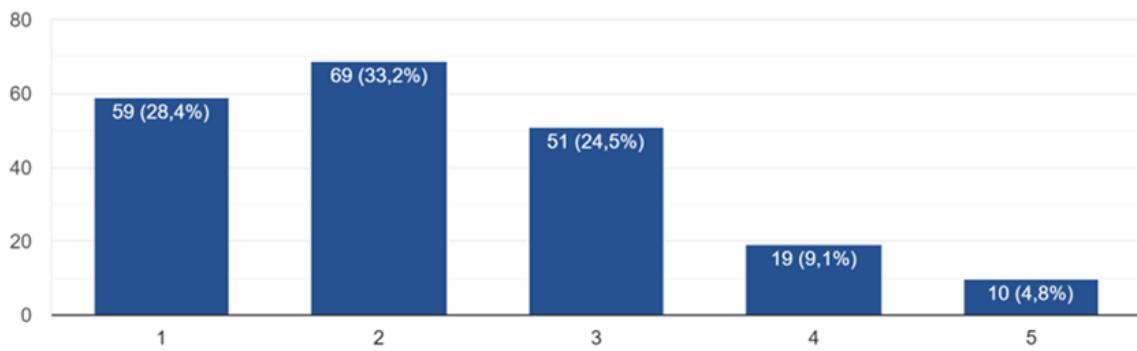
Por fim, havia um espaço livre no formulário para que os alunos pudessem expressar suas opiniões de forma mais livre. Após a leitura de todas e a filtragem das opiniões expressas, resumem-se em 4 parabenizações pelo desenvolvimento do presente projeto, 18 reclamações e 16 sugestões. Dentre elas, algumas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

Figura 11 – Notas dadas para as características “Justa”, “Variada” e “Contínua”

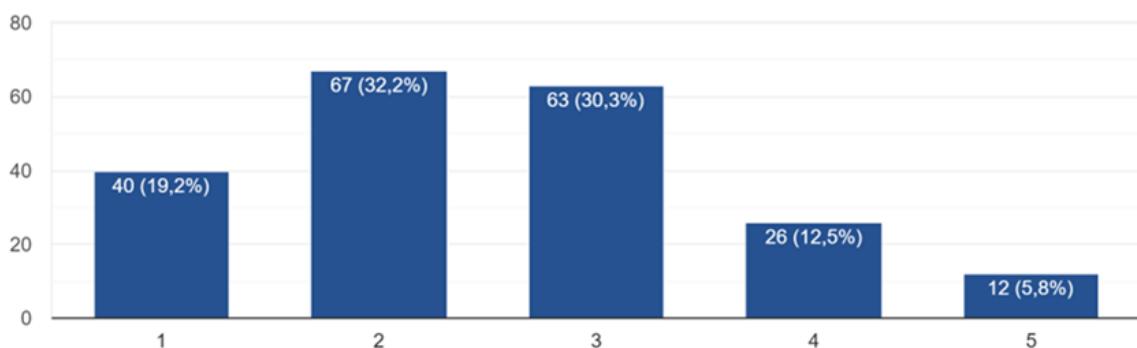
Justa

208 respostas



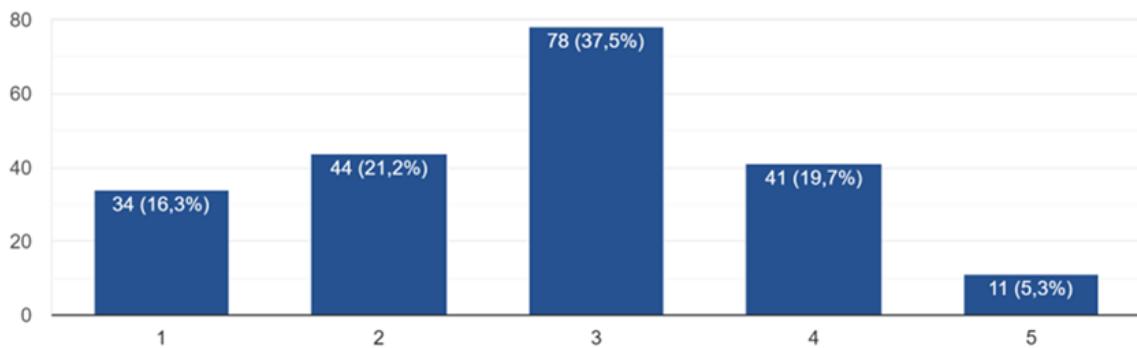
Variada

208 respostas



Contínua

208 respostas

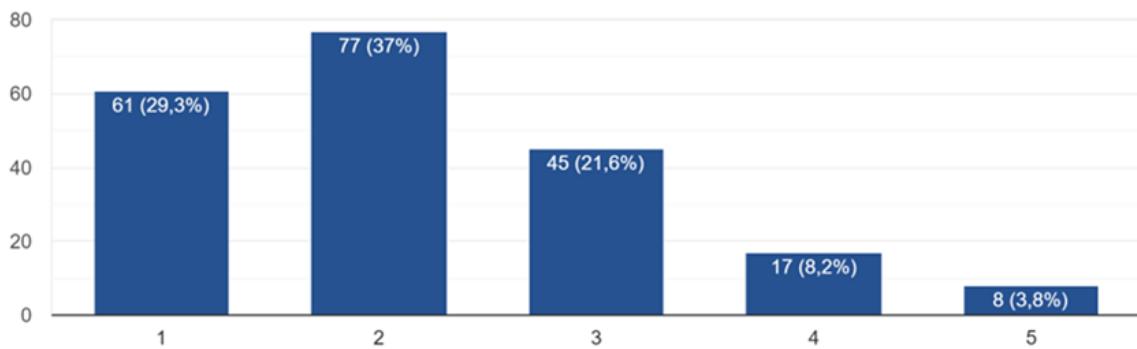


Fonte: autoria própria

Figura 12 – Notas dadas para as características “Eficiente”, “Distribuída” e “Satisfatória”

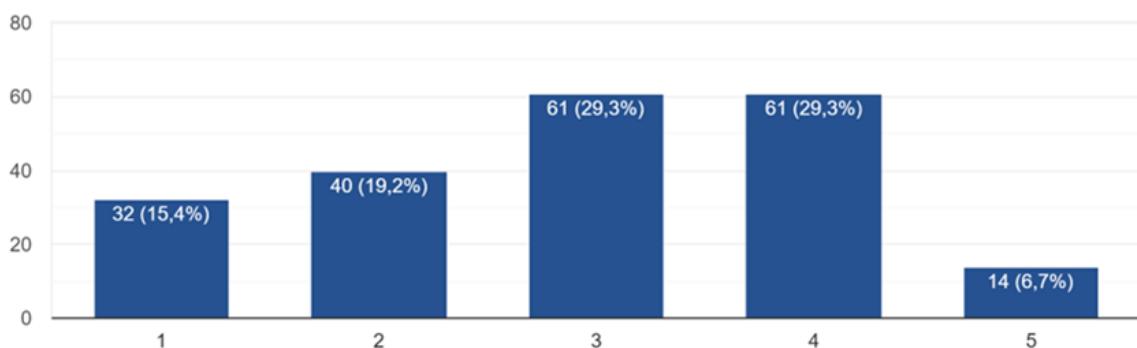
Eficiente

208 respostas



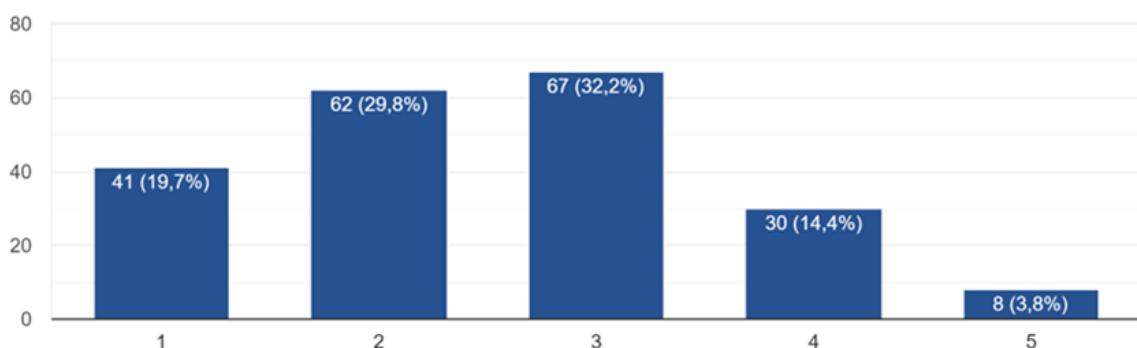
Distribuída

208 respostas



Satisfatória

208 respostas



Fonte: autoria própria

- 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. Reduzir o número de créditos necessários para antecipar a realização do estágio também se mostraria benéfico, visto que atualmente, a conclusão dessa matéria com apenas 9 créditos pendentes inviabiliza a possibilidade de antecipação antes do 9º período. Considera-se viável permitir essa antecipação a partir do 7º período.

[...] 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.”

3.4.7 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.

3.5 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 13](#) e [Figura 14](#).

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 13](#)) aloca uma turma a uma determinada sala, outra seção da mesma instituição (o Centro de Ciência e Tecnologia, ilustrado pela [Figura 14](#)) 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.

3.6 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.

3.6.1 Turmas

As turmas são a base da grade horária, sendo ela a representação da união entre professor, disciplina, alunos, horário e sala. A turma é a representação da oferta de uma disciplina em um determinado horário e local, com um determinado professor e para um determinado grupo de alunos. A turma é a unidade básica da grade horária, e é a partir dela que a grade é construída.

Figura 13 – 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 14 – Disciplina não atribuída à determinada hora e local na grade de horários do CCT

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		Ética Profissional Profa. Maria Alves				
11:00 - 12:00						
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		Metodologia Cient. para Eng. De Produção Prof. Molina	Física II Prof. Everton			
16:00 - 17:00	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

3.6.2 Laboratórios

Os laboratórios são estruturas organizacionais que englobam professores, disciplinas e cursos. 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.

3.6.3 Professores**3.6.4 Centros****3.6.5 Salas****3.6.6 Cursos****3.6.7 Alunos****3.6.8 Disciplinas**

Um dos pontos principais na criação de uma grade horária é a alocação das disciplinas. As disciplinas são a base da grade horária,

3.6.9 Blocos**3.6.10 Pactos Sociais**

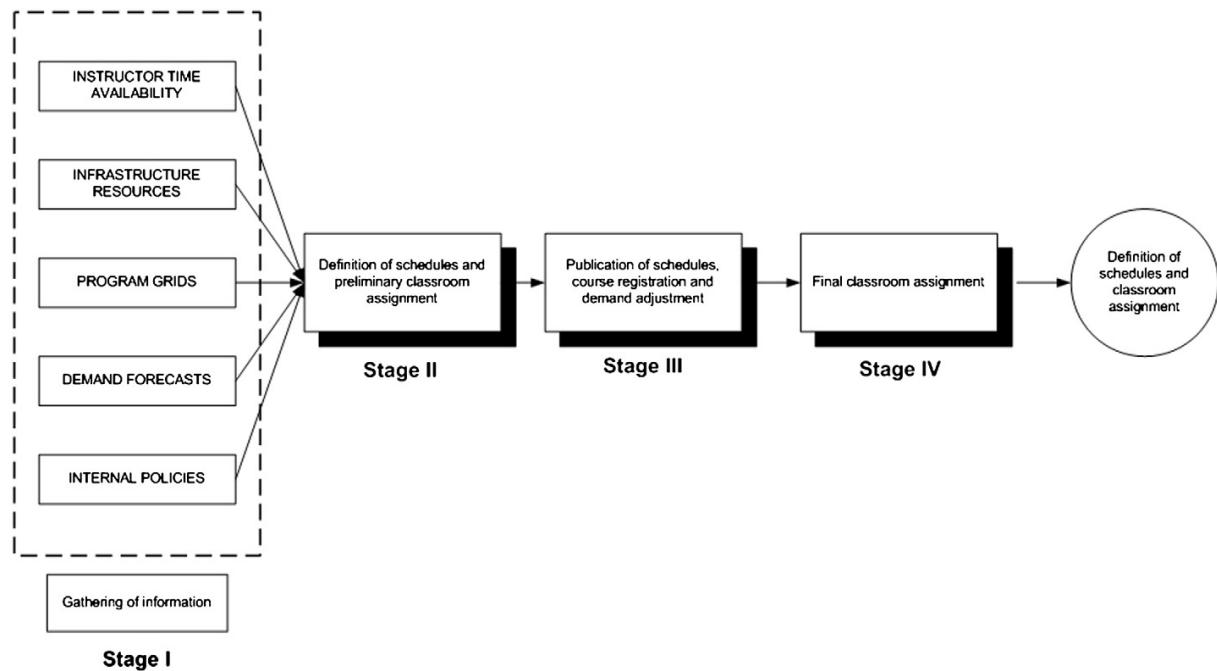
4 Modelagem geral do sistema

Tendo esclarecido sobre as questões gerais do trabalho e da área de estudo. Agora nos aprofundaremos um pouco mais na modelagem e criação de diagramas que ilustrem o funcionamento geral do sistema e a forma como se dará a execução da metodologia proposta.

4.1 Estágios de execução

Em seu trabalho de aplicação prática, ([MIRANDA; REY; ROBLES, 2012](#)) estruturou estágios que compõem o processo necessário para que enfim se alcance a definição de *timetables* ótimas.

Figura 15 – Estágios para a obtenção de grade horária ótima



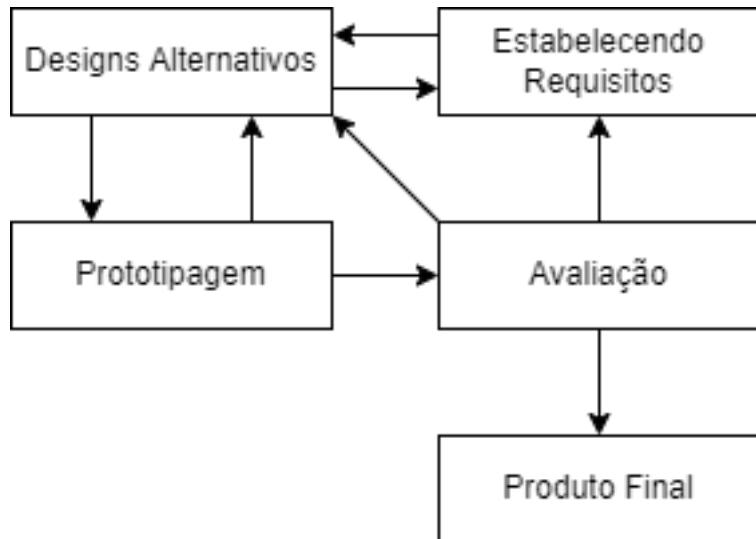
Fonte: ([MIRANDA; REY; ROBLES, 2012](#))

Na [Figura 15](#), estão dispostos 4 estágios principais. O primeiro dispõe da aquisição de informações. O meio de aquisição não é relevante para o momento atual, apenas considera-se que esta informação será obtida. No segundo estágios são definidas grades horárias preliminares para se atribuir os alunos. No terceiro, os alunos se inscrevem e a demanda é ajustada, por fim, no quarto estágio, ocorre a alocação final das salas.

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; DINATA, 2018).

Figura 16 – Etapas do Design de Interação



Fonte: autoria própria

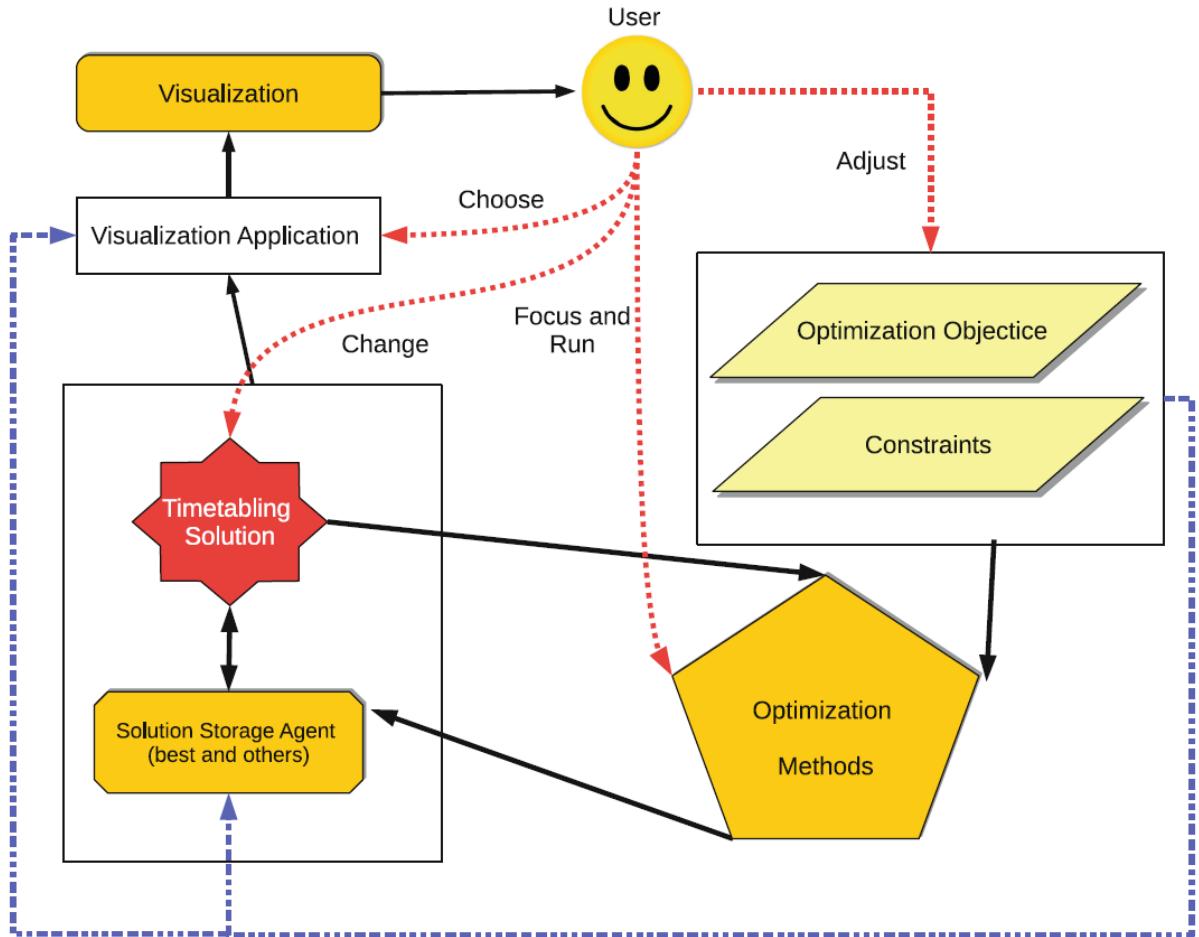
Seguindo o conceito do Design de Interação, a Figura 16 ilustra o ciclo de ações a serem tomadas durante o desenvolvimento do sistema, caso este venha a ser necessário. Neste modelo de pesquisa, os *stakeholders* serão consultados continuamente enquanto lhes é apresentado protótipos do sistema, para 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 finais.

4.3 Funcionamento

O sistema final seguirá uma dinâmica similar à que foi ilustrada por (ALENCAR et al., 2019b) em seu trabalho sobre o uso da Visualização de Informações em relação às Ed-TTPs.

A Figura 17 apresenta o comportamento geral do sistema, como seus diferentes segmentos interagem entre si e de que forma o usuário interage com o mesmo. O usuário poderá ajustar os objetivos da otimização e suas restrições, elas serão utilizadas nos métodos de otimização. Estes métodos serão utilizados para se alcançar soluções para estes critérios, as melhores serão então armazenadas. Em posso destes dados, a aplicação apresentará visualmente estas informações ao usuário, permitindo que ele interaja dinamicamente a fim de alcançar seus objetivos.

Figura 17 – Funcionamento geral do sistema

Fonte: ([ALENCAR et al., 2019b](#))

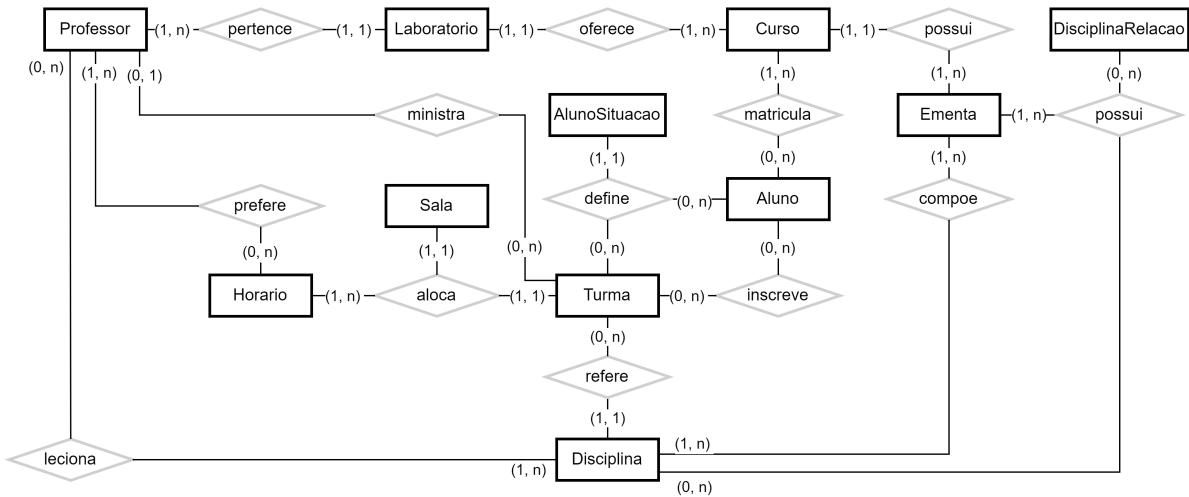
4.4 Modelo de Banco de Dados

Considerando as informações necessárias para o presente trabalho, e também o preparo de campo para potenciais aplicações futuras, foi elaborado um diagrama conceitual de banco de dados, que pode ser visto na [Figura 18](#).

O diagrama conceitual foi elaborado utilizando a ferramenta [draw.io](#) 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, como a relação entre professores e laboratórios, e a de disciplinas e ementas.

Figura 18 – Diagrama Conceitual do banco de dados



Fonte: autoria própria

Embora o diagrama apresente uma visão mais completa de todas as 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 uma parte do diagrama conceitual.

Neste modelo, mais enxuto, temos apenas as entidades principais, onde temos uma turma de determinada disciplina, ministrada por um professor e que ocorre em uma sala em um determinado horário.

4.4.1 Diagrama de Entidade e relacionamento

Neste diagrama vemos as entidades principais, que são *Turmas*, *Disciplinas*, *Professores*, *Horários* e *Salas*. As propriedades escolhidas para cada entidade são compostas por uma mistura de critérios. Por exemplo, o nome do professor, o código da disciplina, e a junção de código e bloco auxiliam primordialmente na identificação real dos professores, disciplinas e salas. Já as informações “período”, “apelido” e “comment”...

E também é notável a presença da entidade *Alunos*, que se apresenta desacoplado das demais entidades. O motivo para isso é que, embora os alunos façam parte do processo de alocação de turmas, ao longo do desenvolvimento, o desenvolvimento de funcionalidades envolvendo os alunos...

5 Desenvolvimento

Para o desenvolvimento do presente trabalho, foram utilizadas diversas ferramentas, tanto para a elaboração do código, quanto para a elaboração do modelo de banco de dados e para a elaboração do protótipo. Todas elas com o intuito de implementar o software necessário para a criação de uma grade horária.

5.1 Projetos anteriores

Antes do desenvolvimento do presente trabalho, foram feitos alguns projetos pessoais que, embora não tenham sido concluídos, serviram como base para o desenvolvimento do presente trabalho.

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. Infelizmente esse projeto não saiu do mundo das ideias. Entretanto, lá permaneceu sendo maturado.

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 realiza 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 [Python](#) que atualmente [se encontra no GitHub](#). 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.

```

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)
17 odt.output_to_txt(demands_and_values)
```

código 5.1 – Obter demanda por extratos em PDF

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 [PDFMiner](#), 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.

Com estes empecilhos, o código acabou abandonado, visto que apesar de útil, não era prático o suficiente para ser utilizado.

5.2 Dados pessoais e a LGPD

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 dos Dados \(LGPD\)](#), 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 acadêmicos](#), 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 mais manual de inserção de dados manualmente por parte dos usuários do sistema.

5.3 Prototipagem

A criação de protótipos, seguindo a abordagem tomada por ([ANDRE; 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 ou de não alcance das expectativas do projeto.

Para este fim, foram feitos os designs iniciais usando o software de design [Figma](#). Algumas telas principais foram concebidas. A primeira, e principal, é a ilustrada pela [Figura 19](#) que permite que o usuário arraste a turma até o horário desejado. A turma ao qual este horário se refere pode ser definida na parte lateral direita.

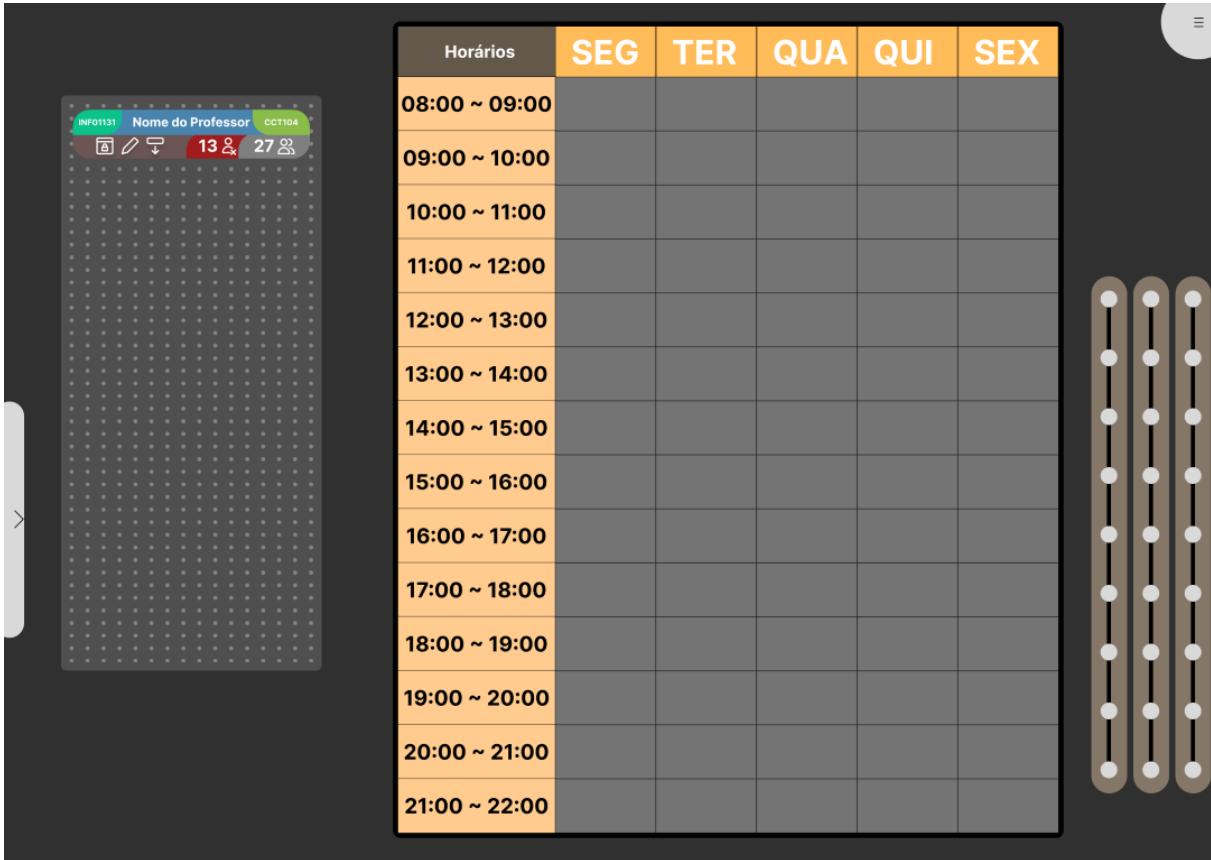
Em seguida, temos a tela base para seleção de dados que deseja-se modificar, ilustrada pela [Figura 20](#), podendo ser de turmas, salas, disciplinas, professores, alunos.

Cada um desses tendo sua própria página de criação, leitura, edição ou deleção de dados.

Quanto às salas, temos a sua alocação atual baseado no ano e semestre. Nessa página pode-se também registrar algumas características da sala, como a quantidade de cadeiras e computadores, e se possui monitor, projetos, quadro de giz e quadro branco. Um exemplo de sala ainda sem turmas alocadas é representado na [Figura 21](#).

[htbp]

Figura 19 – Página principal do sistema



The screenshot shows the main interface of a room booking system. On the left, there is a vertical list of rooms with icons and numbers (e.g., 13, 27). The central part features a large grid where rows represent rooms and columns represent time slots. The top row of the grid contains the days of the week: SEG, TER, QUA, QUI, and SEX. The first column contains the time intervals: 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, and 21:00 ~ 22:00. The grid cells are empty, indicating no bookings.

Fonte: autoria própria

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 22](#) mostra a grade de um aluno inscrito em 2019.1.

Podemos também definir nas disciplinas qual seu código e nome, e o seu período esperado segundo a ementa. Informamos 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 23](#) mostra a página de modificação de disciplinas.

Na seção de professores, temos a relação de disciplinas que os mesmos estão passíveis de ministrar, e também quais são suas preferências de horários ao longo da semana. A [Figura 24](#) mostra a página de modificação de professores.

Por fim, temos a junção de todas as informações registradas acima. Nela, podemos informar em quais horários, dias e em que sala, a turma estará alocada, além de informar qual professor a lecionará e a qual disciplina ela se refere.

Na imagem temos um exemplo das ilustrações de níveis de alerta, informando que

Figura 20 – Página principal de modificação dos dados



Fonte: autoria própria

o tempo de duração do segundo dia de aulas não condiz com a preferência pessoal do professor selecionado, e que na primeira sala estão ocorrendo conflitos. Conflitos esses ressaltados nos nomes dos alunos que demandam tal disciplina, como pode ser visto na [Figura 25](#).

Figura 21 – Página de modificação das informações de salas

Selecionar

Quantidade de cadeiras

Quantidade de Computadores

Código da sala

Sala 2 - P10 - 2º. Semestre 2023
Capacidade: 40 alunos

Horário	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA	SÁBADO
07:00 - 08:00						
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						

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

Ano

Semestre

Fonte: autoria própria

Figura 22 – Página de modificação das informações de alunos



Fonte: autoria própria

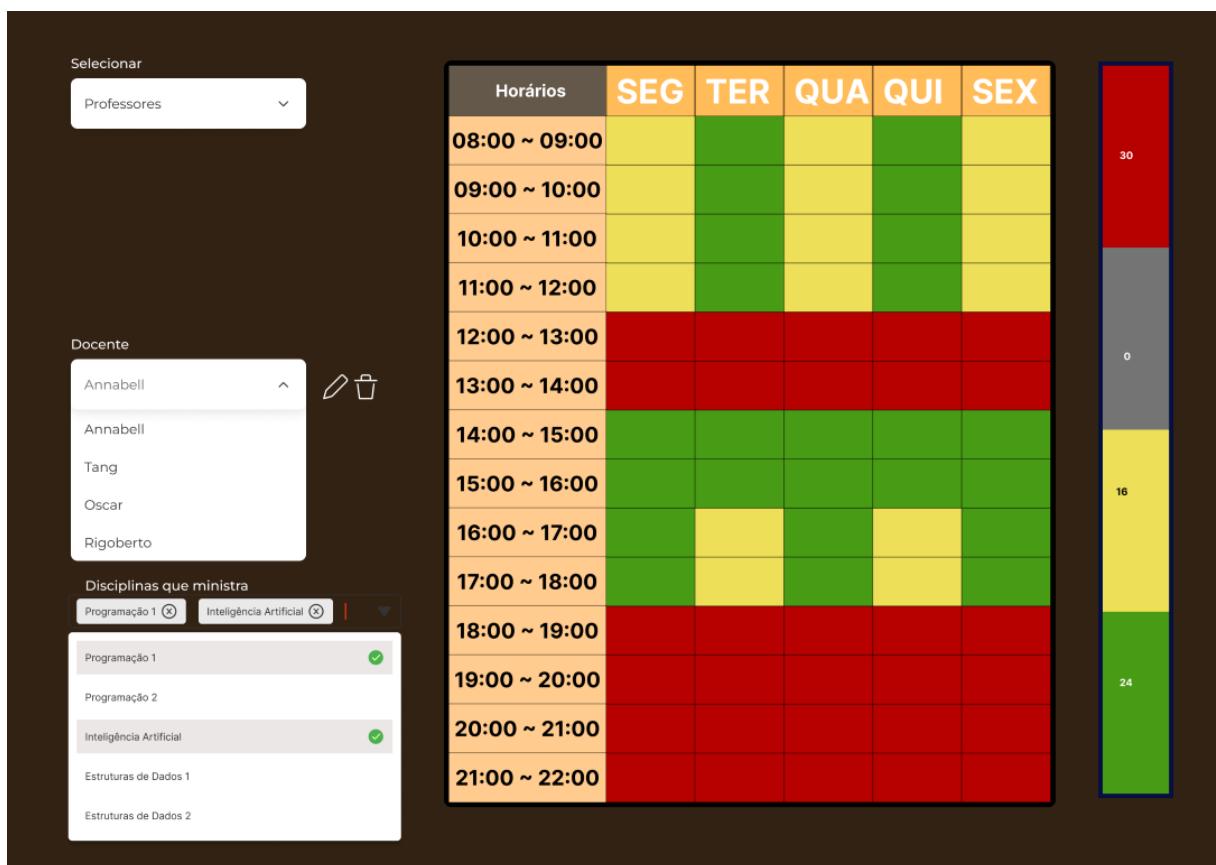
Figura 23 – Página de modificação das informações de disciplinas

The screenshot shows a dark-themed web application for managing academic disciplines. At the top left, there is a dropdown menu labeled "Selecionar" with "Disciplinas" selected. Below this, there are several input fields and dropdown menus:

- Código:** MAT01203 (with edit and delete icons)
- Nome da disciplina:** Cálculo II
- Período esperado:** 2
- Quais cursos:** CC (x), Química (x) | Ciência da Computação, Engenharia de Petróleo, Lic. Química, Licenciatura de Física, Licenciatura de Química
- Pré-requisitos:** Cálculo I (x) | Cálculo II, Cálculo III, Cálculo IV, Cálculo V
- Professores que ministram:** Oscar (x), Rigoberto (x) | Oscar, Elba, Rigoberto, Annabell, Tang
- Requisitos de sala:** Monitor (x) | Monitor, Giz, Projetor, Piloto

Fonte: autoria própria

Figura 24 – Página de modificação



Fonte: autoria própria

Figura 25 – Página de modificação das informações de turmas

The screenshot shows a user interface for managing class information. At the top, there are dropdown menus for 'Selecionar' (Turmas, Ano 2024, Semestre 1). Below this, a card displays 'MAT01203 Rigoberto' with icons for edit, delete, and student counts (4 men, 8 women). The main form fields include 'Nome da disciplina' (Cálculo 2) and 'Código' (MAT01203). Under 'Professor', 'Rigoberto' is selected. The 'Períodos de Horários' section contains two rows: one for 'SEG' (9:00-10:40) and another for 'QUI' (10:00-13:00). A red box highlights a list of student names and IDs:

	Nome	ID
8	(4) M	Abacate de Souza
00119110001	Bebêzinha Ferreira Silva	
00119110002	Bem-caminha da Braga Pinheiro	
00119110003	Damásco Figueiredo das Neves	
00119110004	Elefante Silva	
00119110005	Feliz Augusto Degel	
00119110006	Gato Menezes Oliveira	
00119110007	Helicóptero Dias Ribeiro	
00119110008		

Fonte: autoria própria

5.4 Programação do sistema

Após a elaboração dos protótipos com o Figma, e da conceitualização diagramática do banco de dados, 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 permitiria visualizar as mudanças no código em tempo real. Suas logos estão ilustrados na [Figura 26](#).

Figura 26 – Recursos usados para o desenvolvimento do *Frontend*

Fonte: autoria própria

5.4.1 Primeira Versão

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* para a navegação entre as páginas, e a biblioteca *React Select* para as caixas de seleção.

5.4.1.1 Banco de Dados Primitivo

Os dados contidos no sistema foram inicialmente armazenados em arquivos JSON, que eram importados diretamente para o código. 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 a possibilidade de leitura e mutação temporária, visto que após recarregar ou mudar de página, as mudanças eram perdidas.

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 agragar as informações.

5.4.1.2 Funcionalidades Iniciais

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 5.5](#) denominada **Conflitos**.

Além dessas funcionalidades que se mantiveram até a conclusão do sistema, também foram desenvolvidas funcionalidades que não obtiveram o mesmo êxito e que foram deixadas de lado ao decorrer do caminhar. Dentre elas, podemos citar a definição de níveis de preferência de horários para professores, a definição das características especiais das salas, e o andamento dos alunos em relação às disciplinas. 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.

5.4.1.3 GitHub Pages

Após o desenvolvimento local, como forma de viabilizar o acesso ao sistema por parte de outros usuários, foi feito o *deploy*, ou seja, foi feito o upload 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.

5.4.2 Segunda Versão

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. E ia além: dispunha de funcionalidades que não eram de todo necessárias, e, melhor dizendo, que tinham menor prioridade do que muitas outras.

Mesmo com esta nova dinâmica, outras funcionalidades foram deixadas de lado. Uma das que foram deixadas de lado foi a possibilidade de fixar certas informações. A proposta era que, certas disciplinas como os Cálculos e Álgebra Linear que são ofertadas para múltiplos cursos, pudessem ser fixadas em horários específicos, para que simplificasse aos coordenadores dos cursos a alocação de turmas.

5.4.2.1 GitHub Projects

Com isso, utilizando o GitHub Projects, foi organizado uma tabela de tarefas, vista na [Figura 27](#), 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 27 – Tabela de tarefas do GitHub Projects

The screenshot shows a GitHub Projects board titled 'Tarefas TCC'. The board has three columns: 'MVP', 'MVP3', and 'MVP2'. There are 18 tasks listed, each with a title, assignee (jvfd3), status (Done or Todo), labels (new Feature, quality of Life, bug), milestone (MVP 1 or Future), page (Classes, Database, System, Multiclasses, Students, Professors), and sequence (1 to 20). The interface includes a search bar, filters, and a 'New view' button.

Title	Assign...	Status	Labels	Milestone	Page	Sequence
1. Aprimorar visualização de turmas #1	jvfd3	Done	new Feature	MVP 1	Classes	1
2. Usar um banco de dados simples para armazenar as informações #2	jvfd3	Done	new Feature	MVP 1	Database	20
3. Definir paletas de cores #3		Todo	new Feature, quality of Life	Future	System	
4. Definir requisitos de características de salas das disciplinas #4		Todo	new Feature	Future: conflicts	Subjects	
5. Usar Schema do Banco de Dados do Daniel Brito #5	jvfd3	Done	abandoned	MVP 3	Database	
6. Mostrar a listagem exata de disciplinas demandadas #6		Todo	new Feature	Future: demand	Multiclasses	
7. Definir quais disciplinas cada aluno já fez. #7		Todo	new Feature	Future: demand	Students	
8. Visualizar as disciplinas passíveis de quebra de requisitos #8		Todo	new Feature	Future: demand	Students	
9. Ao adicionar disciplinas em "feitas", "cursando" ou "não feitas", atualizar as outras #9		Todo	new Feature	Future: demand	Students	
10. Cada item selecionável poderia ter seu próprio link #10		Todo	new Feature, quality of Life	Future	System	
11. Cada item em cada página deve ter sua própria página única #11		Todo	new Feature, quality of Life	Future	System	
12. Usar o Select Async para seleção de item #12	jvfd3	Todo	abandoned	Future	System	
13. Usar o JSON base como opções temporárias enquanto não carregam os dados async #13	jvfd3	Done	quality of Life	MVP 1	Database	
14. Reformular o Select para que o JSON que não esteja restrito a Value e Label #14	jvfd3	Done	new Feature, quality of Life	MVP 1	System	
15. Separar o arquivo options que são valores fixos #15	jvfd3	Done	quality of Life	MVP 1	System	1
16. Ajustar roteamento do GitHub Pages #16	jvfd3	Done	bug	MVP 1	System	
17. Padronizar o visual do código #17	jvfd3	Done	quality of Life	MVP 1	System	
18. Visualizar a quantidade de horas que o professor está ministrando #18		Todo	new Feature	Future: conflicts	Professors	

Fonte: autoria própria

Tendo este novo sistema de tarefas em prática, foi possível tranquilizar a mente quanto ao conflito entre 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*, *Pages* e *Sequence*. O *Status* reflete o andamento da tarefa, se ela está disponível, em andamento, ou concluída. O *Pages* reflete em qual página do sistema a tarefa se encontra, e o *Sequence* reflete a ordem de prioridade da tarefa.

5.4.2.2 Permanência 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 dos dados. Para exemplificar o funcionamento geral da permanência dos dados, consideremos o uso de uma *REST API* 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. Ele se encontra em duas formas: a primeira é o chamado “em produção”, que é o sistema

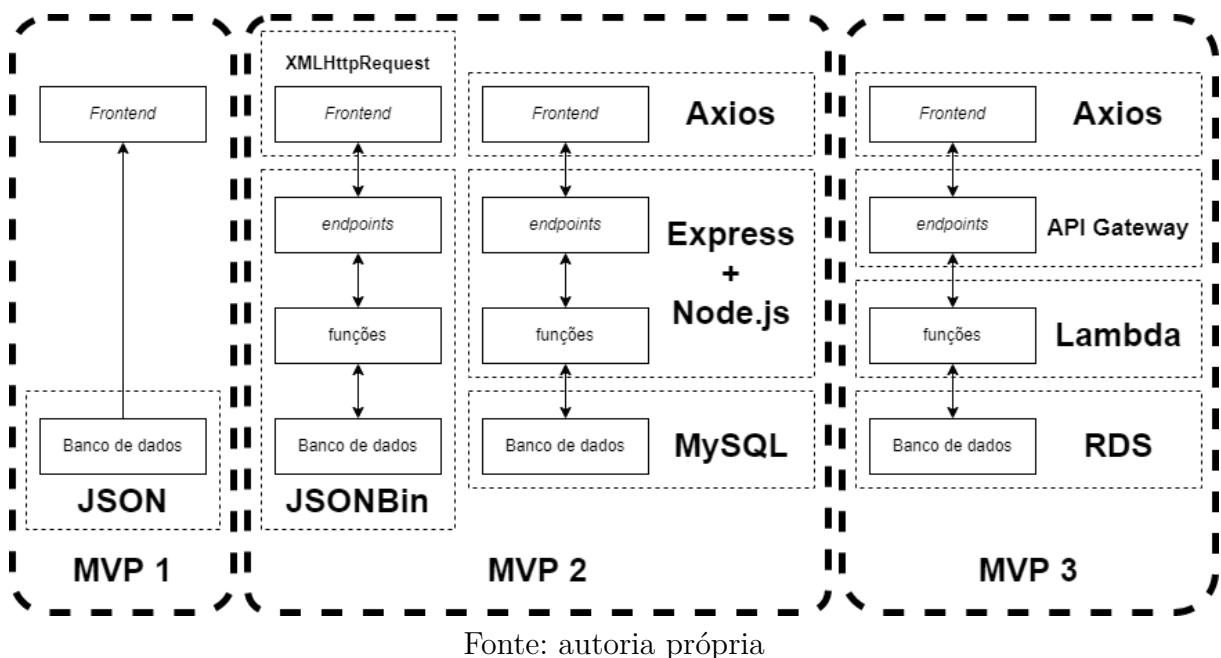
que o usuário final acessa, e a segunda é o chamado “em desenvolvimento”, que é o sistema que o desenvolvedor acessa para realizar as modificações necessárias. Ambas precisam se comunicar com o *backend* para realizar as quatro operações básicas no banco de dados (criação, leitura, atualização e deleção) 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. Se funcionamento é simples: 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.

Figura 28 – Diagrama da progressão funcionamento da permanência dos dados



Fonte: autoria própria

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*. Essa sequência é ilustrada na Figura 28.

JSONBin

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 com JSON, e o escolhido foi o JSONBin.

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.

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. Não se sabe se foi por inexperiência ou por limitações do próprio serviço, mas 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.

MySQL

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, sem haver uma explicação clara do porquê.

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 29](#), foram inicial-

mente criadas. Não seguindo objetivamente a modelagem anteriormente citada. Isso gerou 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 29 – Diagrama primitivo das tabelas de dados SQL

Disciplinas			Professores		
id	iddisciplina	PK	id	idprofessor	PK
int	periodoEsperado		string	laboratorio	
string	codigoDisciplina		string	curso	
string	nomeDisciplina		string	apelidoProfessor	
string	apelidoDisciplina		string	nomeProfessor	

Turmas			Salas		
id	idturma	PK	id	idsala	PK
int	Ano		string	bloco	
int	Semestre		string	codigoSala	
int	demandadeEstimada		string	descricaoBloco	
string	nomeProfessor		int	capacidade	
string	nomeDisciplina		string	blocoSala	

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.

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 *Axios* 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 *Express* para desenvolver um *backend* local executado paralelo ao *frontend*. Essa biblioteca é 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 rodas 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

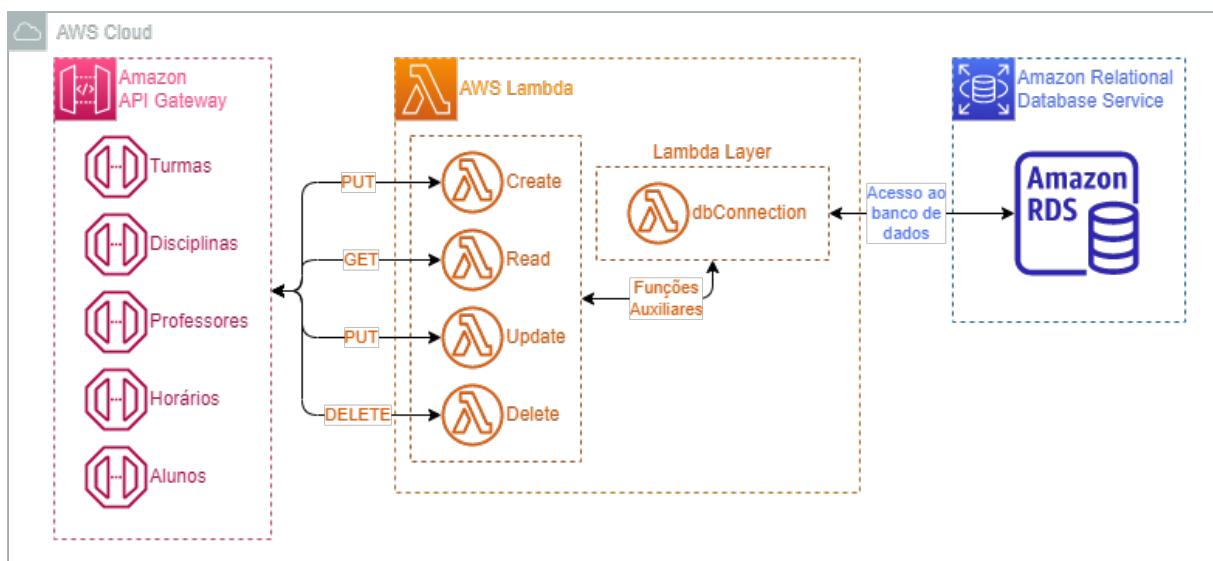
local, não sendo este o desejo final do sistema. Em segunda instância, para haver o acesso aos dados, é necessário que, além do banco de dados, o *backend* também esteja em execução, entretanto, o *GitHub Pages*, onde o sistema está hospedado, não viabiliza essa execução. Com isso viu-se necessária a busca por um novo serviço de hospedagem.

5.4.2.3 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.

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 30, 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.

Figura 30 – API REST no AWS



Fonte: autoria própria

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.

Tanto o API Gateway quanto as funções Lambda precisaram sofrer diversas modificações ao longo do desenvolvimento, e para que essas modificações fossem aplicadas. 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 do API Gateway, a ineficiência não era tão grande, visto que assim que as rotas estiverem configuradas, não há a necessidade de alterá-las. 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.

```
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
```

código 5.2 – Código de *deploy* de Lambda

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.

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

```

código 5.3 – Exemplo de *template.yaml*

Como mostrado no [código 5.3](#), 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.

5.4.2.4 Funcionalidades Adicionais

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 5.5](#) denominada **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 Terceira Versão

Considerando que a segunda versão já apresentava em sua maioria as funcionalidades mínimas desejadas, a terceira versão foi focada em melhorias de usabilidade e na correção de bugs.

5.4.3.1 Mudanças no *GitHub Projects*

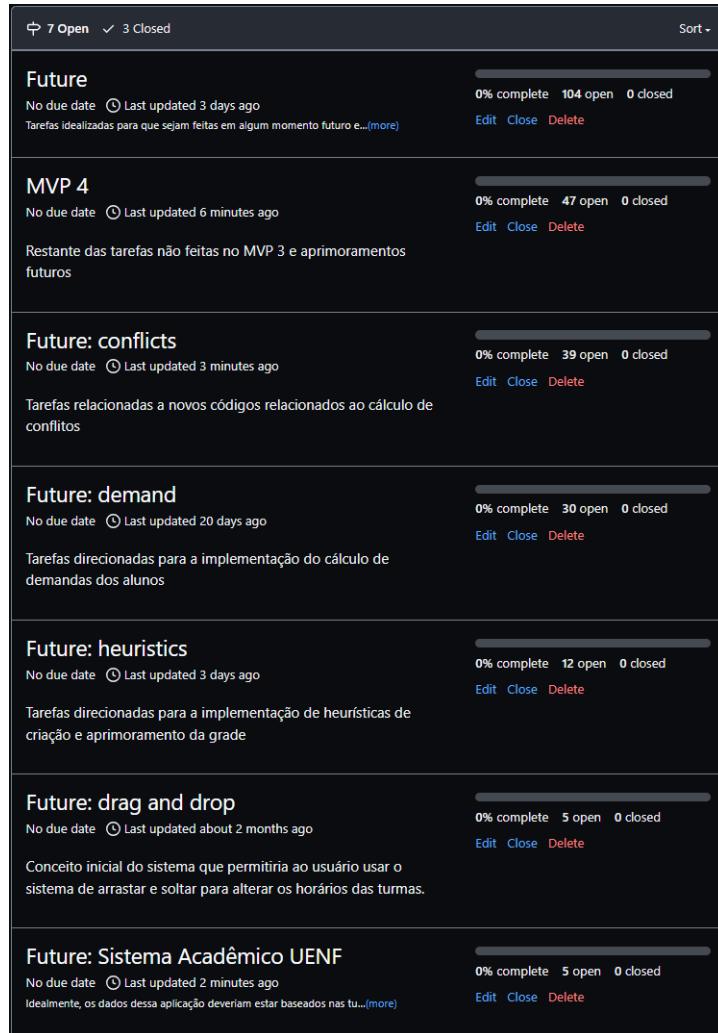
O uso do GitHub Projects se provou como uma excelente forma de organização das tarefas, e assim, foi mantido para a terceira versão. Alguns novos detalhes foram adicionados em sua organização, como a adição da categorização de tarefas por **marcos** (*milestones*, [Figura 31](#)) e **etiquetas** (*labels*, [Figura 32](#)).

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. Então podemos considerar que cada um desses **marcos** teve tarefas atribuídas a si pois...

1. **MVP 1:** foram concluídas na primeira versão;

Figura 31 – Marcos do GitHub Projects



Fonte: autoria própria

2. **MVP 2:** foram concluídas na segunda versão;
3. **MVP 3:** foram concluídas na terceira versão;
4. **Futuro:** foram planejadas para o futuro do sistema;
 - a) **Demandas:** visam calcular a demanda dos alunos por disciplinas;
 - b) **Conflitos:** visam aprimorar a visualização, qualidade e/ou variedade de conflitos;
 - c) **Heurísticas:** visam aprimorar a alocação de turmas através de heurísticas;
 - d) **Arrasta e solta:** visam aprimorar a alocação de turmas através de um sistema de arrastar e soltar;
 - e) **Integração com o Sistema Acadêmico:** visam a integração do atual sistema com o sistema acadêmico da UENF.

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

Figura 32 – Etiquetas do GitHub Projects

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

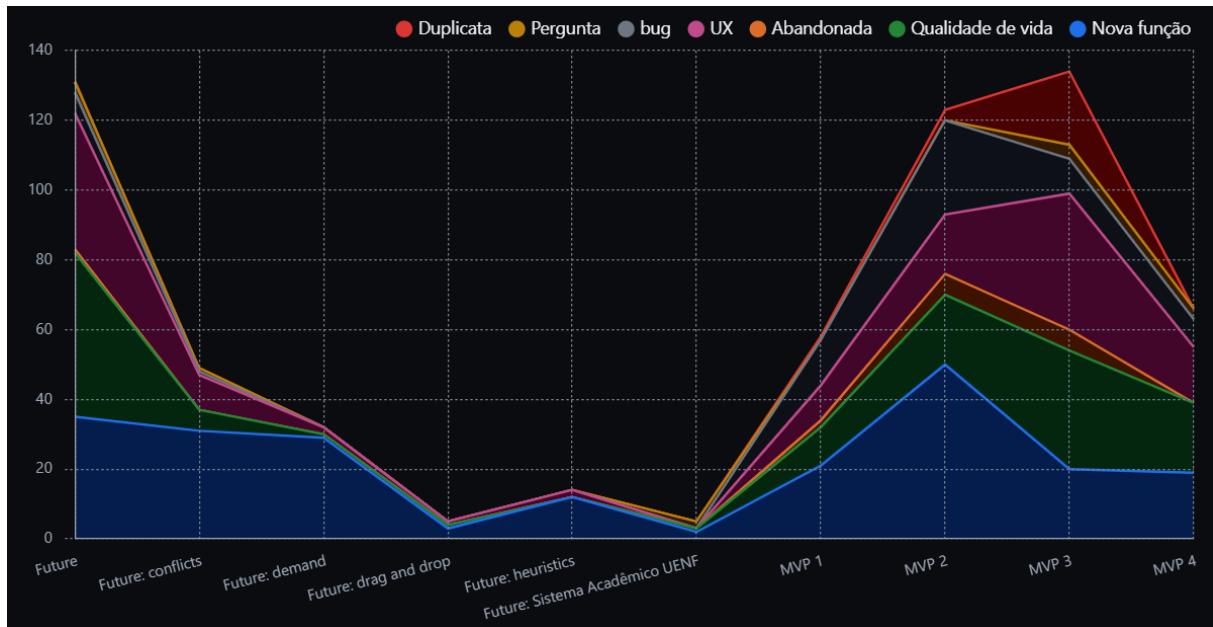
Fonte: autoria própria

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

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:

Figura 33 – Gráfico de Marco *versus* quantidade de tarefas separadas por etiqueta

Fonte: autoria própria

- Quantidade de tarefas de determinada etiqueta a cada marco, exemplificado na [Figura 33](#);
- Quantidade de tarefas por marco;
- Quais páginas receberam tarefas de quais etiquetas;
- Quantas são as tarefas em cada um de seus estados (Completas, em progresso, pendente).

5.4.3.2 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.

Aprimoramento dos identificadores dos conflitos

Os conflitos ocorridos indicavam quais eram os identificadores (ids) das turmas que estavam em conflito, porém, esses ids eram os ids 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.

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. Essa característica 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 se manter 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.

Solução inicial

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.

O cálculo da demanda estimada é dado pela média de demandas de todas as turmas anteriores que possuem demanda estimada. Entretanto, atualmente apresenta-se desativada.

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.

É esperado, entretanto, que esta solução apresente problemas em sua execução, visto que nem todas as alocações de turmas apresentam um padrão. Então, fica a cargo do usuário a verificação e a correção dos possíveis erros alertados pelo sistema.

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 inapropriadas 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.

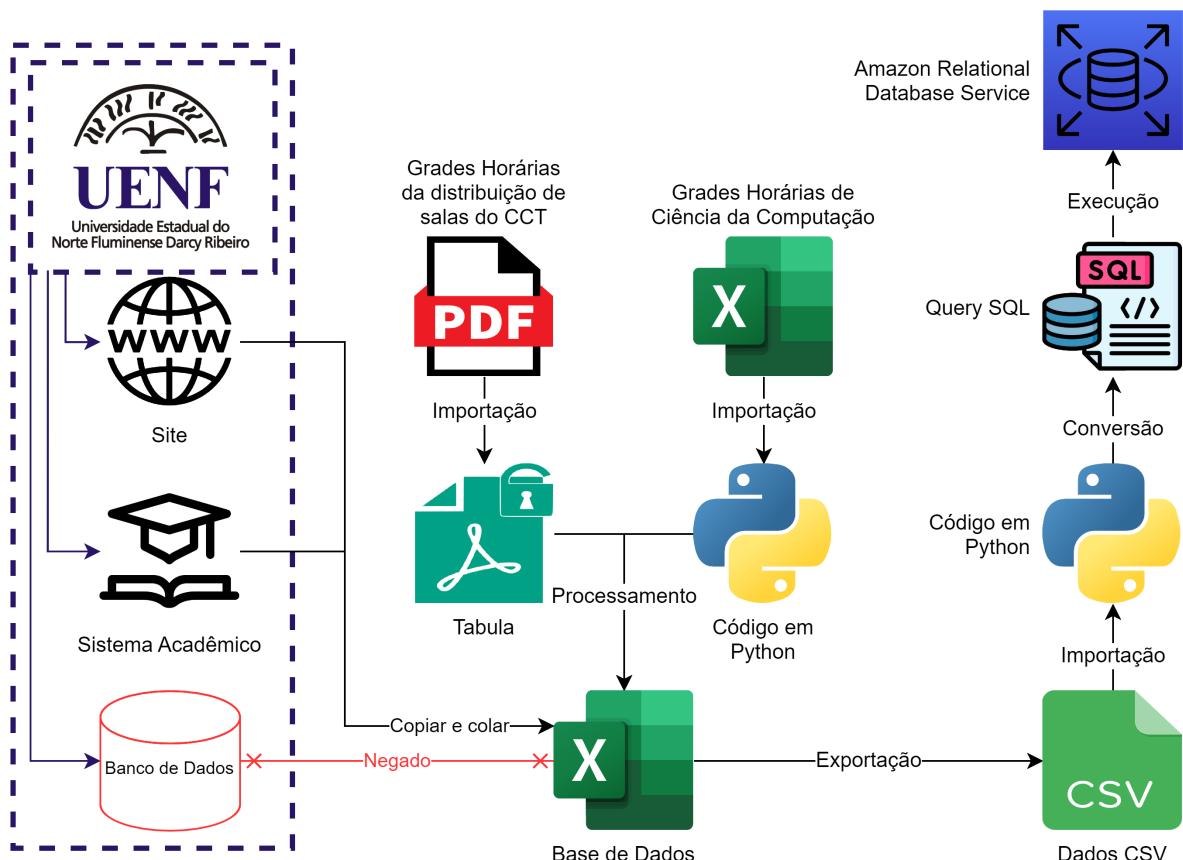
Preenchimento de dados

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 34.

Figura 34 – Diagrama do fluxo de obtenção de dados



Fonte: autoria própria

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 *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 Acadêmico](#), o [site da UENF](#) 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.

Gerais

Neste tópico estão descritas algumas melhorias gerais do sistema que não se enquadraram 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 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 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. A flexibilidade é necessária, visto que o processo de alocação é feito de forma manual. 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.

5.5.1 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 diminuto 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 revolvem 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.

5.5.2 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.

Os conflitos que são representados por cores, têm sua paleta de cores representada na [Figura 35](#). Nessa paleta, dispõe-se de 3 conjuntos principais: a distribuição de cores

Figura 35 – Paleta de cores do sistema

Período da disciplina	Semestre		
	Verão	Correto	Errado
0	#808080	#808080	#808080
1	#FFFF00	#00FF00	#FF0000
2	#EFEF00	#00EF00	#EF0000
3	#DFDF00	#00DF00	#DF0000
4	#CFCF00	#00CF00	#CF0000
5	#BFBF00	#00BF00	#BF0000
6	#AFAF00	#00AF00	#AF0000
7	#9F9F00	#009F00	#9F0000
8	#8F8F00	#008F00	#8F0000
9	#7F7F00	#007F00	#7F0000
10	#6F6F00	#006F00	#6F0000

Entidade	Conflito		
	Nulo	Sem	Com
Disciplina	#E0E0E0	#CCFF33	#8B0000
Professor	#D0D0D0	#47D902	#C72508
Demandas	#C0C0C0	#3EC200	#000000
Sala	#B0B0B0	#38B000	#DD3333
Dia	#A0A0A0	#008000	#E3580E
Hora	#909090	#007200	#C72508
Duração	#808080	#006400	#C72508

Categoria	Disciplina
Não Computação	#D66615
Sem Paridade	#808080
Eletiva optativa	#0055FF
Eletiva livre	#4800FF
Sem categoria	#EECOCO

Fonte: autoria própria

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”.

5.5.2.1 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 36 – Exemplo de conflito de alocação de professor

	Sala	Dia	Hora de início	Duração
Segunda	Sala v	8	x v	1 hora
Quarta	Sala v	8	x v	2 horas
Sexta	Sala v	8	x v	3 horas

	Sala	Dia	Hora de início	Duração
Segunda	Sala v	9	x v	2 horas
Quarta	Sala v	9	x v	2 horas
Sexta	Sala v	9	x v	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 36](#), 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.

5.5.2.2 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 37](#) representa um caso de conflito de alocação de salas.

Figura 37 – 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

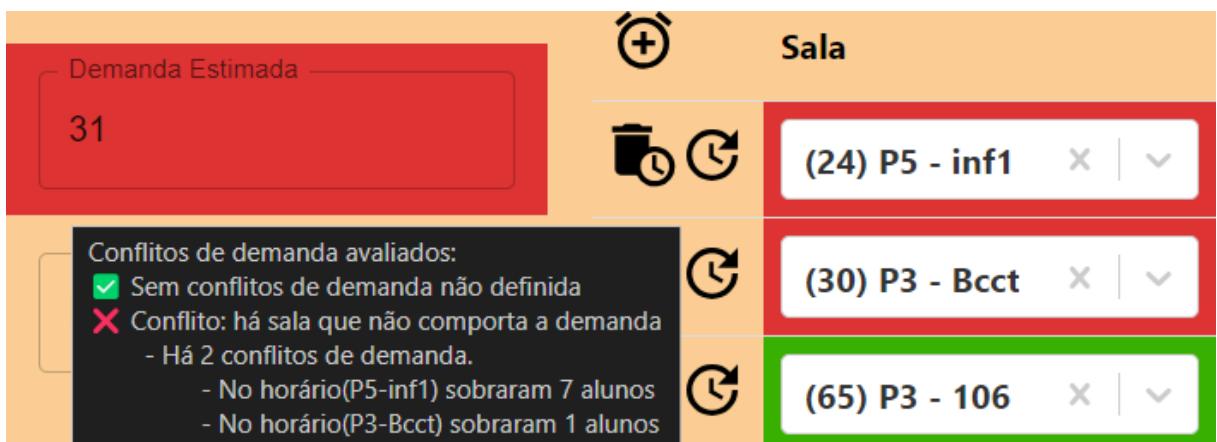
Fonte: autoria própria

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 38](#), a turma fictícia que 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 38 – Exemplo de conflito de capacidade na sala



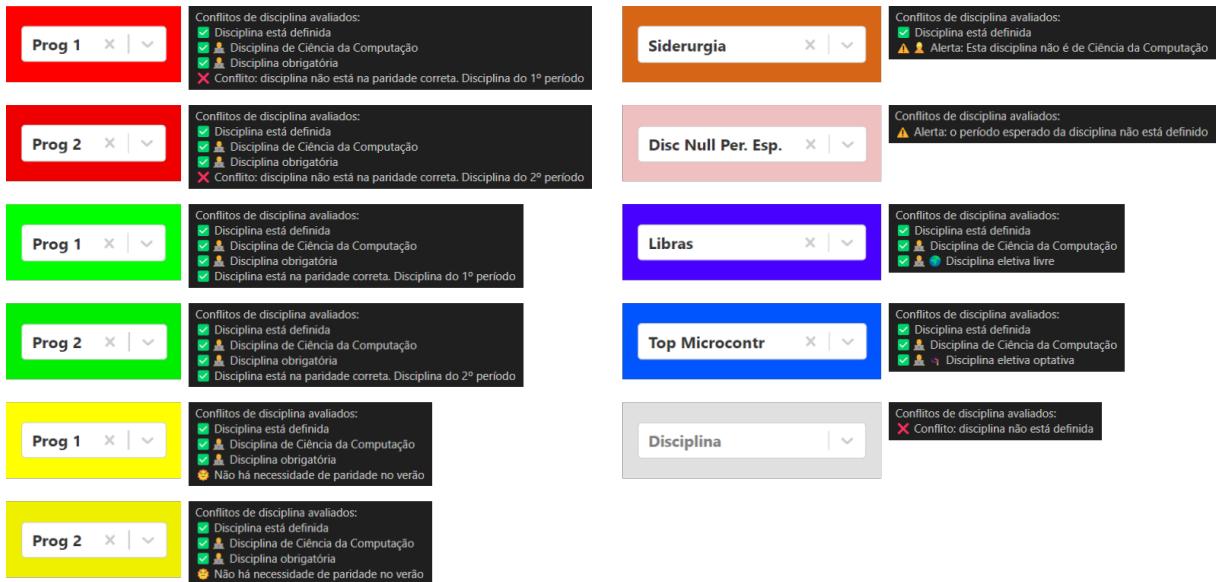
Fonte: autoria própria

5.5.2.3 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 39](#).

Já no lado esquerdo da [Figura 39](#), 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

Figura 39 – Avisos flutuantes dos conflitos de disciplinas



Fonte: autoria própria

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 semestres pares, são ilustradas com a cor avermelhada, seguindo a mesma lógica de gradiente escuro nos últimos períodos.

6 Experimentos

Como forma de testar o sistema desenvolvido, elaborou-se situações hipotéticas que se assemelhem às reais como forma de testar a eficiência do sistema. Para isso, utilizou-se como base os dados disponibilizados pelo coordenador do curso e pelo diretor do CCT. Em seguida, montou-se a estrutura de dados com a qual o sistema trabalhará analisando conflitos, e por fim, elaborou-se uma grade horária com base nos dados coletados.

6.1 Aquisição dos dados

Como o acesso aos dados reais é restrito, vê-se necessário o uso de alternativas para que se possa validar os casos de uso do software desenvolvido.

Para se manter o mais próximo possível da realidade dos dados, utilizou-se os seguintes métodos como base os dados:

- Disciplinas e requisitos: ementa de Ciência da Computação de 2015;
- Capacidades de salas e disciplinas ministradas por professores: pdf gerado pelo diretor do CCT quanto à oferta de salas;
- Alunos e progressão: dados disponibilizados pelo coordenador do Curso de computação;
- Preferências de alunos: formulário quantitativo.

Considerando as tabelas de dados necessárias para a criação de uma grade horária, utilizou-se do site [Mockaroo](#) para a geração de dados aleatórios que restam, sendo eles a preferência dos professores e as informações das disciplinas ministradas pelos professores não listados

Tendo os dados em mãos, resta então o uso prático do software para a alocação de turmas.

6.2 Cenários

Utilizando dos dados obtidos, elaborou-se então um cenário hipotético de criação de grade horária. Considerando assim a demanda de cada um dos alunos, a preferência de horários dos professores, a capacidade das salas e as disciplinas ministradas pelos professores.

Dispondo de todas essas informações e do software desenvolvido, foi possível então inicialmente distribuir as disciplinas em salas e horários segundo suas distribuições existentes nos semestres anteriores. Assim já podendo visualizar os conflitos que ocorreram neste período.

Com as alterações na tabela inicial de distribuição de disciplinas, foi possível obter uma grade horária com menos conflitos e que se aproxima mais da preferência dos professores.

7 Resultados

Como esperado, encontrar uma solução ótima para o problema de criação de grade horária tende à impraticabilidade, visto a dificuldade de se definir se de fato há tal solução a ser atingida. Mesmo assim, ao se utilizar de um método de manipulação manual dos dados, foi possível obter uma solução que se aproxima da ótima, e que pode ser utilizada na prática.

Em termos quantitativos, o sistema desenvolvido apresenta uma redução de conflitos em relação à solução inicial. O que mostra um resultado satisfatório.

Esse resultado, entretanto, não diz respeito à uma conclusão absoluta, visto que o sistema desenvolvido não foi testado em um ambiente real, e sim em um ambiente hipotético. Com isso, considerável parte de informações encontra-se faltante e foi substituída por dados aleatórios, o que pode ter influenciado no resultado final.

7.1 Soluções 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 a solução 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.

7.1.1 Tempo de elaboração das grades

Durante as entrevistas, 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 duas 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 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.

7.1.2 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 anuênciapor 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 anuênciada Coordenação de Curso e a ciência do Chefe do Laboratório responsável pela disciplina”.

Sendo assim possível alterar os horários de aula caso seja necessária para que haja uma melhora geral na distribuição das turmas na grade horária, mais uma vez sendo necessário superar o processo burocrático necessário.

8 Conclusões

O problema de organização de grade horária no ensino superior tem sido amplamente estudado por diversos pesquisadores. Devido sua natureza multidimensional e com forte tendência a especificidades, este campo de estudo se mostra como amplo e complexo.

Através da revisão bibliográfica, foi possível observar que a maioria dos trabalhos se foca em um método heurístico de solução, onde se busca uma solução ótima, ou próxima do ótimo, através de um método de busca. Entretanto, o presente trabalho se propõe a 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.

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, sendo elas a criação, leitura, edição e deleção de dados. 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 e problemas. O que consequentemente tende a agilizar o processo de busca por novas soluções e a redução dos conflitos.

8.1 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.

Assim como os modelos anteriores apresentaram diversas incongruências com a

realidade prática da universidade estudada, 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.

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.

Considerando que as duas tentativas anteriores resultaram em métodos que embora atingissem seu objetivo, não foram implementados na prática, tem-se como esperado que o mesmo ocorra com este trabalho. Com isso, espera-se que em trabalhos futuros se estude e analise 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.

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 semestralmente.

Referências

- ALEGRE, P. Desenvolvimento de um modelo para o school timetabling problem baseado na meta-heurística simulated annealing. *Simulated Annealing*, 2012. Citado na página 22.
- ALENCAR, W. 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: <<http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0007375802750281>>. Citado 5 vezes nas páginas 20, 21, 22, 23 e 24.
- ALENCAR, W. D. S. et al. Information visualization for highlighting conflicts in educational timetabling problems. In: BEBIS, G. et al. (Ed.). *Advances in Visual Computing*. Cham: Springer International Publishing, 2019. v. 11844, p. 275–288. ISBN 978-3-030-33719-3 978-3-030-33720-9. Series Title: Lecture Notes in Computer Science. Disponível em: <http://link.springer.com/10.1007/978-3-030-33720-9_21>. Citado 2 vezes nas páginas 57 e 58.
- ANDRE, A.; DINATA, H. Interaction design to enhance ux of university timetable plotting system on mobile version. *IOP Conference Series: Materials Science and Engineering*, v. 407, p. 012174, set. 2018. ISSN 1757-899X. Disponível em: <<https://iopscience.iop.org/article/10.1088/1757-899X/407/1/012174>>. Citado 5 vezes nas páginas 17, 24, 35, 57 e 62.
- ARRATIA-MARTINEZ, N. M.; MAYA-PADRON, C.; AVILA-TORRES, P. A. 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://www.hindawi.com/journals/mpe/2021/6617177/>>. Citado 3 vezes nas páginas 20, 22 e 23.
- BARHAM, A. M.; WESTWOOD, J. B. A simple heuristic to facilitate course timetabling. *The Journal of the Operational Research Society*, v. 29, n. 11, p. 1055, nov. 1978. ISSN 01605682. Disponível em: <<https://www.jstor.org/stable/3009353?origin=crossref>>. Citado na página 15.
- BOURQUE, P.; FAIRLEY, R. E. (Ed.). *SWEBOk: guide to the software engineering body of knowledge*. Version 3.0. Los Alamitos, CA: IEEE Computer Society, 2014. OCLC: 880350861. ISBN 978-0-7695-5166-1. Citado na página 17.
- 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://linkinghub.elsevier.com/retrieve/pii/S0167923611001746>>. Citado 3 vezes nas páginas 21, 52 e 56.
- 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: <http://link.springer.com/10.1007/978-3-540-77345-0_13>. Citado 2 vezes nas páginas 24 e 28.

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: <<http://ieeexplore.ieee.org/document/5298764/>>. Citado na página 15.

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. 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: <http://link.springer.com/10.1007/3-540-61794-9_51>. Citado 2 vezes nas páginas 20 e 21.

APÊNDICE A – Formulário de pesquisa quantitativa

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

Seção 1: 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.

Seção 2: Sobre você

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

- **Pergunta:** Qual o seu nome?
- **Opções de resposta**
 1. Administração Pública
 2. Agronomia
 3. Biologia (Licenciatura)
 4. Ciência da Computação

5. Ciências Biológicas (bacharelado)
6. Ciências Sociais
7. Engenharia Civil
8. Engenharia de Exploração e Produção de Petróleo
9. Engenharia de Produção
10. Engenharia Metalúrgica
11. Engenharia Meteorológica
12. Física (licenciatura)
13. Matemática (Licenciatura)
14. Medicina Veterinária
15. Pedagogia (Licenciatura)
16. Química (Licenciatura)
17. Zootecnia
18. Outro

- **Pergunta:** Em que ano você ingressou na UENF?
- **Opções de resposta**

1. 2023
2. 2022
3. 2021
4. 2020
5. 2019
6. 2018
7. 2017
8. 2016
9. 2015
10. 2014
11. 2013
12. Outro

Seção 3: 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.

- **Perguntas**

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?

- **Opções de resposta**

1. Sim
2. Não
3. Outro

Seção 4: Preferências pessoais

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

- **Pergunta:** Você prefere disciplinas distribuídas ao longo da semana ou acumuladas em poucos dias?

- **Opções de resposta**

1. Distribuídas ao longo da semana

2. ~
3. Não tenho preferência
4. ~
5. Acumuladas em poucos dias

• **Pergunta:** Você prefere disciplinas na parte da manhã ou na parte da tarde?

• **Opções de resposta**

1. na parte da manhã
2. ~
3. Não tenho preferência
4. ~
5. na parte da tarde

• Como você lida com conflitos de horário entre as disciplinas que deseja cursar?

• **Opções de resposta** (Permite múltiplas escolhas)

- 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...

Seção 5: 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.

- **Pergunta:** Quanto tempo (em períodos) você já teve que esperar para fazer uma disciplina da sua grade?
- Descrição: Exemplo hipotético: estou no 6º período e estou desde o 4º período tentando me inscrever em uma disciplina, mas ela não foi oferecida ou não teve vaga, então tive que esperar 2 períodos.

- **Opções de resposta:** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- **Pergunta:** Qual foi a quantidade máxima de períodos que você se distanciou de uma disciplina de determinado período?
- **Descrição:** Exemplo hipotético: estou no 6º período da faculdade, mas ainda estou cursando uma disciplina do 3º período, pois escolhi não fazer antes, ou ainda não obtive a aprovação, logo, me distanciei 3 períodos do esperado.
- **Opções de resposta:** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Seção 6: Você acha que a distribuição de disciplinas semestrais é...

- Classificações
 - 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)
 - Satisfatória (agradável aos meus desejos pessoais)
- **Descrição:** Exemplo hipotético: estou no 6º período da faculdade, mas ainda estou cursando uma disciplina do 3º período, pois escolhi não fazer antes, ou ainda não obtive a aprovação, logo, me distanciei 3 períodos do esperado.
 1. Discordo completamente
 2. ~
 3. ~
 4. ~
 5. Concordo completamente

Seção 7: Opcional

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.

- Campo de texto livre