



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS
CURSO DE BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

GEORGINIS MATHEUS DO NASCIMENTO

**DESENVOLVIMENTO DE UM AMBIENTE DE GESTÃO DE HORÁRIOS
ACADÊMICOS**

PAU DOS FERROS - RN

2023

GEORGINIS MATHEUS DO NASCIMENTO

**DESENVOLVIMENTO DE UM AMBIENTE DE GESTÃO DE HORÁRIOS
ACADÊMICOS**

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito parcial para obtenção do título de Bacharel em Tecnologia da Informação.

Orientador: Prof. Dr. Reudismam Rolim de Sousa

PAU DOS FERROS - RN

2023

GEORGINIS MATHEUS DO NASCIMENTO

**DESENVOLVIMENTO DE UM AMBIENTE DE GESTÃO DE HORÁRIOS
ACADÊMICOS**

Monografia apresentada à Universidade
Federal Rural do Semi-Árido como requisito
parcial para obtenção do título de Bacharel em
Tecnologia da Informação.

Defendida em: 11/10/2023.

BANCA EXAMINADORA

Documento assinado digitalmente
 **REUDISMAM ROLIM DE SOUSA**
Data: 13/10/2023 07:46:49-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Reudismam Rolim de Sousa - UFERSA

Presidente

Documento assinado digitalmente
 **PEDRO THIAGO VALERIO DE SOUZA**
Data: 17/10/2023 17:43:29-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Pedro Thiago Valério de Souza - UFERSA

Membro Examinador

Documento assinado digitalmente
 **JOSE FERDINANDY SILVA CHAGAS**
Data: 13/10/2023 11:27:22-0300
Verifique em <https://validar.iti.gov.br>

Prof. Me. José Ferdinandy Silva Chagas - UFERSA

Membro Examinador

Dedico este trabalho a minha família, minha mãe Maria Lusinete, ao meu irmão Lucas, minha esposa Liliana e minha filha Sophia, que sempre me apoiaram e sempre estiveram presentes para me apoiar e me levantar quando necessário. Essa é uma conquista de todos vocês.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a minha mãe, Maria Lusinete, que sempre me incentivou, cobrou, superou todas as diversidades da vida e continua sempre sendo a mulher e ser humano maravilhoso que é, e que me ensinou tudo e me orientou a ser o homem que sou hoje.

Agradeço também ao meu irmão, Lucas que sempre esteve ao meu lado e que em muitas vezes durante minha carreira profissional e acadêmica esteve disposto a fazer o necessário para me ajudar.

Agradeço também a minha esposa, Liliana, que me complementa e me motiva todos os dias a ser a melhor versão de mim, a minha filha Sophia que desde que chegou ao mundo mudou todas as minhas prioridades, me obrigou a sair da zona de conforto e está proporcionando maravilhas na minha vida.

Sou grato também a toda a minha nova família que construí desde 2014 quando pisei na UFERSA pela primeira vez. Foram muitas idas e vindas, mas as relações que cultivei aqui serão marcadas para a vida toda.

Agradeço também a todo o corpo docente da UFERSA, em especial, ao meu querido orientador, o Prof. Dr. Reudismam. É unanimidade o quão gentil, carismático é este homem. Uma pessoa 100%. Obrigado pela orientação e por ter me guiado tanto nesse processo.

"O homem pode tanto o quanto sabe"

Francis Bacon.

RESUMO

A automação de tarefas desempenha um papel fundamental nas operações modernas, proporcionando eficiência, economia de tempo e redução de erros em uma variedade de contextos, desde a indústria até o mundo acadêmico. No âmbito educacional, a automatização de processos desempenha um papel crucial, especialmente quando se trata de otimizar o planejamento e a gestão dos semestres letivos. Nesse contexto, esse trabalho tem como objetivo o desenvolvimento da interface proposta em Junior (2021). Inicialmente, foi discutida a importância da Interação Humano-Computador (IHC) no desenvolvimento de sistemas, ressaltando seu papel na usabilidade e na eficácia do projeto. Em seguida, exploramos as Metodologias Ágeis, como o Scrum e o Kanban, e como essas abordagens podem proporcionar maior flexibilidade, adaptabilidade e foco na entrega de valor. Destacamos o estudo prévio realizado na mesma universidade, que contribuiu para o levantamento de requisitos, a definição de regras de negócio e o *design* da base do *frontend* do projeto. Adicionalmente, foi analisado o Google Agenda como referência, destacando como essa ferramenta oferece *insights* valiosos para aprimorar a usabilidade do sistema, especialmente na criação de agendamentos e na apresentação de informações contextuais. Por fim, é apresentada a aplicação desenvolvida, uma interface gráfica para que os coordenadores realizem os agendamentos dos horários.

Palavras-chave: Interação Humano-Computador. Metodologias Ágeis. Agendamento de Aulas

ABSTRACT

Task automation plays a key role in modern operations, providing efficiency, time savings and reduced errors in a variety of contexts, from industry to academia. In the educational sphere, process automation plays a crucial role, especially when it comes to optimizing the planning and management of academic semesters. In this context, this work aims to develop the interface proposed in Junior (2021). Initially, the importance of Human-Computer Interaction (HCI) in systems development was discussed, highlighting its role in the usability and effectiveness of the project. We then explore Agile Methodologies such as Scrum and Kanban and how these approaches can provide greater flexibility, adaptability and focus on delivering value. We highlight the previous study carried out at the same university, which contributed to the gathering of requirements, the definition of business rules and the design of the project's frontend base. Additionally, Google Calendar was analyzed as a reference, highlighting how this tool offers valuable insights to improve the usability of the system, especially when creating schedules and presenting contextual information. Finally, the proposed solutions are presented, a graphical interface for coordinators to schedule appointments.

Keywords: Human-Computer Interaction. Agile Methodologies. Class Scheduling

LISTA DE FIGURAS

Figura 1: Planejamento do semestre via planilha.....	13
Figura 2: Tela principal do ambiente.....	23
Figura 3: Navegação entre cursos e semestres.....	24
Figura 4: Diálogo de alerta.....	25
Figura 5: Adicionar disciplina.....	25
Figura 6: Menu professores.....	26
Figura 7 - Utilização do Quadro Kanban no software da Trello.	28
Figura 8: Tela principal do sistema.	37
Figura 9: Múltiplos agendamentos e espaço vazio para novo cadastro.....	38
Figura 10: Modal de cadastros de horários de aula	39
Figura 11: Lista de opções para o usuário selecionar.....	40
Figura 12: Tela de carga horária dos professores.....	41
Figura 13: Tela de cadastro de disciplinas.....	42
Figura 14: Listagem de disciplinas.....	42

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 JUSTIFICATIVA.....	16
1.2 OBJETIVOS.....	17
1.2.1 OBJETIVO GERAL.....	17
1.2.2 OBJETIVOS ESPECÍFICOS:.....	17
1.3 ORGANIZAÇÃO DO TRABALHO.....	18
2 FUNDAMENTAÇÃO TEÓRICA.....	19
2.1 METODOLOGIA ÁGIL.....	19
2.1.1 QUADRO KANBAN.....	21
2.1.2 SCRUM.....	21
2.1.3 PERSONAL SCRUM.....	22
2.2 INTERAÇÃO HUMANO-COMPUTADOR (IHC).....	24
2.3 DESENVOLVIMENTO WEB.....	25
2.3.1 JAVASCRIPT E TYPESCRIPT.....	26
2.3.2 NODE.JS.....	26
2.3.3 REACT.JS E NEXT.JS.....	27
2.3.4 FIREBASE.....	28
3 PROPOSTA DE DESIGN PARA UM AMBIENTE DE GESTÃO DE HORÁRIOS ACADÊMICOS.....	29
3.1 REQUISITOS.....	29
3.1.1 REQUISITOS FUNCIONAIS.....	29
3.1.2 REQUISITOS NÃO FUNCIONAIS.....	30
3.2 PROTOTIPAÇÃO E DESIGN CENTRADO NO USUÁRIO.....	30
4 METODOLOGIA.....	36
4.1 IMPLEMENTAÇÃO DO KANBAN EM CONJUNTO COM UMA SPRINT SCRUM.....	36
5 PROPOSTA DE SOLUÇÃO.....	39
5.1 TECNOLOGIAS UTILIZADAS.....	39
5.2 REFATORAÇÃO DOS REQUISITOS.....	39
5.2.1 NOVOS REQUISITOS FUNCIONAIS.....	40
5.2.2 NOVOS REQUISITOS NÃO FUNCIONAIS.....	43
5.2.3 REGRAS DE NEGÓCIO.....	43
5.3 DESENVOLVIMENTO DA SOLUÇÃO PROPOSTA.....	45
6 CONSIDERAÇÕES FINAIS.....	51
7 REFERÊNCIAS.....	52
APÊNDICE A - PESQUISA COM PROFISSIONAIS ASSOCIADOS À GERAÇÃO DE HORÁRIOS NA UFERSA, CAMPUS PAU DOS FERROS.....	54
4.1 CARACTERIZAÇÃO GERAL DOS PARTICIPANTES.....	54
4.2 CARACTERIZAÇÃO GERAL DA GERAÇÃO DE HORÁRIOS.....	56
4.3 CARACTERIZAÇÃO GERAL SOBRE A FACILIDADE DE GERAÇÃO DE HORÁRIOS	58

4.4 CARACTERÍSTICAS GERAIS SOBRE A APARÊNCIA DA PLANILHA DE HORÁRIOS.....	59
---	----

1 INTRODUÇÃO

Na atualidade, a automação de tarefas desempenha um papel fundamental na transformação de todos os aspectos da sociedade. A disseminação da tecnologia e o avanço constante das capacidades computacionais permitiram uma revolução silenciosa, mas profunda na maneira como realizamos tarefas cotidianas (CARDOSO, 2016).

O desenvolvimento de software já possui décadas de inovação e progresso. Desde os primeiros dias da programação, quando as instruções eram fornecidas e processadas em cartões perfurados, até os sistemas e redes complexas e interconectados, que existem hoje em dia, o desenvolvimento de software tem sido um fator determinante na evolução da sociedade. À medida que a tecnologia avança, notamos cada vez mais as necessidades humanas e a capacidade das máquinas se relacionarem, tornando os sistemas de software uma parte onipresente e indispensável de nossas vidas (FILHO, 2007).

Muitas tarefas antes desempenhadas por seres humanos estão sendo cada vez mais automatizadas e em algumas empresas existem setores onde todo o processo é executado sem interferência de um operador, totalmente de forma automática (CARDOSO, 2016). Desde a automação de processos industriais até a gestão de informações em instituições acadêmicas, o desenvolvimento de sistemas para executar tarefas complexas está revolucionando a eficiência, a precisão e a economia de recursos em todos os setores.

Na educação, a tecnologia desempenha um papel crucial. As instituições de ensino, como a Universidade Federal Rural do Semi-Árido (UFERSA), buscam constantemente maneiras de otimizar os processos acadêmicos. Os sistemas de gerenciamento integrados agora são a espinha dorsal da administração acadêmica, armazenando informações de alunos, professores, funcionários e processos internos. Esses sistemas podem variar de gestão de bibliotecas e controle de recursos administrativos a transações de pagamento de salários, entre outras atividades.

1.1 JUSTIFICATIVA

No contexto da UFERSA, Campus Pau dos Ferros, um dos desafios mais preocupantes é o planejamento dos semestres letivos. Antes do início de cada semestre, os coordenadores e chefes de departamentos acadêmicos enfrentam essa tarefa árdua de planejar cada período vigente, garantindo que cada disciplina seja distribuída corretamente e que os professores cumpram sua carga horária. Este processo envolve a complexa alocação de horários e recursos, evitando conflitos de horários entre turmas, professores e salas, garantindo que as disciplinas estejam disponíveis para os alunos nos períodos corretos.

No entanto, atualmente, esse processo é conduzido manualmente, com o uso de planilhas eletrônicas. Essa abordagem, além de ser demorada, é propensa a erros humanos, o que pode resultar em choques de horários para os alunos, falta de professores disponíveis e problemas no planejamento dos semestres.

Figura 1: Planejamento do semestre via planilha

1	A	B	C	D	E	F	G	H
2	1º SEMESTRE 2020.1							
3		Horários	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
4		18:40 - 19:30	AMBIENTE - T01 BCT	CÁLCULO I - T01 BCT	GA - T01 BCT	ALGORITMOS E PROGRAMAÇÃO I	ANÁL E EXPR - T01 BCT	----
5		19:30 - 20:20	AMBIENTE - T01 BCT	CÁLCULO I - T01 BCT	GA - T01 BCT	ALGORITMOS E PROGRAMAÇÃO I	ANÁL E EXPR - T01 BCT	----
6	BCT	20:20 - 21:25	CÁLCULO I - T01 BCT	AMBIENTE - T01 BCT	ALGORITMOS E PROGRAMAÇÃO I	ANÁL E EXPR - T01 BCT	GA - T01 BCT	----
7		21:25 - 22:20	CÁLCULO I - T01 BCT	AMBIENTE - T01 BCT	ALGORITMOS E PROGRAMAÇÃO I	ANÁL E EXPR - T01 BCT	GA - T01 BCT	----
8					Noite			
9		18:40 - 19:30	CÁLCULO I - T02 BCT	AMBIENTE - T02 BCT	ALGORITMOS E PROGRAMAÇÃO I	ANÁL E EXPR - T02 BCT	GA - T02 BCT	----
10		19:30 - 20:20	CÁLCULO I - T02 BCT	AMBIENTE - T02 BCT	ALGORITMOS E PROGRAMAÇÃO I	ANÁL E EXPR - T02 BCT	GA - T02 BCT	----
11	BCT	20:20 - 21:25	AMBIENTE - T02 BCT	CÁLCULO I - T02 BCT	GA - T02 BCT	ALGORITMOS E PROGRAMAÇÃO I	ANÁL E EXPR - T02 BCT	----
12		21:25 - 22:20	AMBIENTE - T02 BCT	CÁLCULO I - T02 BCT	GA - T02 BCT	ALGORITMOS E PROGRAMAÇÃO I	ANÁL E EXPR - T02 BCT	----
13					Noite			
14		Horários	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
15		07:00 - 07:55			Manhã			
16		07:55 - 08:50	PEX0101 - CÁLCULO I - T01	PAC0008 - ÉTICA E LEGISLAÇÃO - T01	PEX0101 - CÁLCULO I - T01	PEX1236 - ALGORITMOS - T01 (REUDISMAN)		
17		08:50 - 09:45	PEX0101 - CÁLCULO I - T01	PAC0008 - ÉTICA E LEGISLAÇÃO - T01	PEX0101 - CÁLCULO I - T01	PEX1236 - ALGORITMOS - T01 (REUDISMAN)		
18		09:45 - 10:40	PEX0101 - CÁLCULO I - T01	PAC0008 - ÉTICA E LEGISLAÇÃO - T01	PEX0101 - CÁLCULO I - T01	PEX1236 - ALGORITMOS - T01 (REUDISMAN)		
19	BTI	10:40 - 11:35	PEX0101 - CÁLCULO I - T01	PAC0008 - ÉTICA E LEGISLAÇÃO - T01	PEX0101 - CÁLCULO I - T01	PEX1236 - ALGORITMOS - T01 (REUDISMAN)		
20								
21								
22		07:00 - 07:55	PEX0101 - CÁLCULO I - T02	PAC0008 - ÉTICA E LEGISLAÇÃO - T02	PEX0101 - CÁLCULO I - T02	PEX1236 - ALGORITMOS - T02 (SUB PATRICK)		
23		07:55 - 08:50	PEX0101 - CÁLCULO I - T02	PAC0008 - ÉTICA E LEGISLAÇÃO - T02	PEX0101 - CÁLCULO I - T02	PEX1236 - ALGORITMOS - T02 (SUB PATRICK)		
24		08:50 - 09:45	PEX0101 - CÁLCULO I - T02	PAC0008 - ÉTICA E LEGISLAÇÃO - T02	PEX0101 - CÁLCULO I - T02	PEX1236 - ALGORITMOS - T02 (SUB PATRICK)		
25		09:45 - 10:40	PEX0101 - CÁLCULO I - T02	PAC0008 - ÉTICA E LEGISLAÇÃO - T02	PEX0101 - CÁLCULO I - T02	PEX1236 - ALGORITMOS - T02 (SUB PATRICK)		
26	BTI	10:40 - 11:35	PEX0101 - CÁLCULO I - T02	PAC0008 - ÉTICA E LEGISLAÇÃO - T02	PEX0101 - CÁLCULO I - T02	PEX1236 - ALGORITMOS - T02 (SUB PATRICK)		
27								
28								
29		13:00 - 13:55			Tarde			
30		13:55 - 14:50				PAC0050 - ANÁLISE E EXPRESSÃO TEXTUAL - T02	PAC0050 - ANÁLISE E EXPRESSÃO TEXTUAL - T02	
31		14:50 - 15:45				PAC0050 - ANÁLISE E EXPRESSÃO TEXTUAL - T02	PAC0050 - ANÁLISE E EXPRESSÃO TEXTUAL - T02	
32		15:45 - 16:40				PEX1237 - LAB ALGORITMOS - T02 (SUB. JARBELE)	PEX1237 - LAB ALGORITMOS - T02 (SUB. JARBELE)	
33	BTI	16:40 - 17:35				PEX1237 - LAB ALGORITMOS - T02 (SUB. JARBELE)	PEX1237 - LAB ALGORITMOS - T02 (SUB. JARBELE)	
34								
35								
36								
37		SEGUNDA-FEIRA	TERÇA-FEIRA	QUARTA-FEIRA	QUINTA-FEIRA	SEXTA-FEIRA	SÁBADO	
38		07:00 - 07:55	ARTES 1 - GABRIEL	01 - Cleidice Bessa		SOCIEDADE - Turma 01 - Claudia Muniz		
39		07:55 - 08:50	ARTES 1 - GABRIEL	01 - Cleidice Bessa		SOCIEDADE - Turma 01 - Claudia Muniz		
40		08:50 - 09:45	ARTES 1 - GABRIEL	01 - Cleidice Bessa		SOCIEDADE - Turma 01 - Claudia Muniz		
41		09:45 - 10:40	ARTES 1 - GABRIEL	01 - Cleidice Bessa	AO CURSO - RAFAELA BALBI	SOCIEDADE - Turma 01 - Claudia Muniz		
42		10:40 - 11:35	ARTES 1 - GABRIEL	01 - Cleidice Bessa	AO CURSO - RAFAELA BALBI	SOCIEDADE - Turma 01 - Claudia Muniz		
43								
44		13:00 - 13:55						
45		13:55 - 14:50	DE MONIQUE	CHICO	PSH1599-INTRODUÇÃO AO DESENHO T01 - RAISSA	definir		
46		14:50 - 15:45	DE MONIQUE	CHICO	PSH1599-INTRODUÇÃO AO DESENHO T01 - RAISSA	definir		
47		15:45 - 16:40	DE MONIQUE	CHICO	PSH1599-INTRODUÇÃO AO DESENHO T01 - RAISSA	definir		
48		16:40 - 17:35	DE MONIQUE	CHICO	PSH1599-INTRODUÇÃO AO DESENHO T01 - RAISSA	definir		
49								
50		18:40 - 19:30						
51		19:30 - 20:30						
52		20:30 - 21:25						
53		21:25 - 22:20						
54								

Fonte: autoria própria

Diante desse cenário desafiador, o desenvolvimento de uma solução computacional para contornar esses problemas. É nesse contexto que este trabalho segue com base no protótipo criado no estudo anterior de Júnior (2021), que propôs com uma interface intuitiva e eficiente, voltada para o planejamento acadêmico na UFERSA, visando não apenas

simplificar esse processo complexo, mas também aumentar a eficiência e a precisão do planejamento dos semestres.

Ao desenvolver um sistema que automatiza a geração de horários, a distribuição de disciplinas e a alocação de recursos, o objetivo é reduzir significativamente a margem de erro e eliminar o retrabalho associado ao planejamento manual. Essa solução não apenas beneficiará os coordenadores e chefes de departamentos acadêmicos, mas também terá um impacto positivo direto na qualidade da experiência acadêmica dos alunos.

Neste contexto, este trabalho não visa abordar apenas um desafio acadêmico específico, mas também contribuir para o avanço das práticas de gestão acadêmica, destacando a importância contínua da automação de tarefas, metodologias ágeis e do desenvolvimento de software no cenário educacional moderno.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O objetivo principal deste trabalho é implementar um sistema que simplifique o processo de planejamento dos semestres acadêmicos na UFERSA, buscando agilizar a geração de horários e a distribuição de disciplinas entre o corpo docente, reduzindo a margem de erro com relação ao planejamento manual.

1.2.2 OBJETIVOS ESPECÍFICOS:

- Realizar uma revisão detalhada do trabalho anterior, identificando requisitos funcionais e não funcionais, entidades do sistema, regras de negócio e protótipos que servirão como base para o desenvolvimento da interface;
- Planejar minuciosamente as funcionalidades do sistema, a fim de evitar retrabalho durante a fase de desenvolvimento;
- Implementar um ambiente para a geração de horários conforme elencado por Júnior (2021).

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho segue a seguinte estrutura: no Capítulo 2 é apresentada a fundamentação teórica, em que será descrita a metodologia utilizada para o desenvolvimento, tecnologias utilizadas e uma revisão do estudo prévio realizado; no Capítulo 3 é descrita a proposta de *design* proposta por Júnior (2021). No Capítulo 4 é descrita a metodologia seguida no trabalho. No Capítulo 5 é apresentada a implementação do sistema de gestão de horários acadêmicos; por fim, no Capítulo 6 são apresentadas as considerações finais sobre o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos e princípios fundamentais que sustentam o desenvolvimento do sistema de planejamento acadêmico proposto, considerando as informações fornecidas anteriormente, incluindo as tecnologias utilizadas.

2.1 METODOLOGIA ÁGIL

As metodologias ágeis são abordagens de desenvolvimento de software que se destacam por sua flexibilidade, colaboração intensiva entre as equipes e foco na entrega de valor ao cliente (BECK et al., 2001). Elas surgiram como uma alternativa às metodologias tradicionais de desenvolvimento de software, que muitas vezes eram rígidas, burocráticas e não conseguiam acompanhar as demandas crescentes da indústria de tecnologia (BECK et al., 2001).

As metodologias ágeis tiveram suas raízes em um encontro de 17 desenvolvedores de software em 2001, nas montanhas de Utah, nos Estados Unidos. Esse encontro resultou na criação do Manifesto Ágil, que definiu os princípios e valores fundamentais das metodologias ágeis (BECK et al., 2001). Os signatários do manifesto buscavam uma abordagem mais eficiente e adaptativa para o desenvolvimento de software.

Os quatro valores fundamentais do Manifesto Ágil são:

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

A partir desses valores, várias metodologias e estruturas ágeis foram desenvolvidas, algumas conhecidas são: o Scrum, o *Extreme Programming* (XP), o Kanban, entre outras (SOMMERVILLE, 2011). As metodologias ágeis trouxeram uma nova fórmula para o desenvolvimento de software, e com isso trouxeram vários benefícios como (GOMES, 2013):

- **Flexibilidade e Adaptabilidade:** Às metodologias ágeis permitem que as equipes de desenvolvimento se adaptem rapidamente às mudanças nos requisitos do projeto. Isso é essencial em um mundo onde as necessidades dos clientes podem mudar rapidamente;

- **Entrega Contínua de Valor:** O foco nas necessidades do cliente e na entrega de software em funcionamento significa que os clientes obtêm valor mais cedo e com mais frequência. Isso ajuda a melhorar a satisfação do cliente e a manter os projetos alinhados com as metas de negócios;
- **Maior Colaboração:** As metodologias ágeis promovem a colaboração intensiva entre os membros da equipe, bem como com os clientes. Isso leva a uma comunicação mais eficaz e à resolução mais rápida de problemas;
- **Qualidade Melhorada:** Práticas ágeis, como desenvolvimento orientado a testes (TDD) e integração contínua, contribuem para melhorar a qualidade do software, uma vez que os problemas são identificados e resolvidos precocemente;
- **Maior Satisfação da Equipe:** As equipes ágeis geralmente têm mais autonomia e participação nas decisões, o que pode levar a uma maior satisfação e motivação dos membros da equipe.

No dia a dia, as equipes de desenvolvimento ágil trabalham em iterações curtas e regulares, geralmente de 2 a 4 semanas, durante as quais desenvolvem, testam e entregam funcionalidades (GOMES, 2013). Algumas características comuns do uso das metodologias ágeis incluem:

- **Reuniões Diárias:** As equipes realizam reuniões diárias curtas, conhecidas como *Daily Standup* ou *Daily Scrum*, para sincronizar o trabalho, discutir impedimentos e planejar o dia;
- **Priorização de *Backlog*:** As funcionalidades são priorizadas em um *backlog* de produtos, e as mais importantes são desenvolvidas primeiro;
- **Feedback Contínuo:** Os clientes e as partes interessadas são envolvidos continuamente no processo de desenvolvimento, fornecendo *feedback* regular sobre o software em desenvolvimento;
- **Refinamento de Requisitos:** Os requisitos são refinados à medida que o projeto avança, permitindo uma melhor compreensão das necessidades do cliente;
- **Entrega Contínua:** O software é entregue em incrementos regulares, permitindo que os clientes o utilizem e forneçam *feedback* valioso;
- **Mudanças Bem-Vindas:** As mudanças nos requisitos são bem-vindas e as equipes estão preparadas para ajustar o trabalho conforme necessário;

- **Melhoria Contínua:** As equipes realizam retrospectivas no final de cada iteração para identificar maneiras de melhorar o processo.

2.1.1 QUADRO KANBAN

O Quadro Kanban é uma ferramenta visual que ajuda a gerenciar o fluxo de trabalho e as tarefas de forma eficiente. Cada tarefa é representada por um cartão e esses cartões são movidos através de colunas que representam os estágios do processo, desde a definição até a conclusão. No contexto do desenvolvimento de software, o quadro Kanban é uma escolha sensata para manter o controle das tarefas, identificar gargalos e otimizar o processo de desenvolvimento.

A utilização do Quadro Kanban traz vários benefícios ao processo, permite que todos na equipe visualizem e saibam o que está acontecendo e onde cada tarefa está no fluxo de trabalho. Também ajuda a manter o fluxo de trabalho equilibrado, evita gargalos e sobrecargas, é bastante flexível e pode ser aplicado em vários contextos.

O Quadro Kanban é utilizado em uma variedade de setores e muitas equipes melhoram sua performance ao adotar a utilização desta ferramenta. Também é possível utilizar o Kanban com outras ferramentas e metodologias ágeis pois é possível personalizar da forma que for mais eficiente para o processo de trabalho a ser realizado.

2.1.2 SCRUM

O Scrum é uma metodologia ágil amplamente adotada no desenvolvimento de software. Ele se baseia em princípios de transparência, inspeção e adaptação e promove a colaboração entre equipes multidisciplinares. O Scrum divide o trabalho em iterações curtas chamadas "*sprints*", que geralmente duram de duas a quatro semanas. Durante cada *sprint*, a equipe se concentra na entrega de funcionalidades valiosas e testáveis.

A aplicação do Scrum no contexto do desenvolvimento do sistema de planejamento acadêmico oferece vantagens significativas. Ele permite uma abordagem interativa e adaptativa, onde os requisitos podem ser refinados ao longo do tempo com base no *feedback* dos usuários. Além disso, o Scrum promove a colaboração próxima entre os membros da equipe, o que é fundamental para o sucesso do projeto. Os principais conceitos do Scrum são:

1. **Equipe:** O scrum opera com uma equipe auto-organizada e multifuncional, composta por membros com habilidades específicas para concluir os objetivos;

2. **Dono do Produto (*Product Owner*):** O dono do produto é quem representa o *stakeholder*. Ele é responsável por definir as funcionalidades desejadas para o produto final, priorização de tarefas e deve garantir que a equipe esteja sempre trabalhando nas tarefas mais importantes para o cliente;
3. **Scrum Master:** O Scrum *master* atua como facilitador e *coach* da equipe Scrum. Ele ajuda a equipe na implementação das práticas do scrum, remove impeditivos e garante que o fluxo de desenvolvimento siga sem problemas.
4. **Backlog do Produto:** O *backlog* nada mais é do que a lista de tarefas e melhorias planejadas para o produto. É mantido pelo Dono do Produto e é bem flexível, podendo ser ajustado de acordo com a necessidade.
5. **Sprints:** De forma bem resumida *sprints* são os ciclos de desenvolvimento. São iterações geralmente com duração de 2 a 4 semanas. Durante a *sprint* a equipe seleciona um conjunto de itens do *backlog* para concluir no período estabelecido. O objetivo de cada *sprint* é entregar um incremento no produto, ou seja, ao final de cada *sprint* deve haver uma funcionalidade adicional.
6. **Reuniões do Scrum:** O Scrum possui várias reuniões dentre elas a reunião de planejamento da *sprint*, onde a equipe seleciona as tarefas para serem realizadas, reuniões diárias onde a equipe compartilha as atualizações sobre o processo de desenvolvimento, a revisão da *sprint* onde o trabalho concluído é demonstrado e a retrospectiva, na qual a equipe revisa o processo, desempenho e identifica pontos de melhoria no processo.

O Scrum é conhecido por sua abordagem flexível, adaptativa e colaborativa para o gerenciamento de projetos. Ele ajuda as equipes a serem mais produtivas, responderem rápido às mudanças e a entregarem um produto de alta qualidade que atende as necessidades do cliente.

2.1.3 PERSONAL SCRUM

O Scrum como conhecido foi originalmente desenvolvido para equipes de softwares e este sistema está sendo construído por apenas um desenvolvedor. Porém, existe uma adaptação desse *framework* que permite que indivíduos apliquem os princípios e práticas do Scrum em seus próprios projetos pessoais ou profissionais. Esta adaptação é chamada de Personal Scrum, pois apesar do Scrum ter sido concebido para ser usado em equipes, os

princípios ágeis subjacentes podem ser muito eficientes para gerenciar tarefas e projetos individuais. A seguir estão alguns conceitos chaves do Scrum que podem ser aplicados no contexto do desenvolvimento em um Personal Scrum:

1. **Personal Product Backlog:** Assim como no Scrum tradicional, deve começar com um “*Product Backlog*” pessoal. Esse item possui a mesma característica do tradicional. Nada mais é do que uma lista priorizada de todas as tarefas, metas ou itens que devem ser realizadas para a finalização do objetivo. Cada item no *backlog* representa um objetivo claro;
2. **Personal Sprints:** A diferença é apenas que são estabelecidas metas para si mesmo em intervalos de tempos definidos. As *sprints* podem variar de duração, mas podem ter um período de uma a quatro semanas. Nesse projeto trabalhamos com *sprints* de duas semanas. A cada *sprint* era definida uma quantidade de tarefas que já estivessem refinadas que pudessem ser finalizadas dentro do período da *sprint*;
3. **Quadro kanban:** Foi utilizado um quadro kanban para visualizar o fluxo de desenvolvimento onde cada coluna apresentada no quadro representa os estágios do processo. A medida que um processo era finalizado a tarefa referente a ela ia sendo movida pelas colunas do quadro;
4. **Planejamento:** Durante o desenvolvimento das *sprints*, as próximas tarefas iam sendo refinadas com base nos próximos objetivos do sistema. As tarefas iam sendo priorizadas de maneira que iam complementando o fluxo de desenvolvimento atual, por exemplo, ao concluir a tabela de agendamentos a próxima tarefa poderia ser o cadastro das disciplinas e o comportamento dos componentes na tabela;
5. **Revisão e retrospectiva:** No final de cada ciclo de desenvolvimento, era feita uma checagem das tarefas realizadas, se os objetivos tinham sido alcançados, validações eram feitas e possíveis pontos de melhorias ou *bugs* eram adicionados ao *backlog*;
6. **Adaptação:** Uma das principais vantagens do scrum é a capacidade de se adaptar. Se for identificado um problema, ou algo que pode ser melhorado, deve ser aplicado um processo de melhoria no fluxo de desenvolvimento.

O *Personal Scrum* pode ser uma abordagem altamente eficiente para gerenciar projetos quando se trabalha sozinho, pois fornece estrutura, foco e flexibilidade para o processo de desenvolvimento.

2.2 INTERAÇÃO HUMANO-COMPUTADOR (IHC)

A Interação Humano-Computador (IHC) desempenha um papel vital no desenvolvimento de sistemas interativos para o ambiente educacional. No contexto acadêmico, é crucial que os sistemas sejam altamente usáveis, eficientes e acessíveis. Sistemas eficientes reduzem o tempo de execução, diminuem erros associados e fornecem validações para o processo. Isso garante que alunos, professores e coordenadores de cursos tenham uma experiência positiva ao utilizar a plataforma (JUNIOR, 2021).

Dentro do escopo da UFERSA, onde a gestão de horários acadêmicos é complexa devido à variedade de cursos e turnos, a IHC assume um papel ainda mais importante. A usabilidade e a acessibilidade são princípios fundamentais que asseguram que todos os envolvidos no processo de planejamento acadêmico possam interagir eficazmente com o sistema (JUNIOR, 2021).

Estes são alguns benefícios do uso do IHC no desenvolvimento do projeto:

- Usabilidade Aprimorada: A IHC coloca o usuário no centro do processo de *design*, garantindo que as interfaces e sistemas sejam fáceis de usar e compreender. Isso resulta em uma melhor experiência do usuário, reduzindo erros e aumentando a eficiência;
- Maior Satisfação do Usuário: Quando os sistemas são projetados com as necessidades e preferências dos usuários em mente, a satisfação do usuário aumenta. Isso pode levar a uma adoção mais ampla das soluções propostas;
- Acessibilidade: A IHC também se preocupa em tornar sistemas acessíveis a todos, independentemente de suas habilidades ou limitações. Isso promove a inclusão e garante que todos tenham acesso e consigam utilizar o sistema;
- Eficiência e Produtividade: Interfaces bem projetadas podem aumentar a eficiência das tarefas realizadas pelos usuários. Isso é especialmente importante em contextos acadêmicos, onde a gestão de informações e o acesso a recursos educacionais são essenciais;

- **Coleta de Dados Significativos:** Através de técnicas de IHC, é possível projetar sistemas que coletam dados significativos sobre o comportamento dos usuários. Essas informações podem ser usadas para tomar decisões informadas e melhorar continuamente a solução;
- **Redução de Erros e Conflitos:** A IHC pode ajudar a identificar e mitigar problemas antes que eles ocorram. Isso é crucial para evitar conflitos na programação de cursos e garantir que os recursos educacionais sejam alocados de forma eficaz;
- **Customização e Personalização:** A personalização de sistemas com base nas preferências individuais dos usuários é uma característica da IHC. Isso permite que cada usuário adapte a solução às suas necessidades específicas.

2.3 DESENVOLVIMENTO WEB

O desenvolvimento web é o processo de criação e manutenção de sites e aplicativos que podem ser acessados ou que possuem recursos conectados à internet. Envolve a programação, criação de protótipos e interfaces gráficas, otimização e implementação de funcionalidades para tornar os sites ou aplicativos interativos e funcionais.

Os principais componentes do desenvolvimento web incluem:

- ***Front-end* (ou Desenvolvimento do Lado do Cliente):** Isso se refere à parte visível de um site ou aplicativo, com a qual os usuários interagem diretamente. Envolve a criação do *design*, *layout*, conteúdo e interatividade dos aplicativos. Os desenvolvedores *front-end* se concentram em criar interfaces de usuário atraentes e responsivas (Qual é a diferença entre front-end e back-end no desenvolvimento de aplicações?, 16 de outubro de 2023);
- ***Back-end* (ou Desenvolvimento do Lado do Servidor):** Esta é a parte "invisível" do desenvolvimento web que lida com o processamento de dados, a lógica de negócios e o armazenamento de informações. Os desenvolvedores *back-end* trabalham com linguagens de programação e bancos de dados para garantir que o site ou aplicativo funcione de maneira eficiente e segura (Qual é a diferença entre front-end e *back-end* no desenvolvimento de aplicações?, 16 de outubro de 2023);
- **Banco de Dados:** É o local onde todas as informações são armazenadas, desde textos simples até dados de usuários. Os desenvolvedores web projetam e gerenciam bancos

de dados para garantir que os dados sejam armazenados de maneira organizada e acessados eficientemente (O que é banco de dados?, 16 de outubro de 2023).

2.3.1 JAVASCRIPT E TYPESCRIPT

O Javascript¹ é uma linguagem de programação de alto nível utilizada para desenvolver páginas interativas. O Javascript surgiu inicialmente como uma tecnologia que era executada pelos navegadores para deixar as páginas web mais dinâmicas; porém, devido ao desenvolvimento do software Node.js, hoje é possível utilizar o Javascript fora dos navegadores. Uma das características do Javascript é que ela possui tipagem fraca, ou seja, não há uma declaração explícita do tipo da variável (“O que é Javascript”, 16 de outubro de 2023).

Esta configuração permite que o tipo dessa variável possa ser alterada em tempo de execução. Devido a isso foi desenvolvido um *super set* que adiciona a tipagem ao Javascript, chamado de Typescript. A utilização da tipagem das variáveis permite utilizar algumas ferramentas de análise de código que podem detectar erros relacionados a tipagem, além de podermos garantir que uma variável sempre receberá o mesmo tipo de variável em tempo de execução (“TypeScript is Javascript with syntax for types”, 16 de outubro de 2023).

2.3.2 NODE.JS

O Node.js² é um ambiente de tempo de execução (runtime) JavaScript multiplataforma de código aberto construído no motor JavaScript V8 do Google Chrome. Ele permite a execução de código JavaScript no lado do servidor, em vez de apenas no navegador. Node.js é especialmente adequado para desenvolver aplicativos de rede e servidores da web, mas pode ser usado para uma variedade de aplicações (“Sobre Node.js”, 05 de outubro de 2023).

Principais características e funcionalidades do Node.js:

- JavaScript no servidor: Node.js permite que você use JavaScript no lado do servidor, o que é uma grande vantagem quando você deseja compartilhar código entre o servidor e o cliente;
- Execução assíncrona: Node.js é conhecido por seu modelo de I/O não bloqueante, que permite que operações de I/O sejam executadas de maneira assíncrona, tornando-o eficiente para lidar com tarefas de entrada e saída;

¹ <https://aws.amazon.com/pt/what-is/javascript/>

² <https://nodejs.org/pt-br/about>

- Módulos: Node.js tem um sistema de módulos que facilita a organização e reutilização de código. Existem muitos módulos de terceiros disponíveis que podem ser facilmente integrados em seu projeto;
- Comunidade ativa: Node.js tem uma comunidade de desenvolvedores ativa, o que significa que você pode encontrar uma ampla variedade de bibliotecas e módulos disponíveis para estender as funcionalidades do Node.js;
- Construção de servidores e aplicativos de rede: É particularmente útil para criar servidores web, aplicativos de chat em tempo real, aplicativos de streaming, proxies, e muito mais;
- Cross-platform: Node.js é executado em várias plataformas, incluindo Windows, macOS e várias distribuições Linux;
- Ferramentas de linha de comando: O Node.js também é usado para criar ferramentas de linha de comando que ajudam no desenvolvimento e automação de tarefas;
- Node.js é amplamente adotado em desenvolvimento web e é usado em muitos aplicativos em produção. É uma opção popular para construir aplicativos escaláveis e de alto desempenho, graças à sua capacidade de lidar com muitas conexões simultâneas.

2.3.3 REACT.JS E NEXT.JS

Segundo a própria documentação, o React ³ é uma biblioteca que tem como principal característica a criação de interfaces de usuários. Ela permite a criação de componentes complexos utilizando pequenos trechos de códigos. Com isso o React possibilita a criação de *views* que podem atualizar e renderizar cada componente conforme a mudança dos dados contidos em cada um deles. Essa biblioteca poderosa é escrita em Javascript/Typescript, que permite a utilização dos mais diversos tipos de dados.

O Next⁴ é um framework React que fornece algumas funcionalidades prontas que possibilita a criação de páginas de forma mais rápida. O Next.js lida com as ferramentas e a configuração necessária para o React e fornece estrutura, recursos e otimizações adicionais para o aplicativo web.

³ <https://react.dev/>

⁴ <https://nextjs.org/learn/foundations/about-nextjs/what-is-nextjs>

2.3.4 FIREBASE

O Firebase⁵ É uma plataforma de desenvolvimento de aplicativos móveis e web oferecidos pela Google. Ele fornece uma ampla gama de serviços e recursos para ajudar desenvolvedores a construir, implantar e gerenciar aplicativos com mais facilidade. O Firebase é amplamente utilizado para o desenvolvimento de aplicativos móveis e web, oferecendo uma série de recursos essenciais, como autenticação de usuários, armazenamento em nuvem, banco de dados em tempo real, hospedagem web, notificações em tempo real e muito mais (“Firebase | Google 's mobile and web app development platform”, 17 de outubro de 2023). Os recursos utilizados do Firebase foram:

- **Firestore Authentication:** Permite adicionar autenticação de usuário ao seu aplicativo, suportando autenticação por e-mail, número de telefone, redes sociais (como Google, Facebook, Twitter) e autenticação personalizada. Isso ajuda a proteger seus dados e a personalizar a experiência do usuário com facilidade;
- **Cloud Firestore:** É um banco de dados em tempo real baseado em documentos que oferece armazenamento de dados escalável e sincronização em tempo real para aplicativos. Ele é frequentemente usado em aplicativos que exigem colaboração em tempo real e funcionalidades offline.

⁵ <https://firebase.google.com/>

3 PROPOSTA DE DESIGN PARA UM AMBIENTE DE GESTÃO DE HORÁRIOS ACADÊMICOS

Neste capítulo é apresentado o trabalho prévio conduzido por Júnior (2021) na UFERSA, que lançou as bases para o desenvolvimento do sistema de planejamento acadêmico. Este estudo abordou a problemática da gestão de horários acadêmicos em um contexto similar. Ele identificou a necessidade de uma plataforma que simplificasse o processo de geração de horários, reduzisse erros e proporcionasse uma visão mais clara dos horários de diferentes cursos e turnos.

Os resultados desse estudo prévio, incluindo requisitos levantados, regras de negócio definidas e o *design* da base do sistema, foram utilizados como referência para o desenvolvimento do sistema da UFERSA. Isso demonstra uma abordagem baseada em evidências e experiências anteriores, contribuindo para a eficácia do projeto.

3.1 REQUISITOS

Júnior (2021) define alguns requisitos funcionais e não funcionais.

3.1.1 REQUISITOS FUNCIONAIS

- **RF001** - Cadastrar disciplinas;
 - Descrição: O sistema deve permitir que o usuário cadastre disciplinas.
- **RF002** - Editar disciplinas;
 - Descrição: O sistema deve permitir que o usuário edite disciplinas cadastradas.
- **RF003** - Excluir disciplinas;
 - Descrição: O sistema deve permitir que o usuário exclua disciplinas cadastradas.
- **RF004** - Pesquisar disciplinas;
 - Descrição: O sistema deve permitir que o usuário pesquise por disciplinas cadastradas;
- **RF005** - Adicionar professor a disciplinas;
 - Descrição: O sistema deve permitir que o usuário adicione um professor a uma disciplina
- **RF006** – Editar professor de uma disciplina.;
 - Descrição: O sistema deve permitir que o usuário edite o professor cadastrado na disciplina.
- **RF007** – Excluir professor de uma disciplina;

- Descrição: O sistema deve permitir que o usuário exclua o professor de uma disciplina.
- **RF008** - Pesquisa professor;
 - Descrição: O sistema deve permitir que o usuário pesquise por professores cadastrados em disciplinas.
- **RF009** - Gerar relatório do semestre;
 - Descrição: O sistema deve permitir que o usuário gere relatórios com informações do semestre.
- **RF010** - Avisar disciplinas em choque de horário;
 - Descrição: O sistema deve notificar o choque de horário de disciplinas.
- **RF011** - Avisar disciplinas cadastradas em um único dia;
 - Descrição: O sistema deve notificar disciplinas cadastradas em um único dia.
- **RF012** - Avisar professores com carga horária em excesso;
 - Descrição: O sistema deve notificar se existirem professores com a carga horária além do limite.
- **RF013** - Avisar professores sem carga horária mínima.
 - Descrição: O sistema deve notificar se existirem professores sem a carga horária mínima.

3.1.2 REQUISITOS NÃO FUNCIONAIS

- **RNF001** - Usabilidade (Inteligibilidade);
 - Descrição: O sistema deve conter uma interface intuitiva, ou seja, de fácil uso para o usuário.
- **RNF002** - Confiabilidade (Apreensibilidade);
 - Descrição: O sistema deve permitir a facilidade de aprendizagem para o usuário.
- **RNF003** - Confiabilidade (Operabilidade);
 - Descrição: O sistema deve permitir que o usuário tenha o controle sobre as funcionalidades.

3.2 PROTOTIPAÇÃO E *DESIGN* CENTRADO NO USUÁRIO

O *design* centrado no usuário é uma abordagem crítica para garantir que o sistema de planejamento acadêmico da UFERSA seja intuitivo e atenda às necessidades dos coordenadores de cursos, secretarias e demais envolvidos no processo de planejamento acadêmico.

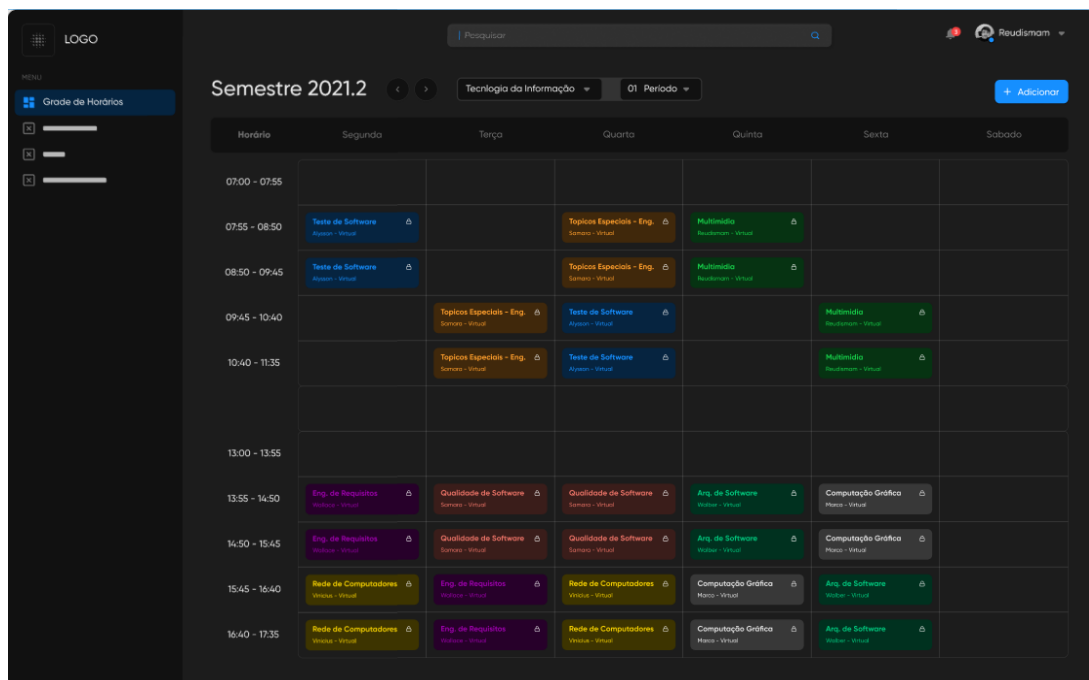
O processo de *design* começou com uma profunda compreensão das necessidades dos usuários. Foram realizadas pesquisas, entrevistas e observações para identificar os desafios enfrentados pelos coordenadores de cursos e as lacunas no sistema de planejamento acadêmico existente. Com base nesses dados, foi possível criar um *design* que aborda

especificamente esses problemas e oferece soluções práticas. A pesquisa realizada por Júnior (2021) pode ser vista no Apêndice A.

A interface de usuário resultante foi projetada para ser não apenas funcional, mas também esteticamente agradável. Isso contribui para a satisfação do usuário e promove a adoção do sistema. A clareza na apresentação das informações, a navegabilidade intuitiva e a facilidade de interação são princípios-chave que orientaram o *design*.

No trabalho de Júnior (2021) são mostradas apenas as telas principais propostas para a solução dos problemas elencados. Na Figura 2 pode ser vista a principal tela do sistema. É nessa tela onde o usuário irá criar os horários das disciplinas e poderá visualizar o planejamento dos horários de cada um dos cursos de acordo com o período e semestre em questão.

Figura 2: Tela principal do ambiente



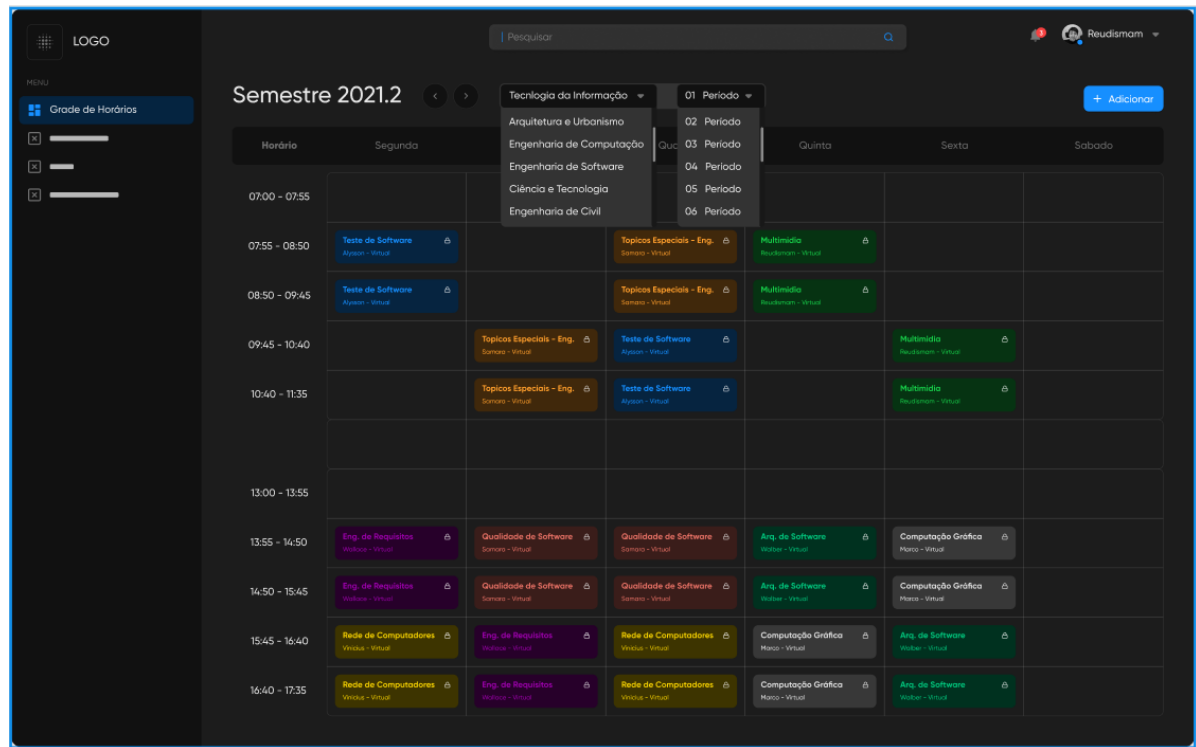
The screenshot displays the main interface of a course scheduling system. The interface is dark-themed and includes a sidebar on the left with a menu, a search bar at the top, and a main grid showing the schedule for Semestre 2021.2. The grid is organized by time slots (Horário) and days of the week (Segunda, Terça, Quarta, Quinta, Sexta, Sábado). The schedule includes various disciplines and courses, such as Teste de Software, Tópicos Especiais - Eng., Multimídia, Eng. de Requisitos, Qualidade de Software, Arg. de Software, and Computação Gráfica. Each entry in the grid is color-coded and includes a small icon representing the discipline or course.

Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:00 - 07:55						
07:55 - 08:50	Teste de Software Alisson - Virtual		Tópicos Especiais - Eng. Sandro - Virtual	Multimídia Reudismann - Virtual		
08:50 - 09:45	Teste de Software Alisson - Virtual		Tópicos Especiais - Eng. Sandro - Virtual	Multimídia Reudismann - Virtual		
09:45 - 10:40		Tópicos Especiais - Eng. Sandro - Virtual	Teste de Software Alisson - Virtual		Multimídia Reudismann - Virtual	
10:40 - 11:35		Tópicos Especiais - Eng. Sandro - Virtual	Teste de Software Alisson - Virtual		Multimídia Reudismann - Virtual	
13:00 - 13:55						
13:55 - 14:50	Eng. de Requisitos Mafra - Virtual	Qualidade de Software Sandro - Virtual	Qualidade de Software Sandro - Virtual	Arg. de Software Mafra - Virtual	Computação Gráfica Mafra - Virtual	
14:50 - 15:45	Eng. de Requisitos Mafra - Virtual	Qualidade de Software Sandro - Virtual	Qualidade de Software Sandro - Virtual	Arg. de Software Mafra - Virtual	Computação Gráfica Mafra - Virtual	
15:45 - 16:40	Rede de Computadores Mafra - Virtual	Eng. de Requisitos Mafra - Virtual	Rede de Computadores Mafra - Virtual	Computação Gráfica Mafra - Virtual	Arg. de Software Mafra - Virtual	
16:40 - 17:35	Rede de Computadores Mafra - Virtual	Eng. de Requisitos Mafra - Virtual	Rede de Computadores Mafra - Virtual	Computação Gráfica Mafra - Virtual	Arg. de Software Mafra - Virtual	

Fonte: Júnior (2021).

Já na Figura 3 pode-se ver como o usuário poderá selecionar o curso e o período. O semestre pode ser alterado ao clicar em um dos ícones no canto superior esquerdo acima do cabeçalho da tabela.

Figura 3: Navegação entre cursos e semestres



Fonte: Júnior (2021).

Durante o processo de criação dos horários acadêmicos podem acontecer erros e cairmos em uma das situações propostas no estudo prévio, como choques de horários, professores com carga horária fora do intervalo correto, etc. Sempre que o sistema identificar um destes problemas, o usuário poderá ver um alerta emitido na tela. Esse alerta é mostrado na Figura 4.

Figura 4: Diálogo de alerta

The screenshot displays a web application for managing academic schedules. The interface includes a sidebar menu on the left with a 'LOGO' and a 'Grade de Horários' section. The main area shows a calendar for 'Semestre 2021.2' with columns for days of the week and time slots. A conflict alert is shown as a tooltip over the 'Topicos Especiais - Eng.' course, stating: 'Disciplina em choque! A disciplina "Topicos Especiais" da profa. "Samara" esta em choque com "Metodos Formais" no 03 Período.' The interface also features a search bar, a user profile 'Reudismom', and a '+ Adicionar' button.

Horário	Segunda	Terça	Quarta	Quinta	Sábado
07:00 - 07:55					
07:55 - 08:50	Teste de Software		Topicos Especiais - Eng.	Multimidia	
08:50 - 09:45	Teste de Software		Topicos Especiais - Eng.	Multimidia	
09:45 - 10:40		Topicos Especiais - Eng.	Teste de Software	Multimidia	
10:40 - 11:35		Topicos Especiais - Eng.	Teste de Software	Multimidia	
13:00 - 13:55					
13:55 - 14:50	Eng. de Requisitos	Qualidade de Software	Qualidade de Software	Arq. de Software	Computação Gráfica
14:50 - 15:45	Eng. de Requisitos	Qualidade de Software	Qualidade de Software	Arq. de Software	Computação Gráfica
15:45 - 16:40	Rede de Computadores	Eng. de Requisitos	Rede de Computadores	Computação Gráfica	Arq. de Software
16:40 - 17:35	Rede de Computadores	Eng. de Requisitos	Rede de Computadores	Computação Gráfica	Arq. de Software

Fonte: Júnior (2021).

Em Júnior (2021), a interface contém um botão, onde o usuário irá clicar para abrir um modal e nesse modal o usuário preencherá os dados, como por exemplo, nome da disciplina, professor, local, horários e poderá selecionar a cor do agendamento da disciplina na tabela (Figura 5).

Figura 5: Adicionar disciplina

The image shows a web application interface for managing a semester schedule. A modal titled "Adicionar Disciplina" is open in the center, allowing a user to add a new course. The modal contains the following fields:

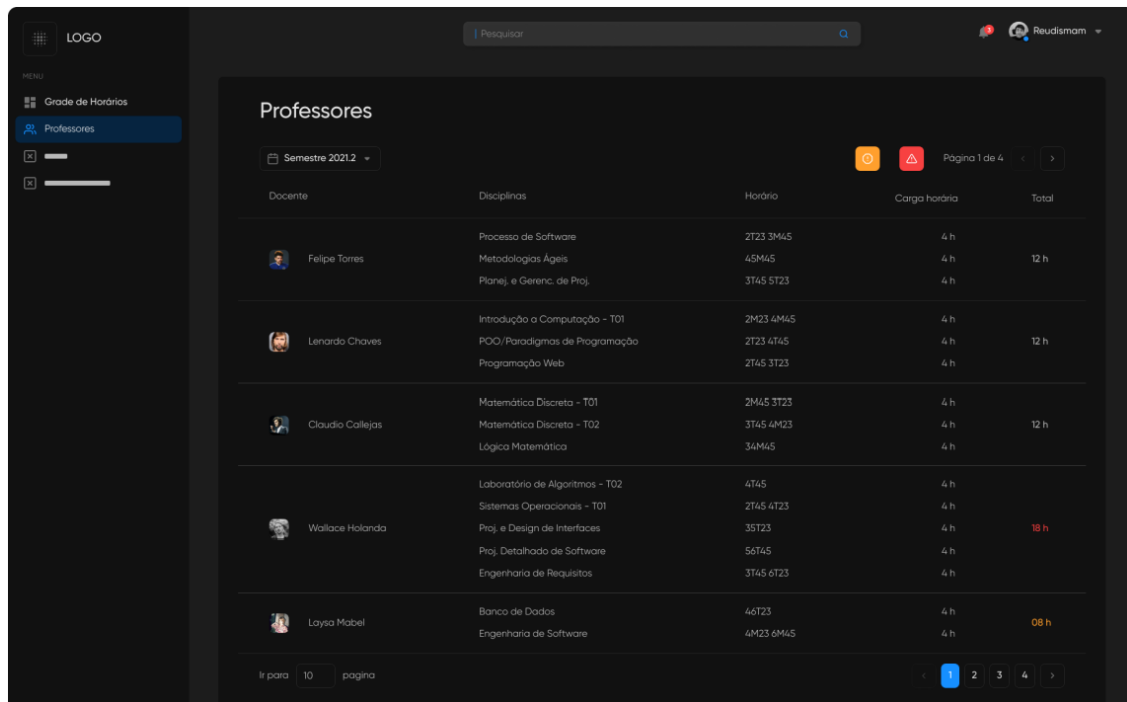
- Nome da Disciplina:** Teste de Software
- Professor:** Samara Martins
- Local:** Virtual
- Horário de início:** 07 : 55 AM
- Horário de término:** 09 : 45 AM
- Comentarios:** (empty text area)
- Color selection:** A row of seven colored circles (red, orange, yellow, green, blue, purple, white).

At the bottom of the modal are "Cancelar" and "Salvar" buttons. The background shows a "Semestre" schedule grid with columns for "Horário", "Sexta", and "Sabado". The grid contains various course entries with their respective colors and names, such as "Rede de Computadores", "Eng. de Requisitos", "Computação Gráfica", and "Arq. de Software".

Fonte: Júnior (2021).

Outro ponto importante é a visualização das informações dos professores. O sistema deve mostrar quais disciplinas os professores têm cadastrados com seus respectivos horários, carga horária de cada disciplina e carga horária total. A Figura 6 mostra a solução proposta.

Figura 6: Menu professores



Docente	Disciplinas	Horário	Carga horária	Total
Felipe Torres	Processo de Software	2T23 3M45	4 h	12 h
	Metodologias Ágeis	4SM45	4 h	
	Planej. e Gerenc. de Proj.	3T45 5T23	4 h	
Lenardo Chaves	Introdução a Computação - T01	2M23 4M45	4 h	12 h
	POO/Paradigmas de Programação	2T23 4T45	4 h	
	Programação Web	2T45 3T23	4 h	
Claudia Callejas	Matemática Discreta - T01	2M45 3T23	4 h	12 h
	Matemática Discreta - T02	3T45 4M23	4 h	
	Lógica Matemática	3M465	4 h	
Wallace Holanda	Laboratório de Algoritmos - T02	4T45	4 h	18 h
	Sistemas Operacionais - T01	2T45 4T23	4 h	
	Proj. e Design de Interfaces	35T23	4 h	
	Proj. Detalhado de Software	56T45	4 h	
	Engenharia de Requisitos	3T45 6T23	4 h	
Laysa Mabel	Banco de Dados	46T23	4 h	08 h
	Engenharia de Software	4M23 6M45	4 h	

Fonte: Júnior (2021).

Na Figura anterior também pode-se ver um *input select* para selecionar o semestre e na parte da direita podemos visualizar dois ícones, um em amarelo e outro em vermelho. O alerta vermelho aparece quando os professores excedem a carga horária máxima e o alerta em amarelo alerta professores que ainda não atingiram a carga horária mínima.

Júnior (2021) objetiva propor um protótipo do sistema para ser desenvolvido definindo bases sólidas para o desenvolvimento. O trabalho realizou análises, levantou requisitos com os *stakeholders* e no final propôs um *design* de acordo com as necessidades e expectativas dos clientes.

4 METODOLOGIA

Para desenvolver o ambiente para auxiliar na geração de horários acadêmicos foram empregadas abordagens e ferramentas voltadas ao desenvolvimento ágil, conforme descrito neste capítulo.

4.1 IMPLEMENTAÇÃO DO KANBAN EM CONJUNTO COM UMA SPRINT SCRUM

Essa abordagem combina elementos do Scrum com os princípios do Kanban para melhorar a gestão do fluxo de trabalho. Durante o desenvolvimento foram utilizados os seguintes conceitos:

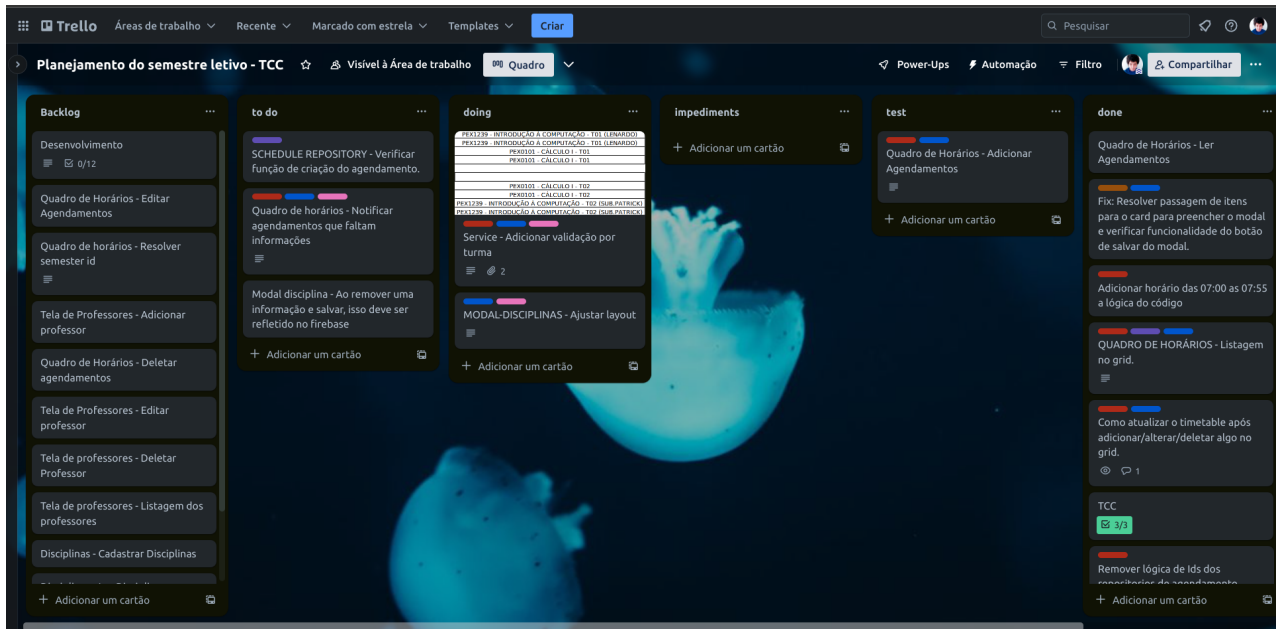
- *Backlog*: Definição e priorização das tarefas relacionadas a desenvolvimento, correção ou refatoração de funcionalidades;
 - *Product Backlog*: Contém funcionalidades, requisitos e melhorias para o projeto. Geralmente é mantido pelo dono do produto, mas como foi implementado o *personal scrum* o *backlog* foi mantido pelo desenvolvedor;
 - *Sprint Backlog*: Contém a seleção de tarefas que a equipe se compromete a entregar ao final de uma *sprint*.
- Quadro Kanban: Começamos criando um quadro Kanban utilizando o software web da Trello⁶ que pode ser visto na Figura 7. O trello foi escolhido devido ser uma ferramenta simples, intuitiva e gratuita. Os cartões representavam as tarefas que deveriam ser desenvolvidas. As colunas representam as etapas pelas quais as tarefas passavam, como:
 - *Backlog*: É o *Product Backlog*;
 - *To do*: É o *Sprint backlog*;
 - *Doing*: Contém as tarefas que estão sendo desenvolvidas no momento;
 - *Impediments*: Contém as tarefas que após terem seu desenvolvimento iniciado sofreram alteração de requisitos ou estão com algum débito técnico ou de informação;

⁶ <https://trello.com>

- *Test*: Após a fase de desenvolvimento ser concluída a tarefa passa por um processo de teste para validar se os objetivos iniciais foram alcançados;
 - *Done*: Tarefa finalizada e validada.
- Limites de Trabalho em Progresso (WIP): Uma das principais práticas do Kanban é a limitação do trabalho em progresso (GOMES, 2013). Definimos limites para o número de tarefas que poderiam estar em uma coluna específica a qualquer momento. Isso nos ajudou a evitar a sobrecarga de trabalho e a manter um fluxo de trabalho mais equilibrado. Para o *backlog*, *to do* e a coluna de *done* não há limite de cards e entre as colunas de *doing*, *impediments* e *test*, que vamos definir como processo de desenvolvimento, foi definido 3 histórias por vez. A quantidade de histórias no processo de desenvolvimento ainda terá de obedecer às definições da *sprint* que serão definidas mais à frente;
- Entrega Contínua de Valor: Foi mantido uma lista de tarefas prioritárias, mas sempre permitindo que novos itens fossem adicionados a qualquer momento, refletindo a abordagem Kanban de priorização contínua;
- Definição das *sprints*: As *sprints* foram definidas para um período de duas semanas com meta de 8 pontos. A pontuação é definida durante o planejamento da *sprint* e depende da complexidade, riscos, esforços envolvidos e estimativa de tempo necessário para desenvolver cada tarefa. Para este projeto as tarefas podem receber pontuação com os valores: 1, 2, 3, 5 e 8. A pontuação representa o tempo necessário para o desenvolvimento da tarefa, lembrando que o Limites de Trabalho em Progresso limita a quantidade de tarefas que estão em processo de desenvolvimento simultaneamente, não a quantidade de tarefas que deve ser desenvolvida durante toda *sprint*. Para a *sprint* foi definida a seguinte relação:
 - 8 pontos: Uma semana e 3 dias a 2 semanas, ou seja uma, *sprint* completa;
 - 5 pontos: Uma semana há uma semana e 2 dias;
 - 3 pontos: 3 dias a uma semana;
 - 2 pontos: até 2 dias;
 - 1 ponto: alteração rápida, 1 dia.

- Melhoria Contínua: Ao final de cada *sprint* é realizado uma avaliação sobre o processo de desenvolvimento e analisados problemas e possíveis melhorias.

Figura 7 - Utilização do Quadro Kanban no software da Trello



Fonte: autoria própria

5 PROPOSTA DE SOLUÇÃO

Neste Capítulo é apresentada a solução proposta para esse trabalho. Como a metodologia utilizada para o desenvolvimento foram o *Personal Scrum* e Kanban, que são metodologias ágeis focadas na entrega, a documentação não foi o foco do desenvolvimento. Desta forma essa seção abordará apenas a parte prática do processo de desenvolvimento, elencando apenas as alterações nos requisitos e regras de negócio do sistema proposto.

5.1 TECNOLOGIAS UTILIZADAS

No desenvolvimento do sistema de planejamento acadêmico da UFERSA foram empregadas tecnologias modernas e eficazes. O TypeScript será a linguagem de programação escolhida. O Typescript pode ser utilizado tanto no React, Next e Node e além de tudo há uma familiaridade pessoal com a linguagem. O Next.js será o *framework* para o desenvolvimento do *front-end*, fornecendo recursos avançados de criação de interfaces de usuário web.

Para utilizar o Typescript e os *frameworks* mencionados foi escolhido o Node.js devido à sua eficiência e capacidade de lidar com múltiplas conexões simultâneas, o que é essencial para uma aplicação web em tempo real como a planejada para a UFERSA. Além disso, o Firebase será a solução de banco de dados, pois oferece várias funcionalidades prontas e bem definidas além de oferecer escalabilidade e armazenamento seguro de dados.

5.2 REFATORAÇÃO DOS REQUISITOS

Durante o planejamento do desenvolvimento e na análise detalhada do protótipo, foi identificada a necessidade de fazer algumas melhorias nos requisitos e nas regras de negócio do sistema. Nessa fase, foi feita uma validação dos requisitos propostos no estudo conduzido por Junior (2021) para aprimorar suas definições, incluir novos elementos e esclarecer regras de negócio que ainda estavam nebulosas.

O objetivo principal do nosso projeto é aperfeiçoar o planejamento dos semestres letivos. Embora o protótipo elaborado por Junior (2021) seja intuitivo e agradável, detectamos alguns desafios a serem superados. Por exemplo, ao examinarmos o modal apresentado na Figura 5, percebemos que ele carece de informações essenciais. Em particular, notamos a falta da opção de selecionar o dia da semana, o que é fundamental para o agendamento. Seria

bastante útil se o modal também fornecesse informações contextuais sobre a operação que está prestes a ser realizada, como detalhes do curso, período e semestre associados ao cadastro.

Outro ponto de melhoria é a gestão de disciplinas, que podem ter uma ou várias turmas. No modelo atual, não está claro como essa operação será executada ou se será viável na estrutura proposta. Apesar de Junior (2021) mencionar nos requisitos que as informações das disciplinas podem ser alteradas ou excluídas, o método exato para realizar essas ações não está claro no estudo ou no protótipo.

Além de se basear no protótipo de Junior (2021), este projeto também utilizou o Google Agenda como referência. O Google Agenda é uma poderosa ferramenta de gerenciamento de calendário oferecida pela Google como parte do conjunto de serviços do Google *Workspace*. O Google Agenda permite que os usuários organizem compromissos, eventos e tarefas de forma eficiente, sendo uma aplicação versátil usada nos âmbitos acadêmico, profissional e pessoal.

Embora o escopo do Google Agenda seja substancialmente maior do que o deste projeto, são destacados alguns aspectos que merecem atenção. Notavelmente, uma funcionalidade importante do Google Agenda é a abertura de um Modal com informações pré-preenchidas sempre que o usuário clica em qualquer parte da tabela. Ao clicar em um espaço vazio, o Modal exibe informações como horário, dia da semana e outros detalhes. Se o usuário seleciona um evento já agendado, o Modal apresenta todas as informações do evento, permitindo edições ou exclusões de informações. Essa abordagem pode ser adaptada a este projeto para melhorar a eficiência na criação de agendamentos. Em contraste, no protótipo de Junior (2021), para criar um agendamento, é necessário clicar em um botão "Adicionar" no canto superior direito da Figura 2, que é funcional, mas pode ser aprimorado com base na abordagem do Google Agenda.

De acordo com isso os novos requisitos ficaram da seguinte forma:

5.2.1 NOVOS REQUISITOS FUNCIONAIS

- RF001 - Na tela principal o sistema deve fornecer uma tabela de horários onde o usuário possa visualizar os agendamentos de aulas e horários vazios;
 - Esse é o ponto principal do sistema, é o motivo do sistema existir. Na tabela gerada os valores horizontais representam os dias da semana de segunda a sábado e os verticais representam os horários das aulas.

- RF002 - Na tela principal o sistema deve oferecer uma forma de entrada de dados para que o usuário repasse a informação de semestre, período e curso;
 - O sistema vai apresentar os horários de aula de acordo com as informações fornecidas.
- RF003 - O sistema só deve imprimir a tela quando o usuário informar o semestre, período e curso;
 - Ao selecionar essas informações o sistema faz a uma requisição ao Firebase e recebe os horários de aulas agendados apenas para o curso, período e semestre em questão.
- RF004 - Quando o usuário clicar em um espaço vazio ou ocupado, o sistema deve abrir um modal e preencher com algumas informações para que o usuário possa alterar, deletar ou criar um novo agendamento;
 - Esse foi um dos insights do Google Agenda. Nesse modal o usuário vai preencher as informações e criar/atualizar/deletar um agendamento. Nos casos onde o usuário clicar em um espaço vazio o sistema deve preencher as informações de horário, dia da semana, semestre, período e curso. Nos casos onde o usuário clicar em um horário agendado para uma aula o modal deve ser preenchido com todas as informações disponíveis sobre o agendamento.
- RF005 - No modal durante o cadastro/atualização, quando o usuário preencher as informações de professor, disciplina, sala, o sistema deve listar opções de acordo com os caracteres digitados e apenas aceitar o cadastro de uma das opções listadas;
 - Não há requisitos funcionais propostos no trabalho anterior sobre essa funcionalidade. Com essa funcionalidade o coordenador ou responsável pela criação dos horários terá que selecionar uma dessas opções que foram previamente cadastradas no sistema.
- RF006 - No modal o usuário deve escolher uma cor para o agendamento de aula na tabela;
 - No Gráfico 10, no apêndice A do trabalho do Junior é evidenciado que boa parte dos clientes utilizam cores para auxiliar o processo. Nesse caso, o sistema deve fornecer algumas opções de cores para que o usuário possa selecionar.
- RF007 - O sistema deve permitir cadastro de usuário;
 - Nas regras de negócio anteriores não estavam definidos requisitos funcionais para a criação de usuário.

- RF008 - O sistema deve permitir que o usuário apague um agendamento de aula;
- RF009 - O sistema deve permitir cadastro de professores;
 - Na **RF005** dos requisitos propostos pelo Junior (2021) os professores serão adicionados às disciplinas, mas adicionar para cada disciplina vários professores pode não ser eficiente o sistema necessita que os professores sejam armazenados de forma independente.
- RF010 - O sistema deve conter uma tela para listagem da carga horária atual de cada professor do semestre selecionado;
 - Descrição: O sistema deve listar a carga horária dos professores e nessa listagem, para cada professor, o sistema deve informar quais disciplinas o professor está cadastrado para ministrar no semestre, o código do horário , carga horária de cada disciplina e a carga horária total do professor no semestre.
- RF011 - O sistema deve permitir atualizar as informações dos professores;
- RF012 - O sistema deve permitir listagem dos professores cadastrados;
 - Em **RF008**, Junior (2021) sugere uma pesquisa de professores, mas no **RF005** dos novos requisitos já define uma pesquisa durante o cadastro e este define a listagem.
- RF013 - O sistema deve permitir exclusão dos professores;
 - Em **RF007**, Junior (2021) define a exclusão de professores de uma disciplina, mas como nos novos requisitos, em RF009, foi definido que os professores devem ser cadastrados de forma independente, o sistema deve permitir a exclusão destes professores.
- RF014 - O sistema deve permitir cadastro de disciplinas;
 - Requisito **RF001** proposto por Junior (2021).
- RF015 - O sistema deve permitir atualização de disciplinas;
 - Requisito **RF002** proposto por Junior (2021).
- RF016 - O sistema deve permitir exclusão de disciplinas;
 - Requisito **RF003** proposto por Junior (2021).
- RF017 - O sistema deve permitir a listagem de disciplinas;
 - Requisito **RF004** proposto por Junior (2021).
- RF018 - O sistema deve permitir a criação de salas de aula;
- RF019 - O sistema deve permitir exclusão de salas de aula;
- RF020 - O sistema deve permitir cadastro de salas de aula;

- RF021 - O sistema deve permitir listagem de salas de aula;
- RF021 - O sistema deve permitir cadastro de blocos de sala;
- RF023 - O sistema deve permitir listagem de blocos de sala;
- RF024 - O sistema deve permitir exclusão de blocos de sala;
- RF025 - O sistema deve notificar o usuário quando um ou mais professores não atingirem a carga horária ou ficarem abaixo da carga horária mínima;
 - **RF012** e **RF013** proposto por Junior (2021).
- RF026 - O sistema só deve permitir que emails liberados previamente consigam acessar os serviços.

5.2.2 NOVOS REQUISITOS NÃO FUNCIONAIS

- RNF001 - Usabilidade (Inteligibilidade);
 - Descrição: O sistema deve conter uma interface intuitiva, ou seja, de fácil uso para o usuário.
- RNF002 - Confiabilidade (Apreensibilidade);
 - Descrição: O sistema deve permitir a facilidade de aprendizagem para o usuário.
- RNF003 - Confiabilidade (Operabilidade);
 - Descrição: O sistema deve permitir que o usuário tenha o controle sobre as funcionalidades.
- RNF004 - Segurança (Confidencialidade);
 - O sistema deve autenticar os usuários.

5.2.3 REGRAS DE NEGÓCIO

- RN001 - As disciplinas podem ter mais de uma turma;
- RN002 - Pelo menos uma turma de cada disciplina de cada período deve estar disponível para os alunos de cada período no semestre;
- RN003 - Se uma disciplina possui mais de uma turma, nenhuma dessas turmas devem ter choque de horário com alguma disciplina que só possua apenas uma turma;
- RN004 - Se mais de uma disciplina possui mais de uma turma no período. Elas podem ter choque de horário se os números das turmas forem diferentes entre si;
 - Descrição: O sistema deve assegurar a prevenção de conflitos de horários para os usuários, embora exceções possam ocorrer. Por exemplo, o Cálculo 1 turma 1 não deve coincidir com a Geometria Analítica turma 1, pois, em teoria, as

mesmas turmas possuem os mesmos alunos em ambas as disciplinas. No entanto, é possível que o Cálculo 1 turma 1 coincida com a Geometria Analítica turma 2.

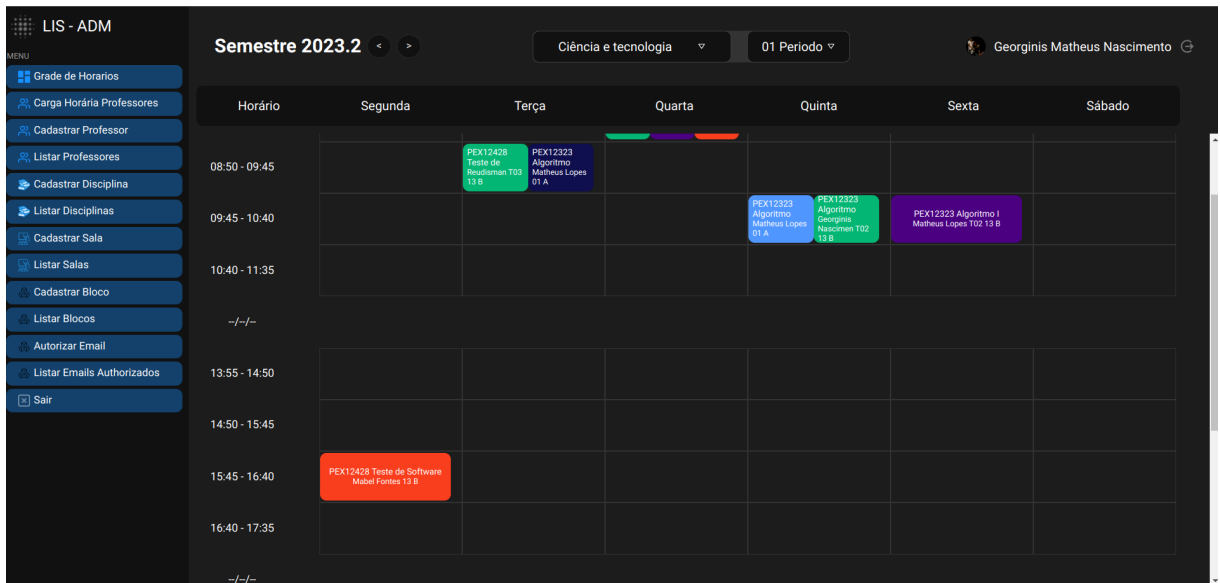
- RN005 - Mais de uma turma pode ser agendada para o mesmo horário desde que siga as regras anteriores;
- RN006 - Não deve ser possível cadastrar um professor para dois agendamentos no mesmo horário;
 - Descrição: Sempre que for cadastrar uma aula para um professor o sistema deve garantir que o professor não esteja cadastrado para outra aula no mesmo dia e horário.
- RN007 - Sempre que for cadastrar uma aula e atribuir uma sala o sistema deve garantir que a sala não esteja cadastrada para outra aula no mesmo dia e horário;

5.3 DESENVOLVIMENTO DA SOLUÇÃO PROPOSTA

Nessa seção são apresentados os resultados referentes ao desenvolvimento da interface proposta.

Na Figura 8 é apresentado o resultado da tela principal. Análogo ao tela proposta em Júnior (2021), essa tela apresenta a tabela das disciplinas dos semestres letivos. A partir dela é possível alterar e criar novos agendamentos e bem como navegar pelos semestres, cursos e períodos. O usuário pode clicar em qualquer lugar da tabela que estiver ocupada por um agendamento para alterá-lo ou clicar em uma parte vazia para criar um novo agendamento.

Figura 8: Tela principal do sistema



Fonte: autoria própria

Como em alguns casos os coordenadores necessitam criar mais de uma turma no mesmo horário, o sistema oferece uma maneira simples para realizar essa operação. Tomando como base o *layout* do Google Agenda, mesmo quando há um horário agendado para alguma disciplina, o sistema mantém uma parcela do horário na tabela vazia para que o usuário possa clicar para agendar uma nova aula para o mesmo horário. O tamanho de cada agendamento de aula no espaço é dividido pela quantidade de aulas agendadas para aquele horário. A Figura 9 a seguir mostra mais de perto como isso é mostrado na tabela.

Figura 9: Múltiplos agendamentos e espaço vazio para novo cadastro

		PEX15223 Matheus T04 13 B	PEX12428 Reudisma T02 13 B	PEX12428 Reudisma T01 01 A
PEX12428 Teste de Reudisman T03 13 B	PEX12323 Algoritmo Matheus Lopes 01 A			
			PEX12323 Algoritmo Matheus Lopes 01 A	PEX12323 Algoritmo Georginis Nascimen T02 13 B

Fonte: autoria própria

Já na Figura 10 é possível visualizar o modal de cadastro da aula. Na parte superior do modal, o sistema mostra o curso, semestre e o período em que o agendamento ou espaço vazio está vinculado. Mais abaixo o sistema fornece algumas opções de entrada como a disciplina, professor, sala, turma, horário, dia da semana e a cor que o agendamento terá na tabela de horários. No canto superior esquerdo do modal temos um botão para deletar o agendamento. Ele só aparece quando o usuário clica em um horário de aula.

Figura 10: Modal de cadastros de horários de aula

The image shows a web application interface for managing class schedules. A modal window titled "Adicionar disciplinas" is open, allowing the user to add a new class. The modal contains the following fields:

- Ciência e tecnologia** (selected)
- Semestre 2023.2** (selected)
- 01 Período** (selected)
- Disciplina**: PEX12323 Algoritmo I
- Professor**: Matheus Lopes
- Sala**: 01 A
- Turma**: T01
- Horário**: 08:50 - 09:45
- Dia da semana**: Terça

The background shows a calendar grid with various time slots and existing class entries. The modal has a "Delete" button and a "X" button in the top right corner. At the bottom of the modal, there are "Cancelar" and "Salvar" buttons. A row of colored dots is also visible above the "Cancelar" and "Salvar" buttons.

Fonte: autoria própria

Ainda no Modal, nos campos de disciplina, professor e sala, à medida que o usuário digita, o sistema mostra uma lista com os dados previamente cadastrados dessas entidades no sistema. Para cadastrar alguma informação nesses campos o usuário deve selecionar uma das opções que aparecem na lista para ser adicionado ao agendamento, se o usuário só digitar, mas não selecionar o valor não será salvo no Firebase. Esse comportamento do sistema é mostrado na Figura 11.

Figura 11: Lista de opções para o usuário selecionar

The image shows a dark-themed modal window titled "Adicionar disciplinas". It contains several input fields and a list of professor names. At the top right are "Delete" and "X" buttons. The form fields are: "Ciência e tecnologia" (category), "Semestre 2023.2" (semester), "01 Período" (period), "Disciplina" (course) with the value "PEX12323 Algoritmo I", "Professor" (teacher) with a dropdown menu showing "M", "Mabel Fontes", "Georginis Nascimento", "Reudisman", and "Matheus Lopes", "Sala" (room) with the value "01 A", "Turma" (class) with the value "T01", "Horário" (time) with the value "08:50 - 09:45", and "Dia da semana" (day of the week) with the value "Terça". At the bottom are "Cancelar" and "Salvar" buttons. A row of colored dots is visible above the buttons.

Fonte: autoria própria

Quando o usuário clica em salvar os dados são armazenados no Firebase, a tabela é atualizada e o novo agendamento de aula aparece.

Na Figura 12 é mostrada a tela da carga horária dos professores. Nessa tela é possível visualizar quais disciplinas cada um dos professores do semestre tem, qual a carga horária de cada uma delas, horário das aulas e qual a carga horária total do professor. Quando o valor está em vermelho, indica que o professor excedeu a carga horária máxima. Quando está em amarelo indica que o professor não atingiu a carga horária mínima.

Figura 12: Tela de carga horária dos professores

Doscente	Disciplina	Código	Carga Horária	Total
GeorGINIS Nascimento	Algoritmo I	2112 3134	4 h	20
	Teste de Software	3T12	4 h	
	Estrutura de Dados I	N2T25	4 h	
	Teste de Software	3T12	4 h	
	Algoritmo I	N2T25	4 h	
Matheus Lopes	Algoritmo I	M2T23	4 h	12
	Teste de Software	T2T24	4 h	
	Estrutura de Dados I	N2T25	4 h	
Mabel Fontes	Algoritmo I	M2T23	4 h	8
	Estrutura de Dados I	T2T24	4 h	

Fonte: autoria própria

Na Figura 13 tem-se uma tela básica utilizada para preencher os dados necessários para o funcionamento do sistema. As abas de cadastro de disciplinas, professores, sala, bloco e entre outros utilizam o mesmo componente e se alteram apenas na quantidade de campos de entrada e na função que é executada quando o usuário clica no botão de salvar. O usuário preenche as informações, clica em salvar, o sistema salva as informações no Firebase e limpa os campos novamente. Devido ao processo e funcionamento ser praticamente os mesmos para todas as telas de cadastro e como esse não é o foco do estudo, mostraremos apenas uma delas.

Figura 13: Tela de cadastro de disciplinas

LIS - ADM

MENU

- Grade de Horarios
- Carga Horária Professores
- Cadastrar Professor
- Listar Professores
- Cadastrar Disciplina
- Listar Disciplinas
- Cadastrar Sala
- Listar Salas
- Cadastrar Bloco
- Listar Blocos
- Autorizar Email
- Listar Emails Autorizados
- Sair

Cadastrar Disciplina

Nome
Insira o nome da disciplina

Créditos
Insira a quantidade de créditos

Salvar

Georginis Matheus Nascimento

Fonte: autoria própria

De forma parecida tem-se as telas de listagem mostrada na Figura 14, que tem o funcionamento praticamente igual para todas as listagens. Nessas telas os usuário pode listar os dados de alguma das entidades do sistema. No caso das disciplinas a listagem mostra o nome das disciplinas, os créditos de cada uma e mais a direita um botão, em que o usuário pode deletar o registro específico.

Figura 14: Listagem de disciplinas

LIS - ADM

MENU

- Grade de Horarios
- Carga Horária Professores
- Cadastrar Professor
- Listar Professores
- Cadastrar Disciplina
- Listar Disciplinas
- Cadastrar Sala
- Listar Salas
- Cadastrar Bloco
- Listar Blocos
- Autorizar Email
- Listar Emails Autorizados
- Sair

Disciplinas

Nome	Créditos	
PEX12323 Algoritmo I	4	X
PEX12428 Teste de Software	4	X
PEX15223 Estrutura de Dados I	4	X

Georginis Matheus Nascimento

Fonte: autoria própria

6 CONSIDERAÇÕES FINAIS

Ao longo deste estudo, foram explorados diversos aspectos relevantes para o desenvolvimento do projeto para a UFERSA que visa aprimorar o planejamento dos semestres letivos.

Foram utilizadas metodologias ágeis, que desempenharam um papel fundamental no desenvolvimento deste projeto. No contexto específico deste projeto foi destacado o uso do Scrum e do Kanban para o gerenciamento de tarefas e o desenvolvimento iterativo. Devido a flexibilidade, adaptabilidade sempre que havia alguma mudança nos requisitos isso era facilmente mapeado e refatorado e com o foco na entrega de valor os requisitos foram todos alcançados.

Também é importante ressaltar a importância do estudo prévio realizado na mesma universidade que levantou requisitos, desenvolveu *design* da base a ser utilizada no front-end do projeto. Esse estudo forneceu uma base sólida para o desenvolvimento subsequente.

Em resumo, este estudo abordou uma variedade de tópicos importantes para o desenvolvimento de softwares. Apesar de durante o processo de desenvolvimento do projeto uma fonte de serviços como o Firebase ter sido implementada para persistir os dados. Pode-se destacar a importância de se ter um *back-end* mais robusto que possa processar e relacionar os dados de forma mais assertiva. O serviço criado no Firebase serviu para que fosse possível utilizar o sistema ao final deste projeto.

É esperado que o sistema facilite o processo de criação dos semestres e reduza o máximo possível problemas relacionados a isso.

7 REFERÊNCIAS

BECK, K.; BEEDLE, M.; VAN BENNEKUM, A.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. **Manifesto para Desenvolvimento Ágil de Software**. 2001. Disponível em <<https://agilemanifesto.org/>>. Acesso em 01 de outubro de 2023

JÚNIOR, José Edglê Santana de Andrade. **Proposta de Design para um Ambiente de Gestão de Horários Acadêmicos**. Trabalho de Conclusão de Curso. Universidade Federal Rural do Semi-Árido (UFERSA). 2021.

CARDOSO, Marcelo de Oliveira. **Indústria 4.0: A Quarta Revolução Industrial**. 2016. Monografia de Especialização, Curso de Especialização em Automação Industrial, Departamento Acadêmico de Eletrônica, Diretoria de Pesquisa e Pós-Graduação, Universidade Tecnológica Federal do Paraná, Curitiba, 2016.

FILHO, Clézio Fonseca. **História da computação : O Caminho do Pensamento e da Tecnologia**. Porto Alegre : EDIPUCRS, 2007

GOMES, A. F. **Agile Desenvolvimento de software com entregas frequentes e foco no valor do negócio**. São Paulo: Casa do Código, 2013.

MAX, B. R.; PRESSMAN, R. S. **Engenharia de Software uma abordagem profissional**. 8. Ed. São Paulo: AMGH, 2016.

SOMMERVILLE, I. **Engenharia de Software**, São Paulo: Pearson Education, 2011.

COSTA, K.R.N. **Personal Scrum: Uma Alternativa Ágil para Desenvolvimento de Indie Games**. Alegrete, 2016

React - A biblioteca para web e interfaces de usuário nativas, React.js, 2023. Disponível em: <https://pt-br.react.dev/>. Acesso em: 05 out. 2023

SOBRE NODE.JS, Node.js, 2023. Disponível em: <https://nodejs.org/pt-br/about/>. Acesso em: 05 out. 2023

WHAT IS NEXT.JS, Next.js, 2023. Disponível em: <https://nextjs.org/learn/foundations/about-nextjs/what-is-nextjs>. Acesso em: 05 out. 2023

O QUE É JAVASCRIPT?, Amazon Aws, 2023. Disponível em:
<https://aws.amazon.com/pt/what-is/javascript/>. Acesso em: 05 out. 2023

QUAL É A DIFERENÇA ENTRE FRONT-END E BACK-END NO DESENVOLVIMENTO DE APLICAÇÕES?. Amazon AWS, 2023. Disponível em:
<https://aws.amazon.com/pt/compare/the-difference-between-frontend-and-backend/>. Acesso em 16 out. 2023.

O QUE É BANCO DE DADOS?. Amazon AWS, 2023. Disponível em:
<https://aws.amazon.com/pt/what-is/database/>. Acesso em 16 out. 2023.

TYPESCRIPT IS JAVASCRIPT WITH SYNTAX FOR TYPES, Typescript, 2023.
Disponível em: <https://www.typescriptlang.org/>. Acesso em 05 out. 2023

O que você pode fazer no Agenda, Google, 2023. Disponível em:
<https://support.google.com/a/users/answer/9302892?hl=pt-BR>. Acesso em 05 out. 2023

FIREBASE | GOOGLE'S MOBILE AND WEB APP DEVELOPMENT PLATFORM, Google, 2023. Disponível em: <https://firebase.google.com/>. Acesso em 17 out. 2023

APÊNDICE A - PESQUISA COM PROFISSIONAIS ASSOCIADOS À GERAÇÃO DE HORÁRIOS NA UFERSA, CAMPUS PAU DOS FERROS

Neste apêndice pode ser vista a pesquisa realizada por Júnior (2021), com vista a elaborar proposta de design para sistema de geração de horários.

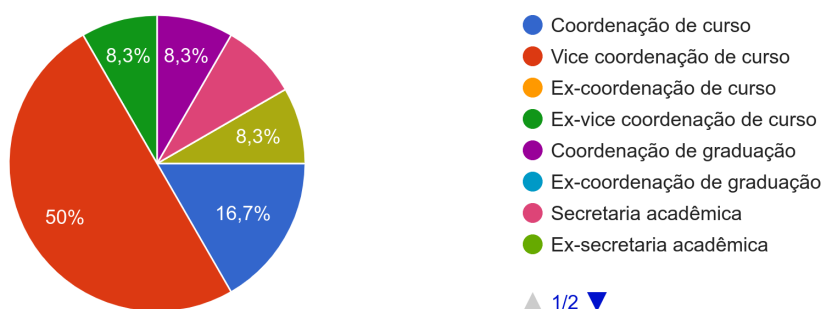
4.1 CARACTERIZAÇÃO GERAL DOS PARTICIPANTES

A maioria dos profissionais que participaram da pesquisa foram vice-coordenadores de curso (50%), conforme pode ser visto no Gráfico 1.

Gráfico 1 - Caracterização atual dos participantes.

Como você se caracteriza atualmente?

12 respostas

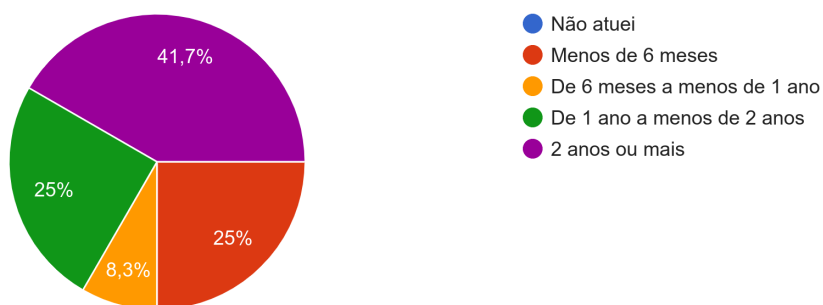


Fonte: Júnior (2021).

Sobre o tempo de atuação nas funções, a maioria informou ter 2 anos ou mais de atuação profissional nessas funções (41,7%), conforme pode ser visto no Gráfico 2.

Gráfico 2: A quanto tempo os participantes atuam na função

Na opção informada anteriormente, a quanto tempo realizou ou realiza a atuação profissional?
12 respostas

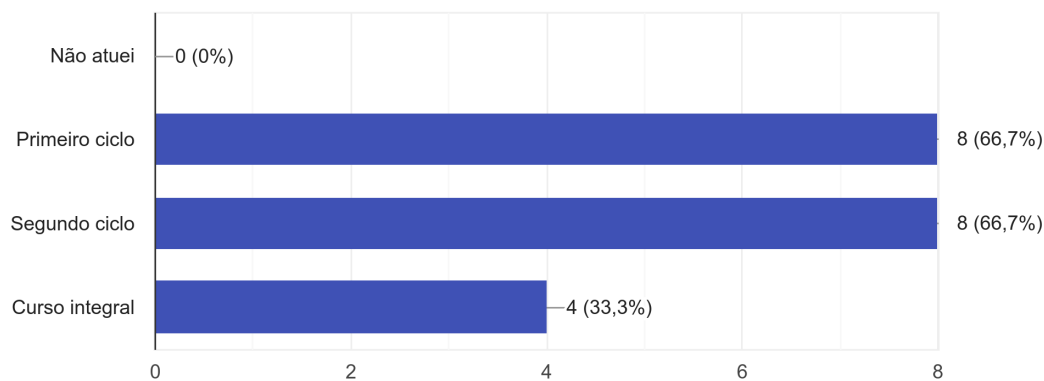


Fonte: Júnior (2021).

A maioria dos respondentes foram provenientes de profissionais relacionados a cursos de primeiro e segundo ciclo (Gráfico 3).

Gráfico 3: Tipos de cursos dos participantes

A sua atuação profissional afeta quais categorias de cursos?
12 respostas



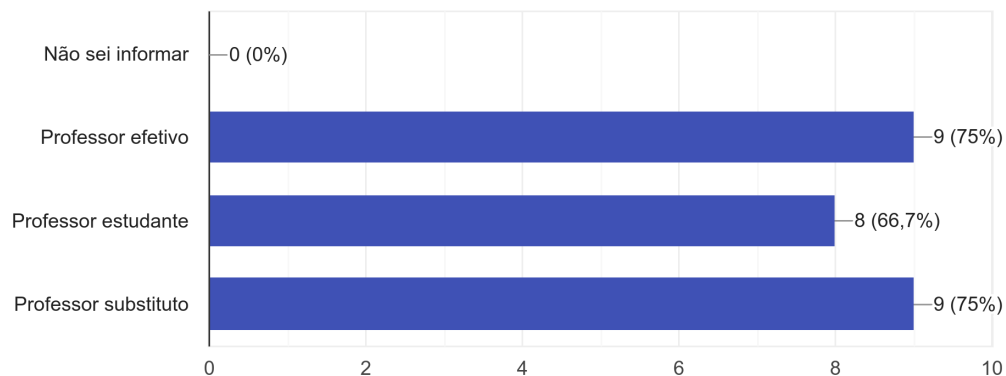
Fonte: Júnior (2021)

Segundo os resultados apontados em Júnior (2021), o tipo de professores que mais afetam os horários são os professores efetivos e substitutos (Gráfico 4).

Gráfico 4: categorias de docentes que impactam a geração de horários

Existem aspectos relacionados às categorias de docentes que impactam na construção da grade de horários? Se sim, quais às categorias de docentes?

12 respostas



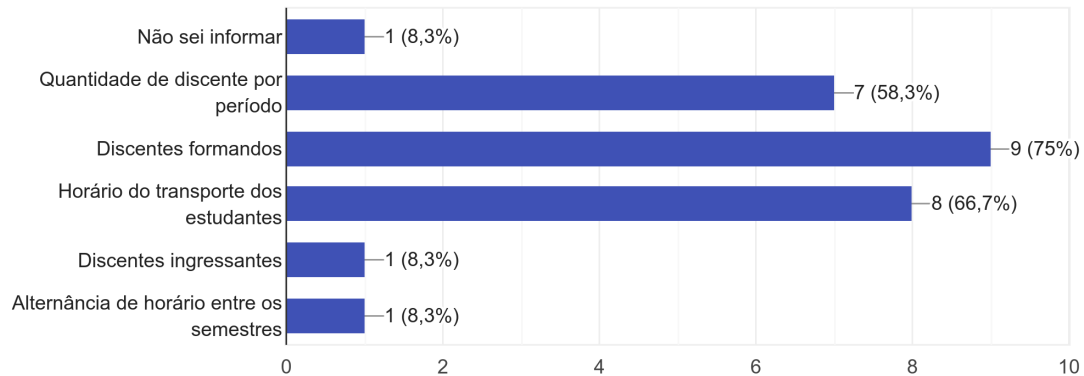
Fonte: Júnior (2021).

4.2 CARACTERIZAÇÃO GERAL DA GERAÇÃO DE HORÁRIOS

Os respondentes da pesquisa informaram que os aspectos relacionados aos discentes que mais afetam a geração de horários são os discentes formandos (75%), o horário do transporte dos estudantes (66,7%), a quantidade de discentes por período (58,3%), conforme Gráfico 5.

Gráfico 5: aspectos relacionados aos discentes que afetam a geração de horário

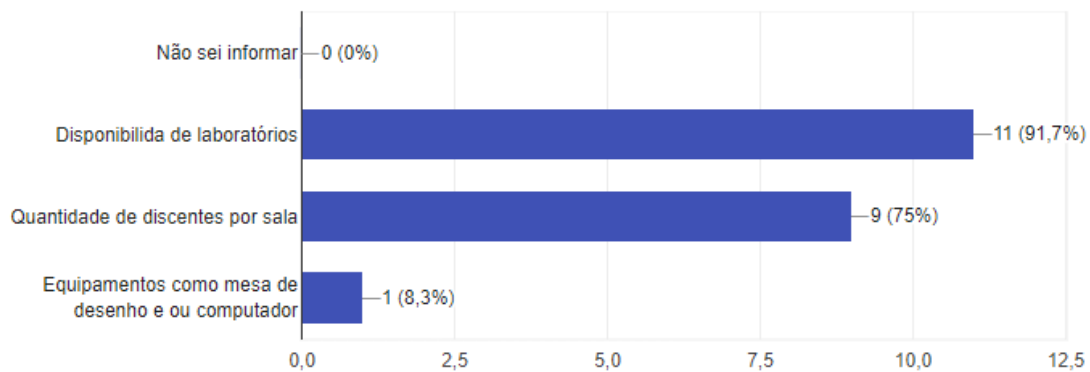
Existem aspectos relacionados aos discentes que impactam na construção da grade de horários?
12 respostas



Fonte: Júnior (2021).

Referente a infraestrutura, o aspecto que mais influência dos horários acadêmicos é a disponibilidade de laboratórios (Gráfico 6).

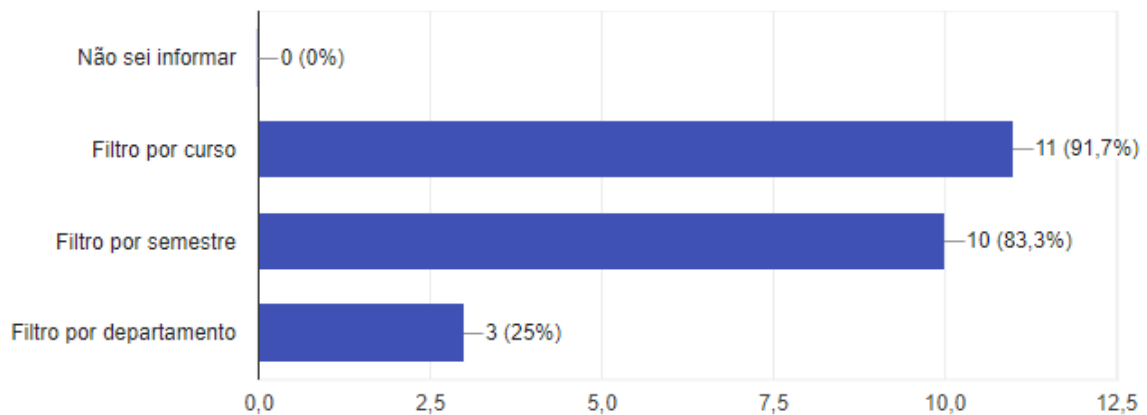
Gráfico 6: aspectos do tipo de sala que influenciam a geração de horários.



Fonte: Júnior (2021).

O filtro que mais pode influenciar o trabalho dos coordenadores, segundo os respondentes, é o filtro por curso (Gráfico 7).

Gráfico 7: tipos de filtros aplicados na geração da grade de horários.

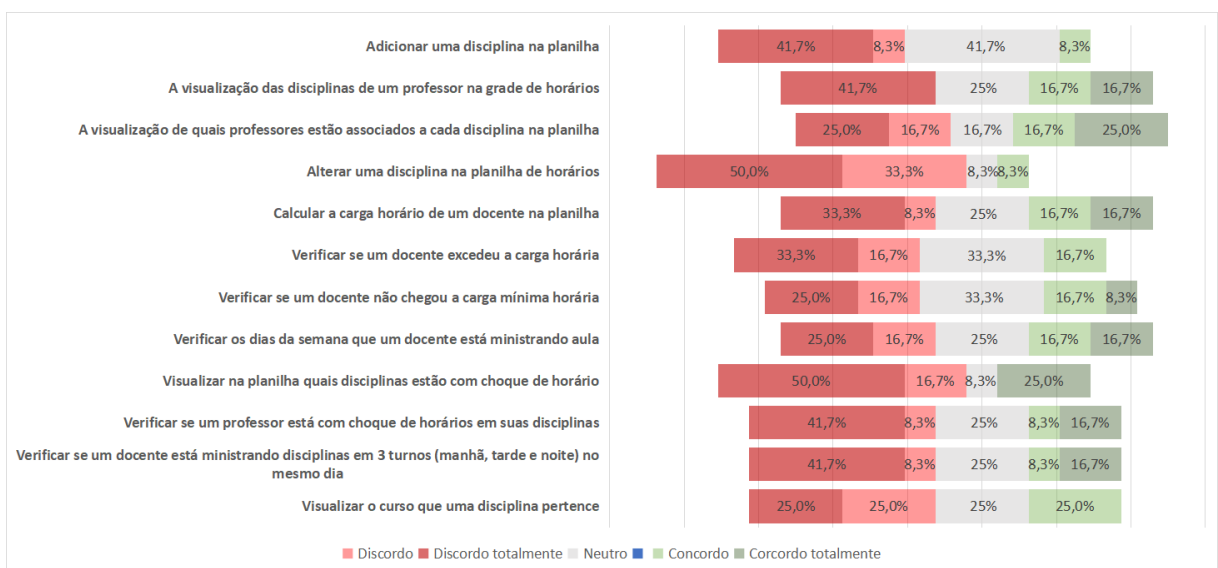


Fonte: Júnior (2021).

4.3 CARACTERIZAÇÃO GERAL SOBRE A FACILIDADE DE GERAÇÃO DE HORÁRIOS

Diversos fatores influenciam na facilidade da geração de horários, conforme pode ser visto no Gráfico 8, em que as áreas mais à esquerda demonstram discordância com a afirmação de facilidade e as áreas coloridas mais à direita representam concordância.

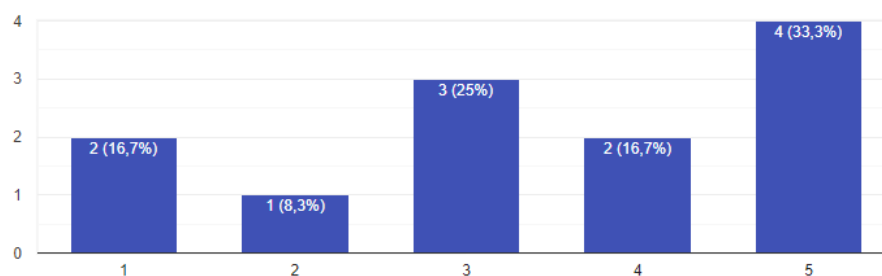
Gráfico 8: facilidades e dificuldades apresentadas na geração de horários.



Fonte: Júnior (2021).

Conforme a suficiência das planilhas no processo de geração de horários, as respostas podem ser vistas no Gráfico 9, sendo a opção 1 referente a discordância total e o valor 5 associado à concordância total.

Gráfico 9: opinião dos participantes sobre o uso de planilhas para gerar horários.



Fonte: Júnior (2021).

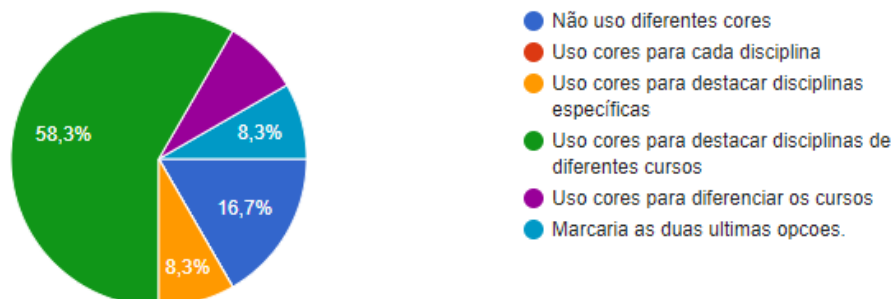
4.4 CARACTERÍSTICAS GERAIS SOBRE A APARÊNCIA DA PLANILHA DE HORÁRIOS

Também foram levantados aspectos visuais que influenciam na geração dos horários, conforme Gráficos 10 e 11.

Gráfico 10: uso de cores no processo de gerar horários.

Sobre o uso cores para destacar alguns itens na planilha de horários, selecione a opção que melhor o descreve.

12 respostas



Fonte: Júnior (2021).

Gráfico 11: uso de fontes para destacar itens na planilha de horários.

Sobre o uso fontes para destacar alguns itens na planilha de horários, selecione a opção que melhor o descreve.

12 respostas



Fonte: Júnior (2021).