

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE
DO NORTE

EDUARDO RIEV DA SILVA OLIVEIRA

MODELAGEM DO SISTEMA - O'TIME

NATAL

2022

EDUARDO RIEV DA SILVA OLIVEIRA

MODELAGEM DO SISTEMA - O'TIME

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo.

Orientador: Prof. Esp. Daniel Walmir dos Santos.

NATAL

2022

Oliveira, Eduardo Riev da Silva.
O48m Modelagem do sistema - O'Time/Eduardo Riev da Silva Oliveira. –
2022.
27 f.: il. Color.

Trabalho de Conclusão de Curso (graduação) – Instituto Federal de
Educação, Ciência e Tecnologia do Rio Grande do Norte, Natal, 2022.
Orientador: Prof. Esp. Daniel Walmir dos Santos Alves.

1. Modelagem de sistema web – Simulação de sistema. 2. Gestão
de horário. 3. SCRUM – Gerenciamento de projetos. I. Título.

CDU: 004.414.23

Catálogo na Publicação elaborada pela Bibliotecária Maria Ilza da Costa – CRB-15/412

Biblioteca Central Sebastião Fernandes (BCSF) – IFRN

EDUARDO RIEV DA SILVA OLIVEIRA

MODELAGEM DO SISTEMA - O'TIME

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo.

Orientador: Prof. Esp. Daniel Walmir dos Santos.

Trabalho de Conclusão de Curso Aprovado em 24/08/2022 pela seguinte Banca Examinadora:

Prof. Esp. Daniel Walmir dos Santos Alves – Orientador
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Prof^ª. Ma. Alyana Canindé Macêdo de Barros – Banca Examinadora
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Prof. Esp. Paulo Eugênio da Costa Filho – Banca Examinadora
Avaliador Externo

AGRADECIMENTOS

A Deus, pois sem ele não seria possível ter chegado até aqui.

Aos familiares e amigos, pois sem o apoio deles, dificilmente teria ido tão longe.

Ao Prof. Esp. Daniel Walmir dos Santos Alves pela orientação, especialmente nessa etapa, apesar da agenda lotada, conseguia disponibilizar para prestar auxílio em vários momentos e sempre buscou formas de motivar seus orientandos.

Aos professores participantes da banca examinadora, Prof^ª. Ma. Alyana Canindé Macêdo de Barros e Prof. Esp. Paulo Eugênio da Costa Filho, pelo tempo e em especial destacando o cuidado e zelo que tiveram em suas anotações, pois suas sugestões foram essenciais para o trabalho.

Aos professores entrevistados durante o PDS Web, pelo tempo concedido e sinceridade durante as entrevistas, pois foram de extrema importância para a compreensão das problemáticas.

Aos colegas da turma do curso, Wallase, Astrogilda e Elvis, e até mesmo aqueles que desistiram do curso, como José, Daniel e Gabriel, proporcionando uma excelente experiência acadêmica, que certamente não será esquecida, principalmente por ajudarem em diversos momentos, gratidão pelo apoio.

RESUMO

O presente trabalho trata da modelagem de um sistema web de gestão de horários, o O'Time, desenvolvido em Vue.js e utilizando uma API Django REST Framework. O principal objetivo é realizar a modelagem de um sistema, capaz de apontar a progressão desde o desenvolvimento do projeto até a avaliação final. A fim de atingir tal proposta, o trabalho irá expor, em um contexto geral, a proposta que foi feita, como se deu o planejamento, as metodologias e ferramentas utilizadas durante cada etapa.

Palavras-chave: modelagem; gestão; sistemas web.

ABSTRACT

The present work deals with the modeling of a web time management system, O'Time, developed in Vue.js and using a Django REST Framework API. The main objective is to carry out the modeling of a system, capable of pointing out the progression from the development of the project to the final evaluation. In order to achieve this proposal, the work will expose, in a general context, the proposal that was made, how the planning took place, the methodologies and tools used during each stage.

Keywords: modeling; management; web systems.

LISTA DE SIGLAS

CNAT	Campus Natal-Central
DIATINF	Diretoria Acadêmica de Gestão e Tecnologia da Informação
IFRN	Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
PAAC	Processo Acadêmico Corporativo
PAAD	Processo Acadêmico Distribuído
PAAS	Processo Acadêmico Simplificado
PAAWEB	Processo Acadêmico Web
SUAP	Sistema Unificado de Administração Pública
TADS	Tecnologia em Análise e Desenvolvimento de Sistemas
UML	Linguagem de Modelagem Unificada

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVO GERAL	10
1.2	OBJETIVOS ESPECÍFICOS.....	10
1.3	METODOLOGIA CIENTÍFICA.....	10
1.4	ESTRUTURA DO TRABALHO.....	11
2	REFERENCIAL TEÓRICO	11
2.1	SCRUM.....	11
2.2	PAAS (PROCESSO ACADÊMICO SIMPLIFICADO).....	12
2.3	UML.....	13
3	ARQUITETURA E MODELAGEM DO O'TIME	14
3.1	REQUISITOS DO O'TIME.....	14
3.2	ARQUITETURA DO O'TIME.....	17
3.3	MÓDULOS DO SISTEMA.....	18
3.3.1	Sala de Aula	18
3.3.2	Disciplina	19
3.3.3	Turma	20
3.3.4	Professor	20
3.4	MODELAGEM DO PROTÓTIPO.....	20
4	RESULTADOS ALCANÇADOS	23
5	CONSIDERAÇÕES FINAIS	23
	REFERÊNCIAS	24

APÊNDICE A	25
-------------------------	----

1 INTRODUÇÃO

A DIATINF (Diretoria Acadêmica de Gestão e Tecnologia da Informação) é uma das diretorias do CNAT (Campus Natal-Central) do IFRN (Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte), e segundo dados extraídos do site oficial da diretoria (<https://sites.google.com/escolar.ifrn.edu.br/diatinf/a-diretoria>) no dia 13 de agosto de 2022, ela possui cerca de 55 professores, 453 alunos do nível técnico e mais 966 estudantes do nível superior, totalizando um quadro com 1484 indivíduos. Com tamanho contingente, a tarefa de gestão não é fácil, especialmente se tratando de horários e salas para gerenciar o efetivo de pessoas de diferentes cursos, disciplinas, turmas e professores sem que haja conflitos, o que se torna um problema ainda maior. Para solucionar essa problemática, foram organizadas comissões de horários para cada diretoria, sendo estas as principais responsáveis por recepcionar as demandas necessárias a fim de resolver esses conflitos e então trabalhar em prol do gerenciamento de horários de forma satisfatória.

De um ponto de vista técnico, se torna inviável realizar essa atividade manualmente, uma vez que quantidade de envolvidos e trabalho necessários podem acabar sendo maiores que o previsto, tornando o processo extremamente lento e aumentando as probabilidades do surgimento de erros diversos, a exemplo de: Montagem incorreta de horário; Alocação de salas já ocupadas; Remanejamento de disciplinas para horários nos quais os profissionais designados não estejam disponíveis; Mudanças repentinas de horários, entre outros.

Refletindo sobre os possíveis problemas que podem surgir, foi idealizado a automatização de todo o processo, evitando assim tais problemas existentes e que outros que possam surgir, facilitando na hora de planejar o contingenciamento, minimizando os danos causados por esse tipo de situação. Nesse contexto, uma das opções mais utilizadas no mercado são os *softwares* de gestão de horários escolares, a exemplo do *ASC TimeTables*, usado no IFRN-CNAT, que é capaz de gerar horários de forma automática, apresentando certas restrições, sejam elas de horários dos professores, ou da ocupação das salas, gerando um arquivo exportável dos horários, que podem ser usados como base na criação dos horários no sistema acadêmico, no caso do IFRN, o Sistema Unificado de Administração Pública (SUAP).

Conforme relatado (descrito no apêndice A) pelos professores da comissão de horários, por não existir uma forma de integração entre esses sistemas, os horários são inseridos manualmente por uma equipe da secretaria ou diretoria, acarretando em alguns problemas, como o fato que o *software* não ser colaborativo, permitindo apenas um único

usuário manipular os arquivos por vez, podendo gerar diversas inconsistências caso não haja comunicação direta entre as pessoas que estão responsáveis por essa tarefa. No ano de 2019, conforme relatos dos membros da comissão de horários da DIATINF, a solução encontrada foi a criação de um grupo no WhatsApp, para notificar os demais integrantes sempre que alterações fossem realizadas, e logo após, lançadas em sincronia com o serviço de nuvem *Dropbox*, evitando-se mexer no arquivo até que essas alterações fossem concluídas.

Pensando em como resolver esse problema, o desafio foi proposto durante a disciplina de Projeto de Desenvolvimento de Software (PDS) aos alunos do curso de Tecnologia em Análise e Desenvolvimento de Sistemas (TADS), onde o O'Time surgiu, sendo um sistema com a proposta de gerar os horários de forma simples, que também possibilite indicar as preferências de horários dos professores e a disponibilidade das salas, além de permitir a colaboração de várias pessoas ao mesmo tempo, eliminando a necessidade de meios de comunicação externos, e também que realize a integração direta ou ao menos gere um arquivo que possa ser importado diretamente para o SUAP.

1.1 OBJETIVO GERAL

Este trabalho tem como objetivo realizar a modelagem do sistema O'Time, através da criação de protótipos de tela com fundamentação em diagramas UML.

1.2 OBJETIVOS ESPECÍFICOS

- a) Realizar a modelagem do sistema O'Time;
- b) Apresentar os resultados alcançados através de protótipos de tela de alta fidelidade;

1.3 METODOLOGIA CIENTÍFICA

A metodologia científica deste trabalho tem como objetivo ser descritivo e tem como finalidade ser uma pesquisa aplicada, pois visa o desenvolvimento da modelagem de um sistema web de forma prática, utilizando o conhecimento adquirido durante a graduação, e voltado a solucionar as problemáticas trabalhadas durante as disciplinas de PDS.

A pesquisa aplicada é aquela em que o pesquisador é movido pela necessidade de conhecer para a aplicação imediata dos resultados. Contribui para fins práticos, visando à solução mais ou menos imediata do problema encontrado na realidade. Na pesquisa aplicada, o pesquisador busca orientação prática à solução imediata de problemas concretos do cotidiano. (BARROS; LEHFELD, 2014).

E para que fosse possível o desenvolvimento do trabalho, foi utilizado um processo adaptado de conceitos utilizados para modelagem e desenvolvimento de *softwares*.

1.4 ESTRUTURA DO TRABALHO

Este trabalho é composto de cinco capítulos. No primeiro temos uma breve introdução e contextualização do trabalho, apontando os objetivos que se deseja alcançar e a metodologia utilizada para o desenvolvimento e aplicação dos mesmos. O segundo capítulo aborda o referencial teórico, onde são expostos tópicos que serão importantes para uma melhor compreensão do trabalho. O terceiro capítulo trata da arquitetura do projeto, explicando quais as ferramentas usadas durante o desenvolvimento do projeto, as mudanças realizadas ao longo desse período e suas justificativas. No quarto capítulo, vemos uma análise dos resultados obtidos no decorrer de todo o processo até o final. Por fim, no quinto capítulo encontramos as considerações finais diante de tudo que foi apresentado.

2 REFERENCIAL TEÓRICO

Nesta seção será abordada a fundamentação teórica deste trabalho, abordando alguns dos principais conceitos a serem utilizados no decorrer do trabalho, de forma a melhorar o entendimento à medida que forem sendo abordados.

2.1 SCRUM

É um framework de gestão de projetos, sendo amplamente reconhecido como metodologia ágil, que consiste em dividir um projeto em ciclos de atividades (sprints), que normalmente duram cerca de um mês, com reuniões frequentes para que a equipe possa alinhar o que vem fazendo e pensar formas de melhorar o processo com agilidade, passando por mudanças de planejamento basicamente o tempo todo, conforme haja necessidade.

SUTHERLAND (2013) afirma que “O Scrum aplica uma abordagem iterativa e incremental para otimizar a previsibilidade e controlar o risco”, o que significa dizer que uma vez que o processo é iniciado, a cada nova etapa, sempre que algo for incrementado, devemos analisar os possíveis riscos de forma a antecipar e manter as situações desses riscos sob controle, de acordo com o autor “É esperado que um Time Scrum se adapte assim que aprender algo novo através da inspeção. “, simbolizando justamente que o time deva estar sempre aberto a mudanças e preparado caso o inesperado ocorra.

2.2 PAAS (PROCESSO ACADÊMICO SIMPLIFICADO)

É o modelo usado dentro do Projeto de Desenvolvimento de Software (PDS), que é o processo aplicado dentro dessa disciplina no curso de Tecnologia em Análise e Desenvolvimento de Sistemas (TADS), consistindo de três sub-processos que o compõem, sendo eles:

1. O PAAWEB (Processo Acadêmico Web), que é um processo baseado nas práticas do Processo Unificado (OpenUP) e da Modelagem Ágil, que tem por objetivo criar uma aplicação web que possa ser acessada por qualquer pessoa com o link e que os usuários possam utilizar suas funções;
2. O PAAD (Processo Acadêmico Distribuído), sendo este um processo iterativo e incremental para o projeto de Software Distribuído, que tem por objetivo desenvolver uma aplicação com o desacoplamento (separação) dos componentes de front-end e back-end do projeto da aplicação, gerando uma API que possa ser acessada externamente;
3. O PAAC (Processo Acadêmico Corporativo), que tem por objetivo lançar uma aplicação que possa ser lançada como uma espécie de build final, que a depender do objetivo da solução, pode simbolizar a etapa de lançamento do produto, até mesmo em formato mobile.

Cada um desses sub-processos é descrito em 4 fases, divididas entre iterações, de duração variável, contendo tarefas que visam cumprir certas metas estabelecidas para o desenvolvimento do projeto, sendo elas:

1. Concepção: O objetivo é estabelecer uma visão inicial para construção do sistema, onde serão produzidos um escopo, delimitando o que faz parte ou não da proposta, e nessa fase o time deve demonstrar que o projeto é viável de um ponto de vista de negócios, elencando alguns possíveis riscos.
2. Elaboração: Tem como objetivo analisar o domínio do problema, desenvolver o plano de projeto e eliminar os elementos que possam colocar o projeto em risco.
3. Construção: Visa finalizar os componentes considerados essenciais e as características principais da aplicação, dependendo da equipe determinar quais serão esses componentes, e então realizar a integração desses componentes,

que devem ser todos testados para garantir que estão funcionando de acordo com o planejado nas etapas anteriores.

4. Validação: O foco desta fase é assegurar que o software esteja disponível para seus usuários finais, atravessando várias iterações e incluindo testes no produto em preparação para release e ajustes pequenos com base no feedback do usuário.

Cada fase tem um conjunto de artefatos que devem ser produzidos, entregues e apresentados ao final da iteração, onde as partes interessadas se encontram e discutem acerca do que foi produzido, sugerindo melhorias e soluções para os problemas enfrentados até aquele ponto.

2.3 UML (Linguagem de Modelagem Unificada)

BOOCH (2006) afirma que: “Construímos modelos para compreender melhor o sistema que estamos desenvolvendo.”, e nesse aspecto, a modelagem nos ajuda a observar mais facilmente pontos passíveis de melhorias, refinamentos e otimizações, uma vez que simplifica toda a abstração documentada de forma mais visível, mais próxima da realidade quanto possível, para que ambas partes interessadas tenham uma melhor compreensão do que se espera atingir durante o desenvolvimento, e se necessário, aplicar correções, modificar ou até remover alguns aspectos, ficando a critério da equipe desenvolvedora juntamente com o cliente decidir o que será ou não mantido no projeto.

Ainda de acordo com o autor, os modelos devem alcançar 4 objetivos após a modelagem do sistema:

1. Visualizar o sistema como é ou se deseja que seja;
2. Especificar a estrutura e comportamento do sistema;
3. Proporcionar um guia para a construção do sistema;
4. Documentar decisões tomadas;

Esses objetivos devem estar em consonância com o que for desenvolvido, uma vez que estabelecem que a modelagem ditará os próximos passos a serem seguidos nas etapas que virão até que a construção da solução seja finalizada, ressaltando que sistemas mais complexos não podem ser entendidos em sua totalidade, então buscamos como resultado entender o básico e o implementar aos poucos, buscando sempre atingir as metas que foram estabelecidas anteriormente pelas partes interessadas.

3 ARQUITETURA E MODELAGEM DO O'TIME

O O'Time tem a proposta de ser um sistema web de montagem de horários escolares, que permita aos gestores, pessoas responsáveis que utilizarão a plataforma, serem capazes de criarem horários de forma simples, e que possibilite a colaboração entre eles, também sendo capaz de realizar integrações com softwares externos e permita a criação de um arquivo que possa ser exportado.

3.1 REQUISITOS DO O'TIME

O glossário de engenharia de software do IEEE (IEE N° 90) define requisito como “Uma condição ou capacidade necessitada por um usuário para resolver um problema ou alcançar um objetivo.” Ou seja, entende-se por requisito quaisquer necessidades que sejam identificadas, e comprovadas como fundamentais para que a solução seja desenvolvida dentro de acordo com os interesses das partes envolvidas, e nesse aspecto, o O'Time apresenta atualmente os seguintes requisitos:

Requisitos Funcionais			
Cód.	Nome	Descrição	Categoria
F01	Cadastro de Usuários	O sistema deve permitir que os usuários se cadastrem no mesmo.	Evidente
F02	Login	O sistema deve permitir que os usuários entrem no sistema usando suas credenciais	Evidente
F03	Recuperação de Senha	O sistema deve permitir que os usuários recuperem suas senhas, se necessário.	Evidente
F04	Definição nos <i>Slots</i> de Horários	O sistema deve permitir que gestores definam <i>slots</i> de horários para as turmas cadastradas.	Evidente
F05	Alteração nos <i>Slots</i> de Horários	O sistema deve permitir que gestores alterem <i>slots</i> de horários já definidos, podendo excluí-los se necessário.	Evidente
F06	Notificação de Mudanças de Horários	O sistema deve notificar os usuários de mudanças nos horários.	Evidente
F07	Cadastro de Disponibilidade de Horários	O sistema deve permitir o cadastro de horários de disponibilidade dos professores.	Evidente
F08	Alteração de	O sistema deve permitir a alteração de	Evidente

	Disponibilidade de Horários	horários de disponibilidade dos professores, podendo excluí-los se necessário.	
F09	Visualização de Horários Cadastrados	O sistema deve permitir aos gestores que visualizem os horários cadastrados por eles.	Evidente
F10	Solicitação de Mudanças de Horário	O sistema deve permitir aos professores solicitarem aos gestores mudanças de horários.	Evidente
F11	Cadastro de Diretorias	O sistema deve permitir ao diretor cadastrar diretorias.	Evidente
F12	Alteração de Diretorias	O sistema deve permitir que o diretor altere as diretorias cadastradas, podendo excluí-las se necessário.	Evidente
F13	Listagem de Diretorias	O sistema deve permitir a listagem de diretorias cadastradas.	Evidente
F14	Cadastro de Gestores	O sistema deve permitir ao diretor cadastrar gestores.	Evidente
F15	Alteração de Gestores	O sistema deve permitir ao diretor alterar gestores cadastrados, podendo excluí-los se necessário.	Evidente
F16	Listagem de Gestores	O sistema deve permitir a listagem de gestores cadastrados.	Evidente
F17	Cadastro de Salas	O sistema deve permitir ao diretor cadastrar salas.	Evidente
F18	Alteração de Salas	O sistema deve permitir ao diretor alterar as salas cadastradas, podendo excluí-las se necessário.	Evidente
F19	Listagem de Salas	O sistema deve permitir a listagem de salas cadastradas.	Evidente
F20	Verificação de Disponibilidade de Salas	O sistema deve permitir a verificação da disponibilidade de salas.	Evidente
F21	Cadastro de Disciplinas	O sistema deve permitir ao gestor cadastrar disciplinas.	Evidente
F22	Alteração de Disciplinas	O sistema deve permitir ao gestor alterar disciplinas cadastradas, podendo excluí-las se necessário.	Evidente
F23	Listagem de	O sistema deve permitir a listagem de cursos	Evidente

	Disciplinas	cadastrados.	
F24	Cadastro de Cursos	O sistema deve permitir ao gestor cadastrar cursos.	Evidente
F25	Alteração de Cursos	O sistema deve permitir ao gestor alterar cursos cadastrados, podendo excluí-los se necessário.	Evidente
F26	Listagem de Cursos	O sistema deve permitir a listagem de cursos cadastrados.	Evidente
F27	Cadastro de Turmas	O sistema deve permitir ao gestor cadastrar turmas.	Evidente
F28	Alteração de Turmas	O sistema deve permitir ao gestor alterar turmas cadastradas, podendo excluí-las se necessário.	Evidente
F29	Listagem de Turmas	O sistema deve permitir a listagem de turmas cadastradas.	Evidente
F30	Cadastro nos Slots de Horários	O sistema deve permitir ao gestor definir os slots de horário de uma turma já cadastrada.	Evidente
F31	Alteração nos Slots de Horários	O sistema deve permitir ao gestor alterar os slots de horário de uma turma já cadastrada, podendo excluí-los se necessário.	Evidente
F32	Visualização dos Slots de Horários	O sistema deve permitir ao professor visualizar slots de horário nos quais ele tem aula.	Evidente
F33	Cadastro de Turmas	O sistema deve permitir ao gestor cadastrar turmas.	Evidente

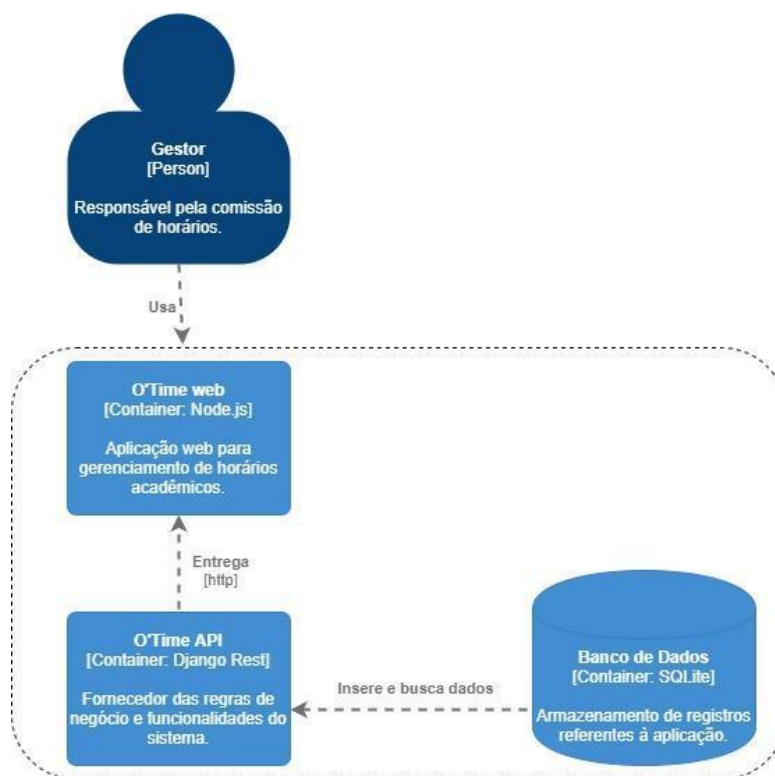
Requisitos Não-Funcionais				
Cód.	Nome	Descrição	Categoria	Obrigatoriedade
NF01	Interface Web	Deve funcionar em uma plataforma web.	Interface	Obrigatório
NF02	Interface Mobile	Deve funcionar em uma plataforma mobile.	Interface	Desejável
NF03	Tecnologias de Desenvolvimento	Será desenvolvido usando o Framework Django, Python, HTML5, JavaScript e CSS.	Implementação	Obrigatório

Requisitos Não-Funcionais				
Cód.	Nome	Descrição	Categoria	Obrigatoriedade
NF01	Interface Web	Deve funcionar em uma plataforma web.	Interface	Obrigatório
NF04	Restrição de horários	Permitir que apenas gestores modifiquem os horários.	Especificação	Obrigatório
NF05	Exportação do arquivo de horários	O sistema deve permitir aos usuários exportar o arquivo de horários para outros formatos.	Especificação	Desejável

3.2 ARQUITETURA DO O'TIME

Quando falamos em modelagem, é preciso entender primeiramente como o sistema funcionaria, uma visão arquitetural, e nesse sentido, o diagrama de container auxilia no entendimento, uma vez que apresenta essa ideia de forma mais simplificada, como pode ser visto na figura 1, onde temos uma explicação da funcionalidade de acesso à plataforma web:

Figura 1 - Diagrama de Container

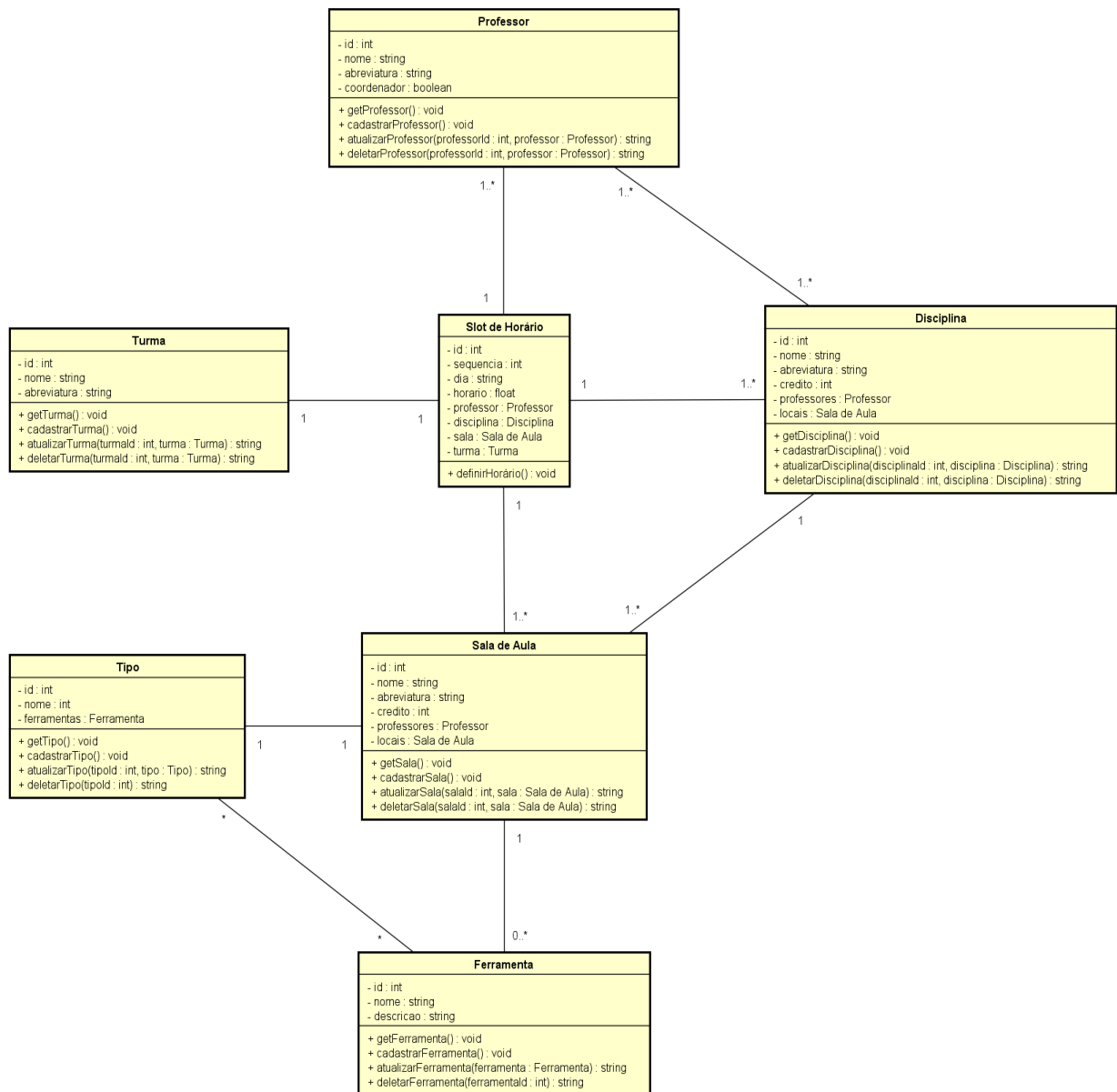


Fonte: Elaborado por Fernanda, integrante do O'Time (2021)

3.3 MÓDULOS DO SISTEMA

O sistema está dividido em quatro módulos principais, que representam as entidades que são integradas durante a montagem de horários, tais são os módulos: Sala de Aula, Disciplina, Turma e Professor, representados relacionalmente na figura 2, com as classes adjuntas Tipo e Ferramentas, que apenas complementam a entidade Sala de Aula, enquanto a entidade Slot de Horário representa a junção de todas, já que um horário é feito a partir da união dessas 4 entidades.

Figura 2 - Diagrama de Classes



Fonte: Elaborado pelo autor (2022)

Cada um deles representa as entidades que juntas irão servir para formar o horário, cada qual contendo algumas atividades relacionadas, como cadastro, listagem, alteração e remoção de dados, que são apresentadas no sistema em forma de casos de uso (CDUs), entrando em funcionamento seguindo o fluxo descrito no CDU e obtendo sucesso ao final, caso todos os passos descritos sejam obedecidos. Cada módulo será abordado mais detalhadamente em sua respectiva seção.

3.3.1 Sala de Aula

O módulo sala de aula é o mais complexo do sistema, sendo ele quem compõe principalmente áreas relativas ao espaço físico da instituição, que serão utilizados durante a criação dos horários para identificar a disponibilidade daquela sala durante determinado período de tempo. Esse módulo contém três diretivas que compõem sua estrutura, sendo de suma importância para sua execução, uma vez que são elas que determinam quais as atividades que podem ser feitas, sendo elas:

1. Sala de Aula: Contém uma descrição da sala, contendo nome, uma abreviatura, qual o tipo da sala e a capacidade numérica de pessoas que a sala comporta;
2. Tipos: Contém uma classificação por tipo de sala, contendo um nome e as ferramentas que uma sala daquele tipo específico deve ter;
3. Ferramentas: Contém a descrição e a identificação por nome de cada uma das ferramentas disponíveis em cada sala.

Levando em consideração as funcionalidades, este módulo engloba as funções de cadastro, alteração, remoção e visualização das entidades sala, tipo e ferramentas, que serão utilizadas dentro da criação de horários para estabelecer o espaço onde determinada aula irá ocorrer e dentro do banco de dados servirá como identificador caso algum erro precise ser corrigido nesse caso específico. Cada sala, tipo e ferramenta são identificados por um ID gerado automaticamente, acessível apenas para o administrador dentro da interface API do banco de dados, e um nome, dado como variável, podendo ser alterado sempre que o usuário desejar.

3.3.2 Disciplina

O módulo disciplina está relacionado diretamente aos módulos de professor e sala de aula, que funcionam como pré-requisito para a criação deste, uma vez que a disciplina só pode existir se houver um professor a quem possamos atribuir a responsabilidade por dar a aula e uma sala onde ele possa exercer tal atividade. Por outro lado, esse módulo tem uma estrutura mais simples, contendo apenas a seção para cadastro, alteração, visualização e remoção das disciplinas, que são exibidas numa lista e identificadas por ID único criado automaticamente e um nome armazenado numa variável.

3.3.3 Turma

O módulo turma é o mais simples do sistema, uma vez que conta apenas com um nome e um turno para sua identificação, semelhante ao módulo disciplina, também possui uma seção para cadastro, alteração, visualização e remoção das turmas cadastradas. Ele se destaca por ser o único que não se relaciona e nem possui dependências de qualquer tipo com outras entidades, funcionando independentemente de estarem cadastradas previamente ou não.

3.3.4 Professor

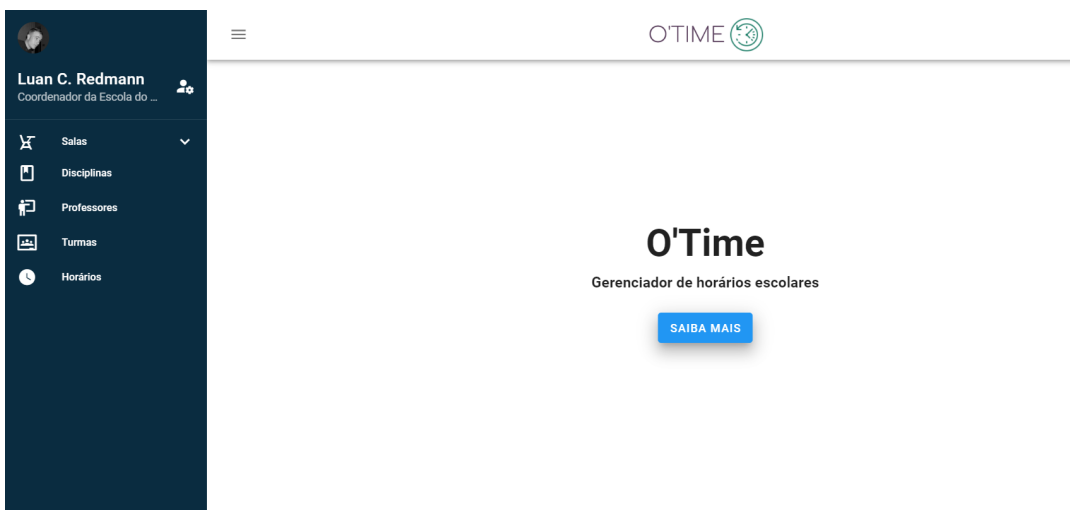
O módulo professor é outro que também possui independência em relação aos outros, sendo necessário apenas o cadastro de um professor para executar suas outras tarefas, que são visualização, alteração e remoção da entidade professor, sendo identificado por ID, nome e uma checkbox indicando se é coordenador do curso ou não, também possuindo um relacionamento um para muitos com a entidade disciplina, pois um professor pode lecionar em várias disciplinas.

3.4 MODELAGEM DO PROTÓTIPO

O protótipo foi construído em Vue.js, tendo como base a imagem idealizada de como seria o sistema, utilizando o sistema de componentes do Vue, foram criadas estruturas estilizadas no Vuetify, aproveitando de elementos visuais já presentes na versão antiga, feita em BootStrap 4, como checkbox, navbar, listas e elementos flutuantes. Já os módulos foram tratados em seções separadas, como se vê na figura 2, onde são a disposição os apresenta em forma de itens listados na barra lateral, onde é possível acessá-los e realizar as operações padrão (cadastro, atualização, remoção e visualização dos dados). Já a relação entre os

módulos foi feita na forma de pré-requisitos, a exemplo de disciplinas, que exige que haja ao menos uma sala e um professor para que possa ser cadastrada no sistema, sendo necessário que seja feito o cadastro dessas informações antes que se possa realizar a criação de uma nova disciplina.

Figura 2 - Tela Inicial



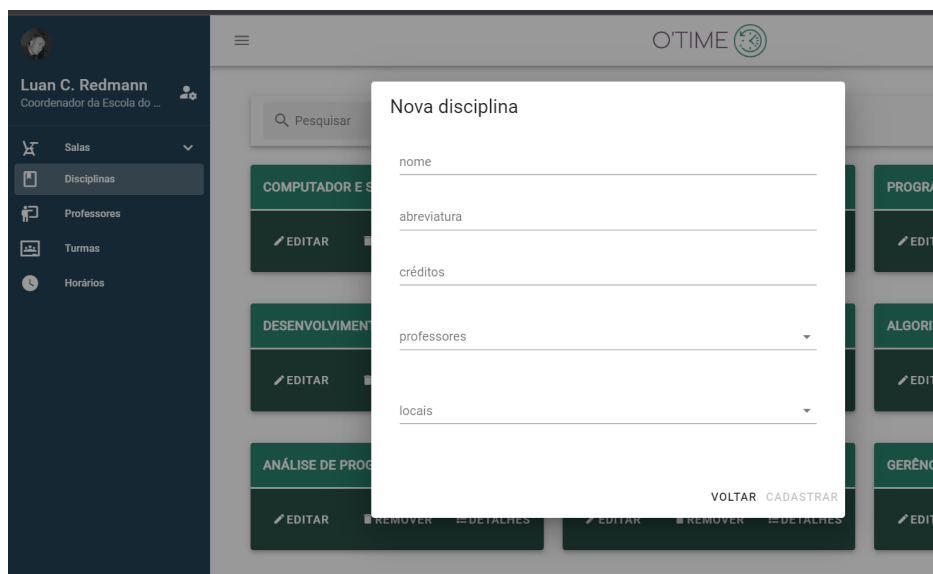
Fonte: Elaborado pelo autor (2022)

Os eventos foram linkados usando o sistema Django HTML, que permite a inserção de elementos do back-end no próprio arquivo HTML, o que segue a lógica SFC (Single-File Component) do Vue, que consiste em separar as funções de cada parte do código de acordo com o escopo ao qual ela pertence.

Na parte organizacional, o SCRUM foi utilizado pela equipe desenvolvedora do projeto como forma de estímulo e principalmente adaptada para a realidade do grupo, de forma que utilizamos as reuniões semanais para discutir o que cada um fez durante a semana, buscar soluções para os problemas que eventualmente surgiram, como a saída de membros, falta de conhecimentos específicos, bugs, descumprimento de prazos, etc.

Na parte de interface, os elementos que diferem da versão anterior são principalmente os modais, que substituem a necessidade de uma página que realize a mesma ação, sendo necessário apenas criar um modal dentro do componente onde se pretende que seja usado e a ferramenta já o aplica automaticamente nessa parte da aplicação, sendo por conta do processo de reatividade do Vue, que identifica as mudanças e executa o processo numa forma de pré-carregamento, notificando sobre quaisquer erros que surjam.

Figura 3 - Modais



Fonte: Elaborado pelo autor (2022)

Por fim, temos na parte de submissão de reserva de horários uma tabela com os horários cadastrados, sendo possível ver os horários de salas, disciplinas, professores e turmas de forma específica, usando um filtro mapeado, onde cada posição representa a entidade “card” no sistema, que é uma variável de estado que armazena as informações de cada horário individualmente.

Figura 4 - Tabela de Horários

Horário	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira ↑
7:00 - 7:45					
7:45 - 8:30					
8:50 - 9:35					
9:35 - 10:20					
10:30 - 11:15					
11:15 - 12:00					
13:00 - 13:45					
13:45 - 14:30					
14:50 - 15:35					

Fonte: Elaborado pelo autor (2022)

4 RESULTADOS ALCANÇADOS

Até o presente momento, o sistema compreende todas as funções de cadastro, alteração, remoção e visualização propostas, e a plataforma se encontra disponível online, sendo possível ser acessada pelo seguinte link: <<http://otime-app.herokuapp.com/#/>>.

Alguns documentos precisavam ser atualizados para concluir determinados tópicos, como o diagrama de classes, que teve de ser refeito para atender aos padrões atuais, e portanto, esses documentos ainda não se encontram disponíveis no repositório onde o projeto está hospedado no presente momento (2022), no caso o Github. Feitos os ajustes necessários o objetivo de modelagem foi alcançado dentro do possível, com ressalvas a pontos não tão aprofundados por conta do tempo.

Já na aplicação, tanto protótipo quanto versão final se encontram em consonância com a modelagem como exposto ao longo do trabalho, logo o objetivo foi alcançado. Porém o sistema ainda falha nos aspectos integração, colaboratividade, criação de um arquivo exportável e login, sendo necessário ainda a implementação destes fatores para que o sistema possa ser considerado como tendo atingido tudo que se propôs.

5 CONSIDERAÇÕES FINAIS

Deve ser levado em consideração algumas dificuldades enfrentadas pelo grupo, principalmente em virtude da Pandemia do COVID, o que fez o projeto ficar parado por cerca de um ano e alguns meses, e ao retornar a equipe não possuía a mesma produtividade inicial, tendo em vista fatores psicológicos e a dificuldade em se readaptar ao ambiente escolar, fora desentendimentos gerados por divergências de pensamento sobre diversos assuntos, falhas de comunicação e de conhecimento por parte de alguns integrantes, o que tornou o processo de desenvolvimento um tanto mais lento que o esperado, sendo impossível realizar as tarefas da forma como se pretendia.

Levando esses e outros problemas em consideração, recomenda-se que sejam feitas melhorias no back-end, tendo em vista que a versão atual gera problemas no cors e no banco de dados com frequência, a intuitividade do site poderia ser melhor destacada, com uma reorganização da forma como os dados são apresentados, também sendo importante mencionar que algumas partes da documentação, tais como documentos de risco, arquitetura, testes, diagramas, e etc., e ainda com relação à implementação, se faz necessário o login e a exportação de dados, para que o sistema seja capaz de realizar as funcionalidades requisitadas.

REFERÊNCIAS

BARROS, Aidil Jesus da Silveira; LEHFELD, Neide Aparecida de Souza. **Fundamentos de Metodologia Científica**. 3. ed. São Paulo: Pearson Prentice Hall, 2014.

BOOCH, Grady. **UML: guia do usuário**. Elsevier Brasil, 2006.

DIRETORIA DE GESTÃO E TECNOLOGIA DA INFORMAÇÃO. **A diretoria**. 25 mai. 2022. Disponível em: <https://sites.google.com/escolar.ifrn.edu.br/diatinf/a-diretoria>. Acesso em: 13 ago. 2022.

IFRN. **PAWeb - Processo Acadêmico Web**. Disponível em: <https://processo-academico.github.io/paweb/>. Acesso em: 13 ago. 2022.

IFRN. **PAAD - Processo Acadêmico Ágil Distribuído**. Disponível em: <https://processo-academico.github.io/paad/>. Acesso em: 13 ago. 2022.

IFRN. **PAAC - Processo Acadêmico Ágil Corporativo**. Disponível em: <https://processo-academico.github.io/paac/>. Acesso em: 13 ago. 2022.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE. **Processo Acadêmico**. 07 nov. 2017. Disponível em: <https://github.com/processo-academico>. Acesso em: 01 jul. 2022.

SCHWABER, Ken; SUTHERLAND, Jeff. La guía de Scrum. **Scrum Guides**. 2013.

SUTHERLAND, Jeff. **SCRUM: A arte de fazer o dobro de trabalho na metade do tempo**. Leya, 2014.

APÊNDICE A – QUESTIONÁRIO SOBRE PROBLEMAS DE HORÁRIO COM O PÚBLICO ALVO

Prof. Alessandro (Professor da DIATINF)

P: Enfrenta problemas com espaço ou horários?

R: Sim, normalmente de espaço, é muito difícil saber quando uma sala já está ocupada.

P: Qual a solução você usa para esse problema?

R: Uso o SUAP e vejo os horários num arquivo .pdf que contém os horários da sala que pretendo usar.

PROBLEMA DIAGNOSTICADO: Escala de horários

Fernando Pedro (Secretário da DIATINF)

P: Enfrenta problemas com espaço ou horários?

R: Sim, de espaço e horários, mudam constante e repentinamente.

P: Como você resolve esse problema?

R: Uso o SUAP para fazer as modificações de acordo com as novas informações passadas pela comissão, mas o fato de ser totalmente manual faz isso ser muito cansativo.

PROBLEMA DIAGNOSTICADO: Inserção de horários no SUAP

Prof. Ronaldo Maia (Integrante da Comissão de Horários da DIATINF)

P: Enfrenta problemas com espaço ou horários?

R: Sim, gerenciar as mudanças em ambos os casos é bem difícil.

P: Qual a solução que você utiliza?

R: Uso o Software ASC TimeTables, mas ainda é bem problemático, apesar dele fazer quase tudo sozinho, o programa não é colaborativo, exigindo a criação de vários arquivos diferentes para cada diretoria, e uma vez que não sincroniza entre elas, usamos um modelo no Excel para criar os horários e jogamos os arquivos no DropBox, informando no WhatsApp aos outros membros da comissão, de forma a impedir que outra pessoa mexa no mesmo arquivo acidentalmente. Outro problema sério é o exportável, que mesmo sendo possível de converter um .pdf, o sistema acadêmico acaba não sendo capaz de importar, exigindo trabalho totalmente manual na inserção desses horários, gerando diversos problemas sempre que há alterações.

PROBLEMA DIAGNOSTICADO: Montagem de horários