



---

Faculdade de Tecnologia de Americana – Ministro Ralph Biasi

**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Bruna Zanão Balthazar

Roberto de Souza Costa

Tiago Henrique de Moraes

**MONITOROU**

**Americana, SP**

**2019**

---

Faculdade de Tecnologia de Americana – Ministro Ralph Biasi

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Bruna Zanão Balthazar

Roberto de Souza Costa

Tiago Henrique de Moraes

### **MONITOROU**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Dr. Kleber Andrade de Oliveira.

Área de concentração: Laboratório de Engenharia de Software

**Americana, SP**

**2019**

**FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS**

**Dados Internacionais de Catalogação-na-fonte**

B158d BALTHAZAR, Bruna Zanão

Desenvolvimento de um sistema web para gestão de monitorias intitulado Monitorou. / Bruna Zanão Balthazar, Roberto de Souza Costa, Tiago Henrique de Moraes. – Americana, 2019.

113f.

Monografia (Curso de Tecnologia em Análise e Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Dr. Kleber de Oliveira Andrade

1 Desenvolvimento de software 2.Engenharia de software I. COSTA, Roberto de Souza II. MORAES, Tiago Henrique de III. ANDRADE, Kleber de Oliveira II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.3.05

---

**Faculdade de Tecnologia de Americana**

Bruna Zanão Balthazar  
Roberto de Souza Costa  
Tiago Henrique de Moraes


**DESENVOLVIMENTO DE UM SISTEMA WEB PARA GESTÃO DE  
MONITORIAS INTITULADO MONITOROU**

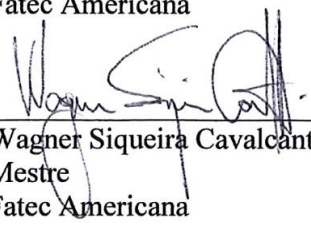
Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana.


Área de concentração: Laboratório de Engenharia de Software.

Americana, 11 de Junho de 2019.

**Banca Examinadora:**

  
\_\_\_\_\_  
Kleber de Oliveira Andrade  
Doutor  
Fatec Americana

  
\_\_\_\_\_  
Wagner Siqueira Cavalcante  
Mestre  
Fatec Americana

  
\_\_\_\_\_  
Eduardo Antonio Vicentini  
Mestre  
Fatec Americana

## RESUMO

Este trabalho tem como objetivo o desenvolvimento de um sistema WEB, destinado a monitoria de alunos aplicadas em instituições de ensino a fim de conclusão da matéria laboratório de engenharia de software, para a Faculdade de Tecnologia de Americana. Esse sistema foi pensado na dificuldade do aluno em se comunicar com os monitores oferecidos pelas instituições como apoio ao conteúdo apresentado em suas grades curriculares, onde no método comum o aluno tem que estar presente na instituição para ser atendido pelo monitor em seu horário de atividade. Com um sistema WEB, o aluno pode ser atendido em qualquer lugar que tenha um computador com conexão à internet e no horário de atividade do monitor disponível. Durante o processo de desenvolvimento foram encontradas dificuldades para conclusão de algumas atividades propostas devido à pouca experiência no desenvolvimento e gerenciamento do tempo pelos integrantes. Isso possibilitou um olhar mais técnico e uma experiência mais próxima do mercado de trabalho.

**Palavras Chave:** monitoria; ensino; universidade.

## **ABSTRACT**

This work aims at the development of a WEB system, aimed at the monitoring of students applied in educational institutions in order to complete the subject of software engineering laboratory, for the Faculty of Technology of Americana. This system was thought of the difficulty of the student to communicate with the monitors offered by the institutions as support to the content presented in their curriculum, where in the common method the student has to be present at the institution to be attended by the monitor in their schedule of activity. With a WEB system the student can be attended to anywhere that has a computer with internet connection and in the activity time of the monitor available. During the development process difficulties were encountered for the completion of some proposed activities due to the lack of experience in the development and time management of the members. This enabled a more technical look and experience closer to the job market.

**Keywords:** monitoring; learning; university;

## Sumário

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUÇÃO .....</b>                           | <b>20</b> |
| <b>2</b> | <b>PROJETO DO SISTEMA MONITOROU.....</b>          | <b>21</b> |
| 2.1      | Levantamento de Requisitos .....                  | 22        |
| 2.1.1    | Requisitos Funcionais .....                       | 22        |
| 2.1.2    | Requisitos Não Funcionais.....                    | 23        |
| 2.2      | Recursos e Ferramentas .....                      | 23        |
| <b>3</b> | <b>MODELAGEM.....</b>                             | <b>27</b> |
| 3.1      | Casos de Uso.....                                 | 28        |
| 3.2      | Documentação dos Casos de Uso .....               | 33        |
| 3.3      | Diagrama de Entidade de Relacionamento (DER)..... | 39        |
| 3.4      | Dicionário de Dados .....                         | 40        |
| 3.5      | Direitos de acesso.....                           | 49        |
| 3.6      | APIs do Sistema de Monitorou.....                 | 54        |
| 3.6.1    | Especificação das bibliotecas utilizadas .....    | 54        |
| 3.6.2    | Endpoints .....                                   | 56        |
| <b>4</b> | <b>DESENVOLVIMENTO .....</b>                      | <b>69</b> |
| 4.1      | Etapas de Desenvolvimento .....                   | 69        |
| 4.1.1    | Protótipo 1 .....                                 | 69        |
| 4.1.2    | Protótipo 2 .....                                 | 71        |
| 4.1.3    | Protótipo 3 .....                                 | 73        |
| 4.1.4    | Protótipo 4 .....                                 | 75        |

|       |  |     |
|-------|--|-----|
| 4.1.5 | Protótipo 5 .....  | 77  |
| 4.1.6 | Protótipo 6 .....  | 79  |
| 4.1.7 | Protótipo 7 .....  | 81  |
| 4.1.6 | Protótipo 8 .....  | 83  |
| 4.1.8 | Protótipo 9 .....  | 85  |
| 4.1.8 | Protótipo 10 .....   | 87  |
| 4.1.9 | Protótipo 11 .....   | 89  |
| 4.2.0 | Protótipo 12 .....   | 91  |
| 4.2.1 | Protótipo 13 .....   | 93  |
| 4.2.2 | Protótipo 14 .....   | 95  |
| 4.2.3 | Protótipo 15 .....   | 97  |
| 4.2.4 | Protótipo 16 .....   | 99  |
| 4.2.5 | Protótipo 17 .....   | 101 |
| 4.2   | Interfaces de Usuário .....                                      | 103 |
| 4.2.1 | Interface <i>Home Page</i> .....                                 | 103 |
| 4.2.3 | Interface de Cadastro de Admin .....                             | 105 |
| 4.2.4 | Interface de Cadastro de Monitor .....                           | 106 |
| 4.2.5 | Interface Cadastro de horários de monitor .....                  | 106 |
| 4.2.6 | Interface Cadastro de Aluno .....                                | 107 |
| 4.2.7 | Interface Cadastro de Professor .....                            | 108 |
| 4.2.7 | Interface <i>Dashboard</i> do Administrador .....                | 108 |
| 4.2.8 | Interface <i>Dashboard</i> Menu Cadastrar do Administrador ..... | 109 |
| 4.2.9 | Interface <i>Dashboard</i> Menu do Monitor .....                 | 110 |



|        |   |     |
|--------|---|-----|
| 4.2.10 | Interface <i>Dashboard</i> Horários do aluno..... | 110 |
| 4.2.11 | Interface Perfil do Aluno.....                    | 110 |
| 4.2.12 | Interface Perfil do Monitor.....                  | 111 |
| 4.2.13 | Interface para grupo de Estudos .....             | 112 |
| 5      | CONSIDERAÇÕES FINAIS .....                        | 115 |
|        | REFERÊNCIAS .....                                 | 117 |

## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1 - Diagrama de caso de uso do cadastro e login de usuário ..... |    |
| 26  |    |
| Figura 2 - Diagrama de caso de uso caminho para monitoria online .....  | 27 |
| Figura 3 - Diagrama de caso de uso de eventos importantes .....         |    |
| 28  |    |
| Figura 4 - Diagrama de caso de uso de presença monitor aluno .....      |    |
| 29  |    |
| Figura 5 - Diagrama de caso de uso do grupo de estudo .....             |    |
| 30  |    |
| Figura 6 - Diagrama de classe de entidade de relacionamento.....        |    |
| 37  |    |
| Figura 7 - JSON de retorno Usuários .....                               |    |
| 54  |    |
| Figura 8 - JSON de retorno de Matérias .....                            |    |
| 55  |    |
| Figura 9 - JSON de retorno de horários .....                            |    |
| 56  |    |
| Figura 10 - JSON de retorno de sessão .....                             |    |
| 57  |    |
| Figura 11 - JSON de retorno de Curso .....                              |    |
| 58  |    |
| Figura 12 - JSON de retorno de Eventos .....                            |    |
| 59  |    |
| Figura 13 - JSON de retorno de Presença .....                           |    |
| 60  |    |

|  |    |
|--|----|
| Figura 14 - JSON de retorno de Falta .....                     | 60 |
| Figura 15 - JSON de retorno de Grupo de Estudos .....          | 62 |
| Figura 16 - JSON de retorno de Conteúdos .....                 | 63 |
| Figura 17 - JSON de retorno de Questionários .....             | 65 |
| Figura 18 - JSON de retorno de Resposta de Questionários ..... | 65 |
| Figura 19 - Gráfico de <i>Burndown</i> da entrega 1 .....      | 67 |
| Figura 20 - Gráfico de <i>Burndown</i> da entrega 2 .....      | 69 |
| Figura 21 - Gráfico de <i>Burndown</i> da entrega 3.....       | 71 |
| Figura 22 - Gráfico de <i>Burndown</i> da entrega 4 .....      | 73 |
| Figura 23 - Gráfico de <i>Burndown</i> da entrega 5 .....      | 75 |
| Figura 24 - Gráfico de <i>Burndown</i> da entrega 6 .....      | 77 |
| Figura 25 - Gráfico de <i>Burndown</i> da entrega 7 .....      | 79 |
| Figura 26 - Gráfico de <i>Burndown</i> da entrega 8 .....      | 81 |

|  |     |
|--|-----|
| Figura 27 - Gráfico de <i>Burndown</i> da entrega 9 .....  | 83  |
| Figura 28 - Gráfico de <i>Burndown</i> da entrega 10 ..... | 85  |
| Figura 29 - Gráfico de <i>Burndown</i> da entrega 11.....  | 87  |
| Figura 30 - Gráfico de <i>Burndown</i> da entrega 12 ..... | 89  |
| Figura 31 - Gráfico de <i>Burndown</i> da entrega 13 ..... | 91  |
| Figura 32 - Gráfico de <i>Burndown</i> da entrega 14 ..... | 93  |
| Figura 33 - Gráfico de <i>Burndown</i> da entrega 15 ..... | 95  |
| Figura 34 - Gráfico de <i>Burndown</i> da entrega 16 ..... | 97  |
| Figura 35 - Gráfico de <i>Burndown</i> da entrega 17 ..... | 99  |
| Figura 36 - Interface Home .....                           | 100 |
| Figura 37 - Interface Home 2 .....                         | 101 |
| Figura 38 - Interface Login .....                          | 102 |
| Figura 39 - Interface Cadastro de Admin .....              | 102 |

|   |     |
|---|-----|
| Figura 40 - Interface Cadastro de Monitor .....                                   | 103 |
| Figura 41 -Interface Cadastro de Horários de Monitor .....                        | 104 |
| Figura 42 - Interface Cadastro de Aluno .....                                     | 104 |
| Figura 43 - Interface cadastro de professor .....                                 | 105 |
| Figura 44 - Interface <i>Dashboard</i> do Administrador .....                     | 106 |
| Figura 45 - Interface <i>Dashboard</i> menu do Administrador .....                | 106 |
| Figura 46 - Interface <i>Dashboard</i> Menu do Monitor .....                      | 107 |
| Figura 47 - Interface <i>Dashboard</i> do Aluno .....                             | 107 |
| Figura 48 - Interface Perfil do Aluno.....  | 108 |
| Figura 49 - Interface Perfil do Monitor .....                                     | 108 |
| Figura 50 – Interface para criar grupo de estudos .....                           | 109 |
| Figura 51 - Interface de visualização dos grupos de estudo (perfil aluno) .....   | 109 |
| Figura 52 - Interface de visualização dos grupos de estudo (perfil monitor) ..... | 110 |

Figura 53 - Interface para criar um Quiz .....  
111

Figura 54 - Interface visualização e responder Quiz .....  
111

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 - Comparativo de funcionalidades entre Moodle e o aplicativo desenvolvido neste trabalho..... | 18 |
| Tabela 2 - Requisitos funcionais do projeto.....   | 19 |
| Tabela 3 - Requisitos não funcionais do projeto.....   | 20 |
| Tabela 4 - Caso de uso “Efetuar cadastro - Aluno” .....  | 31 |
| Tabela 5 - Caso de uso “Efetuar cadastro - Monitor” .....  | 31 |
| Tabela 6 - Caso de uso “Efetuar cadastro - Admin e Professor” .....                                    | 32 |
| Tabela 7 - Caso de uso “Login” .....   | 32 |
| Tabela 8 - Caso de uso “Criar Eventos Importantes” .....   | 33 |
| Tabela 9 - Caso de uso “Visualizar Eventos” .....  | 33 |
| Tabela 10 - Caso de uso “Registrar Presença” .....   | 34 |
| Tabela 11 - Caso de uso “Correção de presença” .....   | 34 |
| Tabela 12 - Caso de uso “Criar Grupo de Estudo” .....  | 35 |
| Tabela 13 - Caso de uso “Entrar Grupo de Estudo” .....   | 36 |

|  |    |
|--|----|
| Tabela 14 - Descrição da tabela usuário .....                    | 38 |
| Tabela 15 – Descrição da tabela horário de monitores .....       | 39 |
| Tabela 16 - Descrição da tabela de matérias .....                | 39 |
| Tabela 17 - Descrição da tabela de Grupo de Estudos .....        | 40 |
| Tabela 18 - Descrição da tabela de Cursos .....                  | 40 |
| Tabela 19 - Descrição da tabela de Instituição .....             | 41 |
| Tabela 20 - Descrição da tabela de Conteúdos .....               | 41 |
| Tabela 21 - Descrição da tabela de Presença.....                 | 42 |
| Tabela 22 - Descrição da tabela de CursoMateria.....             | 42 |
| Tabela 23 - Descrição da tabela de Eventos.....                  | 43 |
| Tabela 24 – Descrição da tabela de Questionário.....             | 43 |
| Tabela 25 - Descrição da tabela de Opção de Questionário.....    | 44 |
| Tabela 26 - Descrição da tabela de Resposta do Questionário..... | 44 |



|   |    |
|---|----|
| Tabela 27 - Descrição da tabela de Grupo de Estudo.....                     | 45 |
| Tabela 28 - Descrição da tabela de GrupoDeEstudosUsuarios.....              | 45 |
| Tabela 29 - Direito de acesso para manipulação de Usuários .....            | 46 |
| Tabela 30 - Direito de acesso para manipulação de Horários de Monitor ..... | 46 |
| Tabela 31 - Direito de acesso para manipulação de Presença/Ausência .....   | 47 |
| Tabela 32 - Direito de acesso para manipulação de Grupo de Estudos .....    | 47 |
| Tabela 33 - Direito de acesso para manipulação do Quiz .....                | 48 |
| Tabela 34 - Direito de acesso para manipulação de Reposta Quiz .....        | 48 |
| Tabela 35 - Direito de acesso para manipulação de Matéria .....             | 49 |
| Tabela 36 - Direito de acesso para manipulação de Curso .....               | 49 |
| Tabela 37 - Direito de acesso para manipulação de Eventos .....             | 49 |
| Tabela 38 - Direito de acesso para manipulação de Conteúdo .....            | 50 |
| Tabela 39 - <i>Endpoints</i> de usuários .....                              | 53 |

|  |    |
|--|----|
| Tabela 40 - <i>Endpoints</i> Matérias .....                    | 55 |
| Tabela 41 - <i>Endpoints</i> Horários de Monitores .....       | 55 |
| Tabela 42 - <i>Endpoints</i> de Sessão .....                   | 56 |
| Tabela 43 - <i>Endpoints</i> de Cursos .....                   | 57 |
| Tabela 44 - <i>Endpoints</i> de Eventos .....                  | 58 |
| Tabela 45 - <i>Endpoints</i> de Presenças .....                | 59 |
| Tabela 46 - <i>Endpoints</i> de Grupo de Estudos .....         | 61 |
| Tabela 47 - <i>Endpoints</i> de Conteúdos .....                | 63 |
| Tabela 48 - <i>Endpoints</i> Quiz .....                        | 64 |
| Tabela 49 - Planejamento realizado para primeira entrega ..... | 67 |
| Tabela 50 - Planejamento realizado para segunda entrega .....  | 68 |
| Tabela 51 - Planejamento realizado para terceira entrega ..... | 70 |
| Tabela 52 - Planejamento realizado para quarta entrega .....   | 72 |

|   |    |
|---|----|
| Tabela 53 - Planejamento realizado para quinta entrega .....          | 74 |
| Tabela 54 - Planejamento realizado para sexta entrega .....           | 76 |
| Tabela 55 - Planejamento realizado para sétima entrega .....          | 78 |
| Tabela 56 - Planejamento realizado para oitava entrega .....          | 80 |
| Tabela 57 - Planejamento realizado para nona entrega .....            | 82 |
| Tabela 58 - Planejamento realizado para décima entrega .....          | 85 |
| Tabela 59 - Planejamento realizado para décima primeira entrega ..... | 86 |
| Tabela 60 - Planejamento realizado para décima segunda entrega .....  | 88 |
| Tabela 61 - Planejamento realizado para décima terceira entrega ..... | 90 |
| Tabela 62 - Planejamento realizado para décima quarta entrega .....   | 92 |
| Tabela 63 - Planejamento realizado para décima quinta entrega .....   | 94 |
| Tabela 64 - Planejamento realizado para décima sexta entrega .....    | 96 |
| Tabela 65 - Planejamento realizado para décima sétima entrega .....   | 98 |

## 1 INTRODUÇÃO

A monitoria acadêmica fornecida pelas instituições de graduação é um serviço de grande utilidade para auxiliar os alunos em sua jornada para a conclusão de um ensino superior. Ela consiste em um aluno que tenha maior conhecimento sobre uma dada matéria se prontifique em prestar o serviço de monitoria para auxiliar os outros alunos com dificuldade na respectiva matéria, podendo usar meios de ensino distintos do docente, mas que impliquem no esclarecimento das dúvidas e melhor compreensão do conteúdo. Embora seja um serviço de grande utilidade, muitos estudantes não têm acesso, devido à dificuldade de locomoção ou tempo disponível para comparecer nos horários da monitoria por conta de sua vida pessoal ou profissional.

Tendo em vista tal problema, decidiu-se criar o sistema *web* “Monitorou”, um sistema que visa facilitar a interação entre aluno e monitor, garantindo que, independentemente do local que esteja, com acesso à internet, o aluno possa ter acesso ao serviço de monitoria podendo conversar diretamente com o monitor presente via *chat*, colocar dúvidas em um fórum na plataforma, bem como ver os horários em que os monitores estarão presentes na instituição de ensino.

A criação desse sistema é voltada para a interação em comunidade entre alunos e monitores ou alunos e alunos, podendo também o docente participar dessa interação, para que ele possa ter uma noção melhor das dúvidas de seus alunos e adotar métodos ou dinâmicas distintas das utilizadas por ele para aprimorar mais suas aulas.

Para o desenvolvimento da plataforma foi utilizado para o *back-end* a linguagem *Ruby on Rails* sera usada a criação das *APIs* e *React* com *Redux* para o desenvolvimento do *front-end*, SGBD *PostgresSQL*, ferramentas de edição de texto *Vim*, *VsCode* e *Sublime Text 3*, *GitHub* para o versionamento do código, ferramentas para a edição de imagem *Adobe Photoshop* *CSS5* e *Adobe Illustrator*, foi selecionado o *Heroku* para a hospedagem gratuita da plataforma e algumas outras bibliotecas para um bom resultado do sistema.

## 2 PROJETO DO SISTEMA MONITOROU

A fim de comparação, existe um sistema semelhante com o projeto desenvolvido, porém, não compartilham exatamente do mesmo objetivo, o sistema a ser apresentado trabalha em uma escala maior no setor acadêmico, o que não é objetivo do sistema desenvolvido neste trabalho. Atualmente, o software mais popular que assemelha ao desenvolvido é o *Moodle*.

Levando estes aspectos em consideração, foi elaborada a Tabela 1, mostrando as principais diferenças entre o *Moodle* e o aplicativo desenvolvido neste trabalho:

**Tabela 1 – Comparativo de funcionalidades entre *Moodle* e o aplicativo desenvolvido neste trabalho.**

| <b>Funcionalidade</b>   | <b><i>Moodle</i></b> | <b>Monitorou</b> |
|-------------------------|----------------------|------------------|
| Plataforma online       | X                    | X                |
| Monitoria               | X                    | X                |
| Fórum/Comunidade        | X                    |                  |
| Chat                    | X                    |                  |
| Pesquisa de opinião     | X                    | X                |
| Tarefas para alunos     | X                    |                  |
| Criar página web (blog) | X                    |                  |
| Feed de notícias        | X                    | X                |
| Upload de arquivos      | X                    | X                |
| Quiz ou Testes          |                      | X                |
| Grupo de estudo         | X                    | X                |

**Fonte: Para recursos do *Moodle* [https://docs.moodle.org/36/en/Main\\_page](https://docs.moodle.org/36/en/Main_page)**

Este trabalho tem como objetivo desenvolver uma alternativa mais eficiente para a interação do aluno e monitor. Quanto aos objetivos específicos, são:

- Melhor interação entre aluno e monitor;

- Garantir o maior controle do professor sobre os alunos que frequentam a monitoria;
- Proporcionar maior aproximação dos alunos da instituição por meio de grupos de estudo disponíveis na plataforma;
- Facilitar o acesso dos alunos aos horários dos monitores, bem como alertas referentes a atrasos e faltas;
- O projeto pode vir atender as necessidades dos alunos universitários facilitando a formação em seu período de formação agilizando a curva de aprendizado.

## 2.1 Levantamento de Requisitos

A engenharia de requisitos (RE – *Requirements Engineering*) é o processo de descobrir, analisar, documentar e verificar requisitos de um sistema. Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007). Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

### 2.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2007). A Tabela 2 apresenta os requisitos funcionais deste projeto.

**Tabela 2 – Requisitos funcionais do projeto.**

| <b>Identificação</b> | <b>Requisito Funcional</b> | <b>Prioridade</b> |
|----------------------|----------------------------|-------------------|
| RF001                | Cadastrar admin            | Essencial         |
| RF002                | Cadastrar monitor          | Essencial         |
| RF003                | Cadastrar aluno            | Essencial         |
| RF004                | Cadastrar professor        | Essencial         |
| RF005                | Exibir matérias do curso   | Essencial         |

|       |   |            |
|-------|---|------------|
| RF006 | Criar Grupo de estudo                                   | Essencial  |
| RF007 | Criar Quiz  | Essencial  |
| RF008 | Exibir dia e horários dos monitores                     | Essencial  |
| RF009 | Controle de presença de monitores                       | Essencial  |
| RF010 | Controle de presença de alunos que frequentam monitoria | Essencial  |
| RF011 | Exibir relatório das respostas do quiz                  | Importante |
| RF012 | Permitir upload de material                             | Importante |
| RF013 | Manter cadastro de eventos                              | Importante |

Fonte: Elaborado pelo autor

### 2.1.2 Requisitos Não Funcionais

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2007). A Tabela 3 apresenta os requisitos não funcionais deste projeto.

Tabela 3 – Requisitos não funcionais do projeto.

| Identificação | Requisito não funcional                            | Categoria      | Prioridade |
|---------------|--|----------------|------------|
| RNF001        | <i>Dashboard</i> intuitivo                         | Usabilidade    | Essencial  |
| RNF002        | Proteção de informações do usuário                 | Confiabilidade | Essencial  |
| RNF003        | O sistema será operado na web                      | Distribuição   | Essencial  |
| RNF004        | Capaz de importar arquivos de imagem               | Usabilidade    | Essencial  |
| RNF005        | Restringir acesso à página de cadastro de usuários | Confiabilidade | Essencial  |
| RNF006        | <i>Timeout</i> após 15 segundos nas consultas      | Desempenho     | Essencial  |

Fonte: Elaborado pelo autor

## 2.2 Recursos e Ferramentas

Esta seção exhibe as ferramentas de programação e os conceitos necessários para o desenvolvimento do Monitorou.

- **Ruby on Rails:** O *Ruby*, que foi criado em 1995 por Yukihiro Matsumoto, é uma das linguagens de programação que tem ganhado popularidade devido a sua versatilidade e curva de aprendizado muito rápida. *Rails* é um *framework* para *Ruby*, lançado em 2004, que ajuda os desenvolvedores a construir

*websites* ao simplificar tarefas repetitivas. É um framework grátis e open source, e, como a maioria dos *frameworks* populares, é distribuído sob uma licença do MIT. O *Ruby on Rails* é baseado no *design Model-View-Control* (MVC).

- **CSS (*Cascade Style Sheet*):** Linguagem utilizada para descrever como os elementos devem ser renderizados em documentos HTML ou XML.
- **JavaScript:** É uma linguagem de programação criada em 1995 por Brendan Eich enquanto trabalhava na *Netscape Communications Corporation*. Originalmente projetada para rodar no *Netscape Navigator*, ela tinha o propósito de oferecer aos desenvolvedores formas de tornar determinados processos de páginas *web* mais dinâmicos, tornando seu uso mais agradável. Um ano depois de seu lançamento, a *Microsoft* portou a linguagem para seu navegador, o que ajudou a consolidar a linguagem e torná-la uma das tecnologias mais importantes e utilizadas na *internet*.
- **PostgreSQL:** É um poderoso sistema gerenciador de banco de dados objeto-relacional de código aberto. Tem mais de 15 anos de desenvolvimento ativo e uma arquitetura que comprovadamente ganhou forte reputação de confiabilidade, integridade de dados e conformidade a padrões. Roda em todos os grandes sistemas operacionais, incluindo *GNU/Linux*, *Unix (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64)*, e *MS Windows*. É totalmente compatível com *ACID*, tem suporte completo a chaves estrangeiras, junções (*JOINS*), visões, gatilhos e procedimentos armazenados (em múltiplas linguagens). Inclui a maior parte dos tipos de dados do *ISO SQL:1999*, incluindo *INTEGER*, *NUMERIC*, *BOOLEAN*, *CHAR*, *VARCHAR*, *DATE*, *INTERVAL* e *TIMESTAMP*. Suporta também o armazenamento de objetos binários, incluindo figuras, sons ou vídeos. Possui interfaces nativas de programação para *C/C++*, *Java*, *.Net*, *Perl*, *Python*, *Ruby*, *Tcl*, *ODBC*, entre outros, e uma excepcional documentação. Como um banco de dados de nível corporativo, o *PostgreSQL* possui funcionalidades sofisticadas como o controle de concorrência multiversionado (*MVCC*, em inglês), recuperação em um ponto no tempo (*PITR* em inglês), *tablespaces*, replicação assíncrona, transações agrupadas (*savepoints*), cópias de segurança a quente (*online/hot*



*backup*), um sofisticado planejador de consultas (otimizador) e registrador de transações sequencial (*WAL*) para tolerância a falhas. Suporta conjuntos de caracteres internacionais, codificação de caracteres *multibyte*, *Unicode* e sua ordenação por localização, sensibilidade a caixa (maiúsculas e minúsculas) e formatação. É altamente escalável, tanto na quantidade enorme de dados que pode gerenciar, quanto no número de usuários concorrentes que pode acomodar. Existem sistemas ativos com o *PostgreSQL* em ambiente de produção que gerenciam mais de *4TB* de dados.

- **Astah Community:** O programa escolhido para auxiliar na criação dos diagramas do sistema foi o *Astah Community*, um software para criação de diagramas em modelagem UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada). Nele estão disponíveis os diagramas de Classes, Caso de Uso, Sequência, Comunicação, Máquinas de Estados, Atividade, Componentes, Implantação e Diagrama de Estrutura Composta. Mesmo com uma interface muito bem elaborada e interativa, o *Astah Community* não é o programa mais fácil de utilizar, pois para desenvolver nele, além de conhecimento em criação de diagramas *UML* é necessário um conhecimento em suas ferramentas, para um melhor proveito e excelência para fazer o trabalho.
- **React:** é “uma biblioteca *JavaScript* declarativa, eficiente e flexível para a criação de interfaces de usuário (*UI*)”. Essa biblioteca surgiu em 2011, no *Facebook*, e passou a ser utilizada na interface do mural de notícias da rede social. No ano seguinte, passou a integrar também a área de tecnologia do *Instagram* e de várias outras ferramentas da empresa. Em 2013, o código foi aberto para a comunidade, o que colaborou para sua grande popularização. O *React* se preocupa apenas com a parte de *view*. Isso quer dizer que ele não assume nada em relação ao seu *stack* e não fornece nenhuma camada de *service*, *controller*, *model*, *collection*, *routes* e etc. O *React* trabalha apenas com o conceito de componente que recebe propriedades, computa o estado e retorna uma representação virtual do DOM. Isso quer dizer que componentes do *React* não trabalham com templates propriamente ditas, a forma como ele lida com isso é estruturando a representação do HTML em objetos.

- **Redux:** É uma biblioteca criada pelo Dan Abramov para tentar otimizar alguns obstáculos que o *Flux* começou a enfrentar, e também veio para simplificar a implementação do mesmo. Inspirada em conceitos da linguagem funcional *Elm*, o *Redux* veio com alguns paradigmas interessantes e um pouco diferenciados do *Flux*, é composto de três princípios que formam e definem o seu conceito. Eles são:
  1. Um único ponto de verdade – Todo o estado da aplicação é mantido em apenas um único objeto chamado de *Store*.
  2. O estado é imutável – O estado da aplicação é inalterável, a única maneira de afetá-lo é emitindo uma *Action* com a mudança.
  3. Mudanças são feitas apenas por funções puras – *Reducers* recebem as *Actions* emitidas e aplicam-nas ao estado. Sempre retornando um novo estado.
- **Material-UI:** É um conjunto de componentes *React* que implementam a especificação de *material design* do *Google*.
- **Vim:** *VI IMproved*) Editor de texto escolhido pelo programador para codificar. Criado por Bram Moolenaar, é uma versão mais poderosa do editor de texto *VI*. ([vim.org](http://vim.org))
- **JSX:** Linguagem de programação orientada a objeto, estaticamente tipada e projetada para rodar em navegadores modernos. Desenvolvido na DeNa como um projeto de pesquisa (2012, DeNA Co., Ltd). Linguagem utilizada de maneira nativa pelo *React*.
- **Fetch API:** Utilizado para realizar requisições *http*, tais como pedidos e respostas.
- **Redux-thunk:** Pacote para realizar ações assíncronas utilizando o *Redux*.
- **React-router:** Pacote para realizar roteamento no *React*, criada e mantida pelo *React Training*.
- **VsCode:** Editor de texto escolhido pelo programador para a codificação dos códigos em rails. Distribuído pela *Microsoft* (*Microsoft Visual Studio Code*,

nome completo) o editor conta com várias ferramentas, add-ons, sistemas de pesquisa e substituição de palavras, multi cursor para escrever em várias linhas ao mesmo tempo, sistema de versionamento integrado com git, podendo ver as alterações e substituí-las antes de enviar o *commit*.

- **Postman:** Ferramenta livre para os testes de *API*, ele simula as chamadas HTTP (POST, PATCH, PUT, GET...) para um *endpoint* que você especificar, podendo configurar os *headers*, *body* e demais variações das *requests*. Como o sistema desenvolvido é uma API RestFul, é a principal ferramenta utilizado no *back-end* para os testes de funcionalidade do sistema.
- **Heroku:** Plataforma para hospedagem do sistema desenvolvido. Gratuito, com configurações práticas e maior facilidade para o *deploy* de um sistema desenvolvido em *ruby on rails*, é a hospedagem utilizada para manter o sistema no ar, para conseguir fazer a integração com as duas linguagens.
- **DbDesigner:** Site utilizado para planejar e desenhar as tabelas do banco de dados. Também com acesso gratuito é uma forma simples e rápida para planejar e modelar o banco, podendo exportar o modelo criado e salvá-lo. Utilizado exclusivamente para a modelagem.
- **Draw.io:** Site para a criação de todo tipo de modelagem e diagramas (caso de uso, diagramas UML...), com acesso gratuito e podendo exportar os diagramas criados, seja em PDF, JPG, JPEG ou arquivo com extensão “.io” própria para o site, que armazena aquilo que foi feito anteriormente no site podendo dar continuidade de onde parou.

### 3 MODELAGEM

Na fase da modelagem é feita a documentação do aplicativo, se trata de diagramas que facilitam na compreensão do projeto de forma padronizada.

A documentação deste trabalho utilizará a linguagem de modelagem *Unified Modeling Language (UML)* para modelar os casos de uso e os diagramas.

### 3.1 Casos de Uso

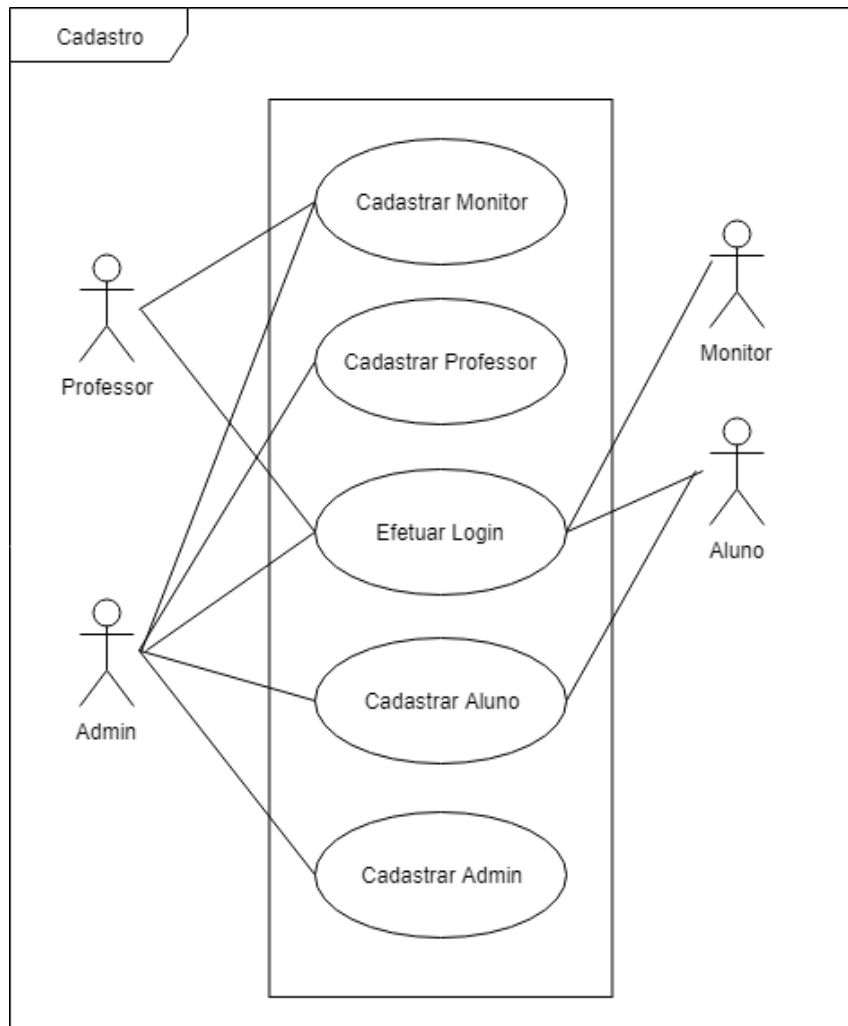
Os diagramas de caso de uso descrevem um cenário de funcionalidades do ponto de vista do usuário, catalogando os requisitos funcionais do sistema. Dentro do diagrama são retratados os atores (representado pelos bonecos), as funcionalidades (representadas pelos balões com a ação escrita por dentro) e as relações (representadas pelas linhas).

Os atores que interagem com o sistema são: o Admin, Monitor, Aluno. O sistema é um caso de uso explícito e se trata do sistema em si em que os casos de uso acontecem.

- **Admin** é o ator que representa o utilizador de gerência deste aplicativo. Um admin pode, por exemplo, cadastrar, acessar, interagir com outros usuários.
- **Monitor** é o ator que dará suporte ao aluno como monitoria.
- **Aluno** representa o usuário utiliza a plataforma para se conectar com o monitor.
- **Professor** é o ator que representa o usuário do sistema com permissões de professor.

A Figura 1 apresenta o caso de uso para a entrada do usuário no sistema.

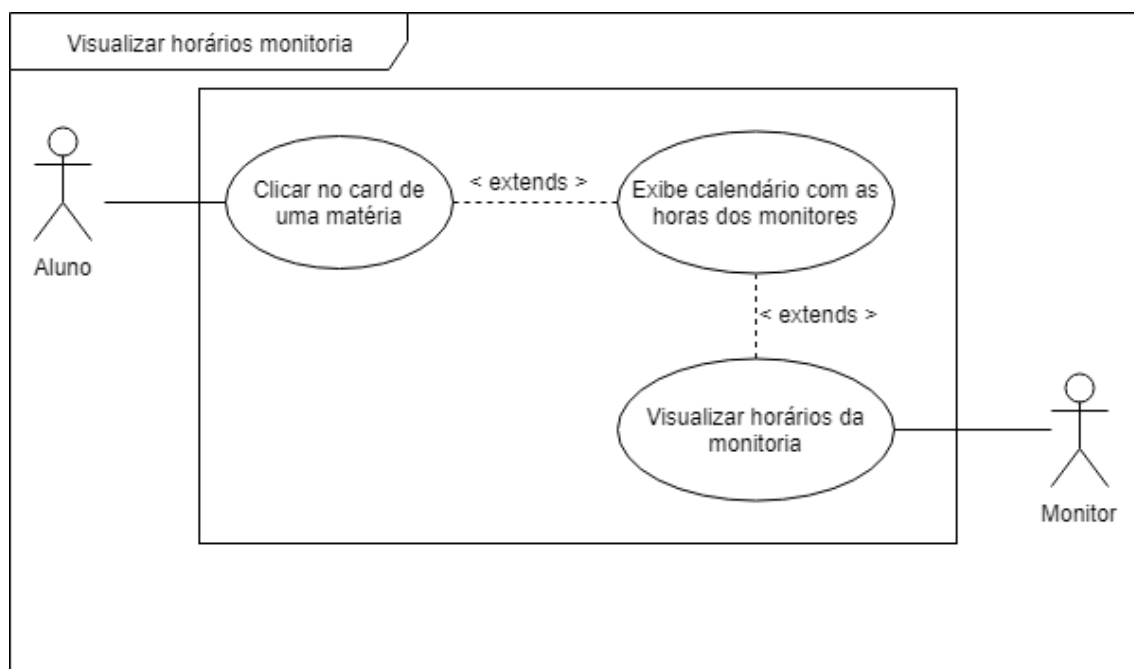
**Figura 1 – Diagrama de caso de uso do cadastro e login de usuário.**



**Fonte: Elaborado pelo autor**

A Figura 2 apresenta o caminho do usuário até chat para monitoria.

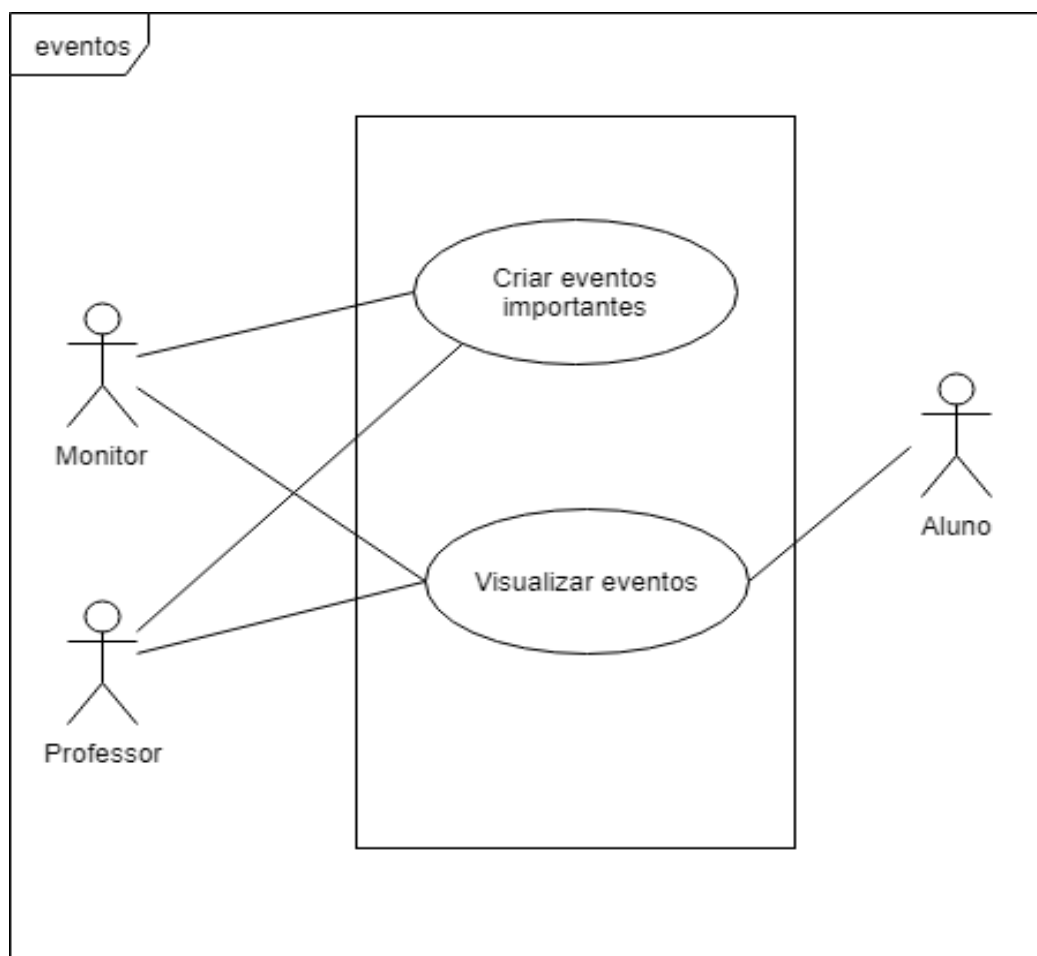
**Figura 2 – Diagrama de caso de uso caminho para monitoria online**



Fonte: Elaborado pelo autor

A Figura 3 representa o caso de uso para a inserção e visualização dos eventos importantes.

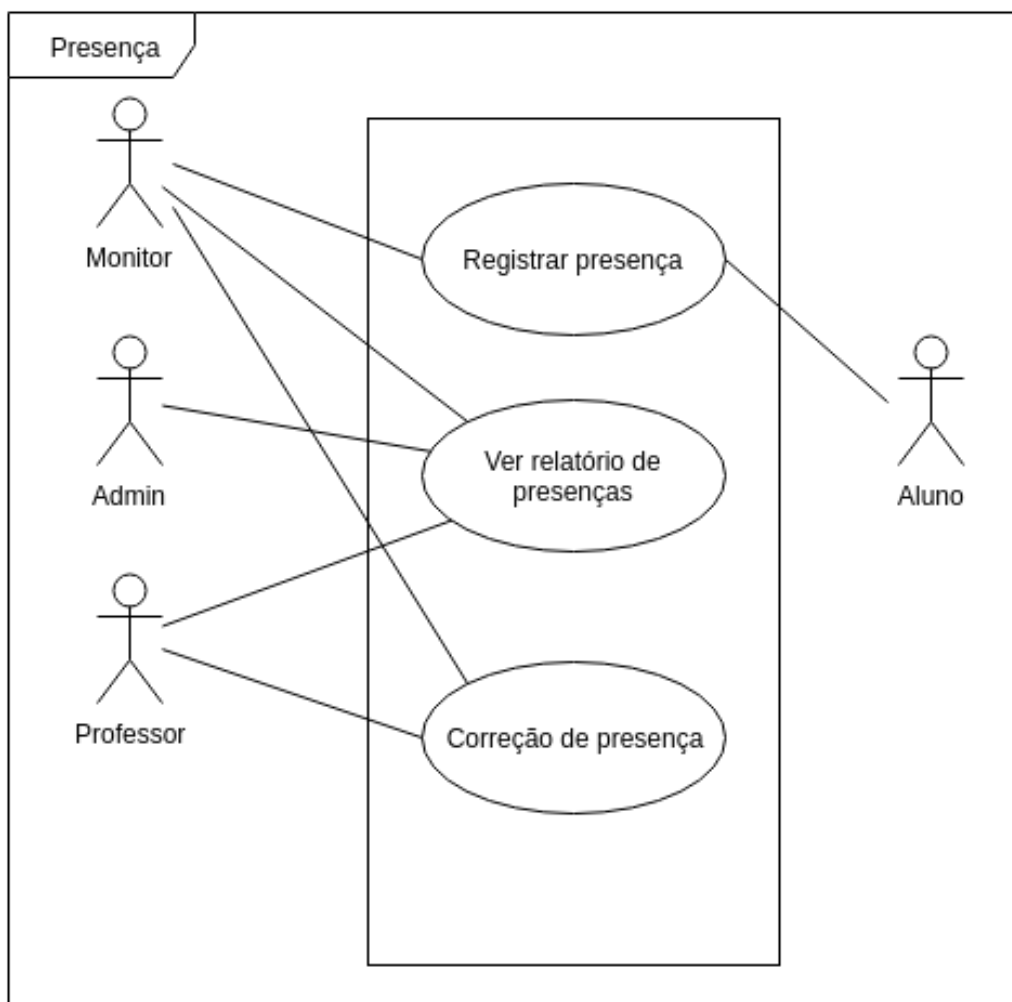
**Figura 3 – Diagrama de caso de uso de eventos importantes.**



Fonte: Elaborado pelo autor

A Figura 4 representa o caso de uso para a inserção e visualização das presenças e ausências dos alunos e monitores.

**Figura 4 – Diagrama de caso de uso de presença monitor aluno.**

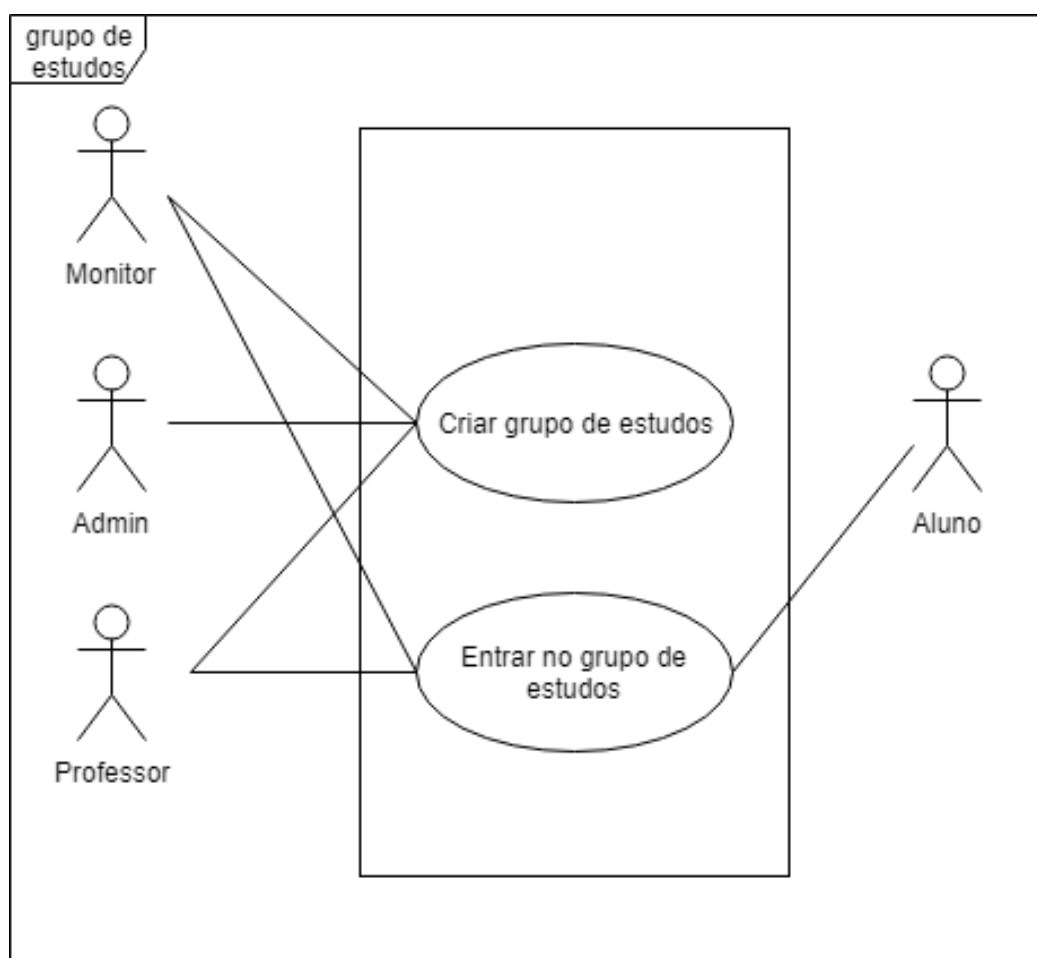


Fonte: Elaborado pelo autor



A Figura 4 representa o caso de uso para as ações relacionadas.

**Figura 5 – Diagrama de caso de uso do grupo de estudo.**



Fonte: Elaborado pelo autor

### 3.2 Documentação dos Casos de Uso

Nesta seção será abordada sobre a descrição de cada caso de uso exposto anteriormente, definindo todo o processo para que ele ocorra com sucesso.

A tabela 4 descreve o caso de uso para efetuar cadastro do aluno.

**Tabela 4 – Caso de uso “Efetuar cadastro - Aluno”.**

|  |  |
|--|--|
| <b>Nome do caso de uso</b>             | Cadastro de Aluno  |
| <b>Atores envolvidos</b>               | Admin, Monitor, Professor, Aluno e <i>API</i> .  |
| <b>Objetivo</b>                        | Este caso de uso descreve os passos do cadastro de um aluno no sistema                       |
| <b>Prioridade de desenvolvimento</b>   | Essencial  |
| <b>Ações do ator para login</b>        | <b>Ações do Sistema</b>  |
| 1. O Aluno clicar em efetuar cadastro. |  |
|  | 2. O sistema leva para a página de cadastro.   |
| 3. O aluno entra com os dados.         |  |
|  | 4. O sistema realiza uma requisição à <i>API</i> que cria o registro do aluno na plataforma. |
|  | 5. Após a autenticação, o sistema redireciona para a página inicial da plataforma.           |
| <b>Validações</b>                      | Para o cadastro seja efetuado, o aluno deve entrar com seu <i>e-mail</i> e senha.            |

**Fonte: Elaborado pelo autor**

A tabela 5 descreve o caso de uso de efetuar cadastro de um monitor.

**Tabela 5 – Caso de uso “Efetuar cadastro - Monitor”.**

|   |  |
|---|--|
| <b>Nome do caso de uso</b>                      | Cadastro de Monitor  |
| <b>Atores envolvidos</b>                        | Admin, Monitor, Professor, Aluno e <i>API</i> .                          |
| <b>Objetivo</b>                                 | Este caso de uso descreve os passos do cadastro de um monitor no sistema |
| <b>Prioridade de desenvolvimento</b>            | Essencial  |
| <b>Ações do ator para login</b>                 | <b>Ações do Sistema</b>  |
| 1. Admin ou professor entra no perfil do aluno. |  |
|   | 2. Clica em alterar perfil e selecione alterar para monitor.             |
|   | 3. A <i>API</i> faz a alteração das permissões do perfil.                |

**Fonte: Elaborado pelo autor**

A tabela 6 descreve o caso de uso de efetuar cadastro de um professor ou admin.

**Tabela 6 – Caso de uso “Efetuar cadastro - Admin e Professor”.**

|   |  |
|---|--|
| <b>Nome do caso de uso</b>                              | Cadastro de Admin Professor  |
| <b>Atores envolvidos</b>                                | Admin, Professor e <i>API</i> .  |
| <b>Objetivo</b>   | Este caso de uso descreve os passos do cadastro de um admin ou professor no sistema                  |
| <b>Prioridade de desenvolvimento</b>                    | Essencial  |
| <b>Ações do ator para login</b>                         | <b>Ações do Sistema</b>  |
| 1. O admin clicar em efetuar cadastro.                  |  |
|   | 2. O sistema leva para a página de cadastro.   |
| 3. O admin escolhe o tipo de perfil admin ou professor. |  |
| 4. Adiciona nome, e-mail e senha.                       |  |
| 5. Clica em cadastrar.                                  |  |
|   | 4. O sistema realiza uma requisição à <i>API</i> que cria o registro do novo cadastro na plataforma. |
|   | 5. Após a autenticação, o sistema redireciona para a página de cadastro novamente.                   |
| <b>Validações</b>                                       | Para o cadastro seja efetuado, deve entrar com e-mail e senha.                                       |

**Fonte: Elaborado pelo autor**

A tabela 7 descreve o caso de uso de efetuar login.

**Tabela 7 – Caso de uso “Efetuar Login”.**

|                                      |   |
|--------------------------------------|---|
| <b>Nome do caso de uso</b>           | Entrar no Sistema / Cadastrar Usuário   |
| <b>Atores envolvidos</b>             | Admin, Monitor, Professor, Aluno e <i>API</i> do sistema.                             |
| <b>Objetivo</b>                      | Este caso de uso descreve os passos do login e / ou cadastro de um usuário no sistema |
| <b>Prioridade de desenvolvimento</b> | Essencial   |

| <b>Ações do ator para login</b>       | <b>Ações do Sistema</b>  |
|---------------------------------------|--|
| 1. O usuário clicar em efetuar login. |  |
|                                       | 2. O sistema leva para a página de login.  |
| 3. Usuário entra com email e senha.   |  |
|                                       | 4. O sistema realiza uma requisição à <i>API</i> que busca as informações de login e senha no sistema de e as autentica. |
|                                       | 5. Após a autenticação, o sistema redireciona para a página inicial da aplicação.  |
| <b>Validações</b>                     | Para o login seja efetuado, o usuário deve entrar com seu e-mail e senha.  |

**Fonte: Elaborado pelo autor**

A tabela 8 descreve o caso de uso de criar eventos importantes.

**Tabela 8 – Caso de uso “Criar Eventos Importantes”.**

|   |  |
|---|--|
| <b>Nome do caso de uso</b>                      | Criar eventos importantes  |
| <b>Atores envolvidos</b>                        | Monitor, Professor, Aluno e <i>API</i> do sistema.                   |
| <b>Objetivo</b>                                 | Este caso de uso descreve os passos para criar um evento no sistema. |
| <b>Prioridade de desenvolvimento</b>            | Essencial  |
| <b>Ações do ator para login</b>                 | <b>Ações do Sistema</b>  |
| 1. Professor, monitor clica em criar um evento. |  |
|   | 2. A <i>API</i> do sistema armazena e exibe o evento para o aluno.   |

**Fonte: Elaborado pelo autor**

A tabela 9 descreve o caso de uso de visualizar eventos.

**Tabela 9 – Caso de uso “Visualizar Eventos”.**

|                            |  |
|----------------------------|--|
| <b>Nome do caso de uso</b> | Visualizar eventos   |
| <b>Atores envolvidos</b>   | Aluno e <i>API</i> do sistema.                                       |
| <b>Objetivo</b>            | Este caso de uso descreve os passos para criar um evento no sistema. |

|   |   |
|---|---|
| <b>Prioridade de desenvolvimento</b>                              | Essencial   |
| <b>Ações do ator para login</b>                                   | <b>Ações do Sistema</b>   |
| 1. Após efetuar o login o aluno acessa sua dashboard de horários. |   |
|   | 2. O sistema irá exibir na dashboard do aluno os eventos importantes cadastrados pelos monitores, professores ou admin. |

**Fonte: Elaborado pelo autor**

A tabela 10 descreve o caso de uso registrar uma presença.

**Tabela 10 – Caso de uso “Registrar Presença”.**

|  |   |
|--|---|
| <b>Nome do caso de uso</b>   | Registrar Presença  |
| <b>Atores envolvidos</b>   | Monitor, Aluno e <i>API</i> do sistema.   |
| <b>Objetivo</b>  | Este caso de uso descreve os passos para criar um evento no sistema.  |
| <b>Prioridade de desenvolvimento</b>   | Essencial   |
| <b>Ações do ator para login</b>  | <b>Ações do Sistema</b>   |
| 1. Monitor clica no botão presenças na tela inicial.   |   |
|  | 2. O sistema abre um modal com opção de presença de monitor e presença de aluno.  |
|  | Modal (monitor: lista o dia atual e um <i>checkbox</i> para marcar presença do dia, aluno: lista todos os alunos cadastrados na matéria, junto com o dia e um <i>checkbox</i> para cada aluno). |
| 3. O monitor <i>check</i> ou não sua presença e dos alunos, e clica no botão salvar, que envia as presenças pro sistema. |   |
|  | 4. A <i>API</i> do sistema salva as presenças.  |
| <b>Validações</b>  | Para alterar a presença do monitor é preciso um perfil de professor ou admin.   |

**Fonte: Elaborado pelo autor**

A tabela 11 descreve o caso de uso correção de uma presença.

**Tabela 11 – Caso de uso “Correção de presença”.**

|                            |                                |
|----------------------------|--------------------------------|
| <b>Nome do caso de uso</b> | Correção de presença.          |
| <b>Atores envolvidos</b>   | Aluno e <i>API</i> do sistema. |

|  |   |
|--|---|
| <b>Objetivo</b>                                    | Este caso de uso descreve os passos para criar um evento no sistema.          |
| <b>Prioridade de desenvolvimento</b>               | Essencial   |
| <b>Ações do ator para login</b>                    | <b>Ações do Sistema</b>   |
| 1. Professor Seleciona o usuário a ser atualizado. |   |
|  | 2. O sistema retorna os valores do usuário solicitado.                        |
| 3. Selecionar o dia de presença a ser corrigido;   |   |
|  | 4. A <i>API</i> do sistema salva as alterações no banco de dados.             |
| <b>Validações</b>                                  | Para alterar a presença do monitor é preciso um perfil de professor ou admin. |

Fonte: Elaborado pelo autor

A tabela 12 descreve o caso de uso criar um grupo de estudo.

**Tabela 12 – Caso de uso “Criar Grupo de Estudo”.**

|   |   |
|---|---|
| <b>Nome do caso de uso</b>  | Criar grupo de estudo.  |
| <b>Atores envolvidos</b>  | Professor, Monitor, Aluno e <i>API</i> do sistema.                                |
| <b>Objetivo</b>   | Este caso de uso descreve os passos para criar um evento no sistema.              |
| <b>Prioridade de desenvolvimento</b>  | Essencial   |
| <b>Ações do ator para login</b>   | <b>Ações do Sistema</b>   |
| 1. Após login, professor ou monitor selecione a opção para criar o grupo de estudos.      |   |
|   | 2. O sistema redireciona para tela de criação do grupo de estudos.                |
| 3. O usuário preenche os campos para a criação do grupo de estudos (nome, matéria e tema) |   |
| 4. Clica em criar grupo.  |   |
|   | 4. A <i>API</i> do sistema cria o registro de grupo de estudos no banco de dados. |
|   | 5. Após ter sido criado o grupo, ele passa a ser exibido para os alunos.          |

|                   |   |
|-------------------|---|
| <b>Validações</b> | Somente visualiza o grupo os alunos que estão matriculados na matéria do grupo de estudo. |
|-------------------|---|

Fonte: Elaborado pelo autor

A tabela 13 descreve o caso de uso entrar em um grupo de estudo.

**Tabela 13 – Caso de uso “Entrar Grupo de Estudo”.**

|   |   |
|---|---|
| <b>Nome do caso de uso</b>  | Entrar grupo de estudo.   |
| <b>Atores envolvidos</b>  | Professor, Monitor, Aluno e <i>API</i> do sistema.  |
| <b>Objetivo</b>   | Este caso de uso descreve os passos para criar um evento no sistema.  |
| <b>Prioridade de desenvolvimento</b>  | Essencial   |
| <b>Ações do ator para login</b>   | <b>Ações do Sistema</b>   |
| 1. O aluno vê o grupo de estudo em sua dashboard.                                     |   |
| 2. Seleciona o grupo de estudo a qual quer entrar e clica no botão “Entrar no grupo”. |   |
|   | 3. É enviado uma requisição para <i>API</i> do sistema que inclui o registro do usuário ao grupo de estudo. |
| <b>Validações</b>   | O aluno só pode entrar no grupo da matéria que está matriculado.  |

Fonte: Elaborado pelo autor

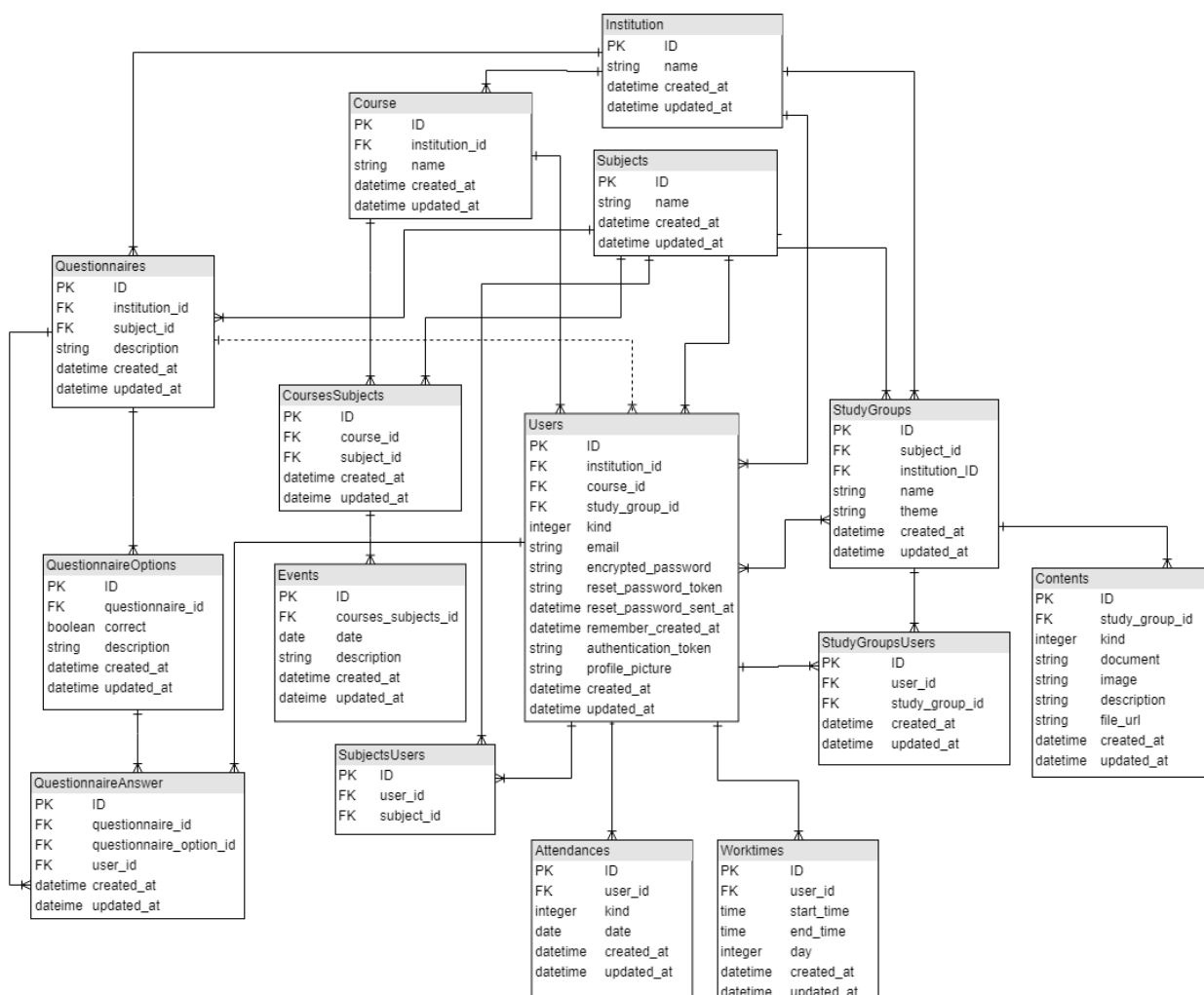
### 3.3 Diagrama de Entidade de Relacionamento (DER)

O diagrama de entidade e relacionamento é responsável por oferecer uma representação da estrutura e relações das classes como também as operações solicitadas pelos atores que servem de modelo para os objetos. Para entender melhor a estrutura funcional do projeto, a Figura 6 apresenta a modelagem do banco do sistema Monitorou.

O nome de todas as tabelas do sistema foi atribuído em inglês por dois motivos, o primeiro é que por convenção da linguagem utilizada para o *back-end* (*Ruby*) declara o nome das tabelas em inglês para ficar mais fácil de manipulá-las durante o desenvolvimento e segundo que no mercado de trabalho normalmente sempre se mantém o inglês como língua para definição dos objetos.

O diagrama de Entidade e Relacionamento é representado pela figura 6, responsável por oferecer uma representação de como todas as entidades se relacionam entre si dentro do sistema.

**Figura 6 – Diagrama de entidade de relacionamento.**



Fonte: Elaborado pelo autor

### 3.4 Dicionário de Dados

O dicionário de dado é importante para que haja um total entendimento de cada campo que compõem as tabelas do banco de dados do sistema, a seguir será informado a descrição de cada tabela e seus campos. Como na linguagem *Ruby* tudo dentro do sistema é um objeto, não há a necessidade de alterar o nome do *ID* de uma entidade visto que para acessá-la deve primeiro passar o objeto. Exemplo: *user.id*



Na tabela 14 é indicado os campos que compõem o *model* de usuários no sistema.

**Tabela 14 - Descrição da tabela usuário**

| Tabela Users (usuários) |          |   |     |          |
|-------------------------|----------|---|-----|----------|
| Campo                   | Tipo     | Descrição   | KEY | NULLABLE |
| ID                      | integer  | Campo de identificação do usuário   | PK  |          |
| name                    | string   | Campo para armazenar o nome do usuário  |     |          |
| institution_id          | bigint   | Identificação da instituição a qual o usuário pertence  | FK  |          |
| course_id               | bigint   | Identificação do curso do qual o usuário pertence   | FK  |          |
| kind                    | integer  | ENUM [0=>student, 1=>monitor, 2=>admin, 3=>teacher] que atribui o tipo ao usuário, variando entre: estudante, monitor, admin e professor, respectivamente. Atributo de tipo inteiro que consegue agregar um valor ao usuário. |     |          |
| created_at              | datetime | Valor atribuído no momento da criação do objeto desta entidade.   |     |          |
| updated_at              | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto.   |     |          |
| email                   | string   | <i>Email</i> do usuário, utilizado para fazer o login na plataforma.  |     |          |
| encrypted_password      | string   | Senha do usuário.   |     |          |
| reset_password_token    | string   | Token de pedido de alteração de senha.  |     |          |
| reset_password_sent_at  | datetime | Data que foi enviado o pedido de restauração da senha.  |     |          |
| authentication_token    | string   | Token de autenticação para garantir a sessão do usuário.  |     |          |
| study_group_id          | bigint   | Número de identificação do grupo a qual o usuário pertence.   | FK  | X        |
| profile_picture         | string   | Url que armazena o diretório da imagem de perfil do usuário   |     | X        |

**Fonte: elaborado pelo autor**

A tabela 15 é indicada a estrutura do *model* de *worktimes* (horários de trabalho dos monitores)

**Tabela 15 – Descrição da tabela horário de monitores**

| Tabela Worktimes (horários dos monitores) |          |   |     |          |
|---|----------|---|-----|----------|
| Campo                                     | Tipo     | Descrição   | KEY | NULLABLE |
| ID  | integer  | ID do registro do horário dos monitores.  | PK  |          |
| start_time                                | time     | Registra a hora de entrada do monitor.  |     |          |
| end_time                                  | time     | Registra a hora de saída do monitor.  |     |          |
| day                                       | integer  | ENUM [0=>monday, 1=>tuesday, 2=>wednesday, 3=>thursday, 4=>friday, 5=>saturday, 6=>sunday] para armazenar o dia da semana a qual o monitor deverá comparecer, iniciando a contagem a segunda. |     |          |
| user_id                                   | bigint   | Identificação a de qual usuário esse horário pertence.  | FK  |          |
| created_at                                | datetime | Valor atribuído no momento da criação do objeto desta entidade.   |     |          |
| updated_at                                | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto.   |     |          |

Fonte: elaborado pelo autor

A tabela 16 representa a estrutura do *model* de *subjects* (matérias) da plataforma.

**Tabela 16 – Descrição da tabela de matérias**

| Tabela Subjects (matérias) |          |   |     |          |
|----------------------------|----------|---|-----|----------|
| Campo                      | Tipo     | Descrição   | KEY | NULLABLE |
| ID                         | integer  | ID da matéria,  | PK  |          |
| name                       | string   | Nome da matéria.  |     |          |
| course_id                  | bigint   | Identificação do curso a qual a matéria pertence.                               | FK  |          |
| created_at                 | datetime | Valor atribuído no momento da criação do objeto desta entidade.                 |     |          |
| updated_at                 | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |     |          |

Fonte: elaborado pelo autor

A tabela 17 representa a estrutura do *model* de *StudyGroups* (Grupo de estudos) da plataforma.

**Tabela 17 – Descrição da tabela de Grupo de Estudos**

| Tabela StudyGroups (grupo de estudos) |          |   |     |          |
|---------------------------------------|----------|---|-----|----------|
| Campo                                 | Tipo     | Descrição   | KEY | NULLABLE |
| ID                                    | integer  | ID do grupo de estudos.   | PK  |          |
| institution_id                        | bigint   | Identificação da instituição a qual o grupo de estudos pertence.                | FK  |          |
| name                                  | string   | Nome do grupo de estudos.   |     |          |
| subject_id                            | bigint   | Matéria a qual o grupo de estudos se trata.                                     | FK  |          |
| theme                                 | string   | Tema a qual o grupo de estudos irá abordar.                                     |     | X        |
| created_at                            | datetime | Valor atribuído no momento da criação do objeto desta entidade.                 |     |          |
| updated_at                            | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |     |          |

**Fonte: elaborado pelo Autor**

A tabela 18 representa a estrutura do *model* de *Courses* (Cursos) da plataforma.

**Tabela 18 - Descrição da tabela de Cursos**

| Tabela Courses (cursos) |          |   |     |          |
|-------------------------|----------|---|-----|----------|
| Campo                   | Tipo     | Descrição   | KEY | NULLABLE |
| ID                      | integer  | ID do curso.  | PK  |          |
| name                    | string   | Nome do curso.  |     |          |
| institution_id          | bigint   | Identificação da instituição a qual o curso pertence.                           | FK  |          |
| created_at              | datetime | Valor atribuído no momento da criação do objeto desta entidade.                 |     |          |
| updated_at              | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |     |          |

**Fonte: elaborado pelo autor**

A tabela 19 representa a estrutura do *model* de *Institutions* (instituição) da plataforma.

**Tabela 19 – Descrição da tabela de Instituição**

| Tabela Institutions (instituições) |          |   |     |          |
|------------------------------------|----------|---|-----|----------|
| Campo                              | Tipo     | Descrição   | KEY | NULLABLE |
| ID                                 | integer  | ID da instituição.  | PK  |          |
| name                               | string   | Nome da instituição.  |     |          |
| created_at                         | datetime | Valor atribuído no momento da criação do objeto desta entidade.                 |     |          |
| updated_at                         | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |     |          |

Fonte: elaborado pelo autor

A tabela 20 representa a estrutura do *model* de *Contents* (Conteúdos) da plataforma.

**Tabela 20 - Descrição da tabela de Conteúdos**

| Tabela Contents (conteudos) |          |   |     |          |
|-----------------------------|----------|---|-----|----------|
| Campo                       | Tipo     | Descrição   | KEY | NULLABLE |
| ID                          | integer  | ID do grupo de estudos.   | PK  |          |
| study_group_id              | bigint   | Identificação do grupo de estudos a qual o conteúdo pertence.   | FK  |          |
| document                    | string   | Campo que armazena o diretório do arquivo, tipo documento que foi feito o <i>upload</i> .                                       |     |          |
| image                       | string   | Campo que armazena o diretório do arquivo, tipo documento que foi feito o <i>upload</i> .                                       |     |          |
| kind                        | integer  | Identificação do tipo de arquivo que foi feito <i>upload</i> , utilizado para filtros e buscas.<br>Variações: [image, document] |     |          |
| created_at                  | datetime | Valor atribuído no momento da criação do objeto desta entidade.   |     |          |
| updated_at                  | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto.   |     |          |

Fonte: Elaborado pelo autor

A tabela 21 representa a estrutura do *model* de *Attendances* (Presenças) da plataforma.

**Tabela 21 - Descrição da tabela de Presença**

| <b>Tabela Attendances (presença)</b> |             |   |            |                 |
|--------------------------------------|-------------|---|------------|-----------------|
| <b>Campo</b>                         | <b>Tipo</b> | <b>Descrição</b>  | <b>KEY</b> | <b>NULLABLE</b> |
| ID                                   | integer     | ID do grupo de estudos.   | PK         |                 |
| user_id                              | bigint      | Identificação do usuário a qual a presença pertence.                                    | FK         |                 |
| date                                 | datetime    | Armazena a data a qual se refere a presença   |            |                 |
| kind                                 | integer     | Identificação se o registro é uma presença ou ausência.<br>Variações: [presence absent] |            |                 |
| created_at                           | datetime    | Valor atribuído no momento da criação do objeto desta entidade.                         |            |                 |
| updated_at                           | datetime    | Valor atribuído no momento da criação e nas posteriores atualizações do objeto.         |            |                 |

**Fonte: elaborado pelo autor**

A tabela 22 representa a estrutura do *model* de *CoursesSubjects* (CursoMateria) da plataforma, para garantir a integridade da relação n:n entre usuário e o curso a qual ele está matriculado.

**Tabela 22 - Descrição da tabela de CursoMateria**

| <b>Tabela CoursesSubjects (CursoMateria)</b> |             |   |            |                 |
|--|-------------|---|------------|-----------------|
| <b>Campo</b>                                 | <b>Tipo</b> | <b>Descrição</b>  | <b>KEY</b> | <b>NULLABLE</b> |
| ID   | integer     | ID da relação.  | PK         |                 |
| course_id                                    | bigint      | Identificação do curso na relação.  | FK         |                 |
| subject_id                                   | bigint      | Identificação da matéria na relação.  | FK         |                 |
| created_at                                   | datetime    | Valor atribuído no momento da criação do objeto desta entidade.                 |            |                 |
| updated_at                                   | datetime    | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |            |                 |

**Fonte: elaborado pelo autor**

A tabela 23 representa a estrutura do *model* de *Events* (Eventos) da plataforma.

**Tabela 23 - Descrição da tabela de Eventos**

| Tabela Events (eventos) |          |   |     |          |
|-------------------------|----------|---|-----|----------|
| Campo                   | Tipo     | Descrição   | KEY | NULLABLE |
| ID                      | integer  | ID do evento  | PK  |          |
| description             | string   | Descrição do evento.  |     |          |
| date                    | date     | Data a qual o evento ocorrerá.  |     |          |
| courses_subject_id      | bigint   | Identificação do curso a qual o evento pertence.                                | FK  |          |
| created_at              | datetime | Valor atribuído no momento da criação do objeto desta entidade.                 |     |          |
| updated_at              | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |     |          |

Fonte: elaborado pelo autor

A tabela 24 representa a estrutura do *model* de *Questionnaire* (Questionário), nomeado “Quiz” em nossa plataforma.

**Tabela 24 - Descrição da tabela de Questionário**

| Tabela Questionnaire (quiz) |          |   |     |          |
|-----------------------------|----------|---|-----|----------|
| Campo                       | Tipo     | Descrição   | KEY | NULLABLE |
| ID                          | integer  | ID do quiz.   | PK  |          |
| subject_id                  | bigint   | Identificação de qual matéria o quiz irá se referenciar.                        | FK  |          |
| institution_id              | bigint   | Identificação de qual instituição o quiz pertence.                              | FK  |          |
| description                 | string   | Descrição da pergunta do questionário.  |     |          |
| created_at                  | datetime | Valor atribuído no momento da criação do objeto desta entidade.                 |     |          |
| updated_at                  | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |     |          |

Fonte: elaborado pelo autor

A tabela 25 representa a estrutura do *model* de *QuestionnaireOptions* (Opção de questionário) na plataforma.

**Tabela 25 - Descrição da tabela de Opção de Questionário**

| Tabela QuestionnaireOption (opção do quiz) |          |   |     |          |
|--|----------|---|-----|----------|
| Campo                                      | Tipo     | Descrição   | KEY | NULLABLE |
| ID   | integer  | ID da opção do quiz.  | PK  |          |
| questionnaire_id                           | bigint   | Identificação do questionário a qual a pergunta pertence.                       | FK  |          |
| correct                                    | boolean  | Identificação se essa é a pergunta correta ou não.                              |     |          |
| description                                | string   | Descrição da opção.   |     |          |
| created_at                                 | datetime | Valor atribuído no momento da criação do objeto desta entidade.                 |     |          |
| updated_at                                 | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |     |          |

**Fonte: elaborado pelo autor**

A tabela 26 representa a estrutura do *model* de *QuestionnaireAnswer* (Resposta do Questionário) da plataforma.

**Tabela 26 - Descrição da tabela de Resposta do Questionário**

| Tabela QuestionnaireAnswer (resposta do quiz) |          |   |     |          |
|---|----------|---|-----|----------|
| Campo   | Tipo     | Descrição   | KEY | NULLABLE |
| ID  | integer  | ID da resposta.   | PK  |          |
| questionnaire_id                              | bigint   | Identificação do questionário a qual a resposta pertence.                       | FK  |          |
| questionnaire_option_id                       | bigint   | Identificação de qual opção de questionário a resposta se refere.               | FK  |          |
| user_id                                       | bigint   | Identificação de qual usuário respondeu.  | FK  |          |
| created_at                                    | datetime | Valor atribuído no momento da criação do objeto desta entidade.                 |     |          |
| updated_at                                    | datetime | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |     |          |

**Fonte: Elaborado pelo autor**

A tabela 27 representa a estrutura do *model* de *SubjectsUsers* (MateriasUsuários) da plataforma, utilizada para garantir a integridade da relação n:n entre usuário e a matéria que ele leciona.

**Tabela 27 - Descrição da tabela de Grupo de Estudo**

| <b>Tabela SubjectUsers (Join table Usuário &gt; matéria)</b> |             |   |            |                 |
|--|-------------|---|------------|-----------------|
| <b>Campo</b>   | <b>Tipo</b> | <b>Descrição</b>  | <b>KEY</b> | <b>NULLABLE</b> |
| ID   | integer     | ID da relação.  | PK         |                 |
| subject_id   | bigint      | Identificação de qual matéria a relação se trata.                               | FK         |                 |
| user_id  | bigint      | Identificação de qual usuário a relação trata.                                  | FK         |                 |
| created_at   | datetime    | Valor atribuído no momento da criação do objeto desta entidade.                 |            |                 |
| updated_at   | datetime    | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |            |                 |

**Fonte: Elaborado pelo autor**

A tabela 28 representa a estrutura do *model* de *StudyGroupUsers* (GrupoDeEstudosUsuário) da plataforma, para garantir a integridade da relação n:n entre usuário e os grupos de estudo aos quais ele pertence.

**Tabela 28 – Descrição da tabela de GrupoDeEstudosUsuarios**

| <b>Tabela StudyGroupUsers (Join table grupo de estudos &gt; usuarios)</b> |             |   |            |                 |
|---|-------------|---|------------|-----------------|
| <b>Campo</b>  | <b>Tipo</b> | <b>Descrição</b>  | <b>KEY</b> | <b>NULLABLE</b> |
| ID  | integer     | ID da relação.  | PK         |                 |
| study_group_id  | bigint      | Identificação de qual grupo a relação se trata.                                 | FK         |                 |
| user_id   | bigint      | Identificação de qual usuário a relação trata.                                  | FK         |                 |
| created_at  | datetime    | Valor atribuído no momento da criação do objeto desta entidade.                 |            |                 |
| updated_at  | datetime    | Valor atribuído no momento da criação e nas posteriores atualizações do objeto. |            |                 |

**Fonte: Elaborado pelo autor**



### 3.5 Direitos de acesso

Para garantir um bom gerenciamento do sistema e controle do sistema, foi atribuído o direito de acesso para cada função do sistema referente a uma atividade. Assim como descritas nas tabelas a seguir:

A tabela 29 representa os direitos de acesso atribuídos para as operações relacionadas a manipulação de usuários no sistema

**Tabela 29 - Direito de acesso para manipulação de Usuários**

| Função           | Atividades |          |              |         |          |             |          |          |          |
|------------------|------------|----------|--------------|---------|----------|-------------|----------|----------|----------|
|                  | Usuário    |          |              |         |          |             |          |          |          |
|                  | Cadastrar? |          |              | Editar? |          | Visualizar? |          | Excluir? |          |
|                  | Professor? | Monitor? | Aluno?       | Outro?  | Próprio? | Outros?     | Próprio? | Outros?  | Próprio? |
| <b>Admin</b>     | Sim        | Sim      | Sim          | Sim     | Sim      | Sim         | Sim      | Sim      |          |
| <b>Professor</b> |            | Sim      |              | Sim     | Sim      | Sim         | Sim      |          |          |
| <b>Monitor</b>   |            |          |              |         | Sim      | Sim         | Sim      |          |          |
| <b>Aluno</b>     |            |          | Se cadastrar |         | Sim      |             | Sim      |          |          |

Fonte: Elaborado pelo autor

A tabela 30 representa os direitos de acesso atribuídos para operações relacionadas aos horários do monitor.

**Tabela 30 - Direito de acesso para manipulação de Horários de Monitor**

| Função           | Atividade          |      |            |          |
|------------------|--------------------|------|------------|----------|
|                  | Horário de Monitor |      |            |          |
|                  | Criar?             | Ver? | Atualizar? | Excluir? |
| <b>Admin</b>     | X                  | X    | X          | X        |
| <b>Professor</b> | X                  | X    | X          | X        |
| <b>Monitor</b>   |                    | X    |            |          |
| <b>Aluno</b>     |                    | X    |            |          |

Fonte: Elaborado pelo autor

A tabela 31 representa os direitos de acesso atribuídos para operações relacionadas as presenças.

**Tabela 31 - Direito de acesso para manipulação de Presenças**

| Função           | Atividade         |            |             |          |
|------------------|-------------------|------------|-------------|----------|
|                  | Presença/Ausência |            |             |          |
|                  | Criar?            | Atualizar? | Visualizar? | Excluir? |
| <b>Admin</b>     | X                 | X          | X           | X        |
| <b>Professor</b> | X                 | X          | X           |          |
| <b>Monitor</b>   | X                 | X (Aluno)  | X           |          |
| <b>Aluno</b>     |                   |            |             |          |

Fonte: Elaborado pelo autor

A tabela 32 representa os direitos de acesso atribuídos para operações relacionadas ao grupo de estudo.

**Tabela 32 - Direito de acesso para manipulação de grupo de estudo**

| Função           | Atividade       |            |             |          |
|------------------|-----------------|------------|-------------|----------|
|                  | Grupo de estudo |            |             |          |
|                  | Criar?          | Atualizar? | Visualizar? | Excluir? |
| <b>Admin</b>     | X               | X          | X           | X        |
| <b>Professor</b> | X               | X          | X           | X        |
| <b>Monitor</b>   | X               | X          | X           |          |
| <b>Aluno</b>     |                 |            | X           |          |

Fonte: Elaborado pelo autor

A tabela 33 representa os direitos de acesso atribuídos para operações relacionadas ao Quiz.

**Tabela 33 - Direito de acesso para manipulação de Quiz**

| Função           | Atividade |      |            |          |
|------------------|-----------|------|------------|----------|
|                  | QUIZ      |      |            |          |
|                  | Criar?    | Ver? | Atualizar? | Excluir? |
| <b>Admin</b>     | X         | X    | X          | X        |
| <b>Professor</b> | X         | X    | X          | X        |
| <b>Monitor</b>   | X         | X    |            |          |
| <b>Aluno</b>     |           | X    |            |          |

Fonte: Elaborado pelo autor

A tabela 34 representa os direitos de acesso atribuídos para operações relacionadas as respostas do Quiz.

**Tabela 34 - Direito de acesso para manipulação de respostas do Quiz**

| Função           | Atividade     |      |            |          |
|------------------|---------------|------|------------|----------|
|                  | Resposta QUIZ |      |            |          |
|                  | Criar?        | Ver? | Atualizar? | Excluir? |
| <b>Admin</b>     |               | X    |            |          |
| <b>Professor</b> |               | X    |            |          |
| <b>Monitor</b>   | X             |      |            |          |
| <b>Aluno</b>     | X             |      |            |          |

Fonte: Elaborado pelo autor

A tabela 35 representa os direitos de acesso atribuídos para operações relacionadas as matérias.

**Tabela 35 - Direito de acesso para manipulação de matéria**

| Função    | Atividade |            |             |          |
|-----------|-----------|------------|-------------|----------|
|           | Matéria   |            |             |          |
|           | Criar?    | Atualizar? | Visualizar? | Excluir? |
| Admin     | X         | X          | X           | X        |
| Professor |           |            | X           |          |
| Monitor   |           |            | X           |          |
| Aluno     |           |            | X           |          |

Fonte: Elaborado pelo autor

A tabela 36 representa os direitos de acesso atribuídos para operações relacionadas aos cursos.

**Tabela 36 - Direito de acesso para manipulação de curso**

| Função    | Atividade |            |             |          |
|-----------|-----------|------------|-------------|----------|
|           | Curso     |            |             |          |
|           | Criar?    | Atualizar? | Visualizar? | Excluir? |
| Admin     | X         | X          | X           | X        |
| Professor |           |            | X           |          |
| Monitor   |           |            | X           |          |
| Aluno     |           |            | X           |          |

Fonte: Elaborado pelo autor

A tabela 37 representa os direitos de acesso atribuídos para operações relacionadas aos eventos.

**Tabela 37 - Direito de acesso para manipulação de eventos**

| Função    | Atividade |             |            |          |
|-----------|-----------|-------------|------------|----------|
|           | Eventos   |             |            |          |
|           | Criar?    | Visualizar? | Atualizar? | Excluir? |
| Admin     | X         | X           | X          | X        |
| Professor | X         | X           | X          | X        |
| Monitor   | X         | X           | X          |          |
| Aluno     |           | X           |            |          |

Fonte: Elaborado pelo autor

A tabela 38 representa os direitos de acesso atribuídos para operações relacionadas ao conteúdo dos grupos de estudo.

**Tabela 38 - Direito de acesso para manipulação de Usuários**

| Tipo de usuário  | Conteúdos |            |        |         |
|------------------|-----------|------------|--------|---------|
|                  | Cadastrar | Visualizar | Editar | Remover |
| <b>Admin</b>     | X         | X          | X      | X       |
| <b>Professor</b> | X         | X          | X      | X       |
| <b>Monitor</b>   |           | X          |        |         |
| <b>Aluno</b>     |           | X          |        |         |

Fonte: Elaborado pelo autor

### 3.6 APIs do Sistema de Monitorou

Para entender melhor a estrutura funcional do projeto, as próximas tabelas apresentam as classes da API do Sistema monitorou, sistema desenvolvido com o padrão arquitetural REST, sendo o webservice responsável pela conexão com o banco de dados para realizar as devidas operações.

#### 3.6.1 Especificação das bibliotecas utilizadas

Na linguagem *Ruby* o termo “*gem*” é utilizado para indicar o nome da biblioteca. Uma “Gema” (*gem*) é um conjunto de códigos e implementações que, a partir do momento que é incluída no projeto, disponibiliza uma nova função. Segue a lista de todas as “*gems*” utilizadas para o desenvolvimento da plataforma:

##### ***Devise:***

Gem utilizada para a autenticação do sistema. Contém todas as ferramentas já disponíveis para criptografia das senhas, validação de email, *tokens* de autenticação (utilizados nesta plataforma por ser API), tudo voltado para autenticação e validação de senhas e login.

##### ***Simple Token Authentication:***

Extensão do *devise* para a criação de *tokens* de acesso para os usuários, utilizados para garantir uma autenticação dupla do usuário e manter a sessão dele, fazendo uma verificação em cada *request* validando o *token* e *e-mail* do usuário, caso estiverem corretos a permissão é concedida, do contrário a permissão é negada.

##### ***Carrierwave:***

Gem utilizada para o upload de imagens e arquivos no projeto, com facilidade para conversão de uma imagem e armazenamento no banco de dados assim como os demais arquivos como PDF, Word e afins, contando com ferramentas de redimensionamento de imagens e demais *callbacks*.

### **Carrierwave-base64:**

Extensão da *gem carrierwave* para facilitar o *upload* de fotos via *API*, por ter a necessidade de envio de imagens via *request* as imagens precisam ser enviadas em base64 para o sistema fazer a conversão correta, base64 é um tipo de criptografia para arquivo que pode ser entendido e processado em sistemas web. Ele é responsável por já fazer a conversão da imagem de base64 para tipo arquivo e vice-versa.

### **Rack-cors:**

*Gem* utilizada para impedir ataques de *Ddos*. Ela impede que um mesmo endereço ou usuário faça 20 requisições por minuto e pode ser configurada para impedir que determinados endereços e *IPs* não façam requisição ao sistema.

### **Has-scope:**

*Gem* utilizada para para facilitar a aplicação filtros da url nos *models* e trazer dados de acordo com os filtros aplicados.

### **Pry-ByeBug:**

*Gem* utilizada para a ação de *debug* do sistema. Ela consegue parar a execução do sistema na linha em que se colocar a *tag* "*binding.pry*". No momento em que o sistema chegar nessa linha de execução, onde a *tag* foi inserida, o sistema para a sua execução e no *log* do servidor é iniciado um *console* onde todas as variáveis e informações até o momento criadas estão disponíveis para manuseio manual e minucioso dos objetos. De todas as pesquisadas para *rails* é uma das *gems* com maior funcionalidade para a ação de debug, de acordo com opinião dos autores deste documento.

### **Pundit:**

*Gem* utilizada para o controle de acesso e autorização, facilita a forma de criar as parametrizações de acesso, como estabelecer o que cada tipo de usuário tem acesso e pode fazer.

### 3.6.2 Endpoints

Por conta de o sistema ter sido desenvolvido em duas linguagens, temos de um lado um *WebService* que responde para as requisições nas telas da plataforma, os *endpoint's* são segmentados, porém os retornos para cada um são semelhantes devido a necessidade de manuseio dos dados no *front-end*.

#### 3.6.2.1 API de Usuários

A API de usuários realiza operações que possuem relações diretas com todos os tipos de usuários do sistema, seja ele aluno, monitor, professor ou admin, assim como descritos na tabela 39.

**Tabela 39 – Endpoints de usuários**

| <i>Endpoints</i> Usuários      |             |  |
|--------------------------------|-------------|--|
| <i>Endpoint</i>                | HTTP Method | Descrição  |
| api/v1/users                   | GET         | Retorna todos os usuários cadastrados.                           |
| api/v1/users?institution=[:id] | GET         | Retorna os usuários cadastrados em uma respectiva instituição.   |
| api/v1/users                   | POST        | Criação de um novo usuário                                       |
| api/v1/users/:id               | GET         | Retorna informações de um usuário específico, baseado em seu ID. |
| api/v1/users/:id               | PATCH / PUT | Atualiza um usuário baseado em seu ID.                           |
| api/v1/users/:id               | DELETE      | Remove o registro do usuário do sistema.                         |

**Fonte: Elaborado pelo autor**



A figura 7 exemplifica os dados retornados pelos *endpoints* de usuários:

**Figura 7 - JSON de retorno Usuários**

```
1 {
2   "id": 10,
3   "name": "Tiago Moraes",
4   "kind": "monitor",
5   "email": "tiago.moraes@monitor.com",
6   "authentication_token": "s93n_tnxRszwrssxhXAJ",
7   "profile_picture": {
8     "url": "/public/uploads/user/profile_picture/7/index.jpeg"
9   },
10  "course": {
11    "name": "Análise e Desenvolvimento de sistemas",
12    "subjects": [
13      {
14        "name": "Matematica"
15      },
16      {
17        "name": "Calculo"
18      },
19      {
20        "name": "Programação"
21      }
22    ]
23  },
24  "worktimes": [
25    {
26      "start_time": "2000-01-01T18:00:00.000Z",
27      "end_time": "2000-01-01T21:00:00.000Z",
28      "day": "friday"
29    }
30  ],
31  "study_groups": [],
32  "attendances": [],
33  "subjects": []
34 }
```

Fonte: Elaborado pelo autor

### 3.6.2.2 API de Matérias

A API de matérias realiza operações que possuem relação com os tipos de matérias do sistema, assim como descritos na tabela 40.

**Tabela 40 – Endpoints Matérias.**

| Endpoints Matérias  |             |   |
|---------------------|-------------|---|
| Endpoint            | HTTP Method | Descrição   |
| api/v1/subjects     | GET         | Retorna todos as matérias cadastradas                             |
| api/v1/subjects     | POST        | Criação de uma nova matéria                                       |
| api/v1/subjects/:id | GET         | Retorna informações de uma matéria específica, baseada em seu ID. |
| api/v1/subjects/:id | PATCH / PUT | Atualiza uma matéria baseada em seu ID.                           |

Fonte: Elaborado pelo autor

A figura 8 exemplifica os dados retornados pelos *endpoints* de matéria:

**Figura 8 - JSON de retorno de Matérias**

```
1 {  
2   "id": 12,  
3   "name": "Cálculo",  
4   "created_at": "2019-04-28T17:34:17.942Z",  
5   "updated_at": "2019-04-28T17:34:17.942Z"  
6 }
```

Fonte: Elaborado pelo autor

### 3.6.2.3 API de Horários de Monitores

A API de horários de monitores realiza operações que possuem relação com horário no calendário de atividades dos monitores, assim como descritos na tabela 41.

**Tabela 41 – Endpoints Horários de Monitores.**

| Endpoint Horários de monitores |             |  |
|--------------------------------|-------------|--|
| Endpoint                       | HTTP Method | Descrição  |
| api/v1/worktimes               | GET         | Retorna todos os horários cadastrados                            |
| api/v1/worktimes/:id           | GET         | Retorna informações de um horário específica, baseada em seu ID. |

|                      |             |  |
|----------------------|-------------|--|
| api/v1/worktimes/:id | PATCH / PUT | Atualiza o horário baseada em seu ID.    |
| api/v1/worktimes/:id | DELETE      | Remove o registro de horário do sistema. |

Fonte: Elaborado pelo autor

A figura 9 exemplifica os dados retornados no *endpoint* de horários de monitor.

Figura 9 - JSON de retorno de horários

```

1  [
2  {
3    "id": 1,
4    "start_time": "2000-01-01T18:00:00.000Z",
5    "end_time": "2000-01-01T21:00:00.000Z",
6    "day": "friday",
7    "user": {
8      "id": 6,
9      "name": "Moraes"
10   }
11 }
12 ]

```

Fonte: Elaborado pelo autor

### 3.6.2.4 API de Sessão

A API de sessão realiza operações de *login* e *logout* na plataforma, assim como descritos na tabela 42.

Tabela 42 – *Endpoints* de Sessão.

| Endpoint de Sessão  |             |   |
|---------------------|-------------|---|
| Endpoint            | HTTP Method | Descrição   |
| api/v1/sessions     | GET         | Inicia uma sessão na plataforma, via login e senha. |
| api/v1/sessions/:id | DELETE      | Finaliza a sessão.                                  |

Fonte: Elaborado pelo autor

A figura 10 exemplifica os dados retornados no *endpoint* de sessão.

**Figura 10 - JSON de retorno de sessão**

```
1 {  
2   "email": "admin@fatec.com",  
3   "authentication_token": "wQkvQGU7PzDfY7GyR_MS"  
4 }
```

**Fonte: Elaborado pelo autor**

### 3.6.2.5 API de Cursos

A API de cursos realiza operações que possuem relação com todos usuários, retornando o curso em que o aluno está matriculado, junto com suas respectivas matérias e horários de monitoria, assim como descritos na tabela 43.

**Tabela 43 – Endpoints de Cursos.**

| Endpoints Cursos   |             |  |
|--------------------|-------------|--|
| Endpoint           | HTTP Method | Descrição  |
| api/v1/courses     | GET         | Retorna todos os cursos cadastrados<br>*Contém filtro de instituição passado por parâmetro na URL. |
| api/v1/courses     | POST        | Criação de um novo curso.  |
| api/v1/courses/:id | GET         | Retorna informações de um curso específico, baseado em seu ID.                                     |
| api/v1/courses/:id | PATCH / PUT | Atualiza um curso baseado em seu ID.   |
| api/v1/courses/:id | DELETE      | Remove o registro do curso do sistema.   |

**Fonte: Elaborado pelo autor.**

A figura 11 exemplifica os dados retornados no *endpoint* de sessão.

**Figura 11 - JSON de retorno de Curso**

```
1 {  
2   "id": 8,  
3   "name": "Cálculo II",  
4   "institution_id": 1,  
5   "created_at": "2019-04-28T17:37:48.156Z",  
6   "updated_at": "2019-04-28T17:37:48.156Z"  
7 }
```

**Fonte: Elaborado pelo autor**

### 3.6.2.6 API de Eventos

A API de eventos realiza operações que possuem relação com eventos cadastrados por um monitor, sendo retornados no mural de horários do aluno, assim como descritos na tabela 44.

**Tabela 44 – Endpoints de Eventos**

| Endpoints Eventos |             |  |
|-------------------|-------------|--|
| Endpoint          | HTTP Method | Descrição  |
| api/v1/events     | GET         | Retorna todos os eventos importantes cadastrados.<br>*Contém filtro de instituição passado por parâmetro na URL. |
| Api/v1/events     | POST        | Criação de um novo evento.   |
| Api/v1/events/:id | GET         | Retorna informações de um evento específico, baseado em seu ID.  |
| Api/v1/events/:id | PATCH / PUT | Atualiza um evento baseado em seu ID.  |
| Api/v1/events/:id | DELETE      | Remove o registro do evento do sistema.  |

**Fonte: Elaborado pelo autor**

A figura 12 exemplifica os dados retornados no *endpoint* de eventos.

**Figura 12 - JSON de retorno de Eventos**

```

1  [
2  {
3      "id": 1,
4      "description": "P1",
5      "date": "2019-05-25",
6      "courses_subject_id": 1,
7      "courses_subject": {
8          "id": 1,
9          "subject": {
10             "name": "Matematica"
11         },
12         "course": {
13             "name": "Análise e Desenvolvimento de sistemas"
14         }
15     }
16 }
17 ]

```

Fonte: Elaborado pelo autor

### 3.6.2.7 API de Presença

A API de presença realiza operações que possuem relação com a confirmação ou não da presença de alunos e monitor no dia em que é acessada a plataforma, assim como descritos na tabela 45.

**Tabela 45 – Endpoints de Presenças.**

| <i>Endpoints</i> Presença      |             |   |
|--------------------------------|-------------|---|
| <i>Endpoint</i>                | HTTP Method | Descrição   |
| api/v1/attendances             | GET         | Retorna todos as presenças e faltas cadastradas.<br>*Contém filtro de instituição passado por parâmetro na URL. |
| api/v1/attendances?users=[:id] | GET         | Retorna as presenças e faltas de um respectivo usuário.   |
| api/v1/attendances             | POST        | Criação de uma nova presença ou falta.  |
| api/v1/attendances/:id         | PATCH / PUT | Atualiza uma falta ou presença baseado  |

|                        |        |  |
|------------------------|--------|--|
|                        |        | em seu ID.   |
| api/v1/attendances/:id | DELETE | Remove o registro de falta ou presença do sistema. |

Fonte: Elaborado pelo Autor

A figura 13 e figura 14 exemplifica os dados retornados pelos *endpoints* de presença, a figura 13 quando é criado uma presença para um aluno e a figura 14 quando é criado uma falta:

Figura 13 - JSON de retorno de Presença

```
1 {  
2   "id": 5,  
3   "user_id": 6,  
4   "kind": "presence",  
5   "date": "2019-05-25",  
6   "created_at": "2019-04-25T16:35:17.786Z",  
7   "updated_at": "2019-04-29T23:40:30.133Z"  
8 }
```

Fonte: Elaborado pelo autor

Figura 14 - JSON de retorno de Falta

```
1 {  
2   "id": 5,  
3   "user_id": 6,  
4   "kind": "absent",  
5   "date": "2019-05-25",  
6   "created_at": "2019-04-25T16:35:17.786Z",  
7   "updated_at": "2019-04-29T23:40:30.133Z"  
8 }
```

Fonte: Elaborado pelo autor

### 3.6.2.8 API de Estudos

A API de estudos realiza operações que possuem relação com criação e participação de um grupo de estudos, disponibilizando conteúdos pertinentes à matéria, assim como descritos na tabela 46.

**Tabela 46 – Endpoints de Grupo de Estudos.**

| <i>Endpoints Grupo de Estudos</i> |             |  |
|-----------------------------------|-------------|--|
| <i>Endpoint</i>                   | HTTP Method | Descrição  |
| api/v1/study_groups               | GET         | Retorna todos os grupos de estudos cadastrados.                          |
| api/v1/study_groups               | POST        | Criação de um novo grupo de estudo.                                      |
| api/v1/study_groups/:id           | GET         | Retorna informações de um grupo de estudo específico, baseado em seu ID. |
| api/v1/study_groups/:id           | PATCH / PUT | Atualiza um grupo de estudos baseado em seu ID.                          |
| api/v1/study_groups/:id           | DELETE      | Remove o registro do grupo de estudos do sistema.                        |
| /api/v1/study_groups/join_group   | POST        | Entra num grupo específico.  |

**Fonte: Elaborado pelo autor.**



A figura 15 exemplifica os dados retornados pelos *endpoints* de grupo de estudos:

**Figura 15 - JSON de retorno de Grupo de Estudos**

```
1 [
2   {
3     "id": 1,
4     "name": "Grupo Alpha",
5     "theme": "Estruturação de um sistema",
6     "institution": {
7       "name": "Fatec"
8     },
9     "contents": [
10      {
11        "id": 1,
12        "study_group_id": 1,
13        "document": {
14          "url": "/uploads/content/document/1/document.pdf"
15        },
16        "image": {
17          "url": null
18        },
19        "kind": "document"
20      },
21      {
22        "id": 2,
23        "study_group_id": 1,
24        "document": {
25          "url": "/uploads/content/document/2/document.pdf"
26        },
27        "image": {
28          "url": null
29        },
30        "kind": "document"
31      }
32    ]
33  }
34 ]
```

Elaborado pelo autor

### 3.6.2.9 API de Conteúdos

A API de conteúdos realiza operações que possuem relação com a criação e alteração de conteúdos para o acesso de alunos, que serão disponibilizados no grupo de estudos, assim como descritos na tabela 47.

**Tabela 47 – Endpoints de Conteúdos.**

| Endpoints Conteúdos |             |   |
|---------------------|-------------|---|
| Endpoint            | HTTP Method | Descrição                                 |
| api/v1/contents     | GET         | Retorna todos os conteúdos cadastrados.   |
| Api/v1/contents     | POST        | Criação de um novo conteúdo.              |
| Api/v1/ontentes/:id | PATCH / PUT | Atualiza um conteúdo baseado em seu ID.   |
| Api/v1/ontentes/:id | DELETE      | Remove o registro do conteúdo do sistema. |

Fonte: Elaborado pelo autor.

A figura 16 exemplifica os dados retornados pelos *endpoints* de Conteúdos:

**Figura 16 - JSON de retorno de Conteúdos**

```
1 {
2   "id": 1,
3   "study_group_id": 1,
4   "document": {
5     "url": "public/assets/formulas_calculo.pdf"
6   },
7   "image": {
8     "url": null
9   },
10  "kind": "document",
11  "created_at": "2019-05-08T23:38:00.600Z",
12  "updated_at": "2019-05-08T23:38:00.600Z"
13 }
```

Fonte: Elaborado pelo autor

### 3.6.2.10 API de Quiz

A API de quiz realiza operações que possuem relação com criação, alteração e visualização de quiz criados pelo monitor. Após ser criado, torna-se disponível aos alunos para serem respondidas, assim como descritos na tabela 48.

**Tabela 48 – Endpoints Quiz**

| <i>Endpoints Quiz</i>                              |             |  |
|--|-------------|--|
| <i>Endpoint</i>                                    | HTTP Method | Descrição  |
| api/v1/questionnaires                              | GET         | Retorna todos os questionários cadastrados.                                  |
| api/v1/questionnaires                              | POST        | Criação de um novo questionário.   |
| api/v1/questionnaires/:id                          | GET         | Retorna informações de um questionário específico, baseado em seu ID.        |
| api/v1/questionnaires/:id                          | PATCH / PUT | Atualiza um questionário baseado em seu ID.                                  |
| api/v1/questionnaires/:id                          | DELETE      | Remove o registro do questionário do sistema.                                |
| /api/v1/questionnaires_answers                     | POST        | Cria uma resposta para um questionário                                       |
| /api/v1/questionnaires_answers?correct_answers=:id | GET         | Retorna todas as respostas corretas de um questionário baseado em seu ID     |
| /api/v1/questionnaires_answers?questionnaire=:id   | GET         | Retorna todas as respostas de um questionário específico, baseado em seu ID. |
| /api/v1/questionnaires_answers?student=:id         | GET         | Retorna todas as respostas de um estudante baseado em seu ID.                |

**Fonte: Elaborado pelo autor**

A figura 17 exemplifica os dados retornados pelos *endpoints* de questionários:

**Figura 17 - JSON de retorno de Questionários**

```
1 {
2   "id": 1,
3   "subject_id": 1,
4   "description": "Qual é a forma mais fácil de fazer um loop em
5     programação?",
6   "institution_id": 1,
7   "questionnaire_options": [
8     {
9       "id": 1,
10      "correct": false,
11      "description": "Utilizando um 'if' "
12    },
13    {
14      "id": 2,
15      "correct": true,
16      "description": "Utilizando 'for' "
17    },
18    {
19      "id": 3,
20      "correct": false,
21      "description": "Utilizando um ponteiro"
22    }
23  ],
24  "subject": {
25    "id": 1,
26    "name": "Programação"
27  },
28  "institution": {
29    "id": 1,
30    "name": "Fatec"
31  }
}
```

Fonte: Elaborado pelo autor

A Figura 18 exemplifica os dados retornados pelos *endpoints* de resposta de questionários:

**Figura 18 - JSON de retorno de Resposta de Questionários**

```
1 {
2   "id": 3,
3   "questionnaire_id": 2,
4   "questionnaire_option_id": 4,
5   "user_id": 2,
6   "created_at": "2019-05-15T00:11:48.637Z",
7   "updated_at": "2019-05-15T00:11:48.637Z"
8 }
```

Fonte: Elaborado pelo autor

## 4 DESENVOLVIMENTO

A metodologia ágil utilizada durante o processo de desenvolvimento é denominada de *Scrum*, isto implica que o projeto é dividido em etapas (*sprints*), cada uma durando três semanas e nas fases finais do projeto, uma semana cada. Essas etapas possuem objetivos e metas a serem concluídas, assim mantendo um melhor fluxo e organização de trabalho, que por fim são compostas de processos e definições para ir criando o sistema de forma estruturada e linear.

### 4.1 Etapas de Desenvolvimento

Desde o início da produção do projeto, foi estabelecida aplicação da metodologia Scrum, para poder ter mais organização e melhor planejamento das entregas. Inicialmente as *sprints* (neste documento determinados como “protótipos”) foram estabelecidas com prazo de 21 dias, após o período de um semestre e, por conta da necessidade de entrega em uma data mais próxima, foram estabelecidas sprints de 7 dias com entregas menores e mais objetivas, garantindo uma boa entrega e maior velocidade na produção do sistema. Cada protótipo era planejado semanalmente e, após a reunião para estabelecer as metas da sprint, a equipe se unia para os alinhamentos finais e início das atividades de desenvolvimento, sempre mantendo uma boa comunicação para que, tanto a parte do *backend*, como a do *frontend* estivesse alinhada e garantindo uma boa fluidez do sistema.

#### 4.1.1 Protótipo 1

No dia 28 de agosto de 2018 o grupo se encontrou para realizar o planejamento da primeira *sprint* (21 dias de desenvolvimento – prazo 18 de Setembro de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas á criação do banco de dados, criação da tela de cadastro, tela de login do aluno, monitor e admin, criação do *back-end* do cadastro e login, documentação do *login* e cadastro. A Tabela 49 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

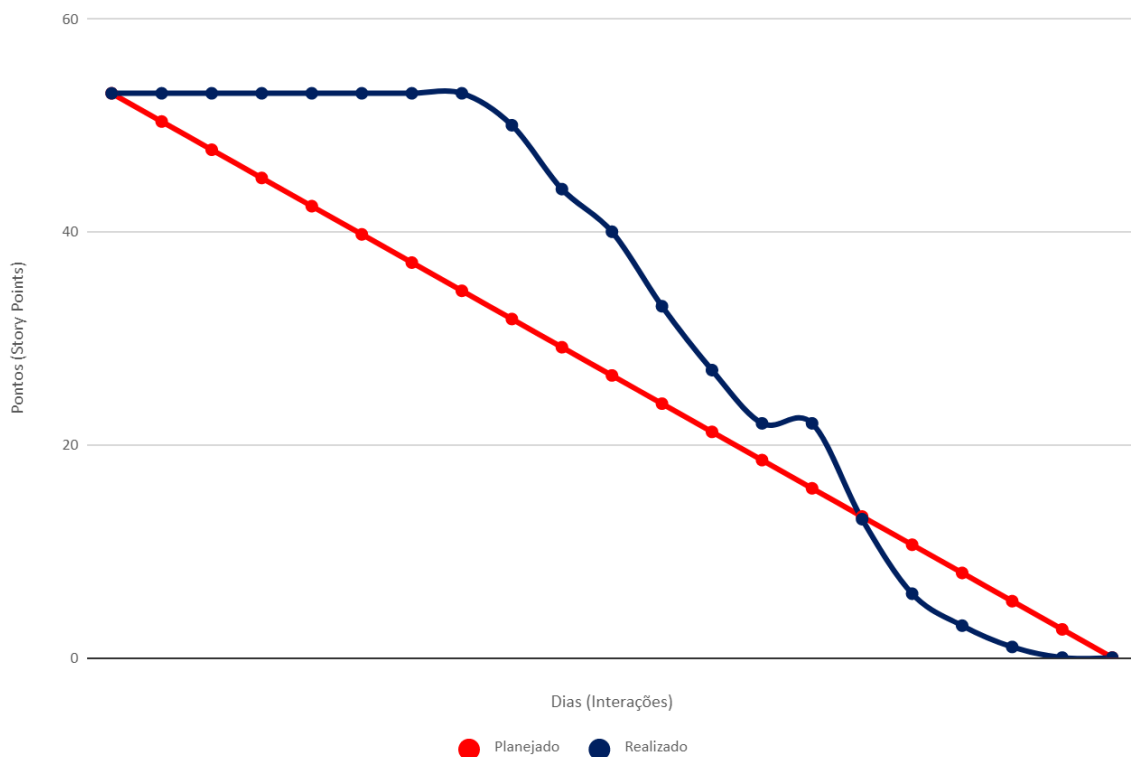
**Tabela 49 – Planejamento realizado para primeira entrega.**

| Atividade                           | Tempo | Pontos |
|-------------------------------------|-------|--------|
| Criação do banco de dados           | 1     | 3      |
| Criação da tela de cadastro         | 3     | 9      |
| Criação da tela de login do aluno   | 1     | 5      |
| Criação da tela de login do admin   | 1     | 5      |
| Criação da tela de login do monitor | 1     | 5      |
| <i>Back-end</i> do cadastro         | 4     | 9      |
| <i>Back-end</i> do login            | 2     | 7      |
| Documentação do login               | 1     | 5      |
| Documentação do cadastro            | 1     | 5      |

Fonte: Elaborado pelo autor (2018).

Durante os 21 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 19 apresenta o gráfico de *Burndown* da *sprint* 1, destacando o planejamento de baixas nos pontos e as baixas realizadas pela equipe. Percebe-se dificuldades no desenvolvimento das atividades ao longo dos dias a equipe teve problemas com o andamento das atividades planejadas – problemas que poderiam ser solucionados com mais comunicação entre a equipe.

**Figura 19 – Gráfico de *Burndown* da entrega 1.**



Fonte: Elaborado pelo autor (2018).

No dia 17 de Setembro de 2018 a equipe se reuniram para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Apesar das dificuldades encontradas a maior parte das atividades planejadas foram concluídas;
- O que deu errado: Houve falta de comunicação entre os integrantes, falta de gerenciamento do tempo e problemas na preparação do ambiente para o início do desenvolvimento;
- Ações de melhorias: Melhorar o gerenciamento do tempo e comunicação entre os membros da equipe;

#### 4.1.2 Protótipo 2

No dia 18 de Setembro de 2018 o grupo se encontrou para realizar o planejamento da segunda *sprint* (21 dias de desenvolvimento – prazo 9 de Outubro de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas á conclusão do *back-end* do *login* e cadastro, criação de *APIs*, criação de *request* de *login* e cadastro, criação da homepage, definição de requisitos funcionais e não funcionais. A Tabela 50 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

**Tabela 50 – Planejamento realizado para segunda entrega.**

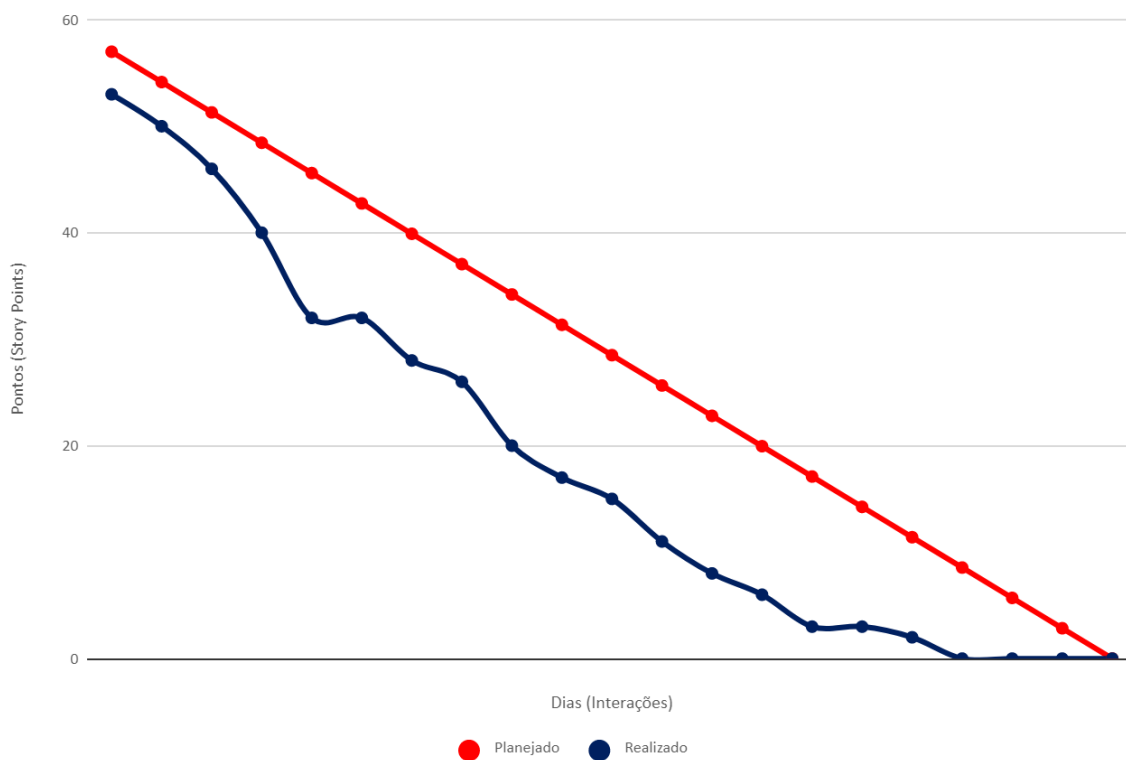
| Atividade                   | Tempo | Pontos |
|-----------------------------|-------|--------|
| Conclusão back-end login    | 5     | 10     |
| Conclusão back-end cadastro | 5     | 8      |
| Criação de APIs             | 5     | 8      |

|  |   |    |
|--|---|----|
| Criação de request login               | 5 | 8  |
| Criação de request cadastro            | 5 | 8  |
| Criação da <i>homepage</i>             | 7 | 10 |
| Requisitos funcionais e não funcionais | 1 | 5  |

Fonte: Elaborado pelo autor (2018).

Durante os 21 dias de desenvolvimento, a equipe não realiza baixas na pontuação conforme as atividades eram concluídas. A Figura 20 apresenta o gráfico de *Burndown* da entrega 2, destacando o planejamento sem baixas nos pontos pela equipe. Percebe-se que no desenvolvimento das atividades ao longo dos dias a equipe não teve problemas com o andamento das atividades planejadas.

Figura 20 – Gráfico de *Burndown* da entrega 2.



Fonte: Elaborado pelo autor (2018).

No dia 8 de Outubro de 2018 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos



próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Todo planejamento ocorreu como esperado;
- O que deu errado: Não houve falhas durante a execução das atividades planejadas;
- Ações de melhorias: Melhorar o tempo das atividades planejadas para aplicar o tempo de sobra em outras atividades;
- Observações: Todas atividades executadas funcionaram corretamente.

#### 4.1.3 Protótipo 3

No dia 9 de Outubro de 2018 o grupo se encontrou para realizar o planejamento da terceira *sprint* (20 dias de desenvolvimento – prazo 30 de Outubro de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a adicionar campo extra na tabela de horários de monitor, correção das *APIs*, melhorar *dashboard*, alterar login para RA ou *e-mail*, tela de castro de aluno independente do admin, diagrama de sequência. A Tabela 51 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

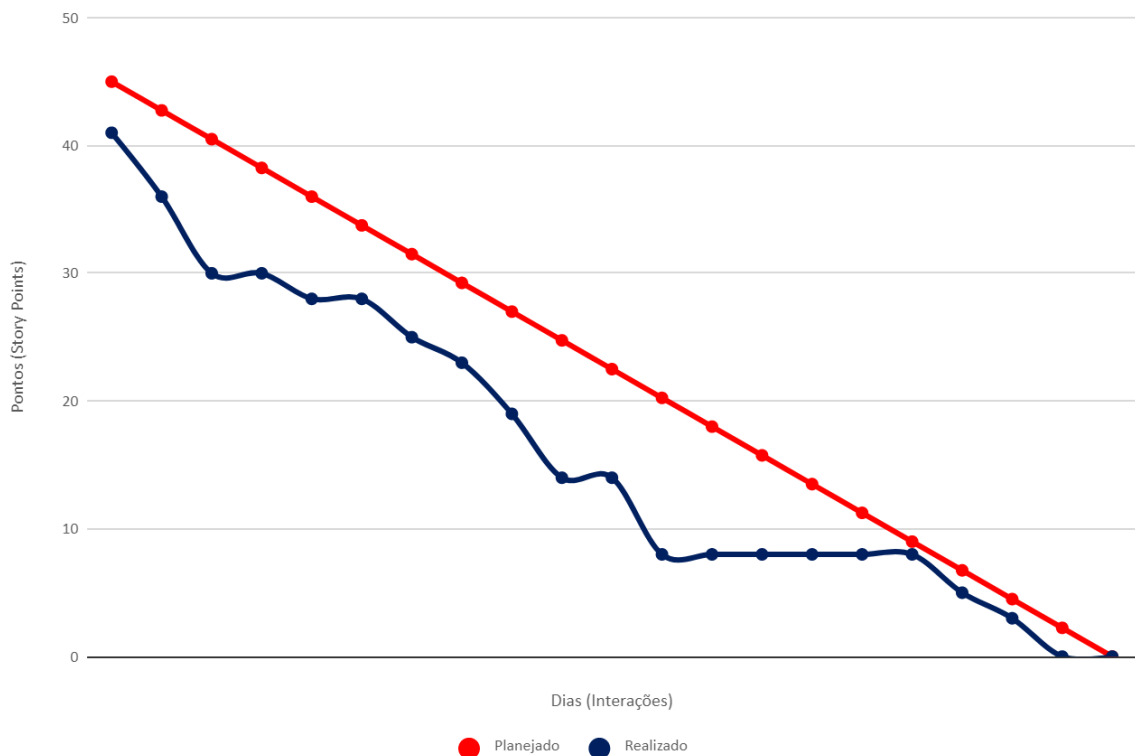
**Tabela 51 – Planejamento realizado para terceira entrega.**

| <b>Atividade</b>                                    | <b>Tempo</b> | <b>Pontos</b> |
|---|--------------|---------------|
| Adicionar campo extra na tabela horários de monitor | 1            | 8             |
| Correção das APIs                                   | 3            | 10            |
| Melhorar dashboard                                  | 3            | 9             |
| Alterar login para RA ou e-mail                     | 1            | 6             |
| Tela de cadastro de aluno (independe do admin)      | 2            | 5             |
| Criação da homepage                                 | 2            | 7             |

**Fonte: Elaborado pelo autor (2018).**

Durante os 20 dias de desenvolvimento, a equipe não realiza baixas na pontuação conforme as atividades eram concluídas. A Figura 21 apresenta o gráfico de *Burndown* da entrega 3, destacando o planejamento sem baixas nos pontos pela equipe. Percebe-se que no desenvolvimento das atividades ao longo dos dias a equipe não teve problemas com o andamento das atividades planejadas.

**Figura 21 – Gráfico de *Burndown* da entrega 3.**



**Fonte: Elaborado pelo autor (2018).**

No dia 29 de Outubro de 2018 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Todo planejamento ocorreu como esperado;

- O que deu errado: Não houve falhas durante a execução das atividades planejadas;
- Ações de melhorias: Implementar campo cidade no registro filtros de retorno de dados;
- Observações: Todas atividades executadas funcionaram corretamente.

#### 4.1.4 Protótipo 4

No dia 30 de Outubro de 2018 o grupo se encontrou para realizar o planejamento da quarta *sprint* (21 dias de desenvolvimento – prazo 20 de Novembro de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a correções na *dashboard*, correções no calendário, criação de *requests*, vincular o horário com a criação de um monitor, *login* de aluno sem instituição, aplicar filtro de curso por instituição, filtro de alunos por instituição. A Tabela 52 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

**Tabela 52 – Planejamento realizado para quarta entrega.**

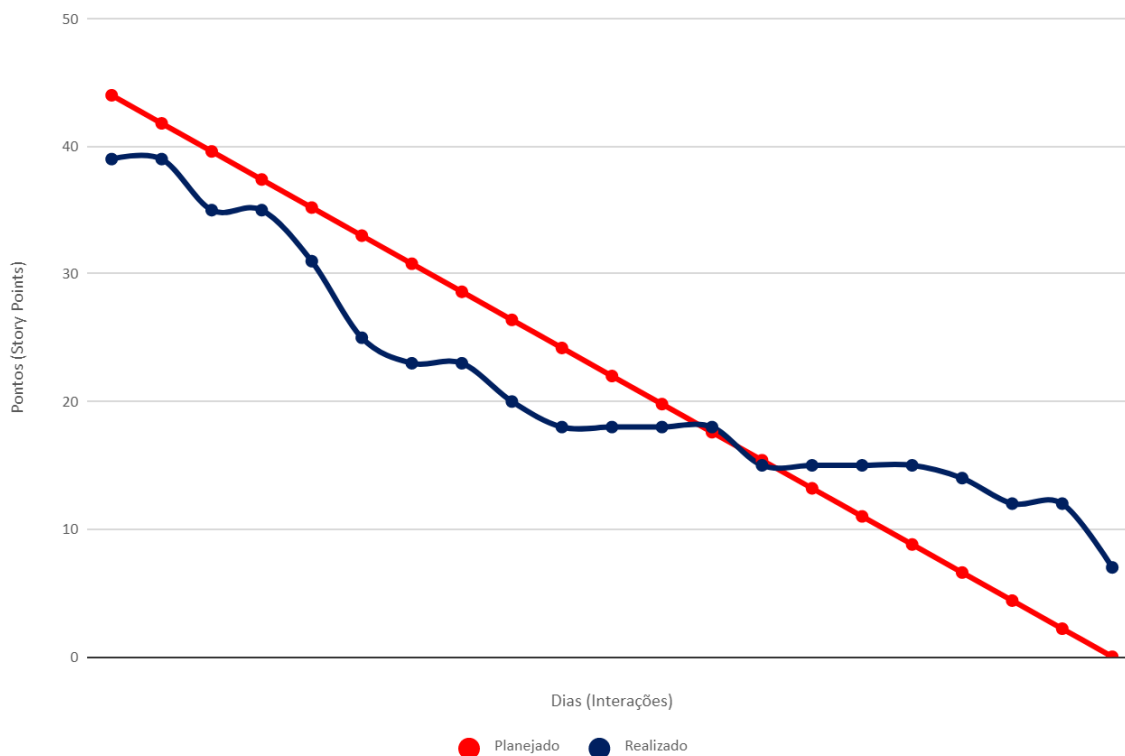
| Atividade                                      | Tempo | Pontos |
|--|-------|--------|
| Correções na dashboard                         | 2     | 5      |
| Correções no calendário                        | 3     | 4      |
| Criação de requests de login                   | 2     | 8      |
| Criação das demais requests                    | 3     | 10     |
| Vincular o horário com a criação de um monitor | 2     | 4      |
| Login de aluno sem instituição                 | 1     | 3      |
| Aplicar filtro de curso por instituição        | 2     | 5      |
| Filtro de alunos por instituição               | 2     | 5      |

**Fonte: Elaborado pelo autor (2018).**

Durante os 21 dias de desenvolvimento, a equipe conseguiu entregar tudo que foi planejado com algumas restrições, tendo uma boa comunicação e auxílio de todos os membros da equipe para a solução dos problemas encontrados. O principal

problema foi o desenvolvimento de uma *feature* onde os integrantes tinha pouco experiência, assim como representado pela figura 22.

Figura 22 – Gráfico de *Burndown* da entrega 4.



Fonte: Elaborado pelo autor (2018).

No dia 19 de Novembro de 2018 a equipe se reuniram para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: A criação das *requests* e melhorias na *deashboard*;
- O que deu errado: A autenticação para o login;
- Ações de melhorias: Efetuar pesquisa referente ao conteúdo a desenvolver.

#### 4.1.5 Protótipo 5

No dia 20 de Novembro de 2018 o grupo se encontrou para realizar o planejamento da quinta *sprint* (20 dias de desenvolvimento – prazo 11 de Dezembro de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas ao *deploy* e integração das linguagens, *requests* e *APIs* do *front-and*, documentar banco de dados, filtro de matéria por curso, dashboard do usuário, conclusão de criação de monitor relacionando horas, caso de uso das funções, resumo na documentação, documentação de testes. A Tabela 53 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

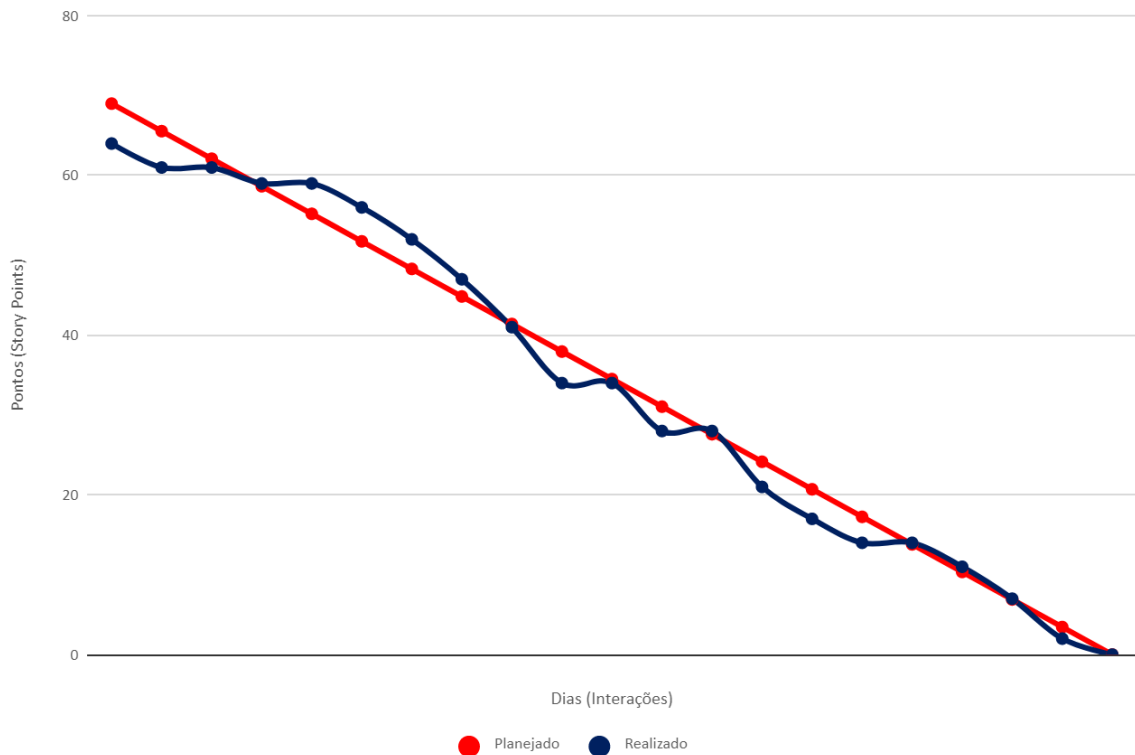
**Tabela 53 – Planejamento realizado para quinta entrega.**

| <b>Atividade</b>                             | <b>Tempo</b> | <b>Pontos</b> |
|--|--------------|---------------|
| Deploy e integração das linguagens           | 2            | 10            |
| Requests e APIs do front-and                 | 4            | 10            |
| Documento banco de dados                     | 1            | 5             |
| Filtro de matéria por curso                  | 5            | 8             |
| Dashboard do usuário                         | 5            | 8             |
| Conclusão criação monitor relacionando horas | 3            | 8             |
| Caso de uso das funções                      | 4            | 10            |
| Resumo documentação                          | 5            | 8             |

**Fonte: Elaborado pelo autor (2018).**

Durante os 20 dias de desenvolvimento, a equipe conseguir realizar com exatidão tudo que foi planejado com pequeno contratempo que foi rapidamente solucionado com auxílio de todos integrantes da equipe. Assim como representado pela figura 23.

Figura 23 – Gráfico de *Burndown* da entrega 5.



Fonte: Elaborado pelo autor (2018).

No dia 10 de Dezembro de 2018 a equipe se reuniram para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Melhoria na criação monitor relacionando horas de trabalho, *deashboard* de usuários;
- O que deu errado: Não houve falhas durante a execução das atividades planejadas;
- Ações de melhorias: continuar o desenvolvimento do projeto fora do período letivo;

#### 4.1.6 Protótipo 6

No dia 7 de Fevereiro de 2019 o grupo se encontrou para realizar o planeamento da sexta *sprint* (7 dias de desenvolvimento – prazo 13 de Fevereiro de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas ao grupo de estudo, erro na edição de hora do monitor, perfil aberto, cadastro, editar perfil, criar dashboard aluno, filtro de matérias e criar responsivo do site. A Tabela 54 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

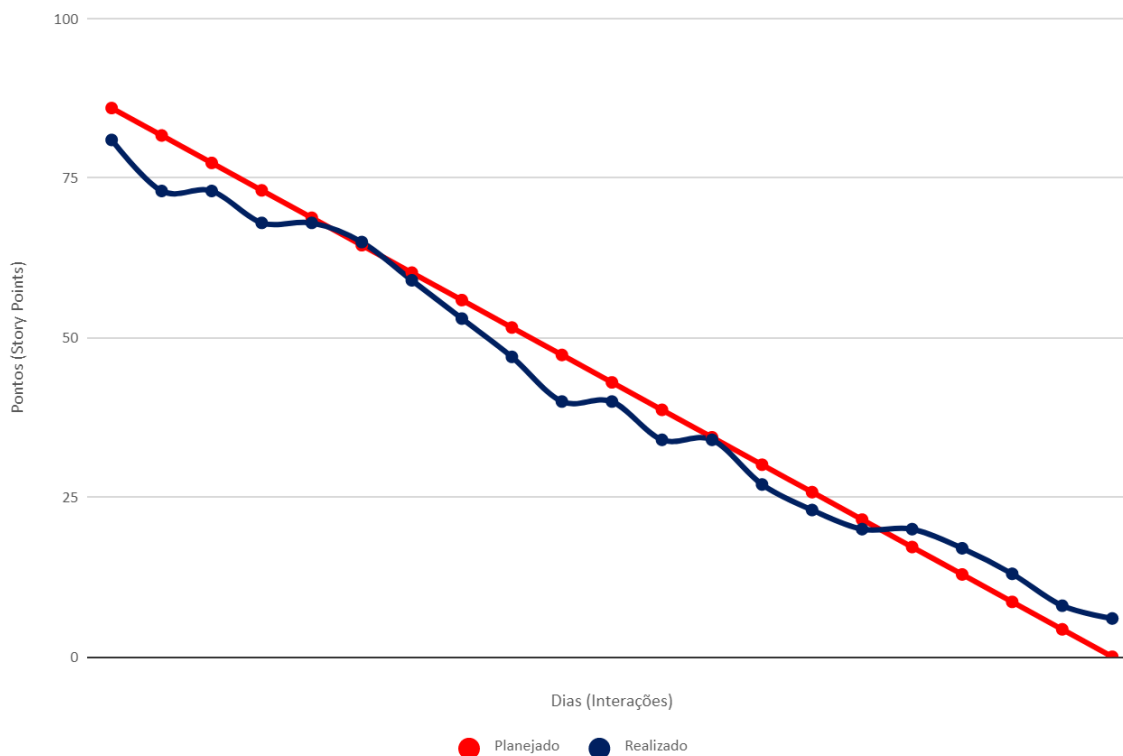
**Tabela 54 – Planeamento realizado para sexta entrega.**

| <b>Atividade</b>                  | <b>Tempo</b> | <b>Pontos</b> |
|-----------------------------------|--------------|---------------|
| Grupo de estudo                   | 3            | 10            |
| Erro na edição de hora do monitor | 2            | 8             |
| Perfil aberto                     | 5            | 9             |
| Cadastro                          | 5            | 9             |
| Criar <i>dashboard</i> aluno      | 2            | 10            |
| Filtro de matérias                | 2            | 10            |
| Criar responsivo do site          | 2            | 10            |

**Fonte: Elaborado pelo autor (2019).**

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado. Assim como representado pela figura 24.

**Figura 24 – Gráfico de *Burndown* da entrega 6.**



**Fonte: Elaborado pelo autor (2019).**

No dia 7 de Fevereiro de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Editar perfil, Criar *dashboard* do aluno, grupo de estudos e edição da hora do monitor;
- O que deu errado: Tivemos grande problema com a criação de perfil aberto;



- Ações de melhorias: Perfil aberto, cadastro, criar responsivo do site e criar filtros de matérias foi removido por não ser necessário;

#### 4.1.7 Protótipo 7

No dia 14 de Fevereiro de 2019 o grupo se encontrou para realizar o planejamento da sétima *sprint* (7 dias de desenvolvimento – prazo 20 de Fevereiro de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas ao *backend* perfil (foto, nome), relação n:n grupo de estudos, criar cadastro aberto, retornar informações do perfil do usuário, opcional: retornar matérias cadastradas aluno, melhorar retorno *endpoint worktimes* (matéria), Adicionar *card* com monitoria nas matérias na *dashboard* do aluno, botão fab presença do monitor, botão fab presença do aluno (marcado pelo monitor), corrigir texto horários do monitor, resolver retorno em inglês no cadastro monitor e opcional: relógio timer tempo de monitoria do monitor. A Tabela 55 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

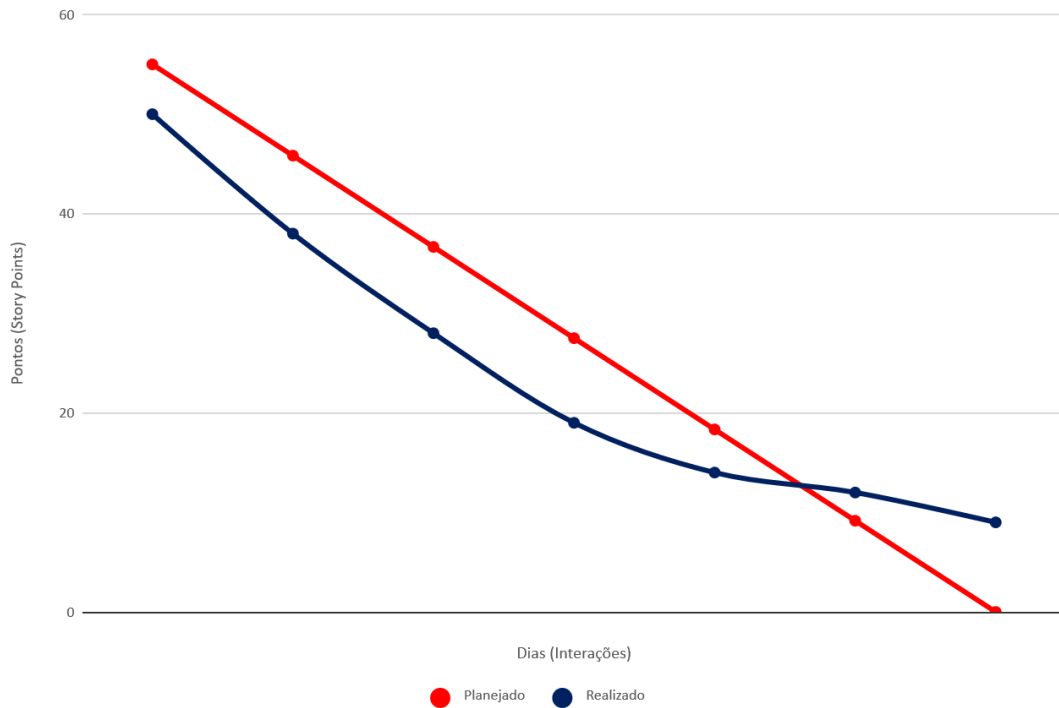
**Tabela 55 – Planejamento realizado para sétima entrega.**

| Atividade   | Tempo | Pontos |
|---|-------|--------|
| Backend perfil (foto, nome)   | 3     | 10     |
| Relação n:n grupo de estudos  | 2     | 5      |
| Criar cadastro aberto   | 1     | 5      |
| Retornar informações do perfil do usuário                                     | 1     | 7      |
| Opcional: Retornar matérias cadastradas aluno                                 | 2     | 8      |
| Melhorar retorno <i>endpoint worktimes</i> (matéria)                          | 1     | 8      |
| Adicionar <i>card</i> com monitoria nas matérias na <i>dashboard</i> do aluno | 2     | 8      |
| Botão fab presença do monitor   | 1     | 5      |
| Botão fab presença do aluno (marcado pelo monitor)                            | 1     | 5      |
| Corrigir texto horário do monitor   | 2     | 9      |
| Resolver retorno em inglês no cadastro monitor                                | 1     | 7      |
| Opcional: relógio timer tempo de monitoria do monitor                         | 1     | 9      |

**Fonte: Elaborado pelo autor (2018).**

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado. Assim como representado pela figura 25.

Figura 25 – Gráfico de *Burndown* da entrega 7.



Fonte: Elaborado pelo autor

No dia 20 de Fevereiro de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Adicionar card com monitoria nas matérias na *dashboard* do aluno, botão fab presença do monitor/aluno foi transformado em um único botão com abas separadas para aluno/monitor, corrigir texto horários do monitor, corrigir retorno em inglês no cadastro de monitor, relação n:n grupo de estudos, *backend* perfil (foto, nome), retornar mais informações do perfil do usuário (matéria, horário - incluir no retorno dos usuários);

- O que deu errado: Cadastro aberto ainda é foi um problema para o desenvolvimento; Relação n:n grupo de estudos;
- Ações de melhorias: Criar cadastro aberto, opcional: relógio timer tempo de monitoria do monitor ainda não foi feito;

#### 4.1.6 Protótipo 8

No dia 21 de Março de 2019 o grupo se encontrou para realizar o planejamento da oitava *sprint* (7 dias de desenvolvimento – prazo 20 de Fevereiro de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a encodar base64 da imagem de perfil do usuário, resolver cadastro aberto, retornar kind no retorno de usuários, criar relação usuário-monitor com matérias 1:n, Lista de Presenças (monitor e aluno), Adicionar eventos importantes no card de matérias do aluno, arrumar o fab de presença, relógio timer de presença monitor, retornar presenças de presença(aluno) e Presença + falta(monitor), adicionar eventos importantes no card de matérias do aluno, validar hora inicio - hora fim na criação de monitor. A Tabela 56 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

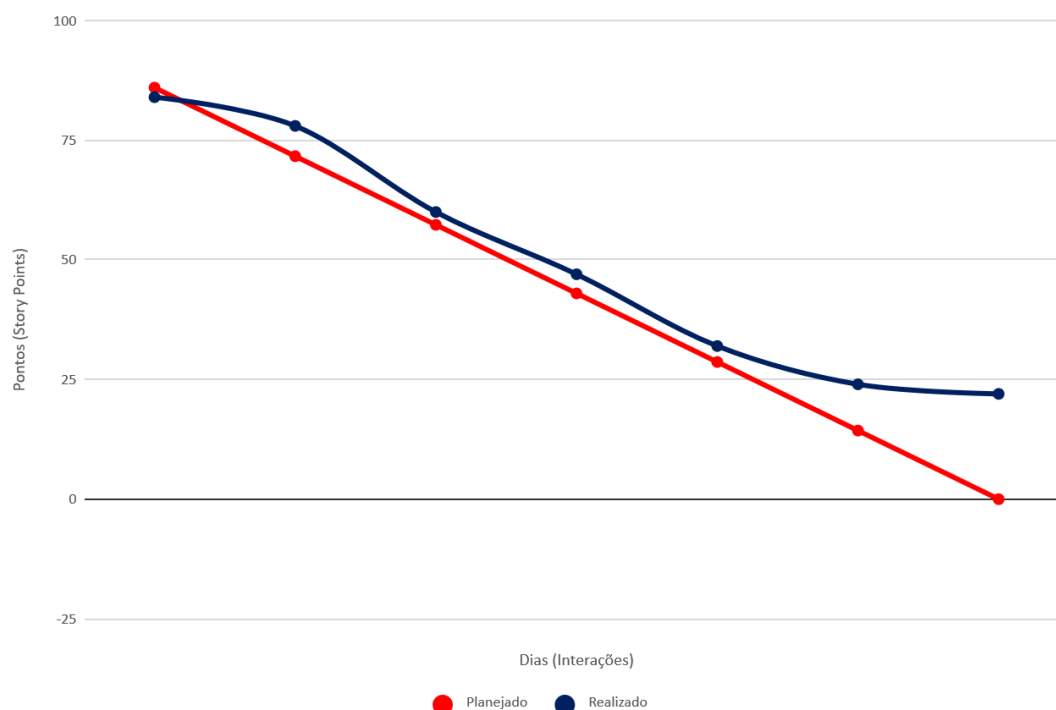
**Tabela 56 – Planejamento realizado para oitava entrega.**

| Atividade   | Tempo | Pontos |
|---|-------|--------|
| Encodar base64 da imagem de perfil do usuário                     | 2     | 10     |
| Resolver cadastro Aberto  | 1     | 5      |
| Retornar kind no retorno de usuários                              | 1     | 7      |
| Criar relação usuário-monitor com matérias 1:n                    | 1     | 5      |
| Lista de Presenças (monitor e aluno)                              | 2     | 8      |
| Adicionar eventos importantes no card de matérias do aluno        | 1     | 7      |
| Relação n:n de curso-matéria                                      | 1     | 8      |
| Adicionar card com monitoria nas matérias do aluno                | 2     | 8      |
| Arrumar o <i>fab</i> de presença                                  | 1     | 5      |
| Relógio timer de presença monitor                                 | 2     | 8      |
| Retornar presenças de presença(aluno) e Presença + falta(monitor) | 1     | 7      |
| Adicionar eventos importantes no card de matérias do aluno        | 1     | 7      |
| Validar hora inicio - hora fim na criação de monitor              | 1     | 9      |

Fonte: Elaborado pelo autor (2019).

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado. Assim com representado pela figura 26

Figura 26 – Gráfico de *Burndown* da entrega 8.



Fonte: Elaborado pelo autor (2019).

No dia 27 de Março de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Adicionar card com monitoria nas matérias do aluno, foi adicionado o nome do monitor responsável pela matéria, juntamente com seus horários, retornar presenças de presença(aluno) e Presença + falta(monitor) foi adicionado como um novo campo no perfil do usuário, junto com suas informações básicas, encodar base64 da

imagem de perfil do usuário, retornar kind no retorno de usuários, criar relação usuário-monitor com matérias 1:n, lista de Presenças (monitor e aluno), adicionar eventos importantes no card de matérias do aluno;

- O que deu errado: Cadastro aberto ainda é um problema para solucionar; Criação de relação entre monitor e matéria foi pensado previamente mas ainda com dúvidas que foram alinhadas com o orientador;
- Ações de melhorias: Perfil aberto/cadastro, Relógio timer de presença monitor foi feito mas ainda não está pegando o horário atual no monitor, apenas um de teste, Adicionar eventos importantes no *card* de matérias do aluno ainda não foi feito, Validar hora início - hora fim na criação de monitor foi feito mas ainda precisa retornar uma mensagem quando preenchido errado;

#### 4.1.8 Protótipo 9

No dia 28 de Março de 2019 o grupo se encontrou para realizar o planejamento da nona *sprint* (7 dias de desenvolvimento – prazo 20 de Fevereiro de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a resolver o cadastro aberto, correção da relação de usuário-monitor com matérias 1:n (*user id* no *model* de *subject*), incluir usuário professor, incluir faltas no *model* de presença (incluir *kind*), adicionar relatório de presenças(aluno) e presenças + faltas(monitor) no perfil do usuário, adicionar eventos importantes no *card* de matérias na *dashboard* do aluno. A Tabela 57 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

**Tabela 57 – Planejamento realizado para nona entrega.**

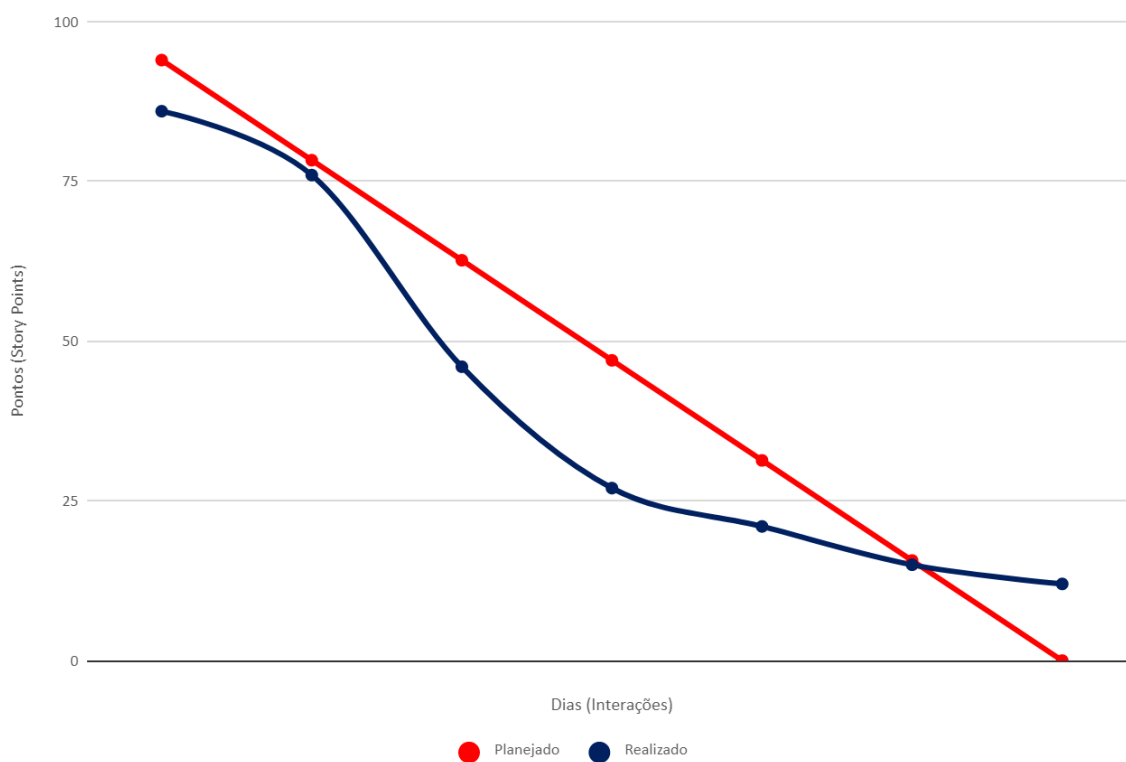
| Atividade  | Tempo | Pontos |
|--|-------|--------|
| Resolver cadastro aberto   | 5     | 10     |
| Correção da relação de usuário-monitor com matérias 1:n ( <i>user id</i> no <i>model</i> de <i>subject</i> ) | 1     | 3      |
| Incluir usuário professor  | 1     | 10     |

|  |   |   |
|--|---|---|
| Incluir faltas no model de presença (incluir <i>kind</i> )                                   | 1 | 5 |
| Adicionar relatório de presenças (aluno) e presenças + faltas (monitor) no perfil do usuário | 1 | 8 |
| Adicionar eventos importantes no <i>card</i> de matérias na <i>dashboard</i> do aluno        | 2 | 7 |

Fonte: Elaborado pelo autor (2019).

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado. Assim como representado pela figura 27.

Figura 27 – Gráfico de *Burndown* da entrega 9.



Fonte: Elaborado pelo autor (2018).

No dia 03 de Abril de 2019 a equipe se reuniu para realizar a revisão do *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Adicionar card com monitoria nas matérias do aluno, foi adicionado o nome do monitor responsável pela matéria, juntamente com seus horários, retornar presenças de presença(aluno) e Presença + falta(monitor) foi adicionado como um novo campo no perfil do usuário, junto com suas informações básicas, encodar base64 da imagem de perfil do usuário, retornar kind no retorno de usuários, criar relação usuário-monitor com matérias 1:n, lista de Presenças (monitor e aluno), adicionar eventos importantes no card de matérias do aluno;
- O que deu errado: Cadastro aberto estava com alguns problemas, mas próximos de ser solucionados; Relação usuário monitor ainda com dúvidas, foi despriorizado para solução de outros problemas mais importantes;
- Ações de melhorias: Perfil aberto/cadastro, Relógio timer de presença monitor foi feito mas ainda não está pegando o horário atual no monitor, apenas um de teste, Adicionar eventos importantes no *card* de matérias do aluno ainda não foi feito, Validar hora inicio - hora fim na criação de monitor foi feito mas ainda precisa retornar uma mensagem quando preenchido errado;

#### **4.1.8 Protótipo 10**

No dia 04 de Abril de 2019 o grupo se encontrou para realizar o planejamento do décima *sprint* (7 dias de desenvolvimento – prazo 10 de Abril de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas dicionário de dados, direitos de acesso do usuário, correção relação usuário-monitor com matérias 1:n, adicionar relatório presenças (aluno) e presenças e faltas (monitor) no perfil do usuário, adicionar eventos importantes no *card* de matérias na *deshboard* do aluno. A Tabela 58 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

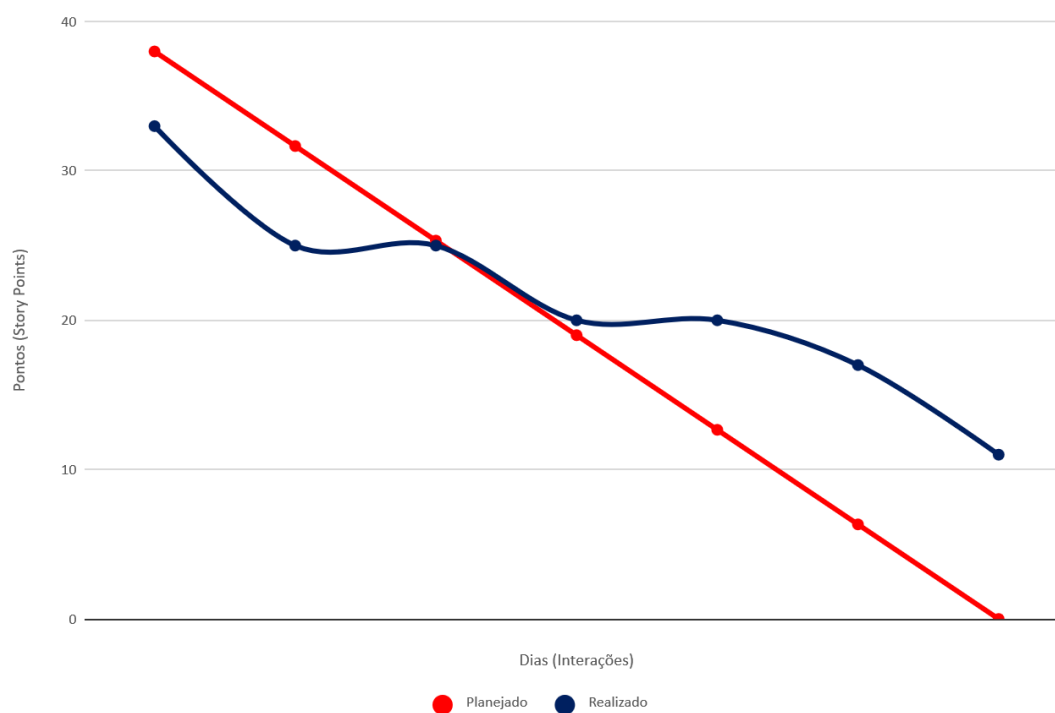
**Tabela 58 – Planejamento realizado para décima entrega.**

| Atividade  | Tempo | Pontos |
|--|-------|--------|
| Dicionário de dados  | 3     | 10     |
| Direitos de acesso do usuário  | 4     | 10     |
| Correção relação usuário-monitor com matérias 1:n  | 1     | 5      |
| Adicionar relatório de presenças (aluno) e presenças e faltas (monitor) no perfil do usuário | 1     | 5      |
| Adicionar eventos importantes no <i>card</i> de matérias na <i>deshboard</i> do aluno        | 1     | 8      |

Fonte: Elaborado pelo autor (2019).

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado. Assim como representado pela figura 28.

**Figura 28 – Gráfico de *Burndown* da entrega 10.**



Fonte: Elaborado pelo autor (2018).

No dia 09 de Abril de 2019 a equipe se reuniu para realizar a revisão do *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos



próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Dicionário de dados, todas as informações e tabelas forma feitas com facilidade;
- O que deu errado: A implementação da relação entre usuário e matéria; Direitos de acesso do usuário levou mais tempo que o esperado.
- Ações de melhorias: Direitos de acesso do usuário, não foi estabelecido ainda as diretrizes de acesso para cada usuário, correção relação usuário-monitor com matérias 1:n, necessita da verificação se está criado corretamente a relação;

#### 4.1.9 Protótipo 11

No dia 11 de Abril de 2019 o grupo se encontrou para realizar o planejamento da décima primeira *sprint* (7 dias de desenvolvimento – prazo 17 de Abril de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas correção relação usuário-monitor com matérias 1:n, direitos de acesso para usuários, descrever recursos e ferramentas, correção no dicionário de dados, adicionar presença de presença(aluno) e presença + falta (monitor), adicionar eventos importantes no *card* de matérias do aluno. A Tabela 59 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

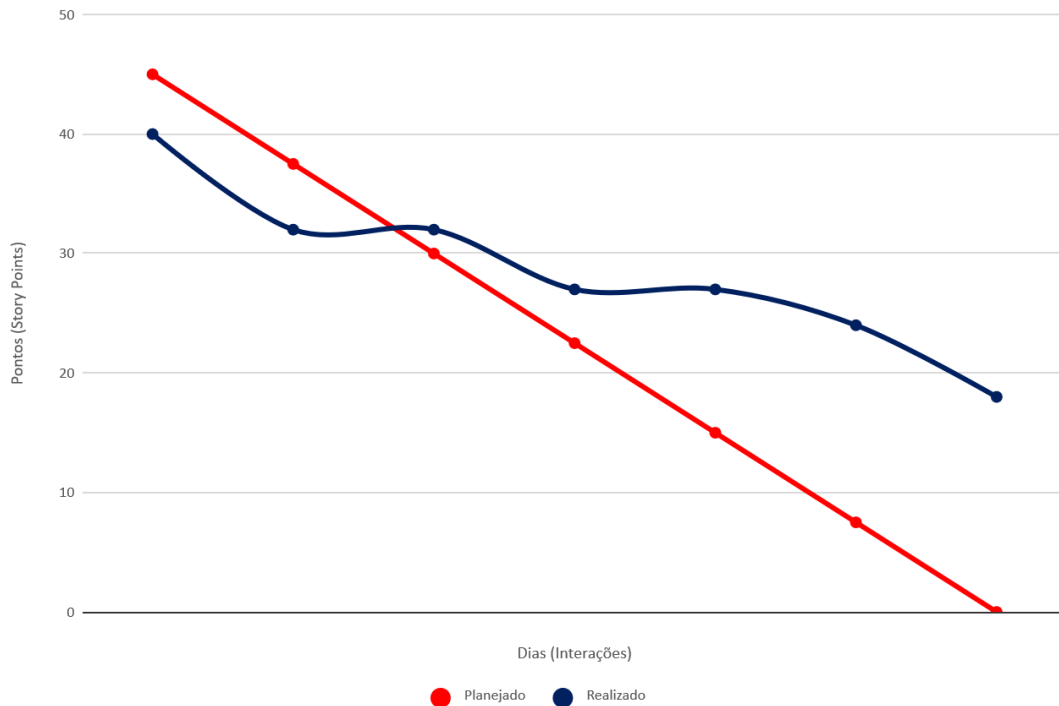
**Tabela 59 – Planejamento realizado para décima primeira entrega.**

| Atividade   | Tempo | Pontos |
|---|-------|--------|
| Correção relação usuário-monitor com matérias 1:n                 | 2     | 5      |
| Direitos de acesso para usuários                                  | 2     | 10     |
| Descrever recursos e ferramentas                                  | 1     | 10     |
| Correção no dicionário de dados                                   | 1     | 5      |
| Adicionar presença de presença(aluno) e presença + falta(monitor) | 2     | 8      |
| Adicionar eventos importantes no card de matérias do aluno        | 1     | 7      |

**Fonte: Elaborado pelo autor (2019).**

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado. Assim como representado na figura 29.

Figura 29 – Gráfico de *Burndown* da entrega 11.



Fonte: Elaborado pelo autor (2018).

No dia 16 de Abril de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Correção no dicionário de dados - Foi incluído uma nova coluna no dicionário de dados para informar os campos que podem ficar nulos quando estiver armazenando o registro. Descrever recursos e ferramentas, todas as demais ferramentas, bibliotecas, programas que foram usados estão agora descritos;

- O que deu errado: Direitos de acesso para usuários: não foi implementado pois ainda não havia sido definido corretamente quais seriam os direitos de acesso para cada tipo de função de usuário. Correção relação usuário monitor com matérias 1:n, como um usuário pode ser matriculado em várias matérias mas também pode lecionar uma matéria terá de ser pensado uma maneira de criar uma relação que não afete a integridade do banco de dados.
- Ações de melhorias: Direitos de acesso para usuários, correção relação usuário-monitor com matérias 1:n;

#### 4.2.0 Protótipo 12

No dia 19 de Abril de 2019 o grupo se encontrou para realizar o planejamento da décima segunda *sprint* (7 dias de desenvolvimento – prazo 25 de Abril de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas correção relação usuário-monitor com matérias 1:n (*user id* no *model* de direitos de acesso para usuários (montar tabela TIPO usuario x função, atribuir o direito de acesso aos usuários de acordo com a tabela.), corrigir tabela de rotas, exemplificar *json* (requisições e respostas), criar filtro para retornar as presenças de um usuário específico, adicionar presença de presença (aluno), presença e falta (monitor), enviar *request*, adicionar eventos importantes no *card* de matérias do aluno. A Tabela 60 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

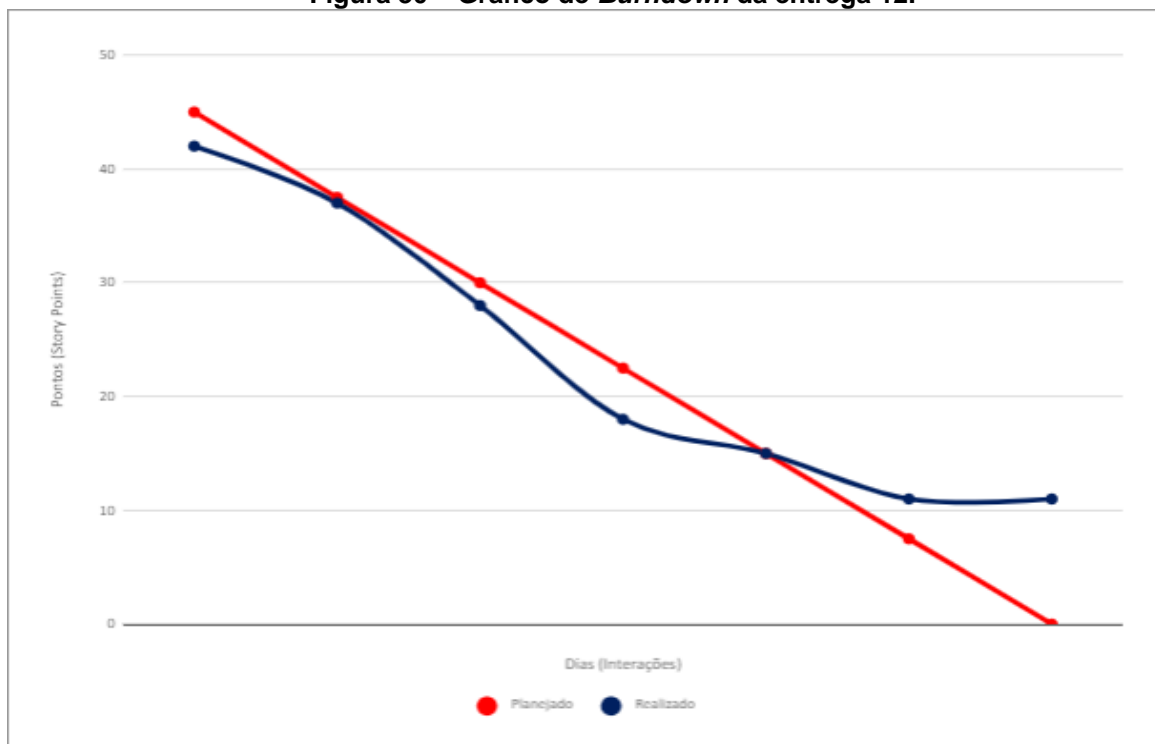
**Tabela 60 – Planejamento realizado para décima segunda entrega.**

| Atividade  | Tempo | Pontos |
|--|-------|--------|
| Correção relação usuário-monitor com matérias 1:n                | 2     | 5      |
| Direitos de acesso para usuários                                 | 2     | 10     |
| Corrigir tabela de rotas   | 1     | 10     |
| Exemplificar json (requisições e respostas)                      | 1     | 5      |
| Criar filtro para retornar as presenças de um usuário específico | 2     | 8      |
| Adicionar presença de presença (aluno)                           | 1     | 7      |
| Presença e falta (monitor)                                       |       |        |
| Enviar request   |       |        |
| Adicionar eventos importantes no card de matérias do aluno       |       |        |

Fonte: Elaborado pelo autor (2019).

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado devido as complicações que surgiram durante o desenvolvimento e a falta de tempo para a solução deles. Assim como representado na figura 30.

Figura 30 – Gráfico de *Burndown* da entrega 12.



Fonte: Elaborado pelo autor

No dia 24 de Abril de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Correção no dicionário de dados - Foi incluído uma nova coluna no dicionário de dados pra informar os campos que podem ficar nulos quando estiver armazenando o registro; Descrever recursos e ferramentas - Todas as demais ferramentas, bibliotecas, programas que foram usados estão agora descritos.

- O que deu errado: Direitos de acesso para usuários - Não foi implementado pois ainda não havia sido definido corretamente quais seriam os direitos de acesso para cada tipo de função de usuário; Correção relação usuário-monitor com matérias 1:n - Como um usuário pode ser matriculado em várias matérias mas também pode lecionar uma matéria terá de ser pensado uma maneira de criar uma relação que não afete a integridade do banco de dados.
- Ações de melhorias: Direitos de acesso para usuário; Correção relação usuário-monitor com matérias 1:n;

#### 4.2.1 Protótipo 13

No dia 02 de Maio de 2019 o grupo se encontrou para realizar o planejamento da décima terceira *sprint* (7 dias de desenvolvimento – prazo 09 de Maio de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas correção relação usuário-monitor com matérias 1:n, atribuir o direito de acesso aos usuários de acordo com a tabela, exemplificar JSON (requisições e respostas), multimídia aula/pdf (grupo de estudo), filtrar aluno por nome para dar presença, criar a tela do perfil do professor, criação do professor (admin), implementar header institution para pesquisar os usuários por nome, adicionar presença de aluno, presença e falta de monitor, interface para cadastrar, editar, excluir eventos e verificar exibição, adicionar eventos importantes no card de matérias do aluno, criação do *faq*. A Tabela 61 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

**Tabela 61 – Planejamento realizado para décima terceira entrega.**

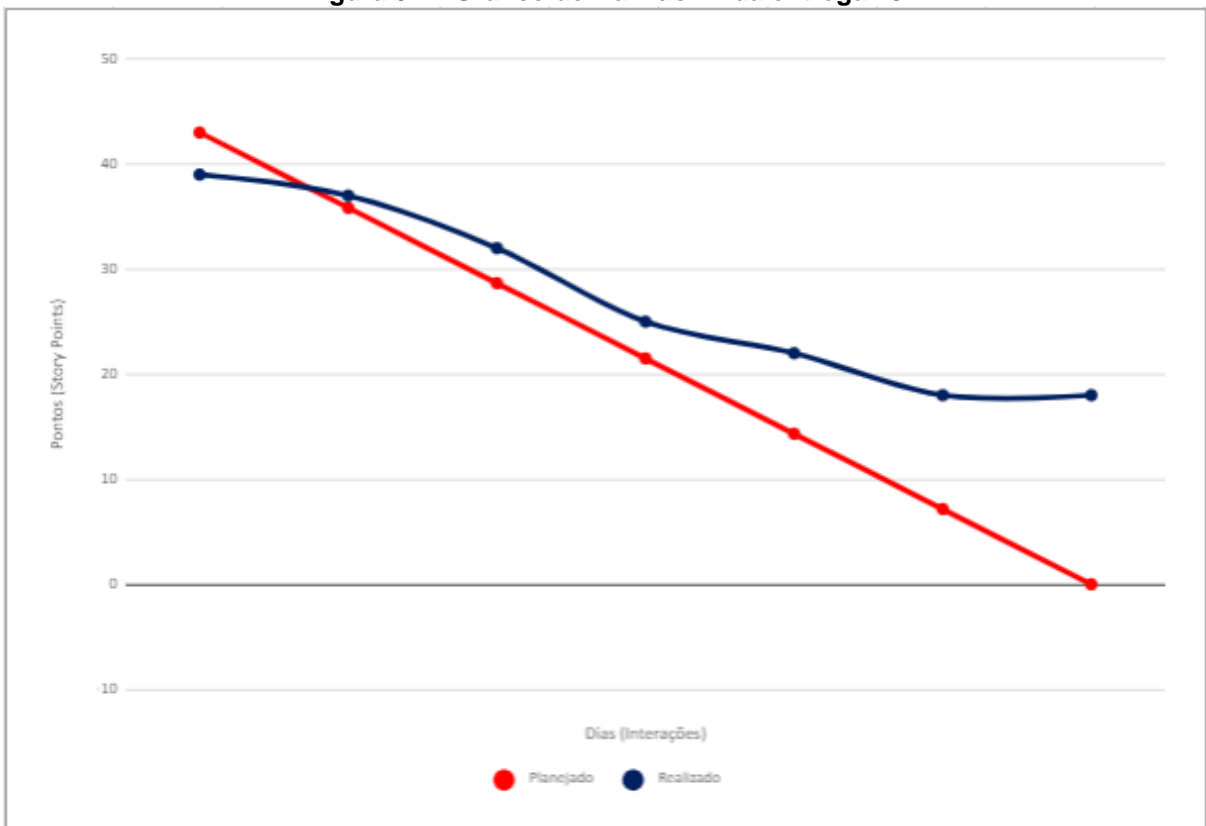
| Atividade  | Tempo | Pontos |
|--|-------|--------|
| Correção relação usuário-monitor com matérias 1:n                | 2     | 10     |
| Atribuir o direito de acesso aos usuários de acordo com a tabela | 1     | 10     |
| Exemplificar json (requisições e respostas)                      | 1     | 5      |
| Multimídia aula/pdf (grupo de estudo)                            | 1     | 8      |
| Filtrar aluno por nome para dar presença                         | 1     | 7      |
| Criar a tela do perfil do professor e criação do professor admin | 1     | 10     |
| Adicionar presença de aluno e falta de monitor                   | 1     | 8      |

|  |   |   |
|--|---|---|
| Interface para cadastrar, editar, excluir eventos e verificar exibição | 2 | 7 |
| Adicionar eventos importantes no card de matérias do aluno             | 1 | 7 |
| Criação do FAQ   | 1 | 6 |

Fonte: Elaborado pelo autor (2019).

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado devido as complicações que surgiram durante o desenvolvimento e a falta de tempo para a solução deles. Assim como representado pela figura 31.

Figura 31 – Gráfico de *Burndown* da entrega 13.



Fonte: Elaborado pelo autor (2018).

No dia 08 de Maio de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Definição dos direitos de acesso; Criar várias presenças de uma vez (back-end); Filtro de presenças para ver as presenças e faltas de um usuário específico;
- O que deu errado: Atribuir direito de acesso de acordo com a tabela;
- Ações de melhorias: Exemplificar JSON - Como a maior parte dessa sprint foi documentar as informações, não tivemos muito acesso a plataforma, então ficou para a próxima sprint adquirir os campos;

#### 4.2.2 Protótipo 14

No dia 09 de Maio de 2019 o grupo se encontrou para realizar o planejamento da décima quarta *sprint* (7 dias de desenvolvimento – prazo 16 de Maio de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas correção relação usuário-monitor com matérias 1:n, multimídia aula e pdf do grupo de estudo, endpoint para *upload* do arquivo, criar nova entidade de relacionamento n:n, criar a tela do perfil do professor, interface para cadastrar, editar, excluir eventos e verificar exibição, adicionar eventos importantes no card de matérias do aluno, criação do *faq*. A Tabela 62 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

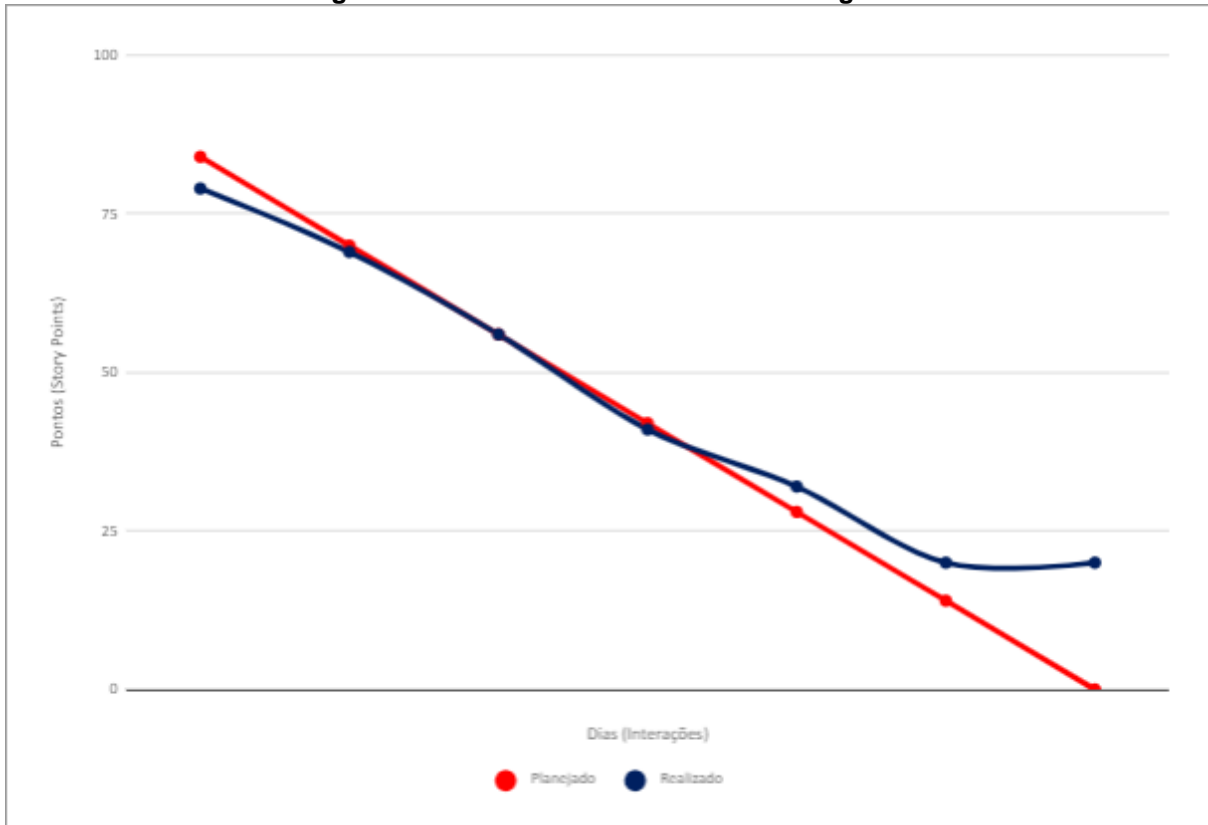
**Tabela 62 – Planejamento realizado para décima quarta entrega.**

| Atividade  | Tempo | Pontos |
|--|-------|--------|
| Correção relação usuário-monitor com matérias 1:n                      | 2     | 10     |
| Multimídia aula/pdf (grupo de estudo)                                  | 1     | 8      |
| Criar nova entidade de relacionamento n:n                              | 1     | 7      |
| Criar a tela do perfil do professor                                    | 1     | 10     |
| Adicionar presença de aluno e falta de monitor                         | 1     | 8      |
| Interface para cadastrar, editar, excluir eventos e verificar exibição | 2     | 7      |
| Adicionar eventos importantes no card de matérias do aluno             | 1     | 7      |
| Criação do FAQ   | 1     | 6      |

**Fonte: Elaborado pelo autor (2019).**

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado devido as complicações que surgiram durante o desenvolvimento e a falta de tempo para a solução deles. Assim como representado pela figura 32.

Figura 32 – Gráfico de *Burndown* da entrega 14.



Fonte: Elaborado pelo autor (2018).

No dia 15 de Maio de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Atribuição do direito de acesso para os usuários de acordo com a tabela; Exemplificar JSON;
- O que deu errado: Upload de imagem/arquivo multimídia para grupo de estudos;



- Ações de melhorias: Correção relação usuário 1:n - Foi deixado para próxima sprint por falta de conhecimento e tempo para implementação; Criação do FAQ;

### 4.2.3 Protótipo 15

No dia 16 de Maio de 2019 o grupo se encontrou para realizar o planejamento da décima quinta *sprint* (7 dias de desenvolvimento – prazo 23 de Maio de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas implementar correção da relação usuário-monitor com matérias 1:n (matéria pertence a usuário), *quiz*, eventuais correções para frontend, enviar *request* de presença, dashboard professor, grupo de estudos, criação do *fag*. A Tabela 63 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

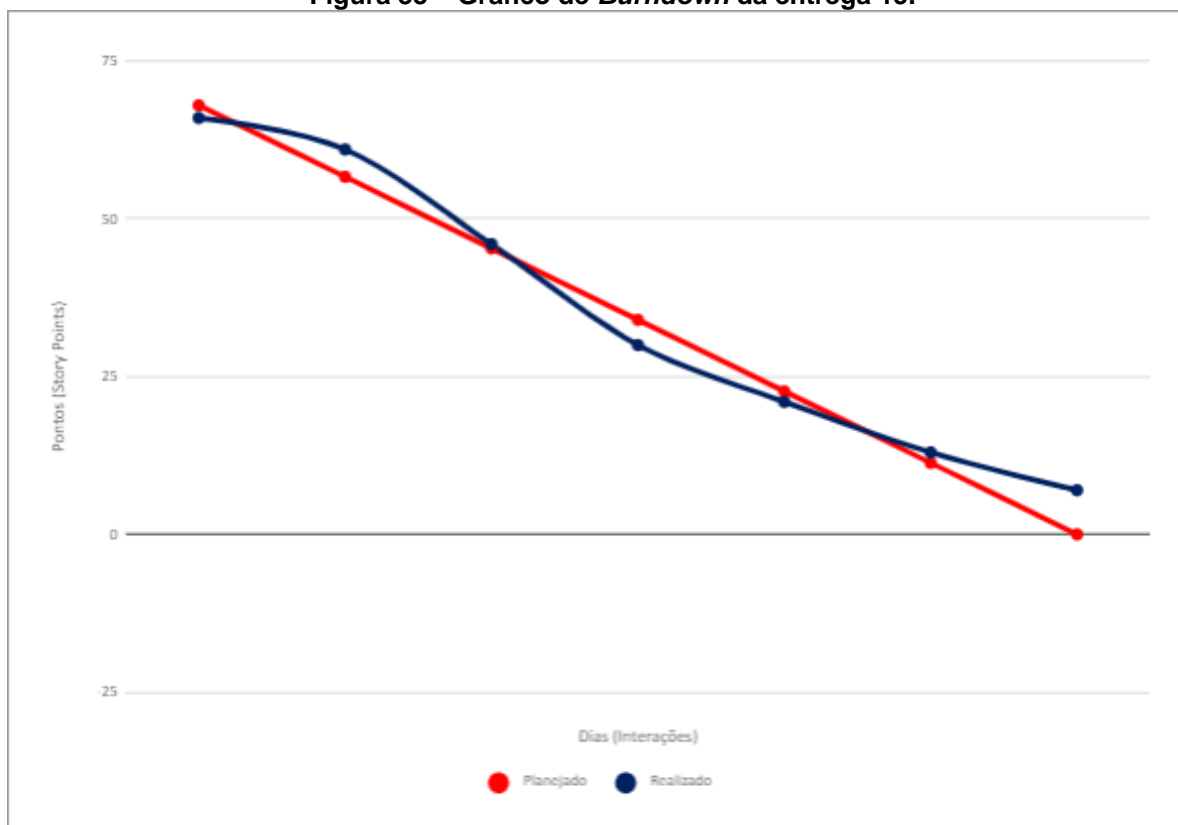
**Tabela 63 – Planejamento realizado para décima quinta entrega.**

| Atividade   | Tempo | Pontos |
|---|-------|--------|
| Correção da relação usuário-monitor com matérias 1:n (matéria pertence a usuário) | 1     | 6      |
| Quiz  | 2     | 10     |
| Eventuais correções para frontend   | 1     | 8      |
| Enviar request de presença  | 1     | 7      |
| Dashboard professor   | 1     | 10     |
| Grupo de estudos  | 1     | 8      |
| Criação do FAQ  | 1     | 6      |

Fonte: Elaborado pelo autor (2019).

Durante os 7 dias de desenvolvimento, a equipe não conseguiu realizar com exatidão tudo que foi planejado devido as complicações que surgiram durante o desenvolvimento e a falta de tempo para a solução deles. Assim como representado pela figura 33.

Figura 33 – Gráfico de *Burndown* da entrega 15.



Fonte: Elaborado pelo autor (2018).

No dia 25 de Maio de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Implementar multimídia e racionamentos, request de presença, eventos importantes;
- O que deu errado: Correção relação usuário 1:n, Criação do FAQ;
- Ações de melhorias: Correção relação usuário 1:n - Foi deixado para próxima sprint por falta de conhecimento e tempo para implementação; Criação do FAQ;

#### 4.2.4 Protótipo 16

No dia 23 de Maio de 2019 o grupo se encontrou para realizar o planejamento da décima sexta *sprint* (7 dias de desenvolvimento – prazo 30 de Maio de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. As atividades planejadas tinham como intuito a resolução do problema da relação de um usuário tipo monitor com matérias 1:n (matéria pertence a usuário), retornar nome do monitor em *worktimes*, direito de acesso para *quiz*, eventuais correções para *frontend*, criação de várias respostas em uma mesma *request*, grupo de estudos, dashboard professor, criação do *faq*. A Tabela 64 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

**Tabela 64 – Planejamento realizado para décima sexta entrega.**

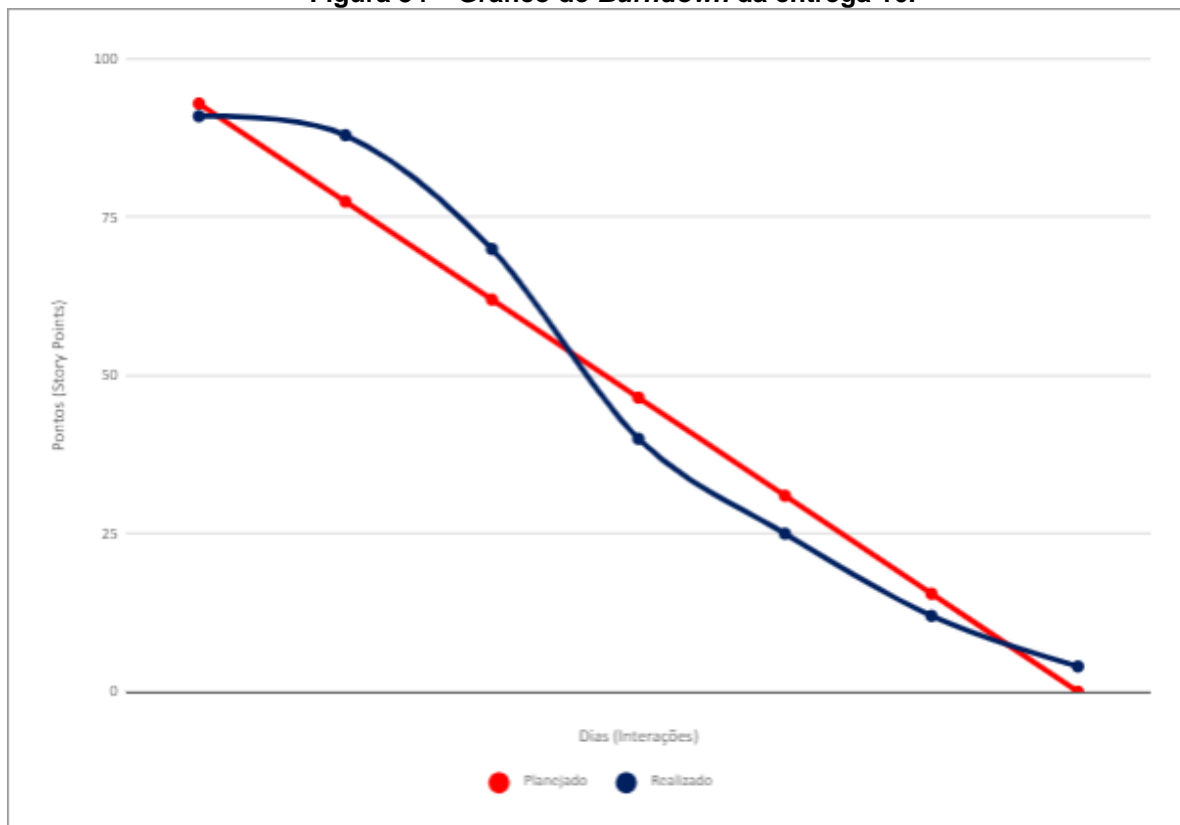
| <b>Atividade</b>  | <b>Tempo</b> | <b>Pontos</b> |
|---|--------------|---------------|
| Correção da relação usuário-monitor com matérias 1:n (matéria pertence a usuário) | 1            | 6             |
| Retornar nome do monitor em <i>worktimes</i>                                      | 2            | 10            |
| Direito de acesso para <i>quiz</i>  | 1            | 8             |
| Criação de várias respostas em uma mesma <i>request</i>                           | 1            | 7             |
| Eventuais correções para <i>frontend</i>  | 1            | 10            |
| Grupo de estudos  | 1            | 8             |
| Quiz  | 2            | 10            |
| Dashboard professor   | 1            | 10            |
| Criação do FAQ  | 1            | 6             |

**Fonte: Elaborado pelo autor (2019).**

Durante os 7 dias de desenvolvimento, de tudo que foi planejado apenas algumas coisas no *frontend* não foram entregues devido a dependência do *backend*

que demorou a entregar as necessidades, assim deixando o tempo muito curto para a implementação no *frontend*. Assim como representado pela figura 34.

Figura 34 – Gráfico de *Burndown* da entrega 16.



Fonte: Elaborado pelo autor (2018).

No dia 29 de Maio de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Quiz, correções para o dashboard professor, tela grupo de estudos;
- O que deu errado: Criação de várias presenças na mesma *request*;

- Ações de melhorias: Mais funções na dashboard de professor e exibição do grupo de estudo;

#### 4.2.5 Protótipo 17

No dia 30 de Maio de 2019 o grupo se encontrou para realizar o planeamento da décima sétima *sprint* (7 dias de desenvolvimento – prazo 6 de Junho de 2019). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas as eventuais correções para *frontend*, retornar *worktimes* com matérias do aluno, conteúdo relacionar com grupo (retorna no grupo), retornar respostas de um quiz específico, grupo de estudos adição de conteúdo, grupo de estudos, *quiz*, criação do *faq*. A Tabela 65 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

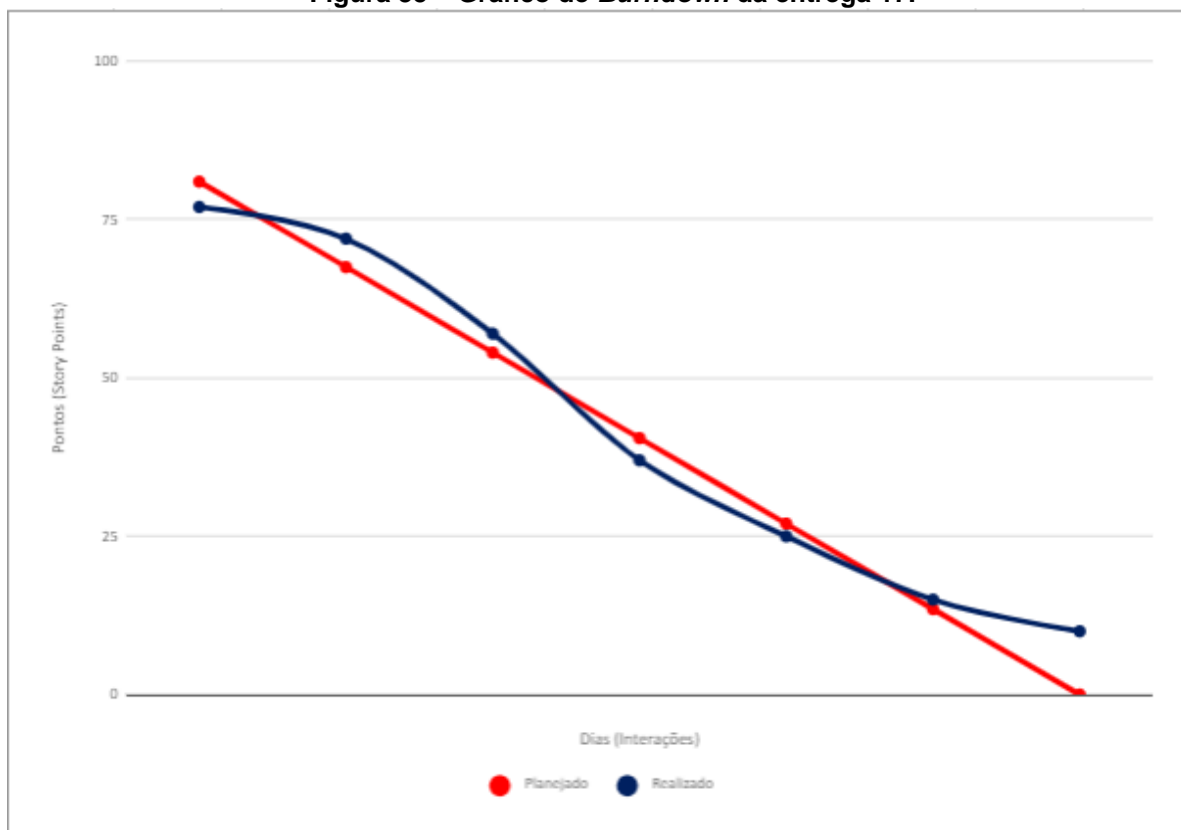
**Tabela 65 – Planeamento realizado para décima sétima entrega.**

| Atividade   | Tempo | Pontos |
|---|-------|--------|
| Eventuais correções para frontend                 | 1     | 10     |
| Retornar worktimes com matérias do aluno          | 2     | 10     |
| Conteúdo relacionar com grupo (retornar no grupo) | 1     | 8      |
| Retornar respostas de um quiz específico          | 1     | 7      |
| Grupo de estudos add conteúdo                     | 1     | 10     |
| Grupo de estudos                                  | 1     | 10     |
| Quiz  | 2     | 10     |
| Criação do FAQ                                    | 1     | 6      |

**Fonte: Elaborado pelo autor (2019).**

Durante os 7 dias de desenvolvimento, tudo ocorreu como o esperado, exceto pela entrega do FAQ, que por conta de ter outras prioridades para o projeto ele foi deixado de lado, no demais, tudo foi entregue. Assim como representado pela figura 35.

Figura 35 – Gráfico de *Burndown* da entrega 17.



Fonte: Elaborado pelo autor (2018).

No dia 5 de Junho de 2019 a equipe se reuniu para realizar a revisão da *sprint*. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma auto avaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: Retornar nome do usuário no *endpoint* de *worktimes*, retorno do conteúdo no *endpoint* de grupo de estudos, filtro de presença, *front end* para quiz;
- O que deu errado: Tudo ocorreu como esperado;
- Ações de melhorias: Dashboard professor, telas do grupo de estudo;

## 4.2 Interfaces de Usuário

Neste ponto serão descritos e exibidos com detalhes todas as telas do sistema Monitorou.

### 4.2.1 Interface Home Page

A Interface *homepage*, apresentada na figura 10, é feita no padrão *one page*, possuindo 4 botões na barra superior da página. O primeiro botão, “Sobre”, ao ser clicado, faz o efeito *scroll*, levando a visualização da página até o ponto onde contém informações sobre o sistema. O segundo botão, “Plataforma”, ao ser clicado, faz o efeito *scroll* levando a visualização da página até o ponto onde contém informações sobre a plataforma. O terceiro botão, “Login”, ao ser clicado, é direcionado para a página onde o usuário acessa as ferramentas da plataforma. O quarto botão, “Cadastrar”, ao ser clicado, é direcionado para a página onde o usuário pode se cadastrar na plataforma, preenchendo um pequeno formulário.

Na figura 36 é apresentado as opções da plataforma e o visual da página.

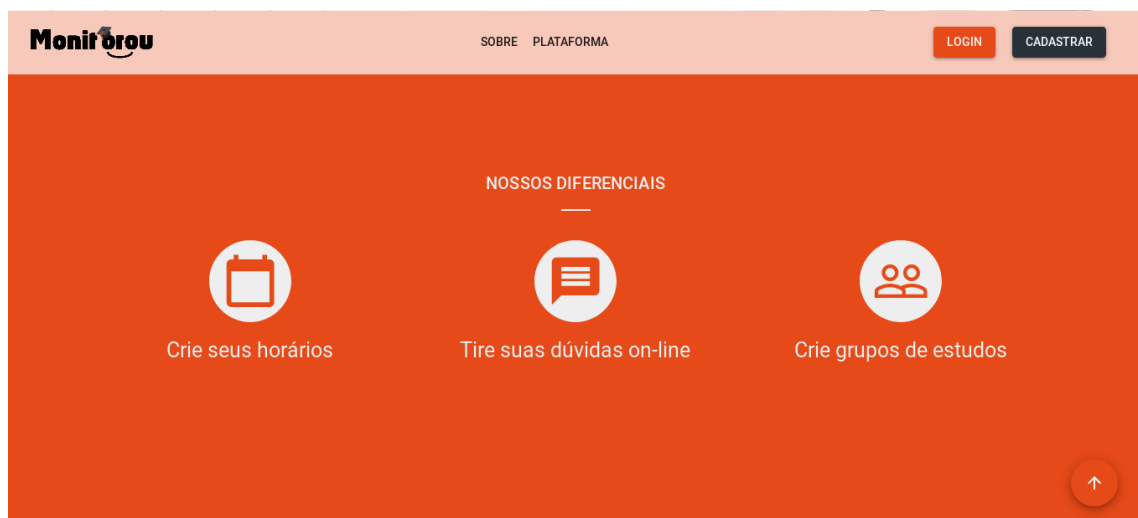
Figura 36 – Interface Home



Fonte: Elaborado pelo autor

Na figura 37 é apresentada a outra parte da página inicial.

Figura 37 – Interface Home 2



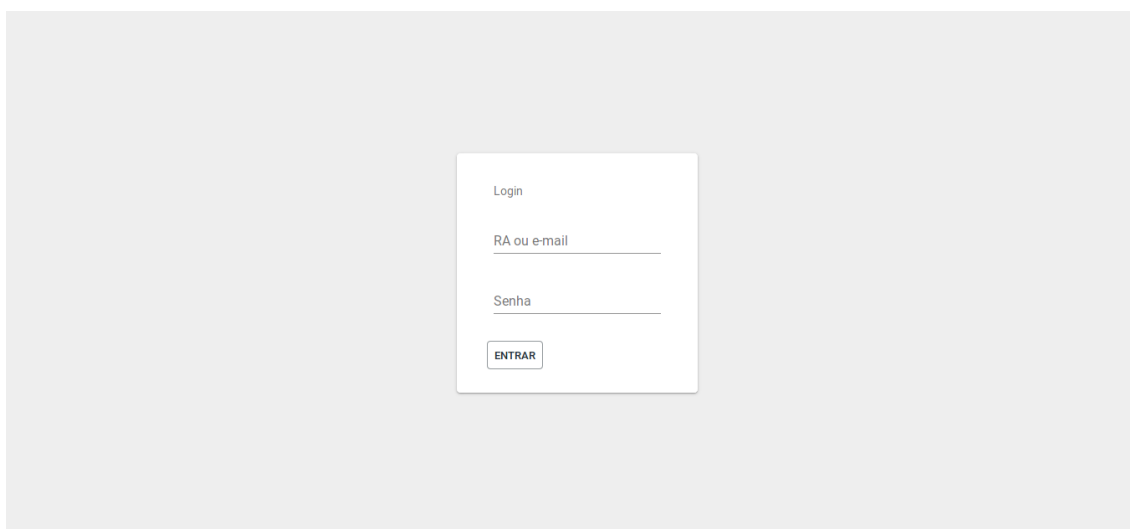
Fonte: Elaborado pelo autor (2018).

#### 4.2.2 Interface Login

A *Interface* login, apresentada na figura 38 é o local onde alunos, monitores e administradores irão ter acesso a plataforma após ter preenchido os campos com RA ou *e-mail* e senha. Logo após inserir os dados clicando no botão “Entrar” é feita uma verificação se os dados estão corretos e caso esteja o usuário é redirecionado ao dashboard com as opções de uso de acordo com cada perfil de usuário, sendo eles administrador, monitor e aluno. Se os dados estiverem errados uma notificação de alerta é exibida.



**Figura 38 – Interface Login**



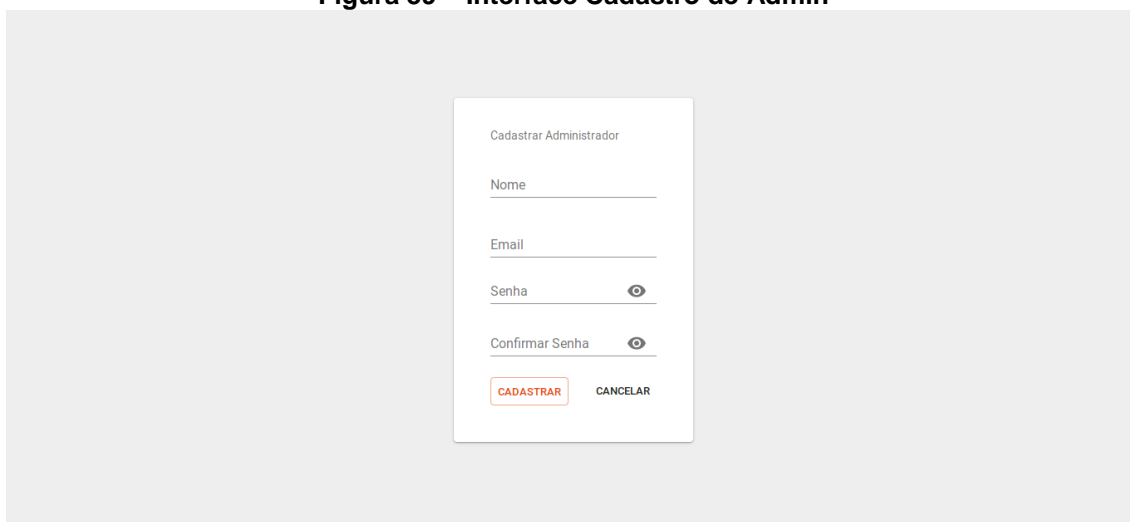
The image shows a login interface centered on a light gray background. The interface is a white rectangular box with the following elements: the word "Login" at the top; a text input field labeled "RA ou e-mail"; another text input field labeled "Senha"; and a button labeled "ENTRAR" at the bottom.

Fonte: Elaborado pelo autor (2018).

#### 4.2.3 Interface de Cadastro de Admin

*Interface* cadastro de admin é apresentada na figura 39, local onde o cadastro de administrador é efetuado, contendo os campos nome, *e-mail*, senha e confirmar senha, com dois botões o primeiro “cadastrar” para confirmar os dados inseridos e o botão “cancelar” para cancelar o cadastro.

**Figura 39 – Interface Cadastro de Admin**



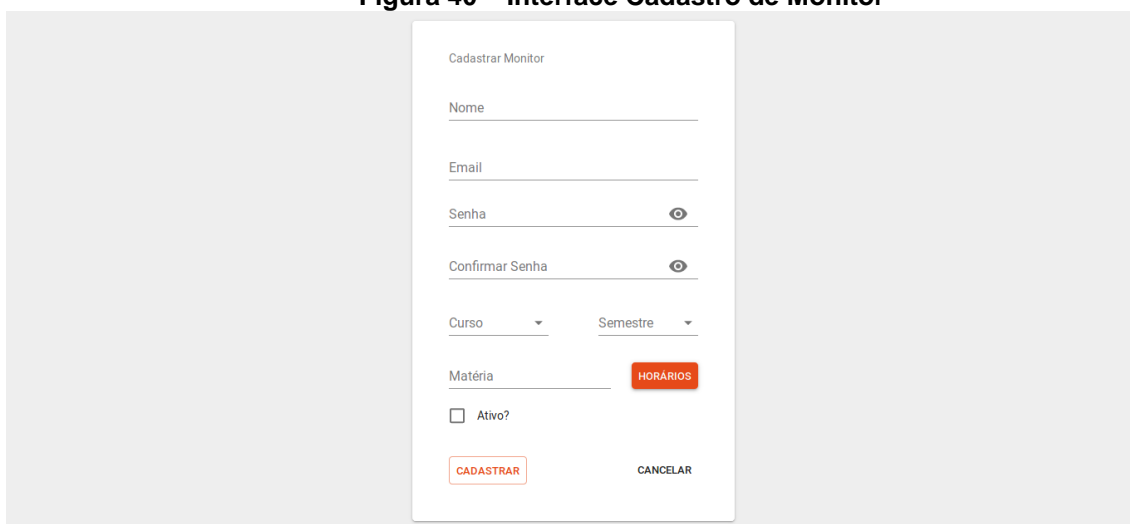
The image shows an admin registration interface centered on a light gray background. The interface is a white rectangular box with the following elements: the title "Cadastrar Administrador" at the top; a text input field labeled "Nome"; a text input field labeled "Email"; a text input field labeled "Senha" with an eye icon to its right; a text input field labeled "Confirmar Senha" with an eye icon to its right; and two buttons at the bottom: "CADASTRAR" (highlighted in orange) and "CANCELAR".

Fonte: Elaborado pelo autor (2018).

#### 4.2.4 Interface de Cadastro de Monitor

Interface cadastro de Monitor é apresentada na figura 40 local onde o cadastro de monitor é efetuado, contendo os campos nome, e-mail, senha e confirmar senha, curso, semestre, matéria e uma *checkbox* para identificar o status do monitor como ativo caso preenchida ou inativo caso vazio, contendo três botões o primeiro horários, ao clicar abre uma janela com horários da semana para confirmar a disponibilidade do monitor em cadastro, segundo “cadastrar” para confirmar os dados inseridos e o terceiro botão “cancelar” para cancelar o cadastro.


**Figura 40 – Interface Cadastro de Monitor**




Cadastrar Monitor

Nome \_\_\_\_\_

Email \_\_\_\_\_

Senha \_\_\_\_\_ 

Confirmar Senha \_\_\_\_\_ 

Curso ▾ Semestre ▾

Matéria \_\_\_\_\_ **HORÁRIOS**

Ativo?

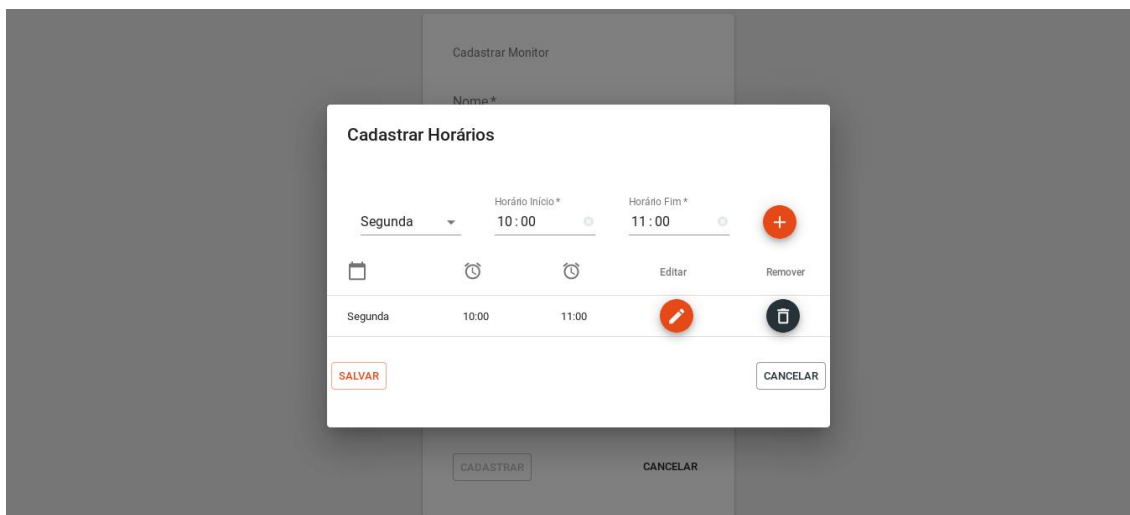
**CADASTRAR** CANCELAR

Fonte: Elaborado pelo autor (2018).

#### 4.2.5 Interface Cadastro de horários de monitor

Interface cadastro de horários de monitor é apresentada na figura 41 demonstra as opções para efetuar cadastros de horários de expediente do monitor, contendo dia da semana, hora de início e fim.

**Figura 41 – Interface Cadastro de Horários de Monitor**

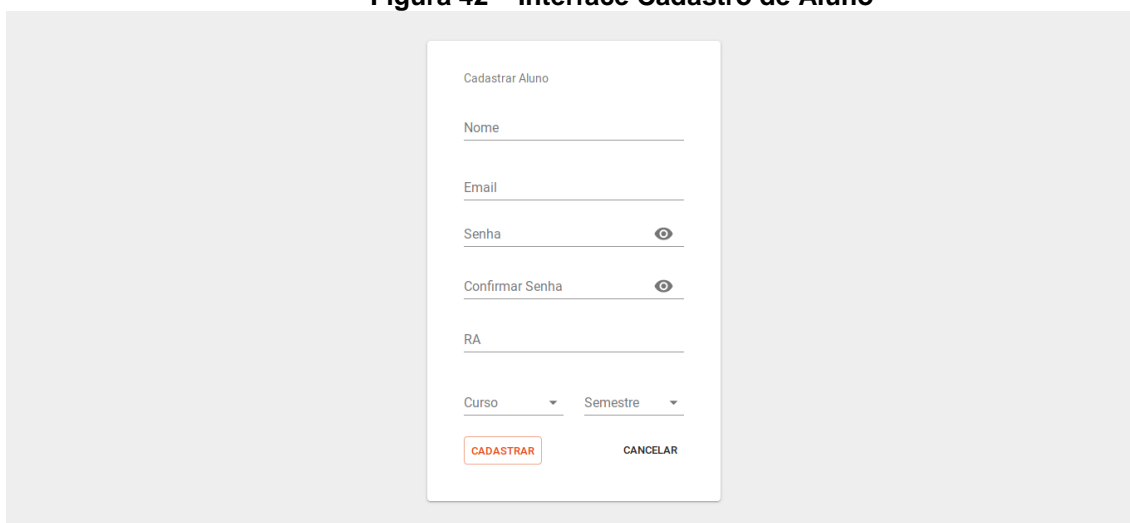


Fonte: Elaborado pelo autor (2018).

#### 4.2.6 Interface Cadastro de Aluno

Interface cadastro de Aluno apresentada na figura 42, local onde o cadastro de aluno é efetuado, contendo os campos nome, e-mail, senha, confirmar senha, curso, semestre, matéria e contendo dois botões, primeiro botão “cadastrar” para confirmar os dados inseridos e o segundo botão “cancelar” para cancelar o cadastro.

Figura 42 – Interface Cadastro de Aluno

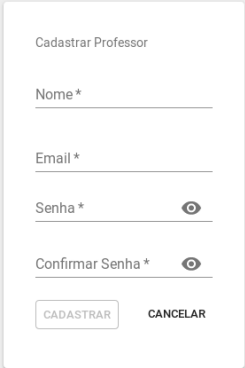


Fonte: Elaborado pelo autor (2018).

#### 4.2.7 Interface Cadastro de Professor

*Interface* cadastro de Aluno apresentada na figura 43, local onde o cadastro de professor é efetuado, contendo os mesmos campos que o cadastro de um monitor só que dessa vez sem a necessidade de informar um horário, primeiro botão “cadastrar” para confirmar os dados inseridos e o segundo botão “cancelar” para cancelar o cadastro.

**Figura 43 – Interface cadastro de professor**



A interface de cadastro de professor é apresentada em uma janela centralizada com o título "Cadastrar Professor". Ela contém os seguintes elementos:

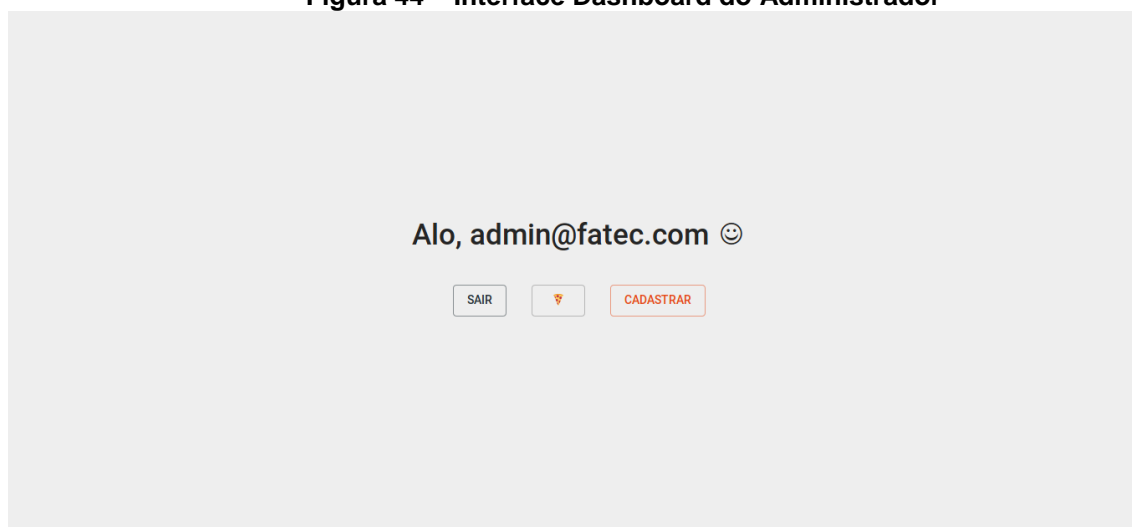
- Um campo de texto rotulado "Nome\*" com um asterisco indicando obrigatoriedade.
- Um campo de texto rotulado "Email\*" com um asterisco indicando obrigatoriedade.
- Um campo de texto rotulado "Senha\*" com um ícone de olho para alternar a visibilidade da senha.
- Um campo de texto rotulado "Confirmar Senha\*" com um ícone de olho para alternar a visibilidade da senha.
- Dois botões de ação: "CADASTRAR" e "CANCELAR".

**Fonte: Elaborado pelo autor**

#### 4.2.7 Interface *Dashboard* do Administrador

*Interface dashboard* do administrador é apresentada na figura 44 o local onde o administrador terá acesso as funções da plataforma, contendo dois botões o primeiro “sair” ao clicar faz o *logout* finalizando o acesso a plataforma e o segundo botão “cadastrar” ao clicar abre uma janela para que o administrador escolha qual tipo de cadastro deseja efetuar.

**Figura 44 – Interface Dashboard do Administrador**

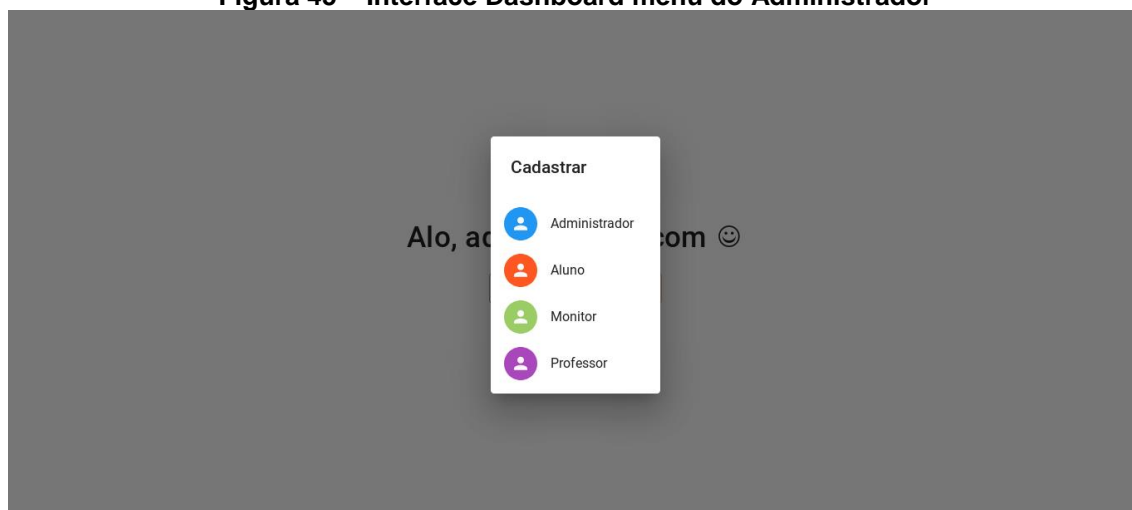


Fonte: Elaborado pelo autor (2018).

#### **4.2.8 Interface *Dashboard* Menu Cadastrar do Administrador**

Interface *dashboard* menu cadastrar do administrador é apresentada na figura 45 demonstra as opções para efetuar cadastros pelo administrador, contendo os perfis de administrador, aluno e monitor. Ao clicar em um perfil em escolha será redirecionado ao formulário de cadastro correspondente ao tipo de perfil.

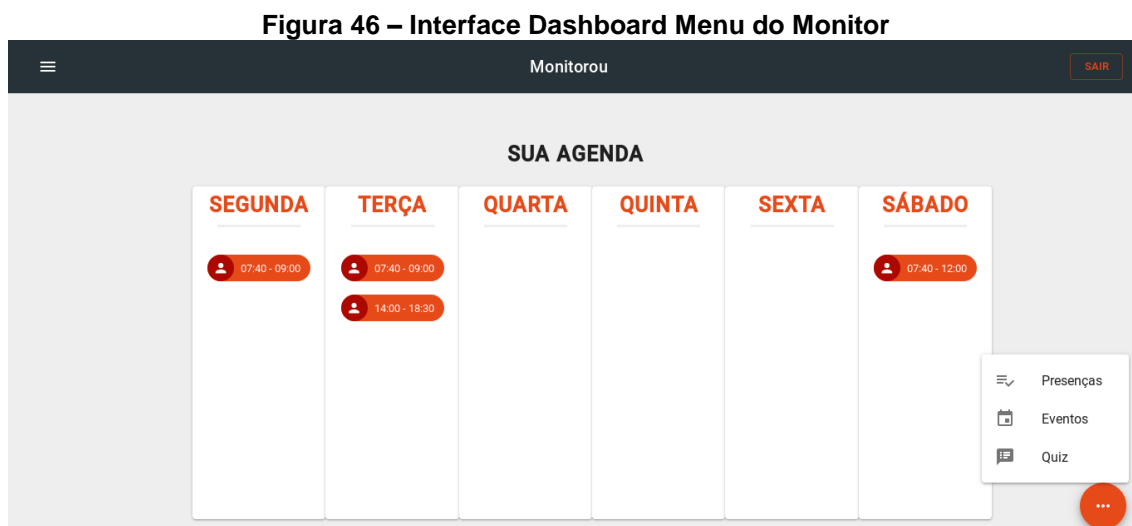
**Figura 45 – Interface Dashboard menu do Administrador**



Fonte: Elaborado pelo autor (2018).

#### 4.2.9 Interface *Dashboard* Menu do Monitor

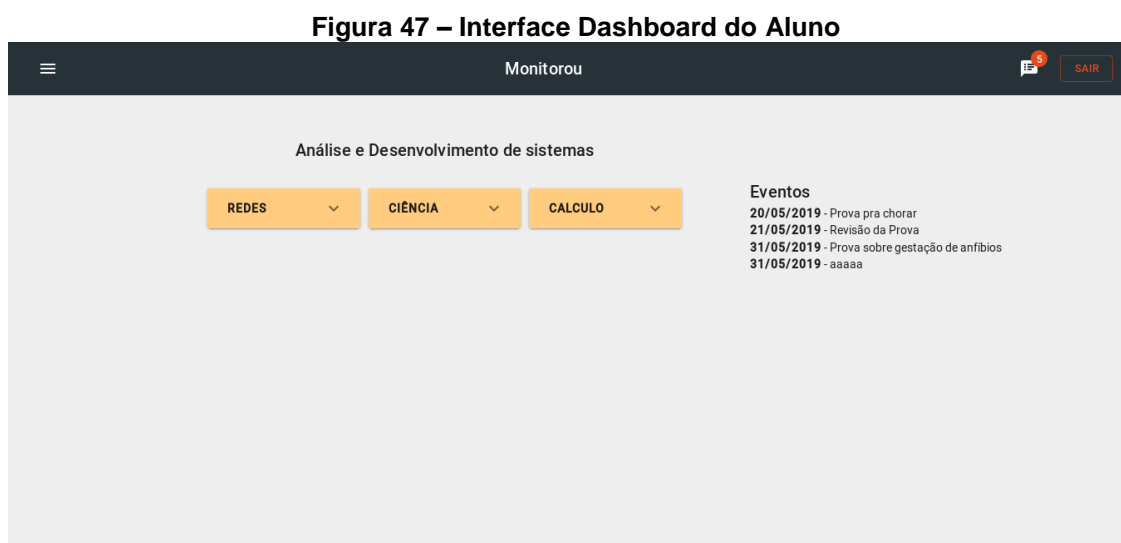
Interface *dashboard* menu do monitor é apresentada na figura 46, demonstra os horários a quais ele está cadastrado, ou seja, os dias que ele deve comparecer a monitoria.



Fonte: Elaborado pelo autor (2018).

#### 4.2.10 Interface *Dashboard* Horários do aluno

Interface dashboard horários do aluno é apresentada na figura 47 demonstra na visão do aluno os horários de expediente do monitor correspondente a cada matéria que o aluno estiver cadastrado.

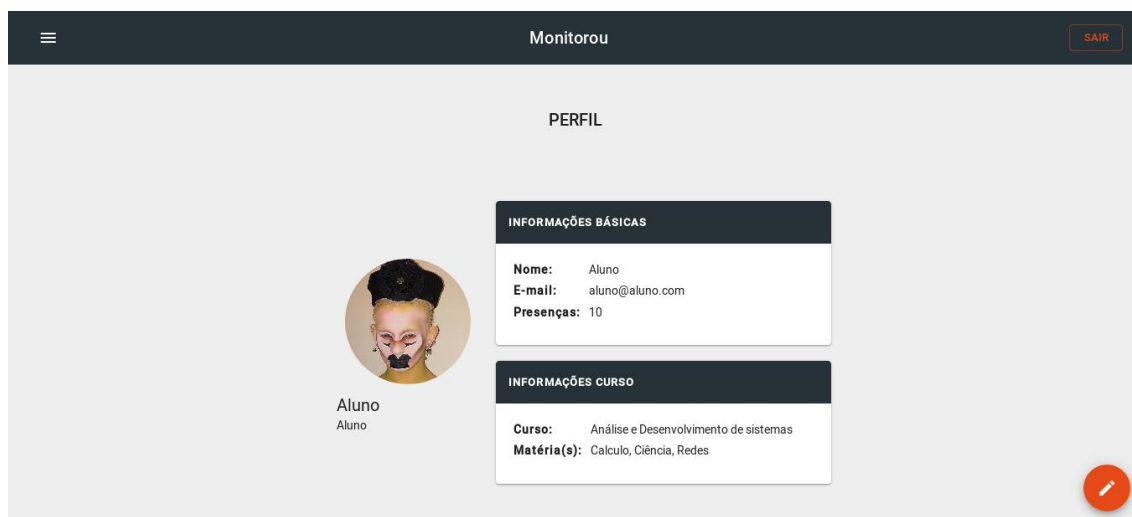


Fonte: Elaborado pelo autor (2018).

#### 4.2.11 Interface Perfil do Aluno

Interface perfil do aluno é apresentada na figura 48 demonstra as informações do aluno cadastrado.

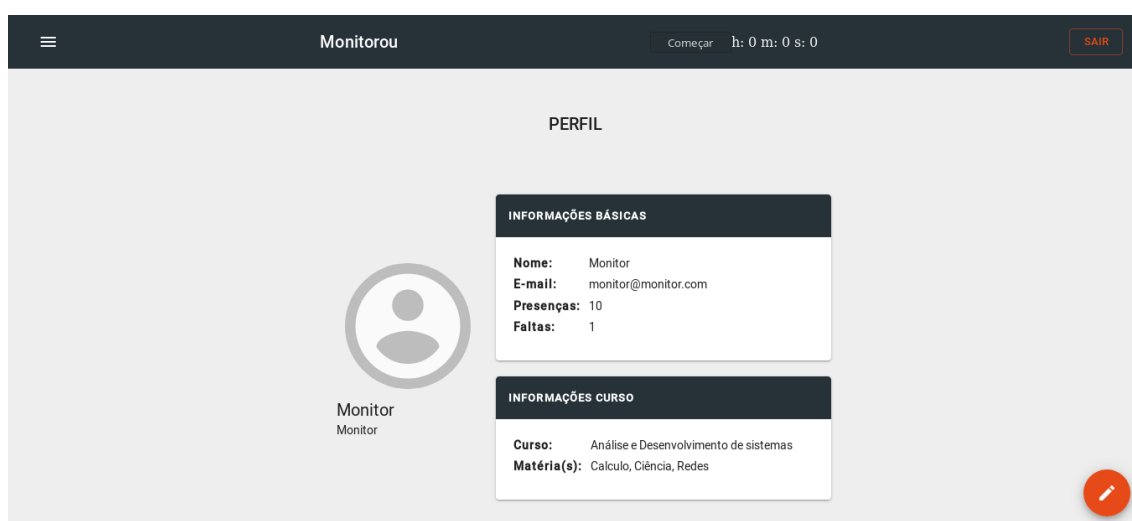
**Figura 48 – Interface Perfil do Aluno**



Fonte: Elaborado pelo autor (2018).

#### 4.2.12 Interface Perfil do Monitor

Interface perfil do aluno é apresentada na figura 49 demonstra as informações do monitor cadastrado.



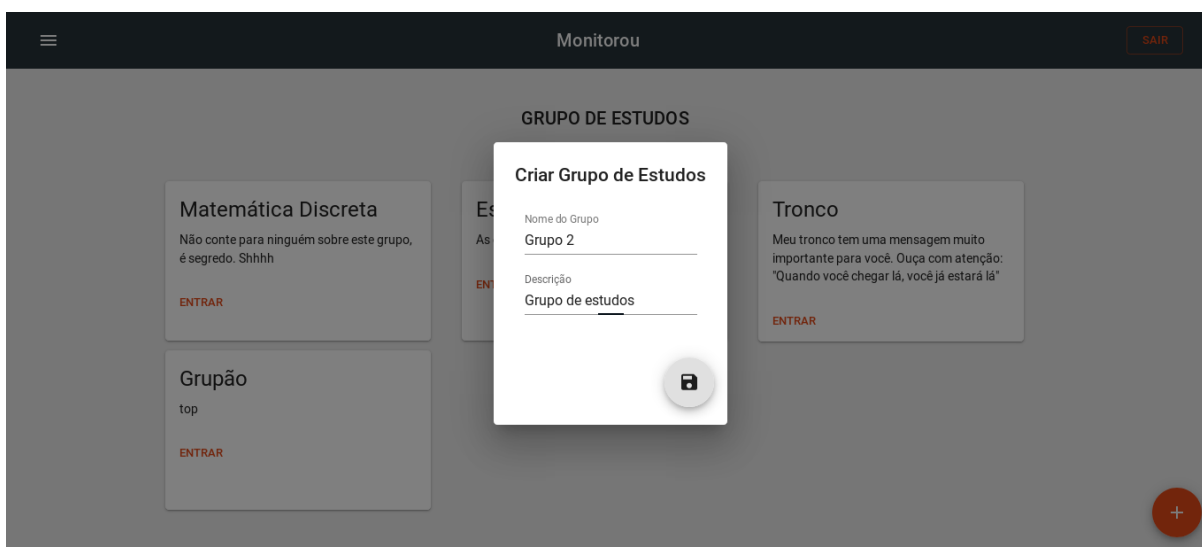
**Figura 49 – Interface Perfil do Monitor**

Fonte: Elaborado pelo autor

### 4.2.13 Interface para grupo de Estudos

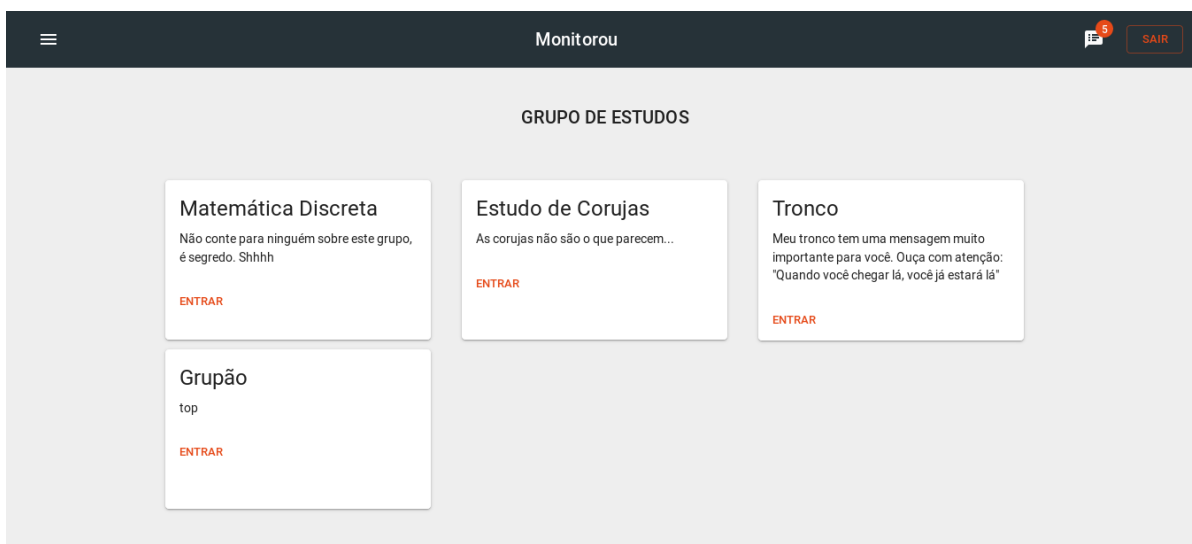
As interfaces para criar e visualizar os grupos de estudo, considerando a visão de um perfil aluno e um perfil monitor, estão representadas nas figuras 50, 51 e 52 respectivamente.

Figura 50 – Interface para criar grupo de estudo



Fonte: Elaborado pelo autor

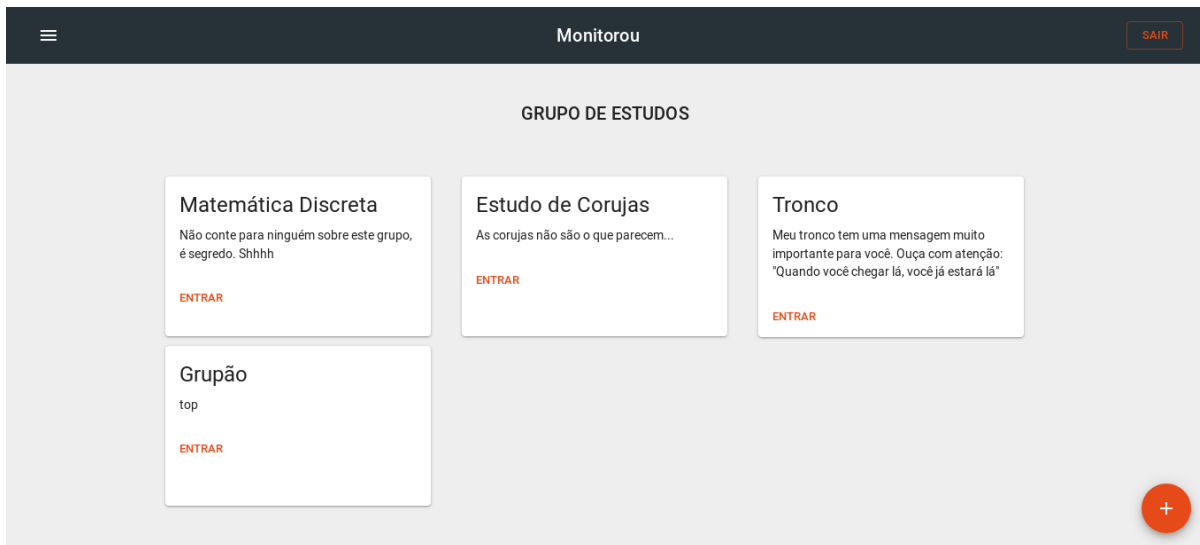
Figura 51 – Interface de visualização dos grupos de estudo (perfil aluno)



Fonte: Elaborado pelo autor

Figura 52 - Interface de visualização dos grupos de estudo (perfil monitor)



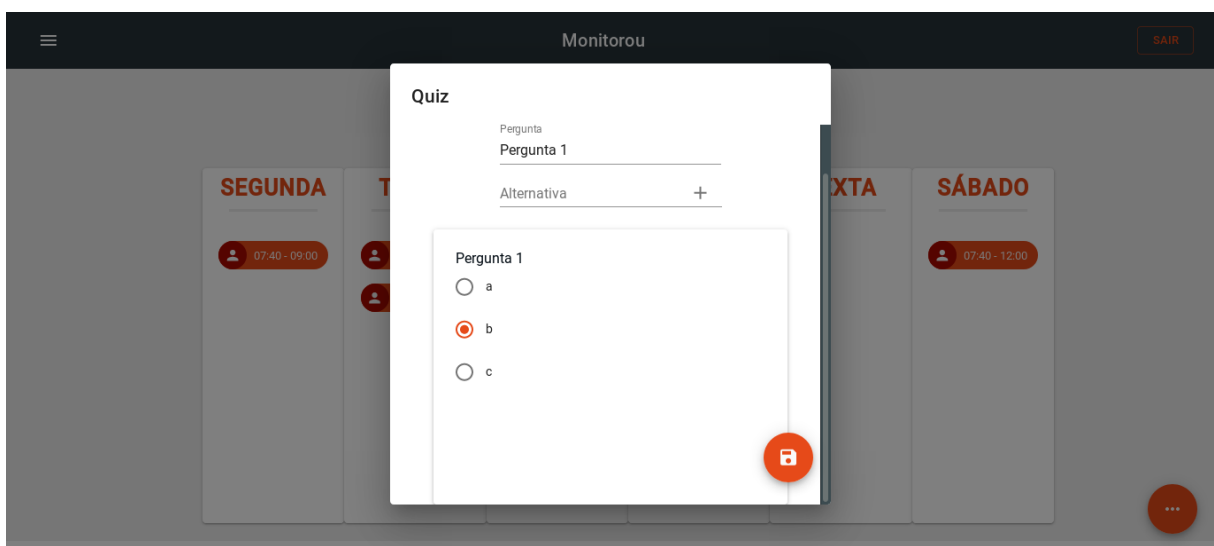


Fonte: Elaborado pelo autor

#### 4.2.14 Interface para Quiz

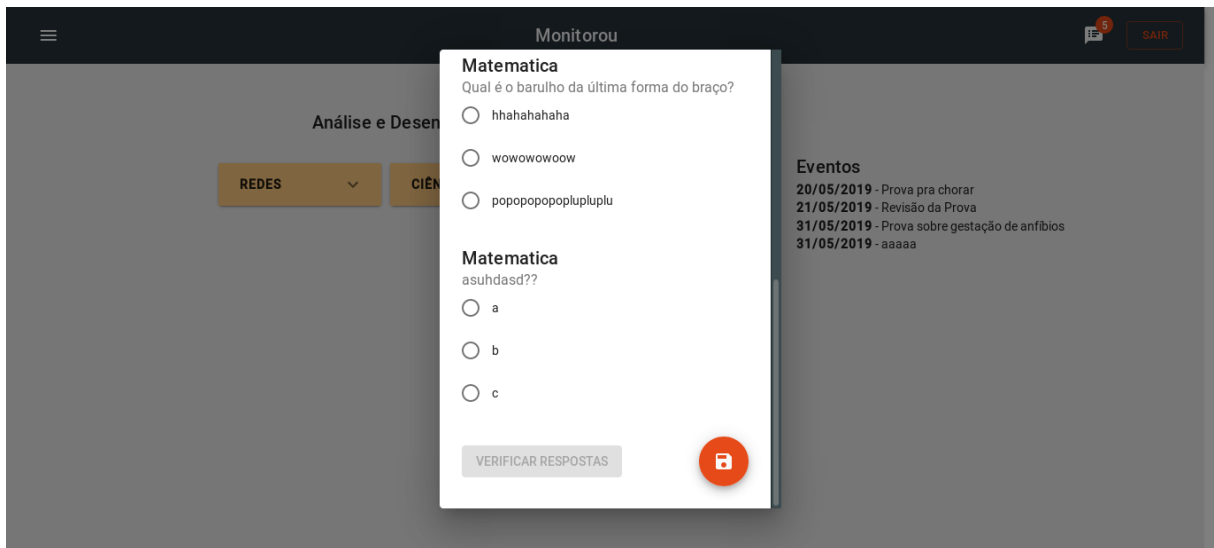
As interfaces que envolvem a criar e visualizar o quiz estão representadas nas figuras 53 e 54 respectivamente.

Figura 53 – Interface para criar um Quiz



Fonte: Elaborado pelo autor

**Figura 54 – Interface visualização e responder Quiz**



**Fonte: Elaborado pelo autor**

## 5 CONSIDERAÇÕES FINAIS

Esse trabalho teve como objetivo o desenvolvimento de um sistema web para que fosse aplicado o que foi aprendido em sala de aula unificando todas as matérias até o momento, tendo como experiência o trabalho em equipe e gerenciamento do tempo, para solucionar a dificuldade encontrada nas pesquisas de campo e garantir um sistema prático e satisfatório que atendesse a necessidade dos universitários em relação a monitoria, bem como o acesso a um conteúdo mais rápido diretamente do professor, possibilitando uma aproximação maior de toda a universidade como um todo.

Durante as fases de desenvolvimento foram encontrados vários problemas devido a não familiaridade com *webservices*, que, mesmo tendo algum conhecimento sobre o assunto, sempre há novos horizontes a serem explorados, com dificuldades que são solucionadas na base de pesquisas e perseverança. As principais dificuldades foram em relação a um controle de sessão para o usuário, algumas definições estruturais que geraram dúvidas que foram sanadas com a orientação do professor e pesquisas em cursos e conteúdos disponíveis na internet. Outras dificuldades foram a elaboração de conteúdo nunca antes feito pela equipe de desenvolvimento, como a o recurso de atribuir presenças para os alunos, demandou uma pesquisa de ambas as partes do sistema (*frontend* e *backend*).

Acredita-se que a plataforma tenha muito potencial para crescimento, tendo em vista que os objetivos foram alcançados com êxito e proporcionaram um grande conhecimento técnico para o desenvolvimento de um sistema. Na plataforma, ainda é focada a inserção de um sistema de *CHAT* entre monitor, professor e aluno, assim garantindo uma maior facilidade de contato entre todos os envolvidos no sistema. Também se planeja a inclusão de um sistema de controle de presença do monitor, para que impossibilite a fraude de presença, garantindo a integridade dos dados fornecidos pelo aluno monitor e um controle mais preciso por parte da coordenação. Considera-se a implementação de um sistema de *upload* de vídeo aula, para que possibilite os professores fazerem *upload* de uma aula ou trecho de aula específico para acesso dos alunos cadastrados na plataforma. Planeja-se também a implementação de um comércio de livros e aulas de conhecimento específico, para

que forneça um acesso mais rápido dos alunos na plataforma onde um *e-commerce* irá facilitar a aquisição de conteúdo para um melhor desempenho acadêmico.

## REFERÊNCIAS

CARRIERWAVE-BASE64. **Carrierwave::Base64**. Disponível em:

<<https://github.com/y9v/carrierwave-base64>>. Acesso em: 24 mar. 2019

ChangeVision Inc. **AstahCommunity**. Disponível em:

<<http://astah.net/editions/community>>. Acesso em 16 de set. 2018.

CODEMY SCHOOL. **Rails API: Token Authentication Part 1 - [006]**. Disponível

em: <<https://www.youtube.com/watch?v=lgdUqtw4weg>>. Acesso em: 6 de set. 2018

Dan Abramov. **REDUX** - Disponível em: <<https://www.devmedia.com.br/react-redux/>>.

Acesso em: 15 jan. 2019

DbDesigner.net. **DbDesinger**. Disponível em: <<https://www.dbdesigner.net/>>.

Acesso em 12 de mar. 2019.

EICH, Brendan. **JavaScript**. Disponível em: <<https://www.javascript.com/>>. Acesso

em 14 de mar. 2019.

HANSSON, David Heinemeier. **Ruby on Rails**. Disponível em:

<<https://rubyonrails.org/>>. Acesso em 20 de ago.2018.

JGraph Ltd. **Draw.io**. Disponível em: <<https://www.draw.io/>>. Acesso em 13 de mar.

2019.

JSX. **JSX - a faster, safer, easier JavaScript**. Disponível em: <<https://jsx.github.io/>>

Acesso em: 13 set. 2018

MATERIAL UI. **MATERIAL UI**. Disponível em: <<https://material-ui.com/>>. Acesso em:

10 de dez. 2018

MATSUMOTO, Yukihiro. **Ruby**. Disponível em: <<https://www.ruby-lang.org/pt/>>.

Acesso em 16 de ago. 2018.

MDS WEB DOCS. **Fetch API - Web APIs**. Disponível em:

<[https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API)>. Acesso em: 28 abr.

2019.

Microsoft. **VsCode**. Disponível em: < <https://code.visualstudio.com/> >. Acesso em 15 de ago. 2018.

MOODLE. Disponível em:<[https://docs.moodle.org/36/en/Main\\_page](https://docs.moodle.org/36/en/Main_page)>. Acesso em: 1 de dez. 2018

NICKLAS, Jonas. **Carrierwave**. Disponível em: <<https://github.com/carrierwaveuploader/carrierwave>>. Acesso em: 19 mar. 2019

PATEL, Gopal. **pry-byebug**. Disponível em: <<https://github.com/deivid-rodriguez/pry-byebug>>. Acesso em: 21 nov. 2018

PLATAFORMATEC. **Devise**. Disponível em: <<https://github.com/plataformatec/devise>> Acesso em: 22 out. 2018

Plataformatec. **HasScope**. Disponível em: <[https://github.com/plataformatec/has\\_scope](https://github.com/plataformatec/has_scope)>. Acesso em: 19 nov. 2018.

Postdot Technologies Pvt. Ltd. **Postman**. Disponível em: < <https://www.getpostman.com/>>. Acesso em 20 de ago. 2018.

PostgreSQL Global Development Group. **PostgreSQL**. Disponível em: < <https://www.postgresql.org/> >. Acesso em 24 de nov. 2018.

RACK-CORS. **Rack CORS Middleware**. Disponível em: <<https://github.com/cyu/rack-cors>>. Acesso em: 29 out. 2019

RAILS GUIDES. **Using Rails for API-only Applications**. Disponível em: <[https://guides.rubyonrails.org/api\\_app.html](https://guides.rubyonrails.org/api_app.html)>. Acesso em: 10 de dez. 2018

REACT TRAINING. **React Router**. Disponível em: <<https://reacttraining.com/react-router/>>. Acesso em: 28 de abr. 2019

REACT. **REACT**. Disponível em: <<https://reactjs.org/>>. Acesso em: 10 de dez. 2018

REDUX. **Async Actions**. Disponível em: <<https://redux.js.org/advanced/async-actions>>. Acesso em: 27 de nov. 2018.

REDUX. **Middleware**. Disponível em: < <https://redux.js.org/advanced/middleware>>. Acesso em: 27 de nov. 2018.

REDUXJS. **Redux Thunk**. Disponível em: <<https://github.com/reduxjs/redux-thunk>> Acesso em: 4 mar. 2019

Salesforce.com. **Heroku**. Disponível em: < <https://www.heroku.com/> >. Acesso em 16 ago. 2018.

SIMPLE-TOKEN-AUTHENTICATION. **Simple Token Authentication**. Disponível em: <[https://github.com/gonzalo-bulnes/simple\\_token\\_authentication](https://github.com/gonzalo-bulnes/simple_token_authentication)>. Acesso em: 22 out. 2018.

SOMMERVILLE, Ian. **Engenharia de Software**: 6 ed. São Paulo: Pearson, 2003.

STACKOVERFLOW. Disponível em: <<https://stackoverflow.com>>. Acesso em: 10 de dez. 2018

UDEMY. **UDEMY**. Disponível em: <<https://www.udemy.com/>>. Acesso em: 22 de ago. 2018

VIM THE EDITOR. **About VIM**. Disponível em: <<https://www.vim.org/>>. Acesso em 13 de jan. 2019

World Wide Web Consortium. **CSS**. Disponível em: < <https://www.w3schools.com/css/> >. Acesso em 28 de out. 2018.