



---

**FACULDADE DE TECNOLOGIA DE AMERICANA - MINISTRO RALPH BIASI**  
**Curso de Tecnologia em Análise e Desenvolvimento de Sistemas**

Bruno Cardoso Ambrosio  
Fernando Daniel Muniz  
Victoria Andressa Santos Macedo de Faria

**PROJETO E DESENVOLVIMENTO DE UM SISTEMA  
COLABORATIVO PARA GERENCIAMENTO DE ATIVIDADES E  
PROJETOS INTITULADO CRONOTE**

**Americana, SP  
2019**



---

**FACULDADE DE TECNOLOGIA DE AMERICANA - MINISTRO RALPH BIASI**  
**Curso de Tecnologia em Análise e Desenvolvimento de Sistemas**

Bruno Cardoso Ambrosio

Fernando Daniel Muniz

Victoria Andressa Santos Macedo de Faria

**PROJETO E DESENVOLVIMENTO DE UM SISTEMA  
COLABORATIVO PARA GERENCIAMENTO DE ATIVIDADES E  
PROJETOS INTITULADO CRONOTE**

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

Área de concentração: Desenvolvimento de Sistema

**Americana, SP**  
**2019**

**FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS  
Dados Internacionais de Catalogação-na-fonte**

A531p      AMBROSIO, Bruno Cardoso

Projeto e desenvolvimento de um sistema colaborativo para gerenciamento de atividades e projetos intitulado: Cronote. / Bruno Cardoso Ambrosio, Fernando Daniel Muniz, Victoria Andressa Santos Macedo de Faria. – Americana, 2019.

93f.

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

Orientador: Prof. Dr. ANDRADE, Kleber de Oliveira Andrade

1 Desenvolvimento de software. I. MUNIZ, Fernando Daniel II. FARIA, Victoria Andressa Santos Macedo de III. ANDRADE, Kleber de Oliveira IV. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.3.05

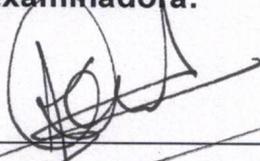
Bruno Cardoso Ambrosio  
Fernando Daniel Muniz  
Victoria Andressa Santos Macedo de Faria

**“PROJETO E DESENVOLVIMENTO DE UM SISTEMA  
COLABORATIVO PARA GERENCIAMENTO DE ATIVIDADES E  
PROJETOS INTITULADO CRONOTE”**

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.  
Área de concentração: Desenvolvimento de Sistema

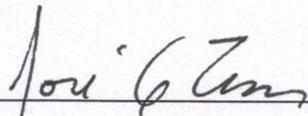
Americana, 09 de dezembro de 2019.

**Banca Examinadora:**



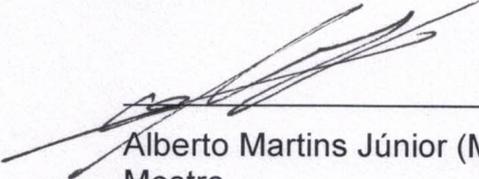
---

Kléber de Oliveira Andrade (Presidente)  
Doutor  
Fatec Americana



---

José Luiz Zem (Membro)  
Doutor  
Fatec Americana



---

Alberto Martins Júnior (Membro)  
Mestre  
Fatec Americana

## **AGRADECIMENTOS**

Em primeiro lugar, agradecemos a Deus, pois sem Ele nós não teríamos forças para a realização deste trabalho.

Ao nosso orientador, Prof. Dr. Kleber de Oliveira Andrade por ter acreditado no projeto, aceitando ser nosso orientador e por ter nos apoiado. A sua participação foi fundamental para nossa formação.

Agradecemos também ao programa do instituto eldorado em conjunto com a Apple, o Apple Developer Academy por abrir nossas visões de negócio, isso com certeza garantiu uma qualidade muito melhor de trabalho.

Agradecemos aos nossos pais e amigos por nos acompanharem e nos incentivarem a nunca desistir.

A todos que nos ajudaram direta ou indiretamente com o projeto.

## EPÍGRAFE

*A menos que modifiquemos a nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo. Albert Einstein (1879 – 1955).*

## RESUMO

Segundo a definição do dicionário, gerenciar consiste em organizar, planejar e executar atividades que auxiliem em determinado processo de trabalho. É comum que as pessoas não dispõem do hábito de planejar, por não existir recursos de fácil acesso e com uma interação intuitiva para o usuário, portanto gastando mais tempo que o necessário para concluir suas atividades. Por este fato, o trabalho apresentado expõe o desenvolvimento de um aplicativo multiplataforma colaborativo, Cronote, um sistema de gerenciamento de atividades e projetos que elabora um balanço e comparações das atividades cadastradas e/ou projetos predeterminados gerando relatórios finais mensais, para consultas futuras e uma possível melhor tomada de decisão. O trabalho apresenta, primeiramente, um estudo sobre os sistemas de gerenciamento de projetos/atividades e, em seguida, o processo de desenvolvimento do sistema utilizando variados recursos como, banco de dados PostgreSQL, tecnologia RESTful, linguagem Java, framework Scrum, padrão de desenvolvimento MVC, design UI/UX, entre outros.

**Palavras-chave:** Planejamento; Gerenciamento de Projetos; Desenvolvimento de Sistemas.

## ABSTRACT

According to the dictionary definition, managing consists in organizing, planning and performing activities that assist in the given work process. People often do not demonstrate the habit of planning because there are no easily accessible features and intuitive user interaction, thus spending more time than necessary to perform their activities. For this reason, the work presented exposes or develops a collaborative multiplatform application, Cronote, a project and activity management system that elaborates a balance and comparisons of the recorded activities that can be executed. The paper presents, reproduced, a study of project / activity management systems and then the system development process using various features such as PostgreSQL database, RESTAPI technology, integrated MVC, UI / UX design, among others.

**Keywords:** Planning. Project Managing. Software Development.

## LISTA DE FIGURAS

Figura 1 – Estrutura ágil: Scrum em resumo . . . . .	19
Figura 2 – Diagrama de caso de uso do login do usuário . . . . .	25
Figura 3 – Diagrama de caso de uso das ações básicas de um usuário. . . . .	26
Figura 4 – Diagrama de caso de uso das ações do líder e dos membros de um time. . . . .	27
Figura 5 – Diagrama de classe de login . . . . .	36
Figura 6 – Diagrama de classe de atividade . . . . .	37
Figura 7 – Diagrama de classe de projeto . . . . .	38
Figura 8 – Diagrama de classe de time . . . . .	39
Figura 9 – Diagrama de sequência de login . . . . .	40
Figura 10 – Diagrama de sequência de cadastro de usuário . . . . .	41
Figura 11 – Diagrama de sequência de recuperação de senha . . . . .	42
Figura 12 – Diagrama de sequência de cadastrar atividade . . . . .	43
Figura 13 – Diagrama de sequência de criar time . . . . .	44
Figura 14 – Diagrama de sequência de criar projeto . . . . .	45
Figura 15 – Diagrama de entidade relacionamento . . . . .	46
Figura 16 – Gráfico de Burndown . . . . .	59
Figura 17 – Gráfico de Burndown - período do dia 23 de junho de 2018 a 7 de julho de 2018 . . . . .	59
Figura 18 – Gráfico de Burndown - período do dia 08 de julho de 2018 a 21 de julho de 2018 . . . . .	60
Figura 19 – Gráfico de Burndown - período do dia 18 de agosto de 2018 a 01 de setembro de 2018 . . . . .	60
Figura 20 – Gráfico de Burndown - período do dia 22 de setembro de 2018 a 06 de outubro de 2018 . . . . .	61
Figura 21 – Gráfico de Burndown - período do dia 25 de fevereiro de 2019 a 17 de março de 2019 . . . . .	61
Figura 22 – Gráfico de Burndown - período do dia 18 de março de 2019 a 07 de abril de 2019 . . . . .	62
Figura 23 – Gráfico de Burndown - período do dia 08 de abril de 2019 a 28 de abril de 2019 . . . . .	62
Figura 24 – Gráfico de Burndown - período do dia 29 de abril de 2019 a 19 de maio de 2019 . . . . .	63
Figura 25 – Gráfico de Burndown - período do dia 27 de maio de 2019 a 16 de junho de 2019 . . . . .	63
Figura 26 – Gráfico de Burndown - período do dia 22 de junho de 2019 a 05 de julho de 2019 . . . . .	64

Figura 27 – Gráfico de Burndown - período do dia 06 de julho de 2019 a 19 de julho de 2019 . . . . .	64
Figura 28 – Gráfico de Burndown - período do dia 20 de julho de 2019 a 02 de agosto de 2019 . . . . .	65
Figura 29 – Gráfico de Burndown - período do dia 03 de agosto de 2019 a 16 de agosto de 2019 . . . . .	65
Figura 30 – Gráfico de Burndown - período do dia 17 de agosto de 2019 a 30 de agosto de 2019 . . . . .	66
Figura 31 – Gráfico de Burndown - dia 31 de agosto de 2019 a 13 de setembro de 2019	66
Figura 32 – Gráfico de Burndown - período do dia 14 de setembro de 2019 a 27 de setembro de 2019 . . . . .	67
Figura 33 – Gráfico de Burndown - período do dia 30 de setembro de 2019 a 13 de outubro de 2019 . . . . .	67
Figura 34 – Gráfico de Burndown - período do dia 18 de outubro de 2019 a 07 de novembro de 2019 . . . . .	68
Figura 35 – logotipo . . . . .	69
Figura 36 – Cores do logotipo . . . . .	69
Figura 37 – Tipografia do Cronote . . . . .	70
Figura 38 – Peleta de cores usadas no sistema . . . . .	71
Figura 39 – Tela Login . . . . .	73
Figura 40 – Tela de cadastro do usuário . . . . .	74
Figura 41 – Tela de recuperação de senha - Etapa 1 . . . . .	75
Figura 42 – Exemplo de email recebido com código de verificação para a recuperação de senha. . . . .	75
Figura 43 – Tela de recuperação de senha - Etapa 2 . . . . .	76
Figura 44 – Tela de atividades. . . . .	77
Figura 45 – Tela de cadastro de atividades. . . . .	78
Figura 46 – Tela de cadastro de categorias. . . . .	78
Figura 47 – Tela de edição de categorias. . . . .	79
Figura 48 – Tela de cadastro de atividade: descrição . . . . .	80
Figura 49 – Tela de projetos. . . . .	80
Figura 50 – Tela cadastro de projetos. . . . .	81
Figura 51 – Tela de visualização de projeto cadastrado. . . . .	81
Figura 52 – Tela de projetos: cadastro de atividades. . . . .	82
Figura 53 – Tela de projetos: visualização de atividades. . . . .	83
Figura 54 – Tela de projetos: gerenciador de dependências. . . . .	83
Figura 55 – Tela de projeto: iniciando atividade de um projeto . . . . .	84
Figura 56 – Tela de projeto: apenas uma atividade inicializada por vez . . . . .	85
Figura 57 – Tela de time . . . . .	85

Figura 58 – Tela cadastro de time . . . . .	86
Figura 59 – Tela cadastro de time: visualização . . . . .	87
Figura 60 – Tela de informações do projeto: Visualização . . . . .	87
Figura 61 – Tela de informações do projeto: Edição . . . . .	88
Figura 62 – Tela de informações do projeto: Adicionar Time . . . . .	88
Figura 63 – Tela de informações do projeto: Visualização com time . . . . .	89
Figura 64 – Tela de time: Visualização do time . . . . .	90

## LISTA DE TABELAS

Tabela 1 – Comparativo de funcionalidades entre os aplicativos mais populares de gestão e elaboração de cronogramas e o sistema desenvolvido neste trabalho.	17
Tabela 2 – Requisitos funcionais do projeto.	19
Tabela 3 – Requisitos não funcionais do projeto.	21
Tabela 4 – Caso de uso: Efetuar Login	28
Tabela 5 – Caso de uso: Manter atividades	28
Tabela 6 – Caso de uso: Responsabilizar pela atividade	29
Tabela 7 – Caso de uso: Acessar relatórios	30
Tabela 8 – Caso de uso: Manter projeto	31
Tabela 9 – Caso de uso: Manter um time	32
Tabela 10 – Caso de uso: Manter membros	33
Tabela 11 – Caso de uso: Manter Categoria	34
Tabela 12 – Caso de uso: Realizar atividade	34
Tabela 13 – Sprint Planning: período do dia 23 de junho de 2018 a 7 de julho de 2018.	48
Tabela 14 – Sprint Planning: período do dia 08 de julho de 2018 a 21 de julho de 2018	48
Tabela 15 – Sprint Planning: período do dia 18 de agosto de 2018 a 01 de setembro de 2018.	49
Tabela 16 – Sprint Planning: período do dia 22 de setembro de 2018 a 6 de outubro de 2018	49
Tabela 17 – Sprint Planning: período do dia 25 de Fevereiro de 2019 a 17 de março de 2019	50
Tabela 18 – Sprint Planning: período do dia 18 de março de 2019 a 07 de abril de 2019	50
Tabela 19 – Sprint Planning: período do dia 8 de abril de 2019 a 28 de abril de 2019	51
Tabela 20 – Sprint Planning: período do dia 29 de abril de 2019 a 19 de maio de 2019	52
Tabela 21 – Sprint Planning: período do dia 27 de maio de 2019 a 16 de junho de 2019	53
Tabela 22 – Sprint Planning: período do dia 22 de junho de 2019 a 5 de julho de 2019	53
Tabela 23 – Sprint Planning: período do dia 6 de julho de 2019 a 19 de julho de 2019	54
Tabela 24 – Sprint Planning: período do dia 20 de julho de 2019 a 02 de agosto de 2019	55
Tabela 25 – Sprint Planning: período do dia 3 de agosto de 2019 a 16 de agosto de 2019	55
Tabela 26 – Sprint Planning: período do dia 17 de agosto de 2019 a 30 de agosto de 2019	56
Tabela 27 – Sprint Planning: período do dia 31 de agosto de 2019 a 13 de setembro de 2019	57
Tabela 28 – Sprint Planning: período do dia 14 de setembro de 2019 a 27 de setembro de 2019	57
Tabela 29 – Sprint Planning: período do dia 30 de agosto de 2019 a 13 de setembro de 2019	57

Tabela 30 – Sprint Planning: período do dia 30 setembro de 2019 a 13 de novembro de 2019. . . . .	58
Tabela 31 – Heurísticas de Nielsen . . . . .	71
Tabela 32 – Comparativo das telas do sistema com as heurísticas de Nielsen . . . . .	72

## SUMÁRIO

<b>1 – INTRODUÇÃO</b> . . . . .	<b>16</b>
<b>2 – PROJETO DO CRONOTE</b> . . . . .	<b>18</b>
2.1 METODOLOGIA DE DESENVOLVIMENTO . . . . .	18
2.1.1 SCRUM . . . . .	18
2.2 LEVANTAMENTO DE REQUISITOS . . . . .	19
2.2.1 REQUISITOS FUNCIONAIS . . . . .	19
2.2.2 REQUISITOS NÃO FUNCIONAIS . . . . .	20
2.3 RECURSOS E FERRAMENTAS . . . . .	21
<b>3 – MODELAGEM</b> . . . . .	<b>24</b>
3.1 CASOS DE USO . . . . .	24
3.2 MODELAGEM DO CASO DE USO . . . . .	24
3.2.1 DOCUMENTAÇÃO DOS CASOS DE USO . . . . .	27
3.3 DIAGRAMA DE CLASSE . . . . .	35
3.4 DIAGRAMA DE SEQUÊNCIA . . . . .	39
3.5 DIAGRAMA DE ENTIDADE RELACIONAMENTO . . . . .	45
<b>4 – DESENVOLVIMENTO</b> . . . . .	<b>47</b>
4.1 CRONOGRAMA DE DESENVOLVIMENTO . . . . .	47
4.1.1 PLANEJAMENTO . . . . .	47
4.1.2 SPRINTS . . . . .	47
4.1.3 BURNDOWN . . . . .	58
4.2 DESIGN . . . . .	68
4.2.1 IDENTIDADE VISUAL . . . . .	68
4.2.2 PALETA DE CORES . . . . .	70
4.2.3 HEURÍSTICAS DE NIELSEN . . . . .	71
4.3 MANUAL . . . . .	72
4.3.1 LOGIN . . . . .	73
4.3.2 CADASTRO . . . . .	73
4.3.3 RECUPERAÇÃO DE SENHA . . . . .	74
4.3.4 ATIVIDADE . . . . .	77
4.3.5 INICIANDO ATIVIDADE . . . . .	84
4.3.6 TIME . . . . .	85
<b>5 – CONSIDERAÇÕES FINAIS</b> . . . . .	<b>91</b>

**Referências . . . . . 92**

## 1 INTRODUÇÃO

No Brasil a população não dispõe do hábito de planejar. “[...] Os brasileiros não estão acostumados a um planejamento de longo prazo, significando que eles passam a maior parte do tempo na fase de execução improvisando à medida que avançam e optando pelas soluções mais pragmáticas” Pearson (2011).

Cada vez mais é evidente a necessidade de precautelar para os diversos imprevistos no dia a dia, seja ele qual for. E para que isso ocorra da maneira correta é necessário que tenha a elaboração de um plano, por menor irrelevância tenha a atividade, saber gerir e estar preparado é fundamental para que se obtenha sucesso alcançando assim um objetivo. Quando se fala em gerenciamento de projetos, é necessário estudar e avaliar todo o contexto no qual os projetos estão inseridos e todos os benefícios que este gerenciamento traz para o seu objetivo estipulado. Para Vieira (2007) “o processo de planejamento define e refina os objetivos e as escolhas das melhores alternativas de ação para atingir os objetivos propostos pelo projeto”. Neste contexto, planejar bem é algo indispensável.

A fim de atender as necessidades relacionadas a planejamento e auxiliar na elaboração de cronogramas, existem alguns softwares que facilitam essa gestão. Este trabalho tem como objetivo desenvolver um sistema para gerenciamento de atividades e projetos. Este sistema foi desenvolvido com a linguagem Java, utilizando JavaFX para interface gráfica juntamente com as bibliotecas Jfoenix, Controlsfx, Fontawesomefx, PostgreSQL para o armazenamento de dados. Para que seja possível realizar este objetivo, foram propostos alguns objetivos específicos, que são:

- Utilizar o framework Scrum e expor os procedimentos realizados;
- Utilizar os princípios RESTful para desenvolver uma API de comunicação entre o banco de dados remoto e o sistema;
- Desenvolver um design eficiente e atrativo utilizando UI/UX, javaFX;
- Utilizar padrões de desenvolvimento MVC

Popular por sua simplicidade, o Trello é uma referência digital para gerenciamento de tarefas que segue o método "kanban", muito usado no desenvolvimento com Scrum. Com ele, o usuário pode criar sua tarefa como cards, definir comentários, anexos, usuários entre outras opções em seus diversos quadros nos quais podemos criar quantas colunas forem necessárias. Podendo também nomear, de acordo com sua realidade [TRELLO](#) (2011).

O RunRun.it se define como uma plataforma completa para ajudar a empresa a controlar tarefas, projetos, fluxo do trabalho e o tempo investido. Ao criar sua tarefa ele associa dependências e sequência de responsáveis envolvidos e, a partir de uma base de dados, é calculado o tempo estimado para cada atividade [RUNRUN.IT](#) (2007).

Asana é conhecido por sua ótima usabilidade e seu principal diferencial é na coordenação do trabalho colaborativo da equipe. Com ela, todos sabem o que precisa ser feito, quem é o

responsável pela atividade e quando o projeto deverá estar finalizado, juntamente com o seu controle de dependências [ASANA \(2008\)](#).

Por fim, o Gantter é responsável por cadastrar todas as tarefas do projeto, sua duração e predecessores tendo acesso a um cronograma bem visual. A ferramenta também conta com o recurso de mapeamento de riscos, entretanto, ele não acompanha a execução das tarefas e a necessidade de um gestor é imprescindível [GANTTER \(2011\)](#).

Levando estes aspectos em consideração, foi elaborada a Tabela 1 mostrando as principais diferenças entre as plataformas mais populares de gestão e elaboração de cronogramas e o aplicativo desenvolvido neste trabalho:

A Tabela 1 representa um comparativo do sistema com os concorrentes, definidos em várias funcionalidades.

Tabela 1 – Comparativo de funcionalidades entre os aplicativos mais populares de gestão e elaboração de cronogramas e o sistema desenvolvido neste trabalho.

<b>Funcionalidade</b>	<b>Cronote</b>	<b>Trello</b>	<b>RunRun.it</b>	<b>Asana</b>	<b>Gantter</b>
Controle de atividades por tempo	X	X	X	X	
Ótima usabilidade	X	X		X	
Cadastro de atividades e projetos	X	X	X	X	X
Controle de dependências	X		X	X	X
Exportação para as demais plataformas: PDF, DOCX, XLS					X
Relatórios mensais comparativos			X	X	X
Facilidade ao adicionar e/ou trocar de conta rapidamente	X				
Sistema colaborativo	X	X	X	X	X
Interface intuitiva	X	X		X	
Versão gratuita	X	X			

Fonte: Elaborado pelos autores

## 2 PROJETO DO CRONOTE

Este capítulo detalha o processo do desenvolvimento de um sistema de gerenciamento de atividades e projetos que elabora um balanço e comparações das atividades cadastradas e/ou projetos predeterminados por meio da engenharia de software. Também descreve todas as tecnologias necessárias para o desenvolvimento do Cronote.

### 2.1 METODOLOGIA DE DESENVOLVIMENTO

Uma das características que é escolhida quando o projeto está sendo planejado é qual será a metodologia de desenvolvimento utilizada. Sendo uma decisão de extrema importância pois ela direcionará a forma como o projeto será desenvolvido, quais pessoas farão parte, quando e quais entregas serão feitas, além de outros pontos importantes.

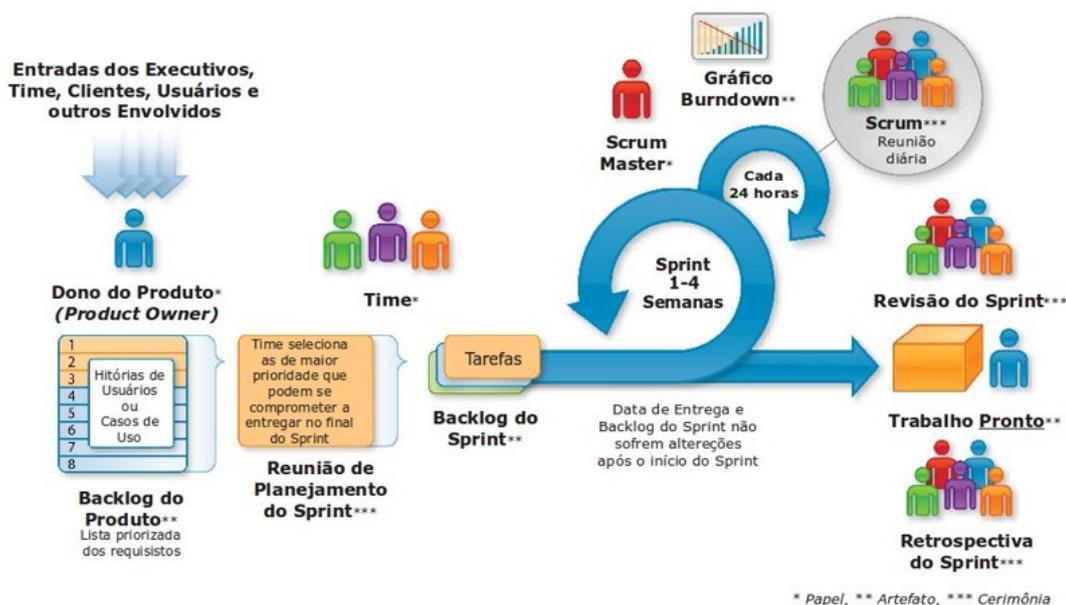
#### 2.1.1 SCRUM

O framework Scrum é bem popular, muito conhecida na área de tecnologia e geralmente é utilizada em projetos de desenvolvimento de software ágil, mas por incrível que pareça, quase completamente desconhecido por profissionais de outras áreas, incluindo os profissionais de planejamento. [SUTHERLAND \(2014\)](#).

Historicamente, o termo Scrum surgiu em um artigo publicado por Hirotaka Takeuchi e Ikujiro Nonaka na Harvard Business Review de 1986. Nesse artigo, intitulado “The new product development game” (“O novo jogo do desenvolvimento de produtos”), Takeuchi e Nonaka descreveram uma abordagem holística, na qual equipes de projeto são compostas de pequenas equipes multifuncionais, trabalhando com sucesso rumo a um objetivo comum, que os autores compararam à formação Scrum do rugby.

Enquanto o PMBoK (Project Management Body of Knowledge) é um guia de boas práticas, dividido em grupos e processos, o Scrum é um framework (estrutura) com cerimônias, valores, princípios e papéis definidos. A metodologia contempla esse conjunto de práticas que podem ser aplicados de formas diferentes de acordo com a realidade de cada projeto. De forma geral, o Scrum se baseia em ciclos de desenvolvimento sucessivos (normalmente quinzenais ou mensais), chamados de Sprints.

Figura 1 – Estrutura ágil: Scrum em resumo



## 2.2 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.

### 2.2.1 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. A Tabela 2 representa os requisitos funcionais do sistema.

Tabela 2 – Requisitos funcionais do projeto.

Identificação	Requisito Funcional	Prioridade
RF001	Um(a) usuário(a) deve ser capaz de manter suas próprias atividades	Essencial
RF002	Um(a) usuário(a) deve ser capaz de manter seus próprios projetos	Essencial
RF003	Um(a) usuário(a) deve ser capaz de criar um time	Essencial
RF004	O(A) líder de cada time deve ser capaz de criar projetos e atividades para os membros.	Essencial

RF005	Cada membro(a) deve ser capaz de definir responsável por uma atividade caso ela não esteja sendo executada por ninguém.	Essencial
RF006	Cada atividade pode ser feita por um(a) membro(a) ou mais (não ao mesmo tempo).	Importante
RF007	As atividades podem ter dependências de outras atividades ou não	Essencial
RF008	O(A) usuário(a) deve ser capaz de adicionar e/ou alterar de conta rapidamente.	Essencial
RF009	O sistema deverá bloquear (edições) atividade após ser iniciada.	Essencial
RF010	O sistema deverá prever o tempo estimado de uma atividade com base na média de tempo real usado para atividades de mesma categoria e prioridade.	Essencial
RF011	As categorias de atividades usadas em um projeto deverão ser comuns a todos os membros do time (Apenas dentro do projeto).	Essencial
RF012	O sistema deve ser capaz de estimar o progresso de um projeto.	Essencial
RF013	O sistema deve ser capaz de contabilizar e exibir o progresso real de um projeto	Essencial

Fonte: Elaborado pelos autores.

### 2.2.2 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.

A Tabela 3 representa os requisitos não funcionais do sistema.

Tabela 3 – Requisitos não funcionais do projeto.

Identificação	Requisito Não Funcional	Categoria	Prioridade
RNF001	Design minimalista e de grande responsividade para as diferentes resoluções de monitores	Usabilidade	Essencial
RNF002	O sistema deve retornar resposta em no máximo 1 minuto	Desempenho	Importante
RNF003	Deve ser utilizado o PostgreSQL como gerenciador do banco de dados.	Ambiental	Essencial
RNF004	Sistema intuitivo e fácil de ser utilizado	Usabilidade	Essencial

Fonte: Elaborado pelos autores

### 2.3 RECURSOS E FERRAMENTAS

**Java:** É atualmente uma das linguagens mais utilizadas em todo o mundo e continua em crescimento nas empresas, através de novas adoções. Um ponto importante é que hoje o Java não é apenas uma linguagem, mas sim uma ambiente de desenvolvimento multiplataforma. Pode ser desenvolvida em qualquer sistema operacional, por usar uma máquina virtual para executar todos os seus aplicativos. É uma linguagem de alto nível [DEVMEDIA \(2017\)](#).

**JavaFX:** Uma nova biblioteca gráfica da linguagem Java. Com ela é possível desenvolver interfaces gráficas, animações com CSS e Java, gráficos, reproduzir áudio e vídeo. Além disso, foram utilizadas as bibliotecas: **Jfoenix**, **Controlsfx**, **Fontawesomefx** – que são coleções de projetos de código aberto que tornam o trabalho com o JavaFX melhor e mais prático. Foram aplicados também, os padrões de UI e UX (*User Interface* e *User Experience*). O UI é o meio que o usuário interage com o produto, todas as partes “físicas” que possibilitam uma determinada ação seja realizada. O UX envolve a parte emocional, o quão amigável essa interação pode ser e os sentimentos que ele tem ao realizá-la. Engloba design e desenvolvimento do produto, como usabilidade, arquitetura da informação e experiência [Wikibooks \(2016\)](#).

**Maven:** é uma ferramenta de integração de projetos. É responsável por gerenciar dependências, controlar versão de artefatos, gerar relatórios de produtividade, garantir execução de testes, manter nível de qualidade do código dentre outras. Maven torna possível isolar as bibliotecas usadas no projeto em um ‘repositório’ compartilhado pela equipe, ou por toda internet no caso do repositório central do Maven. Dessa forma não é necessário qualquer preocupação quanto a duplicidade de dependências entre módulos do projeto e nem em relação à disponibilidade das mesmas no repositório de código [MAVEN \(2018\)](#).

**Eclipse IDE:** O Eclipse é uma IDE (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado) de código aberto utilizado para desenvolvimento em linguagem java, mas é também multilinguagem, ou seja, suporta outras linguagens de programação tais como C/C++ instalando-se os devidos plug-ins adicionais para cada linguagem. É um projeto livre de patentes por ser um software livre. E é portátil, ou seja, sua aplicação funciona

vários ambientes. Permite também a refatoração do código, que é uma forma organizada de reestruturar o código para ser melhorado [IDE \(2019\)](#).

**PostgreSQL:** é um SGBD (Sistema Gerenciador de Banco de Dados) relacional, que utiliza a linguagem SQL (*Structured Query Language*, ou Linguagem de Consulta Estruturada). PostgreSQL também é multiusuário e multitarefas. A escolha do PostgreSQL para o desenvolvimento do sistema Cronote foi por ser de fácil utilização, além de possuir uma interface extremamente simples e ser compatível com grande parte dos sistemas operacionais e totalmente gratuito [POSTGRESQL \(2017\)](#).

**MVC:** é uma arquitetura de desenvolvimento que organiza a aplicação em três camadas: modelo (*model*), visualização (*view*) e controladora (*controller*). Cada uma das camadas tem uma função específica [Celestino \(2014\)](#):

**Modelo:** Sempre que o assunto for manipular dados, entra em ação o modelo. Ele é responsável pela leitura e escrita de dados, e também de suas validações;

**Visualização:** a camada de interação da aplicação com o usuário. Ela apenas faz a exibição dos dados, a interface do sistema;

**Controladora:** O responsável por receber todas as requisições do usuário feitas através da camada de visualização. Cada evento programado na visualização gera uma requisição que a camada controladora é responsável por atender, controlando qual modelo irá processar os dados e qual visualização será exibida ao usuário.

**O termo API** (*Application Programming Interface* ou Interface de Programação de Aplicações) trata-se de um conjunto de rotinas e padrões estabelecidos e documentados por uma aplicação A, para que outras aplicações consigam utilizar as funcionalidades desta sem precisar conhecer detalhes de sua implementação. Desta forma, as APIs permitem uma interoperabilidade entre aplicações. Em outras palavras, a comunicação entre aplicações e entre os usuários [Celestino \(2014\)](#).

**Codacy:** Identifica automaticamente os problemas por meio da análise e revisão do código submetido. Notifica sobre problemas de segurança, cobertura de código, duplicação de código e complexidade de código em todas as solicitações de confirmação e recebimento, diretamente de seu fluxo de trabalho atual [Codacy \(2018\)](#).

**Tomcat:** O software Tomcat, desenvolvido pela Fundação Apache, permite a execução de aplicações para web. Sua principal característica técnica é estar centrada na linguagem de programação Java, mais especificamente nas tecnologias de Servlets e de *Java Server Pages* (JSP) [DEVMEDIA \(2007\)](#).

**REST:** traz uma série de princípios/regras que permitem a criação de um projeto com interfaces bem definidas. Permitindo, por exemplo, que aplicações se comuniquem. RESTful é uma métrica que mede a capacidade do sistema de atender aos princípios REST. Um dos principais frameworks para desenvolver aplicações REST em Java é o **Jersey**, que é um framework que implementa todas as características da arquitetura REST [Dias \(2016\)](#).

**Hibernate:** é o framework para persistência de dados mais utilizado em projetos Java.

Sendo uma das primeiras opções a implementar o conceito de mapeamento objeto-relacional (ORM), em pouco tempo se tornou referência entre os desenvolvedores, tendo influenciado, inclusive, a criação da especificação JPA [hat \(2016\)](#).

**UML:** A Linguagem de modelagem unificada (UML) foi criada para estabelecer uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de software complexos, tanto estruturalmente quanto para comportamentos. Além do desenvolvimento de software, a UML tem aplicações em fluxos do processo na fabricação. É análoga aos modelos utilizados em outros campos, e é composta por diferentes tipos de diagramas. De modo geral, diagramas UML descrevem o limite, a estrutura e o comportamento do sistema e os objetos nele contidos [Guedes \(2009\)](#).

Para isso, utilizou-se a ferramenta **Astah Community** que é um software para modelagem UML, disponibilizado para o desenvolvimento de diagramas de Classes, Casos de Uso, Sequência, Comunicação, Máquina de Estados, Atividade, Componentes, Implantação e Diagrama de Estrutura Composta [ChangeVision \(2019\)](#).

**Git:** Git é um sistema de controle de versão de arquivos. Através deles podemos desenvolver projetos em que diversas pessoas podem contribuir simultaneamente, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas. O grande benefício é conseguir combinar versões de colaboradores diferente. Para o controle no desenvolvimento do Cronote optou-se pela ferramenta **GitHub**, que é um serviço web que oferece diversas funcionalidades extras aplicadas ao git. Podendo utilizar gratuitamente para hospedar o projeto. E também, o **GitKraken**, um cliente gráfico multiplataforma, muito prático e simplificado que torna mais simples as ações cotidianas que envolvem o git [Conservancy \(2016\)](#).

**Gmail biblioteca Commons-email:** Através do commons-email, é possível enviar e-mails simples, com textos puros, em formato HTML e também e-mail com anexos. Desenvolvido como uma camada mais abstraída da API JavaMail (J2EE-API) o que a tornou mais simples e robusto. Utilizou-se essa tecnologia para enviar os códigos gerados na redefinição da nova senha do usuário e outras notificações/confirmações [Commons \(2018\)](#).

Ferramentas complementares:

**WhatsApp e Discord:** Para a comunicação e reuniões de planejamento.

**Trello:** Para o acompanhamento das sprints.

### 3 MODELAGEM

A modelagem é feita a partir da UML (*Unified Modeling Language*), que nos auxilia a definir as características do nosso software, tais como os requisitos e estrutura lógica, facilitando o entendimento sobre o que seria desenvolvido [Guedes \(2009\)](#).

#### 3.1 CASOS DE USO

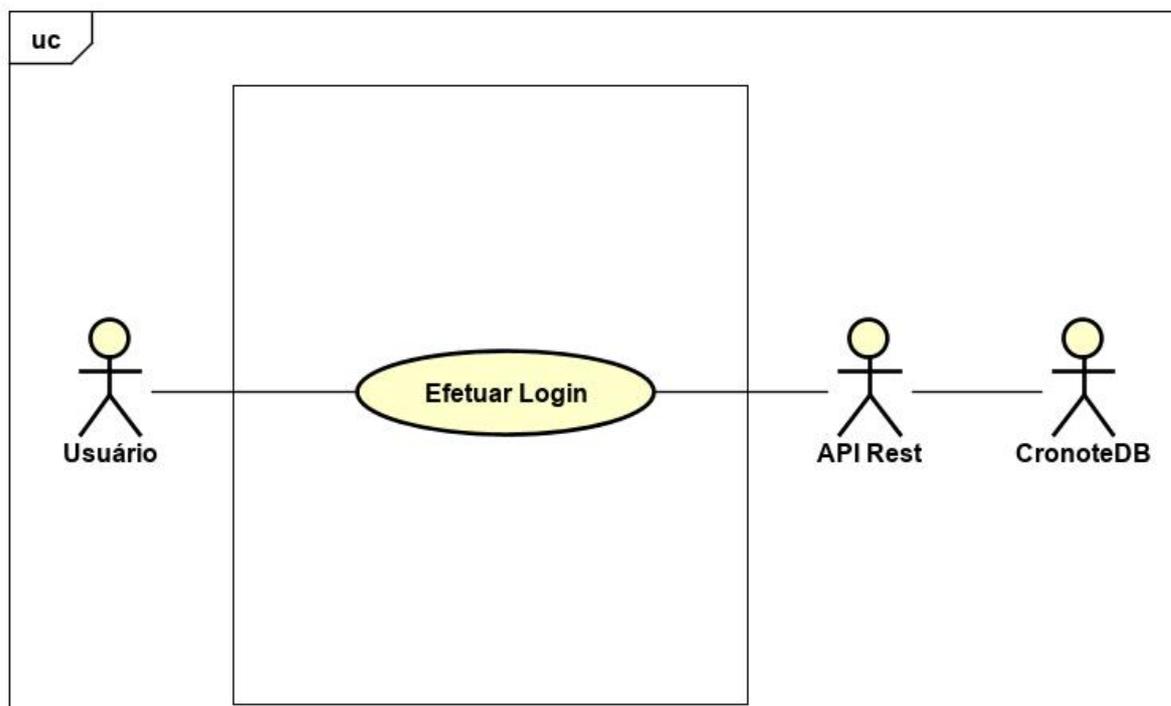
O caso de uso é o diagrama mais geral e informal da UML, nele é apresentado um cenário de funcionalidades do sistema, representando os requisitos funcionais. Os atores (representados por bonecos), são pessoas ou outros programas que se utilizam do software, as elipses representam as diferentes funcionalidades que poderão ser feitas, e por fim, as linhas representam as relações do atores com o sistema, o que cada um terá disponível no sistema [Guedes \(2009\)](#). Os atores que irão interagir com o sistema são: O usuário, a API Rest, CronoteDB, Líder e Membro.

- **Usuário:** é o ator que representa as pessoas que irão utilizar o sistema de forma individual, podendo se tornar líder ou membro.
- **API Rest:** é o ator que representa a API rest que cuida dos dados que estarão indo para o servidor.
- **CronoteDB:** é o ator que representa o banco de dados do sistema.
- **Líder:** é o ator que representa o usuário que criou um time, sendo assim, o dono do time.
- **Membro:** é o ator que representa o usuário que entrou em time não sendo quem o criou.

#### 3.2 MODELAGEM DO CASO DE USO

A Figura 2 apresenta o caso de uso do login do usuário no sistema.

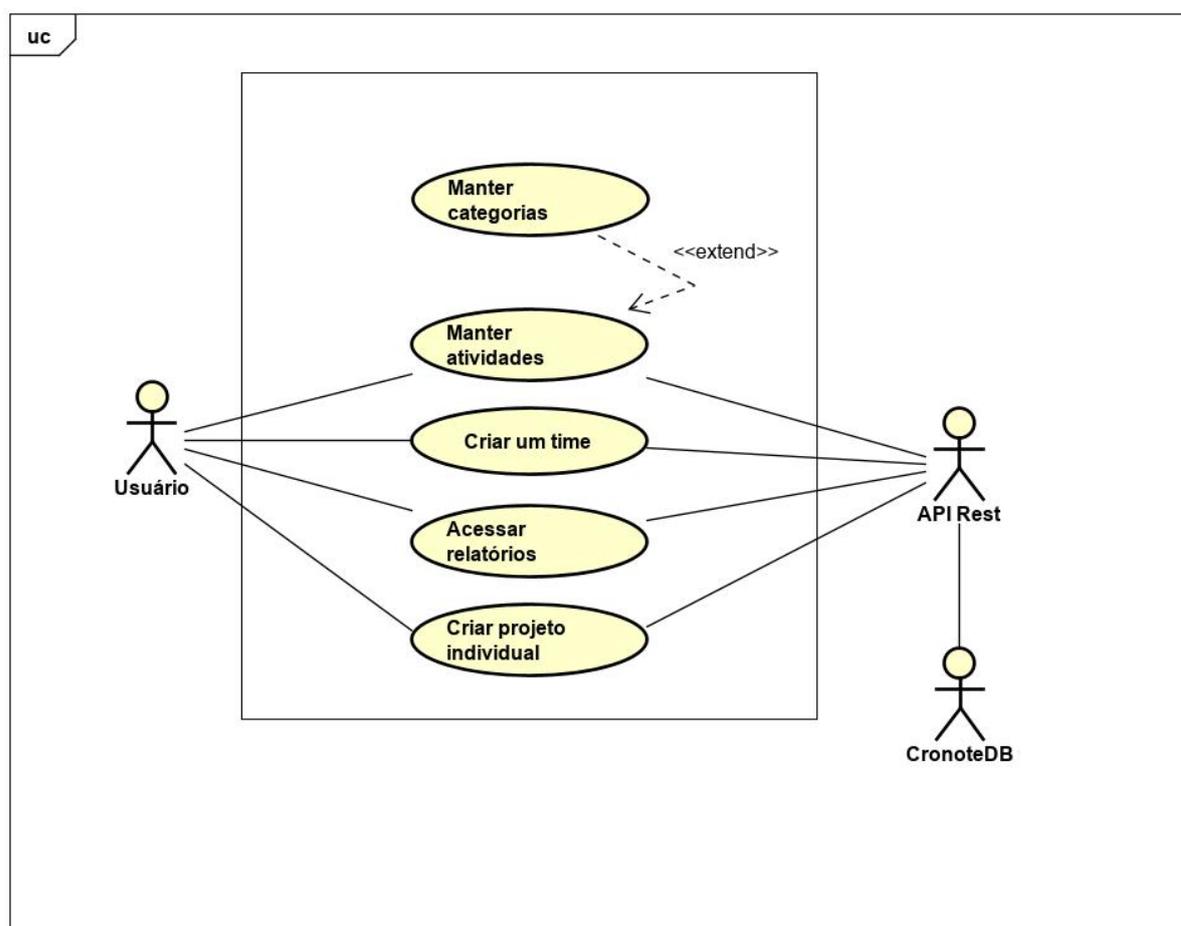
Figura 2 – Diagrama de caso de uso do login do usuário



Fonte: Elaborado pelos autores

A Figura 3 apresenta o caso de uso das ações básicas do usuário que não está em um time.

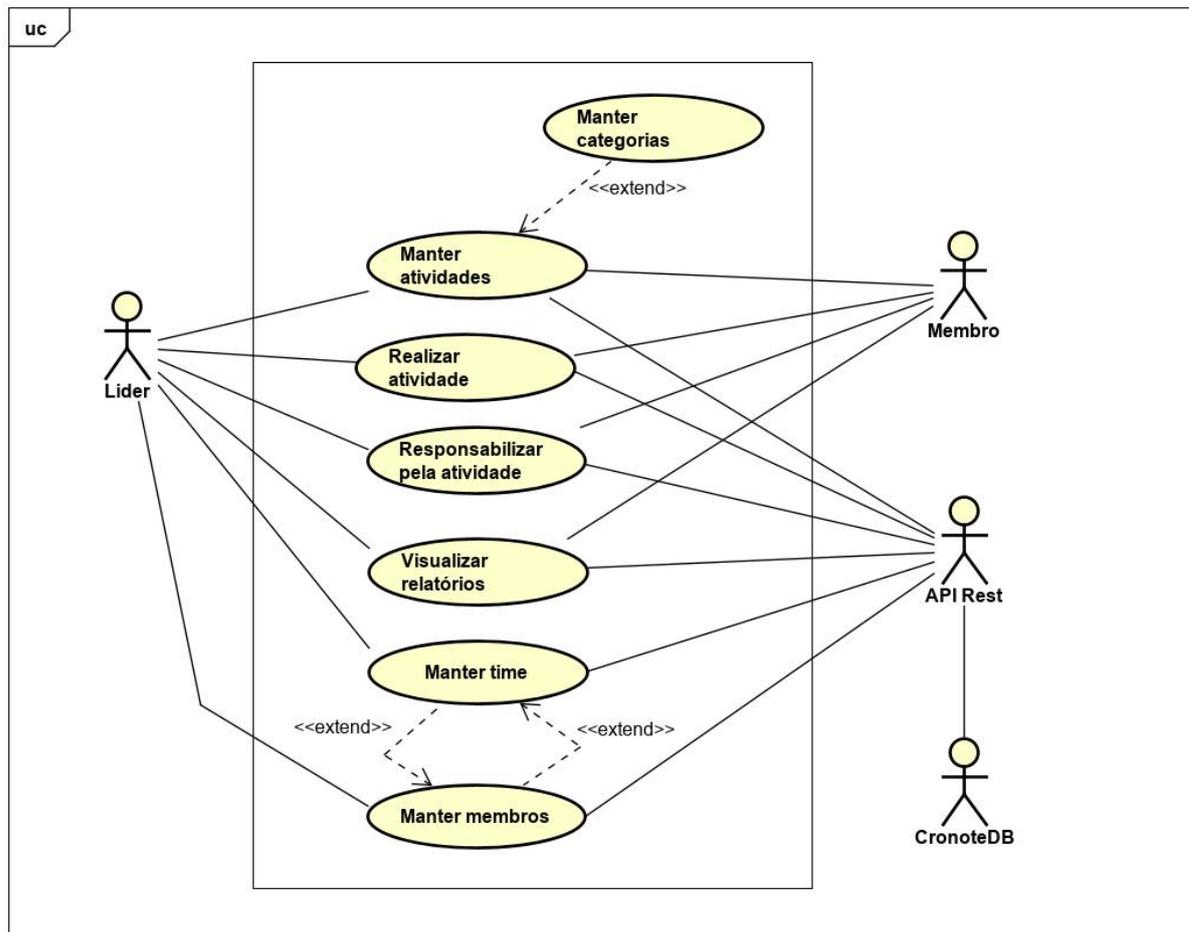
Figura 3 – Diagrama de caso de uso das ações básicas de um usuário.



Fonte: Elaborado pelos autores

A Figura 4 apresenta o caso de uso das ações tanto de um membro e como um líder time.

Figura 4 – Diagrama de caso de uso das ações do líder e dos membros de um time.



Fonte: Elaborado pelos autores

Na seção 3.2.1 é apresentado a documentação dos casos de uso do desenvolvimento deste trabalho.

### 3.2.1 DOCUMENTAÇÃO DOS CASOS DE USO

Cada funcionalidade dos casos de uso é descrita da tabela 4 até a tabela 12. O caso de uso "Efetuar login", utilizará a Api Rest para se comunicar com o banco e assim verificar se o usuário existe ou não, caso exista o sistema iniciará o carregamento e dará as informações ao usuário caso o usuário não exista será mostrada uma mensagem de erro dizendo que as credenciais estão incorretas.

A Tabela 4 representa o caso de uso de efetuar login.

Tabela 4 – Caso de uso: Efetuar Login

<b>Nome do caso de uso</b>	Efetuar Login
<b>Atores envolvidos</b>	Usuário, API REST, Cronote DB.
<b>Objetivo</b>	Este caso de uso descreve os passos do login para os usuários do sistema.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	Ações do sistema
1. O usuário clica em login	
	2. O sistema realiza uma requisição à API REST que pesquisa as informações de usuário e senha do usuário e os autentica.
	3. Depois da autenticação, o sistema direciona o usuário para a página principal do aplicativo.
<b>Validações</b>	Para o login ser efetuado, o usuário deverá digitar seu usuário e senha.

Fonte: Elaborado pelos autores

O caso de uso de "Manter atividades", utiliza o Api Rest para realizar quaisquer das seguintes atividades: criação, atualização ou exclusão de qualquer atividade,

A Tabela 5 representa o caso de uso de manter atividades.

Tabela 5 – Caso de uso: Manter atividades

<b>Nome do caso de uso</b>	Manter atividades
<b>Atores envolvidos</b>	Usuário, API REST, Cronote DB.
<b>Objetivo</b>	Este caso de uso descreve os passos do usuário ao criar e/ou modificar uma atividade.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	Ações do sistema
1. O usuário clica em cadastrar uma atividade.	
	2. O sistema pega os dados que o usuário inseriu e solicita uma requisição à API REST que então guardará as informações no banco de dados.
3. O usuário faz uma busca de uma atividade por nome e/ou categoria.	
	4. O sistema pega a informação de qual dos filtros ele usou e faz uma requisição à API REST para que realize uma busca com os parâmetros passados.

5. O usuário realiza uma alteração na atividade, clicando na atividade e depois no botão de editar.	
	6. O sistema mostra uma nova tela ao usuário, para que então, mude ou adicione novos dados à atividade.
7. O usuário preenche as informações e por fim irá pressionar o botão de confirmar.	
	8. O sistema então pega as informações que foram colocadas e faz a requisição à API REST para que sejam gravadas essas novas informações.
9. O usuário clica em uma atividade que ainda não foi iniciada e então clicar em deletar.	
	10. O sistema exibe uma mensagem de confirmação da ação a ser realizada.
11. O usuário então tem duas opções, e irá escolher entre "sim" ou "não", clicando nos seus respectivos botões.	
	12. Caso o usuário tenha clicado em "sim" a API do sistema muda o status da atividade para "inativo", assim excluindo para o usuário e exibindo uma mensagem de confirmação de exclusão. Caso clique em "não", a mensagem de confirmação se fecha.
<b>Validações</b>	Para cadastro e alteração é feito uma validação dos dados, se foram preenchidos corretamente, respeitando todos os formatos pré-definidos.

Fonte: Elaborado pelos autores

O caso de uso de "Responsabilizar pela atividade", o usuário pode então começar uma atividade e se responsabilizar por ela.

A Tabela 6 representa o caso de uso de se responsabilizar pela atividade.

Tabela 6 – Caso de uso: Responsabilizar pela atividade

<b>Nome do caso de uso</b>	Criar um time
----------------------------	---------------

<b>Atores envolvidos</b>	Usuário, API REST, Cronote DB.
<b>Objetivo</b>	Este caso de uso descreve os passos do usuário ao criar um time.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	Ações do sistema
1.O usuário clica em iniciar a atividade.	
	2. O sistema então faz uma requisição a Api Rest que registra o inicio da atividade e vincula o usuário a atividade.
3. O usuário clica em pausar a atividade.	
	4. O sistema faz uma requisição a Api Rest que registra a pausa da atividade.
<b>Validações</b>	O sistema valida se já existe um time em que o usuário participa com o mesmo nome.

Fonte: Elaborado pelos autores

O caso de uso de "Acessar relatórios", utiliza a Api Rest para acessar o banco de dados do sistema, buscando os dados e os apresentando conforme o filtro e o tipo de relatório requisitado.

A Tabela 7 representa o caso de uso de acessar relatórios

Tabela 7 – Caso de uso: Acessar relatórios

<b>Nome do caso de uso</b>	Acessar relatórios
<b>Atores envolvidos</b>	Usuário, API REST, Cronote DB.
<b>Objetivo</b>	Este caso de uso descreve os passos do usuário ao criar e/ou modificar uma atividade.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	Ações do sistema
1.O usuário clica em visualizar um relatório.	
	2. O sistema abre uma nova tela, mostrando todos os tipos que estão disponíveis no sistema e apresenta os filtros de escolha.

3. O usuário então escolhe corretamente qual o tipo do relatório, assim como os seus devidos filtros, além do formato do arquivo.	
	4. O sistema faz uma requisição à API REST para trazer os dados especificados pelos filtros e então monta o relatório no formato desejado pelo usuário, além de deixar o usuário em uma tela de carregamento enquanto executa a tarefa e depois exibi-lo ao usuário.
5. O usuário depois de ver o seu relatório e se está tudo correto com o que ele especificou, clica em concluir.	
<b>Validações</b>	Os relatórios só serão criados caso todos os dados estejam de acordo com o sistema e tipo de relatório.

Fonte: Elaborado pelos autores

O caso de uso de "Manter projetos", utiliza a Api Rest para acessar o banco de dados cadastrando, editando ou excluindo o projeto.

A Tabela 8 representa o caso de uso de manter projeto.

Tabela 8 – Caso de uso: Manter projeto

<b>Nome do caso de uso</b>	Manter projeto
<b>Atores envolvidos</b>	Usuário, API REST, Cronote DB.
<b>Objetivo</b>	Este caso de uso descreve os passos do usuário ao criar um projeto.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	Ações do sistema
1.O usuário clica em criar um projeto.	
	2. O sistema mostra uma página de cadastro de projetos.
3. O usuário informa todos os dados corretamente, decidirá entre atribuir o projeto a algum time ou não e clica em criar	

	4. O sistema pega todos os dados e envia para a API REST que então se encarrega de cadastrar o projeto no banco de dados.
5. O usuário então poderá realizar o cadastro de atividades para aquele projeto.	
<b>Validações</b>	Cadastro de projetos precisa de todas as informações corretamente preenchidas nos formatos definidos.

Fonte: Elaborado pelos autores

O caso de uso "Manter time", utiliza a API Rest para cadastrar, editar ou excluir as informações do time e que fará sempre a validação dos nomes utilizados.

A Tabela 9 representa o caso de uso de manter um time.

Tabela 9 – Caso de uso: Manter um time

<b>Nome do caso de uso</b>	Manter time
<b>Atores envolvidos</b>	Líder, API REST, Cronote DB.
<b>Objetivo</b>	Este caso de uso descreve os passos do usuário ao criar e/ou modificar um time.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	Ações do sistema
1. O líder entra no modo de edição do time e altera qualquer informação desejada e para manipular os membros ele precisa: arrastar o usuário que ele deseja pra lista de "Membros" ou de "Não Membros" e clica em salvar.	
	2. O sistema faz uma requisição na API REST, que então tira o membro daquele time.
	3. O sistema então pega as informações que o usuário altera e atualiza o banco de dados.
<b>Validações</b>	Para cadastro e alteração é feita uma validação dos dados, se forem preenchidos corretamente, respeitando todos os formatos pré-definidos.

Fonte: Elaborado pelos autores

O caso de uso "Manter membros", utiliza a Api Rest para se comunicar com o banco de dados e neste caso envolve não apenas o sistema, mas também o email de um usuário, por que o convite será enviado no email e o sistema consegue reconhecer a requisição e efetiva o cadastro do novo membro.

A Tabela 10 representa o caso de uso de manter membros.

Tabela 10 – Caso de uso: Manter membros

<b>Nome do caso de uso</b>	Manter membros
<b>Atores envolvidos</b>	Líder, API REST, Cronote DB.
<b>Objetivo</b>	Este caso de uso descreve os passos do usuário para adicionar ou remover membros do time.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	Ações do sistema
1.O líder entra no modo de edição do time e pesquisa por usuários (em caso de adição).	
	2. O sistema faz uma requisição para a API REST que retorna os usuários compatíveis com a pesquisa.
3. o líder então arrastará o usuário que ele deseja pra lista de "Membros"(no caso de adição) ou de "Não Membros"(no caso de exclusão) e clica em salvar.	
	4. O sistema então pega as informações que o usuário alterou e faz uma requisição para que a API atualize o banco de dados.
	5. A API verifica os usuários que estão sendo adicionados e envia emails de convite para eles.
6. O usuário convidado entra em seu e-mail e clica em aceitar convite para efetivar seu cadastro no time.	
	7. O sistema recebe uma requisição desse usuário e mudará o status do usuário para ativo, efetivando seu cadastro.
<b>Validações</b>	Será feita uma validação dos dados, se todos os campos obrigatórios foram preenchidos corretamente, respeitando todos os formatos pré-definidos.

Fonte: Elaborado pelos autores

O caso de uso de "Manter Categoria", utiliza a Api Rest para cadastrar, editar ou deletar as categorias. A validação do sistema é apenas para verificar se não há outra categoria com o mesmo nome que o usuário já tenha cadastrado.

A Tabela 11 representa o caso de uso de manter categoria.

Tabela 11 – Caso de uso: Manter Categoria

<b>Nome do caso de uso</b>	Manter categoria
<b>Atores envolvidos</b>	Usuário, API REST, Cronote DB.
<b>Objetivo</b>	Este caso de uso descreve os passos ao manter uma categoria.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	Ações do sistema
1. O usuário clica em adicionar uma categoria	
	2. O sistema mostra a ele um novo campo em que ele pode digitar o nome da nova categoria.
3. O usuário digita a nova categoria e clica em confirmar	
	4. O sistema pega a informação manda para a API REST e então é cadastrada no banco de dados.
5. O usuário clica em editar a categoria.	
	6. O sistema mostra uma tabela com as categorias já cadastradas pelo usuário.
7. O usuário seleciona a categoria que deseja editar e troca as suas informações.	
	8. O sistema manda para a API REST que valida os dados e confirma a mudança no banco de dados.
<b>Validações</b>	Para se cadastrar ou editar uma categoria, o sistema valida se já existe alguma com o mesmo nome.

Fonte: Elaborado pelos autores

O caso de uso de "Realizar atividade", utilizará a Api Rest para informar ao servidor o andamento de tal atividade, assim como se o usuário pausou ou se já extrapolou o tempo planejado. Assim que a atividade for iniciada

A Tabela 12 representa o caso de realizar atividade.

Tabela 12 – Caso de uso: Realizar atividade

<b>Nome do caso de uso</b>	Realizar atividade
<b>Atores envolvidos</b>	Usuário, API REST, Cronote DB.

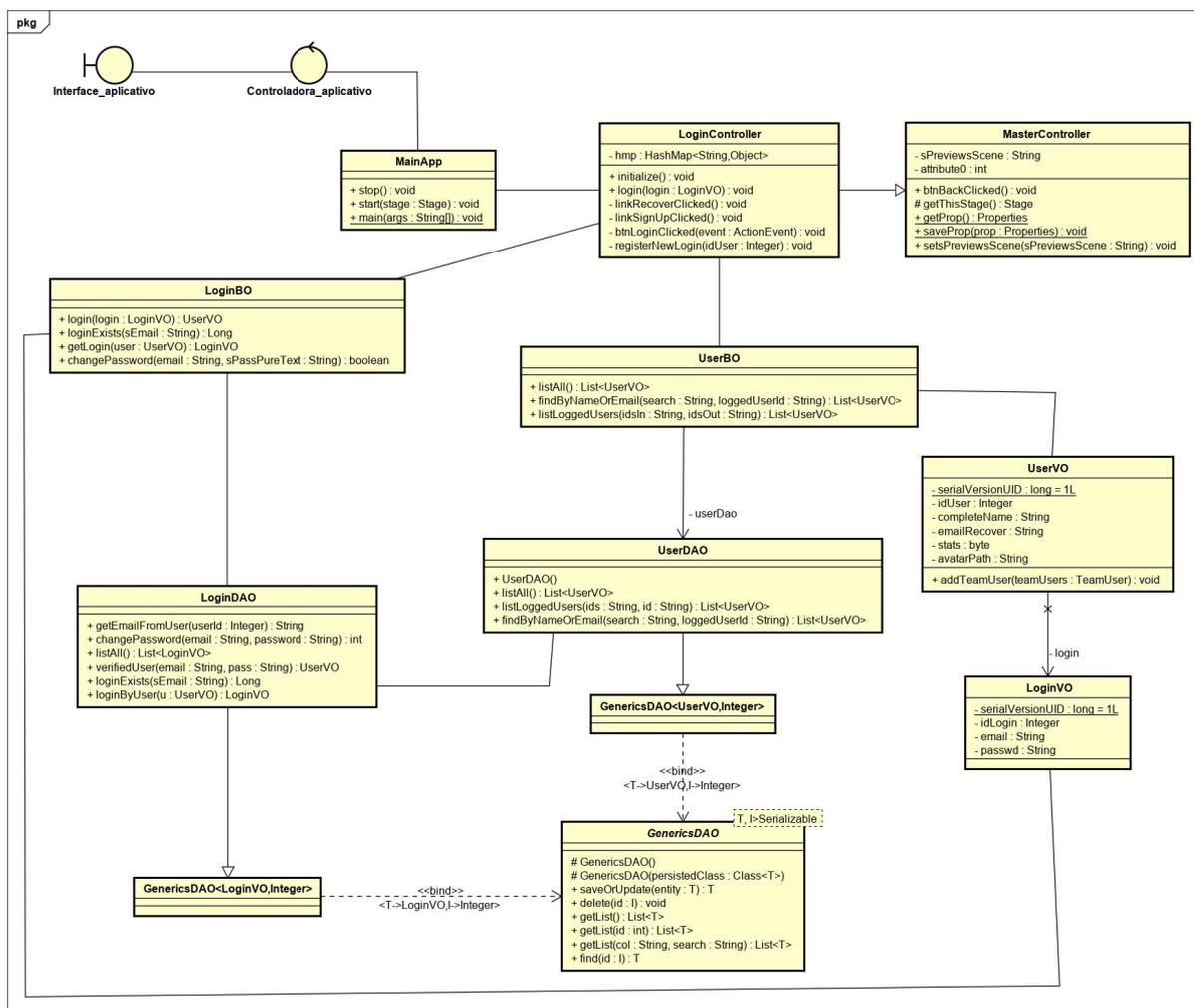
<b>Objetivo</b>	Este caso de uso descreve os passos ao manter uma categoria.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	Ações do sistema
1. O usuário clica em começar uma atividade	
	2. O sistema notificará a Api Rest que aquela atividade está em andamento.
	3. O sistema a cada 10 segundos irá fazer uma requisição à Api Rest com o tempo sendo contabilizado para todos os usuários do time.
4. O usuário clica em pausar a atividade.	
	5. O sistema enviará uma requisição a Api Rest que por sua vez executa o comando de pausa da atividade e notifica todos os usuários conectados do time.
6. O usuário clica em finalizar a atividade.	
	7. O sistema envia uma requisição a Api Rest que identifica e encerra o tempo de execução da atividade.
<b>Validações</b>	Para se cadastrar ou editar uma categoria, o sistema irá validar se já existe alguma com o mesmo nome.

Fonte: Elaborado pelos autores

### 3.3 DIAGRAMA DE CLASSE

O diagrama de classe é responsável por representar a estrutura e relações das classes, como também as operações solicitadas pelos atores dos casos de uso [Guedes \(2009\)](#). A Figura 5 representa as classes necessárias para que o usuário consiga se conectar ao sistema.

Figura 5 – Diagrama de classe de login



Fonte: Elaborado pelos autores

A classe *MainApp* é responsável apenas para conseguir iniciar o programa, ela que dá início e chama as próximas classes.

A classe *LoginController* é responsável por chamar a criação, alteração ou exclusão das informações de login dos usuários e dados pessoais, utilizando outras classes para completar as estruturas.

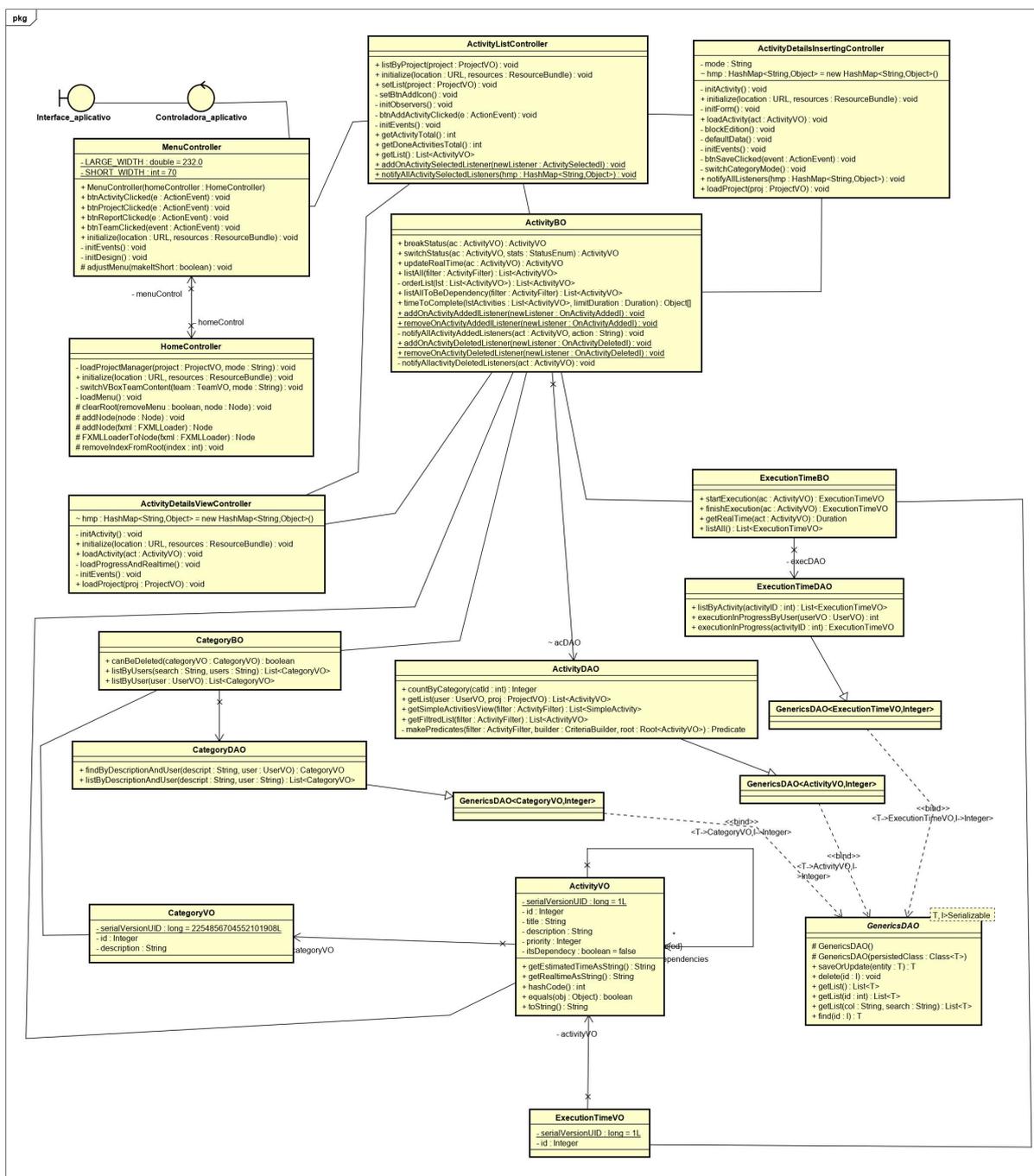
As classes *LoginBO* e *UserBO* são responsáveis por executar os métodos que estão nas classes *LoginDAO* e *UserDAO* que entram em contato com o banco de dados e então realizarão as operações.

As classes *LoginVO* e *UserVO* são uma representação do banco de dados no sistema, facilitando o uso do *Hibernate* para realizar as operações, assim como todas as classes do tipo VO.

A classe *GenericsDAO* é responsável por guardar as informações de acesso ao banco de dados, como o usuário e a senha criptografada, e ela é herdada por todas as classes do tipo DAO.

A Figura 6 representa as classes necessárias para que seja possível executar as atividades.

Figura 6 – Diagrama de classe de atividade



Fonte: Elaborado pelos autores

A classe *MenuController* é responsável por manter as telas, fazendo com que elas troquem de uma para a outra, mantendo o todo o programa em apenas uma tela sem abrir novas.

A classe *ActivityController* é responsável por cuidar da tela de atividades com os métodos que aciona as outras classes afim de realizar as tarefas que precisariam ser executadas.

A classe *ActivityDetailsInsertingController* é responsável por cuidar do cadastro e edição de uma atividade.

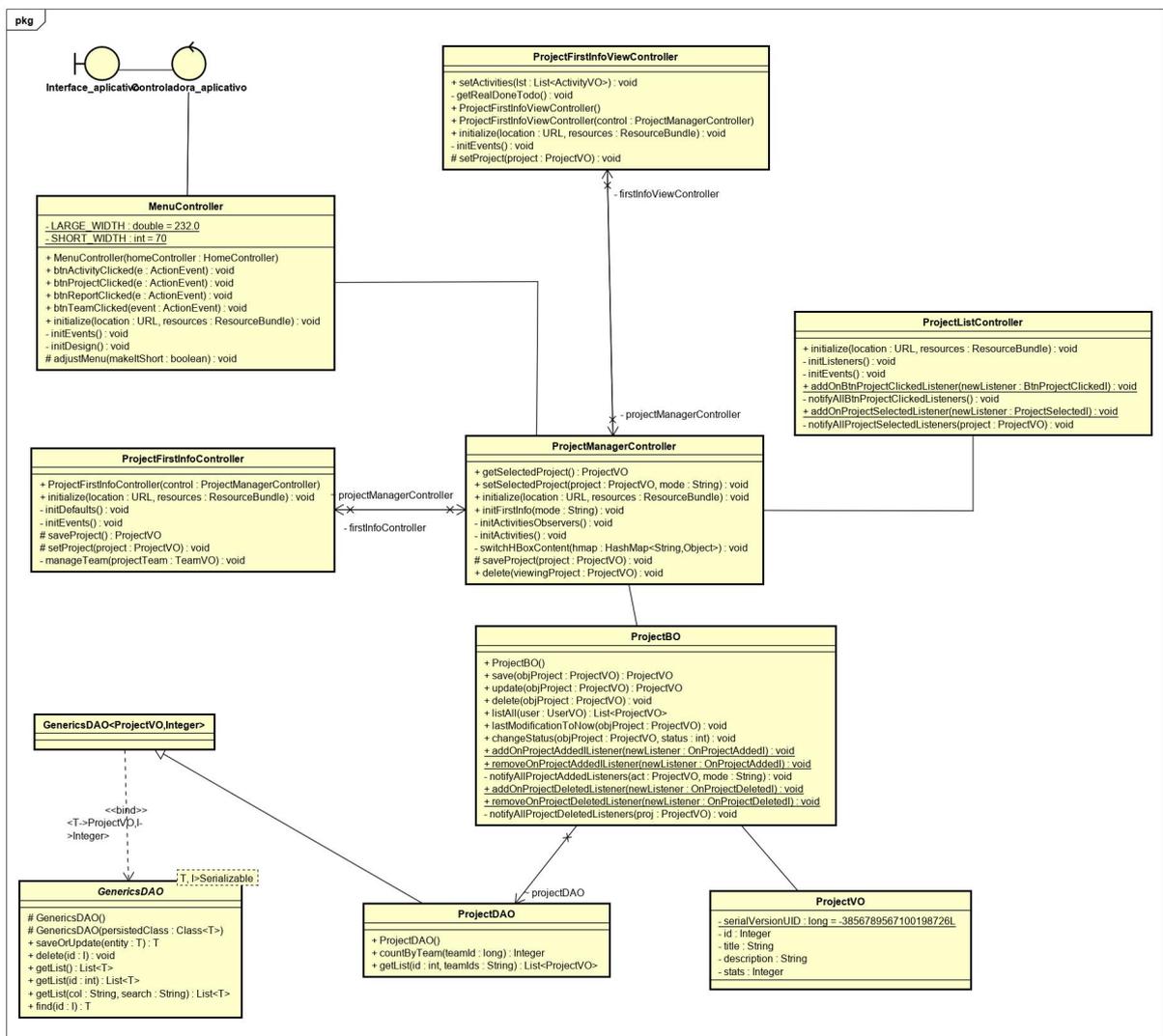
A classe *ActivityBO* é responsável por ser a ponte que realiza todas as tarefas que envolvem as operações básicas de atividades, como: criar, editar e excluir uma atividade.

As classes *ExecutionTimeBO* e *ExecutionTimeDAO* são responsáveis pelo andamento de uma atividade, conseguindo medir o tempo que a mesma já percorreu e recuperando o status em que ela se encontra (em andamento, pausada ou atrasada).

A classe *CategoryBO* é responsável por cuidar das categorias de cada atividade.

A Figura 7 representa todas as classes envolvidas com projeto.

Figura 7 – Diagrama de classe de projeto



Fonte: Elaborado pelos autores

A classe *ProjectManagerController* é responsável por cuidar da tela de projeto, todas as funções básicas são executadas por essa classe.

A classe *ProjectFirstInfoController* é responsável por apresentar os dados do projeto ao usuário assim que acessar a tela.

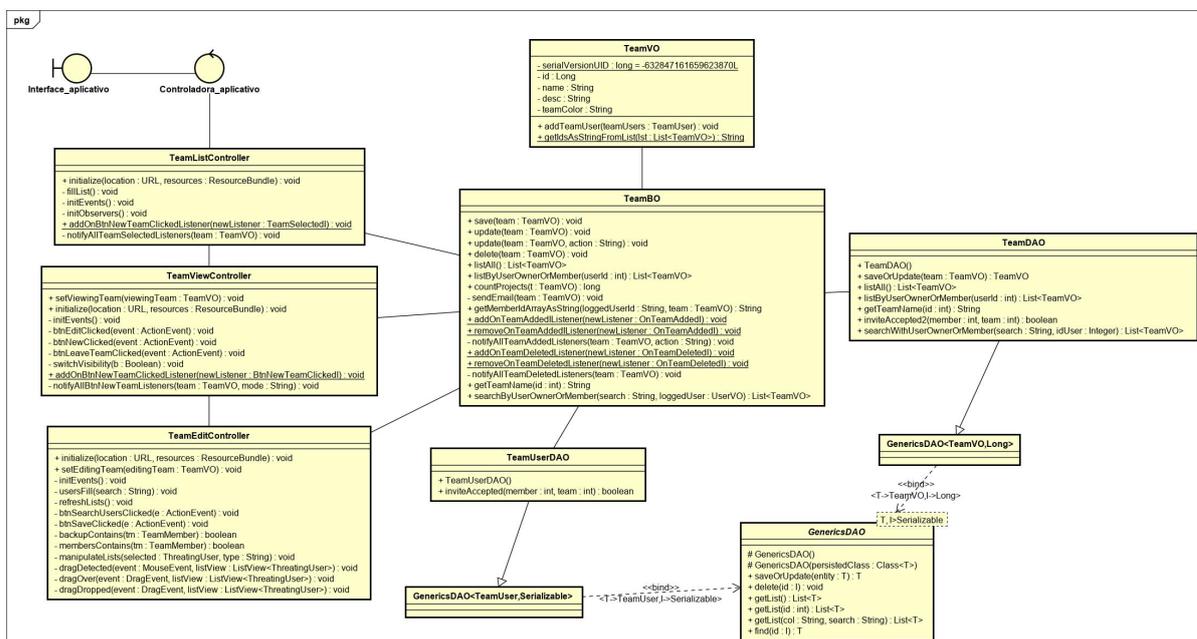
A classe *ProjectFirstInfoViewController* é responsável por cuidar da parte de edição das informações do projeto.

A classe *ProjectListController* é responsável por manter a lista de projetos que existe em cada usuário ou time.

A classe *ProjectBO* é responsável por todas as operações envolvendo os projetos.

A Figura 8 representa todas as classes que fazem parte da estrutura de time.

Figura 8 – Diagrama de classe de time



Fonte: Elaborado pelos autores

A classe *TeamListController* é responsável por mostrar todas os times que aquele usuários esta participando como membro ou líder, e também faz com que consiga criar um novo time.

A classe *TeamViewController* é responsável por mostrar os dados do time para os usuários e atualizar as informações quando necessárias.

A classe *TeamEditController* é responsável por deixar o usuário editar as informações e também como conseguir convidar mais membros para o time.

A classe *TeamBO* é responsável por cuidar de todas as operações que envolvam time.

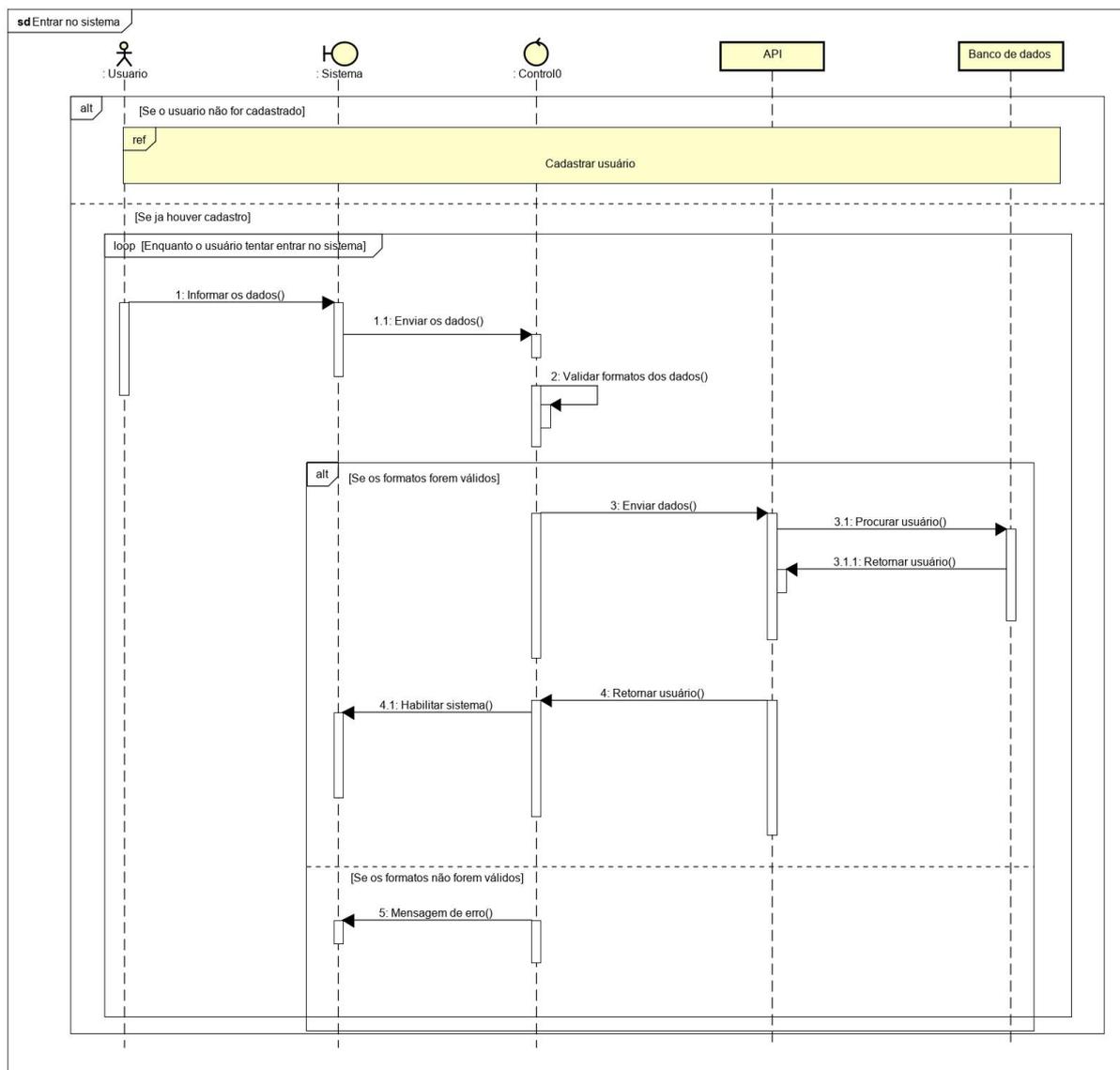
### 3.4 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência identifica o comportamento do sistema e que procura determinar a sequência dos eventos que acontece em cada um dos processos. Este diagrama baseia-se no caso de uso e que então cada função declarada acaba gerando uma nova sequencia a ser

documentada Guedes (2009).

A Figura 9 representa a sequencia de login

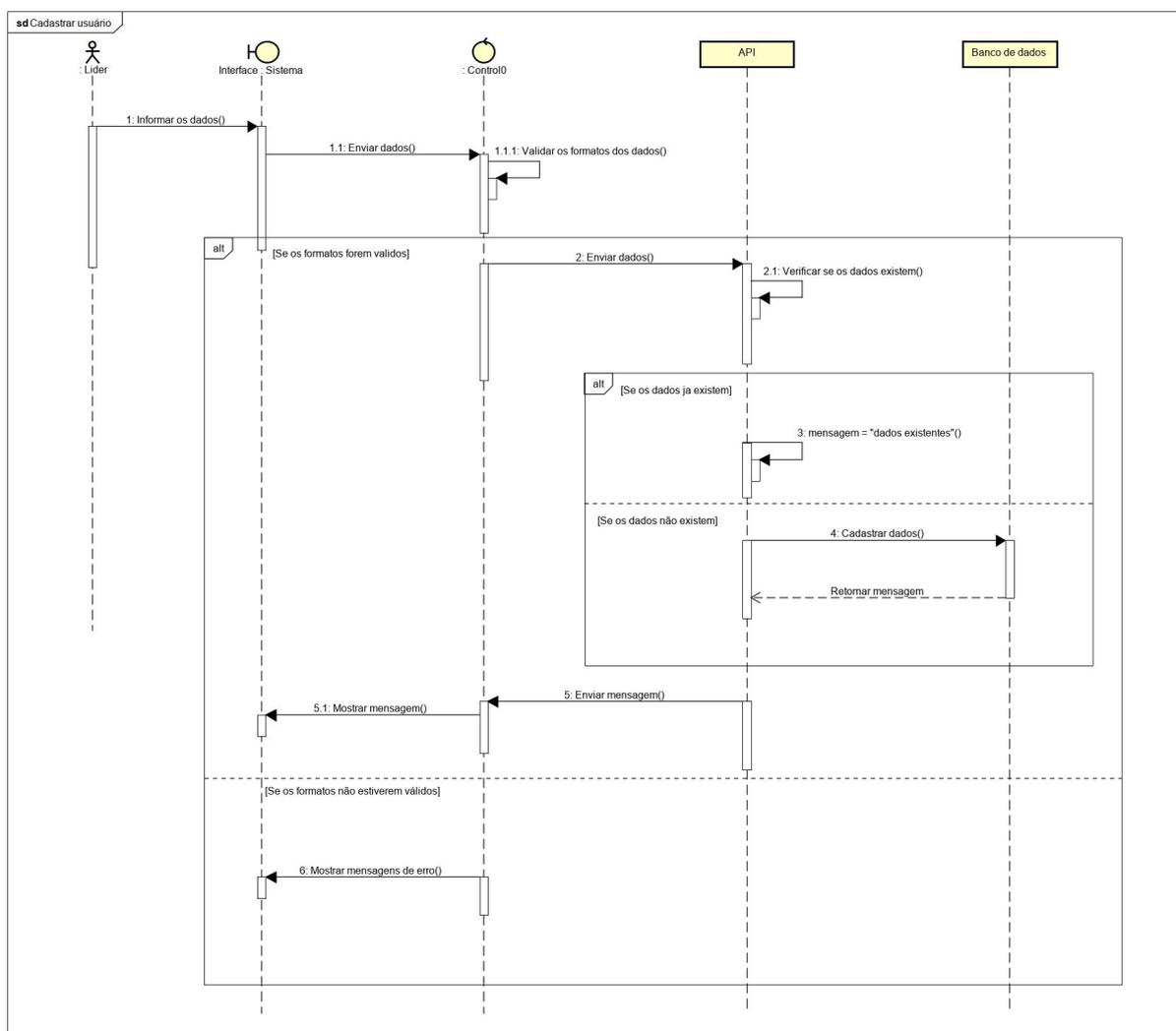
Figura 9 – Diagrama de sequência de login



Fonte: Elaborado pelos autores

A Figura 10 representa a sequencia de cadastro de usuário

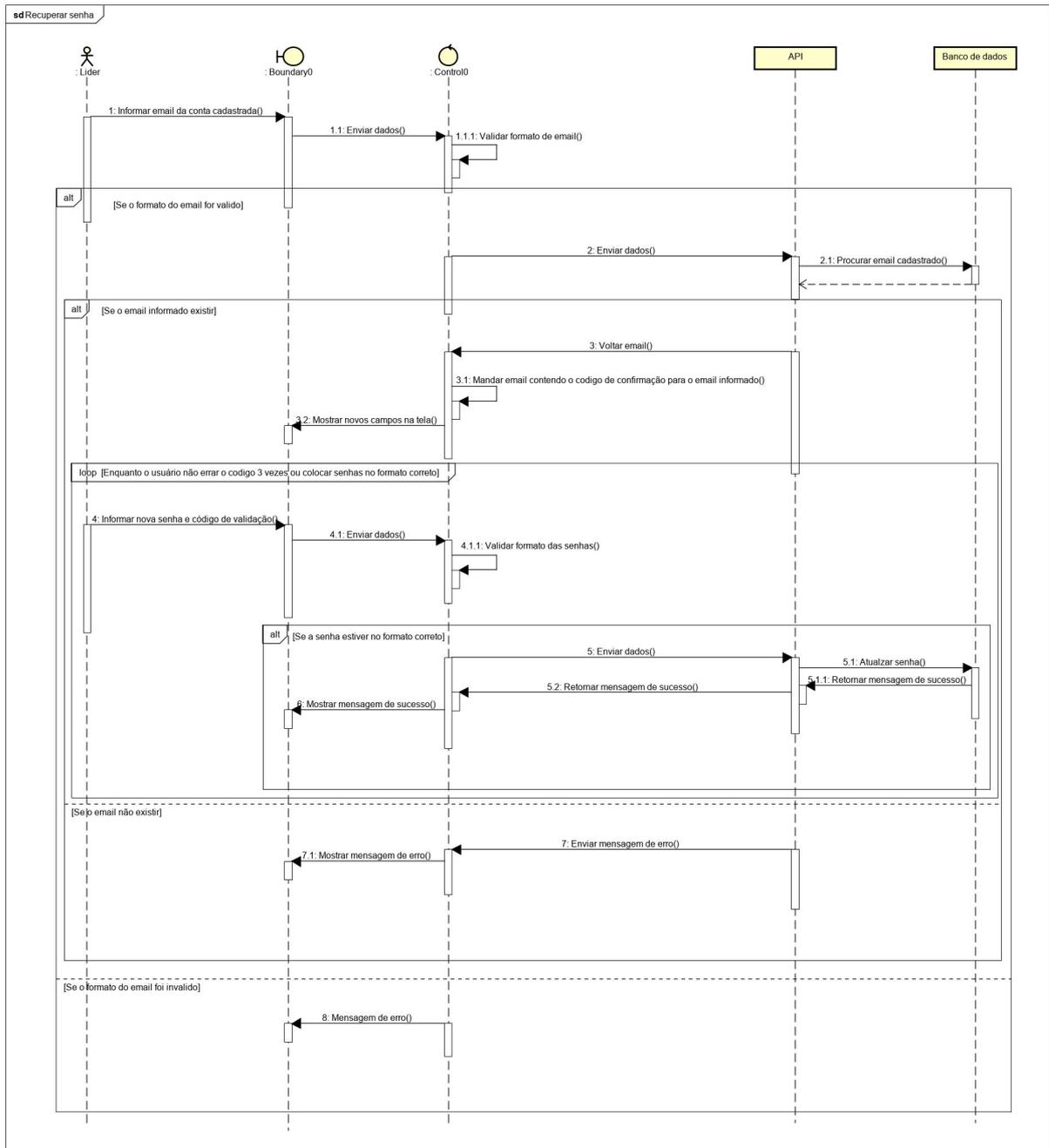
Figura 10 – Diagrama de sequência de cadastro de usuário



Fonte: Elaborado pelos autores

A Figura 11 representa a sequencia de recuperação de senha

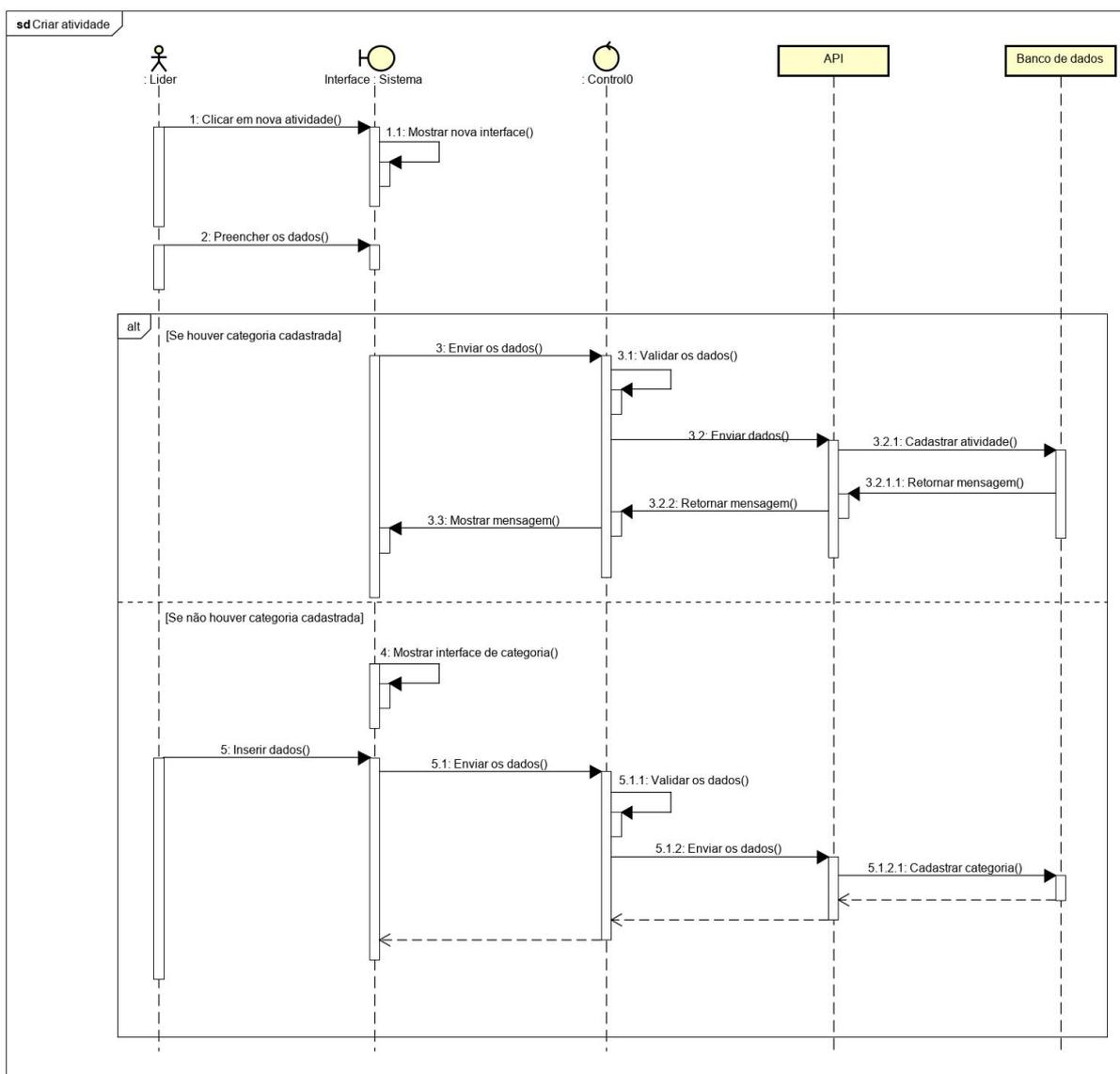
Figura 11 – Diagrama de sequência de recuperação de senha



Fonte: Elaborado pelos autores

A Figura 12 representa a sequencia de cadastrar atividade

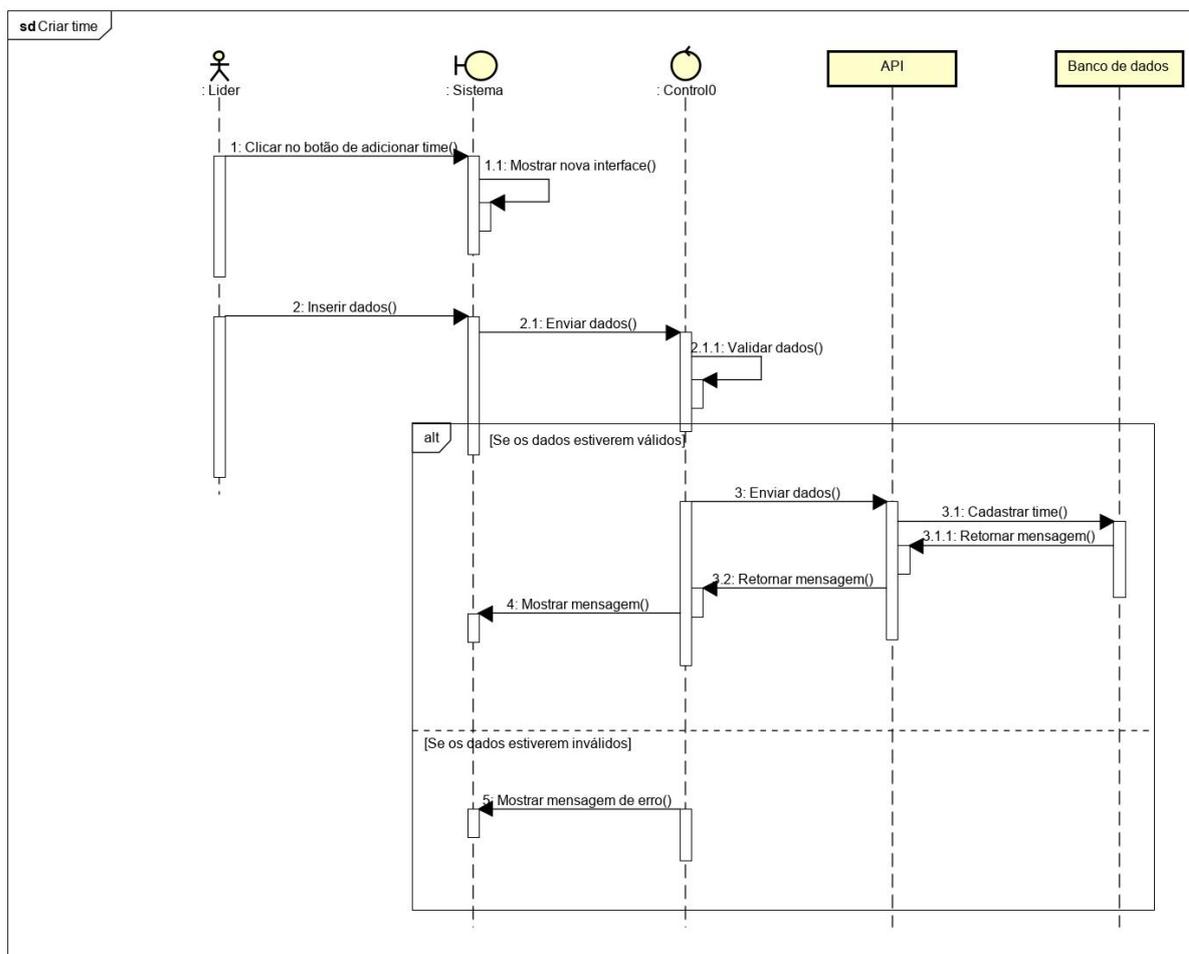
Figura 12 – Diagrama de sequência de cadastrar atividade



Fonte: Elaborado pelos autores

A Figura 13 representa a sequencia de criar time

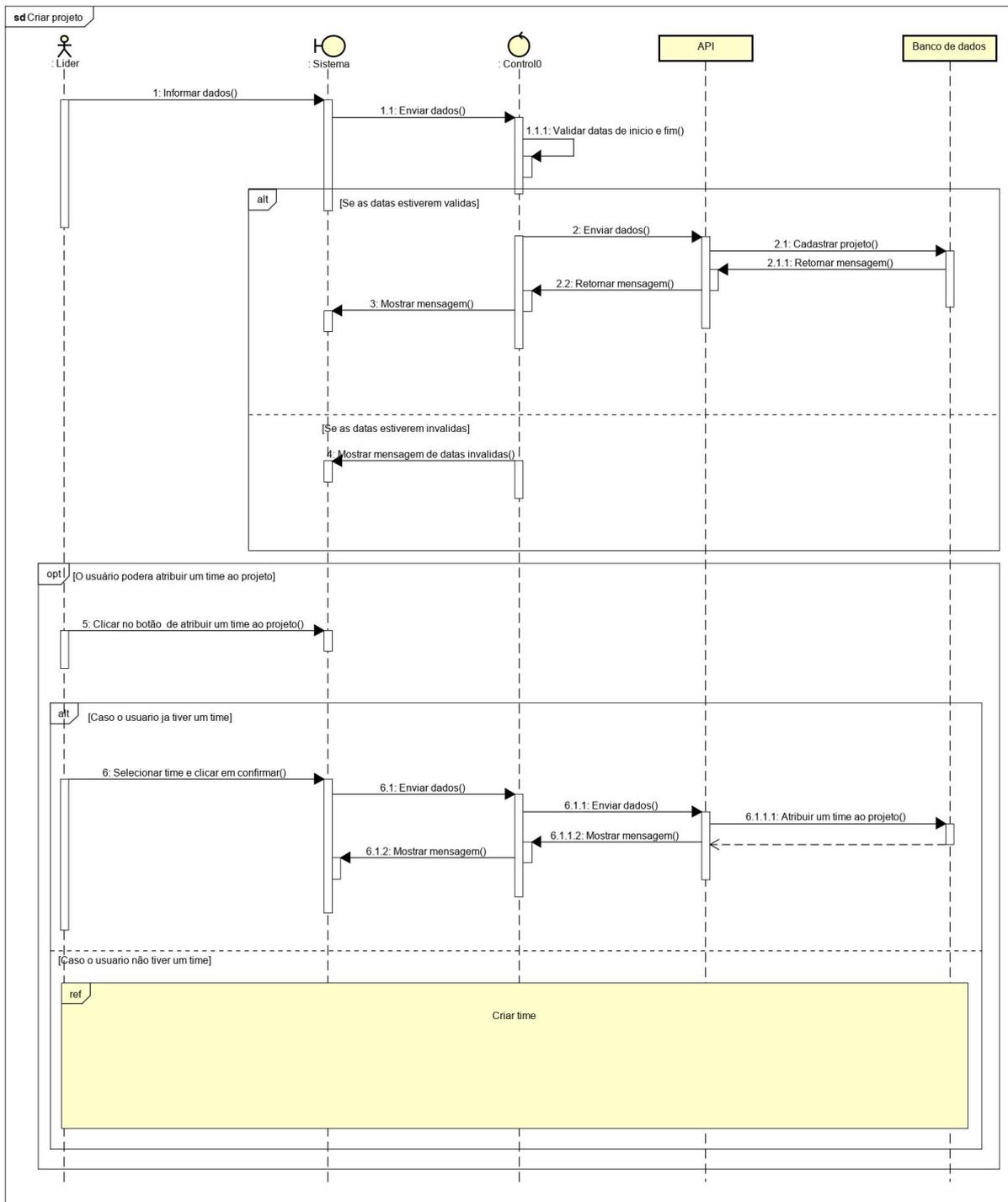
Figura 13 – Diagrama de sequência de criar time



Fonte: Elaborado pelos autores

A Figura 14 representa a sequencia de criar projeto

Figura 14 – Diagrama de sequência de criar projeto



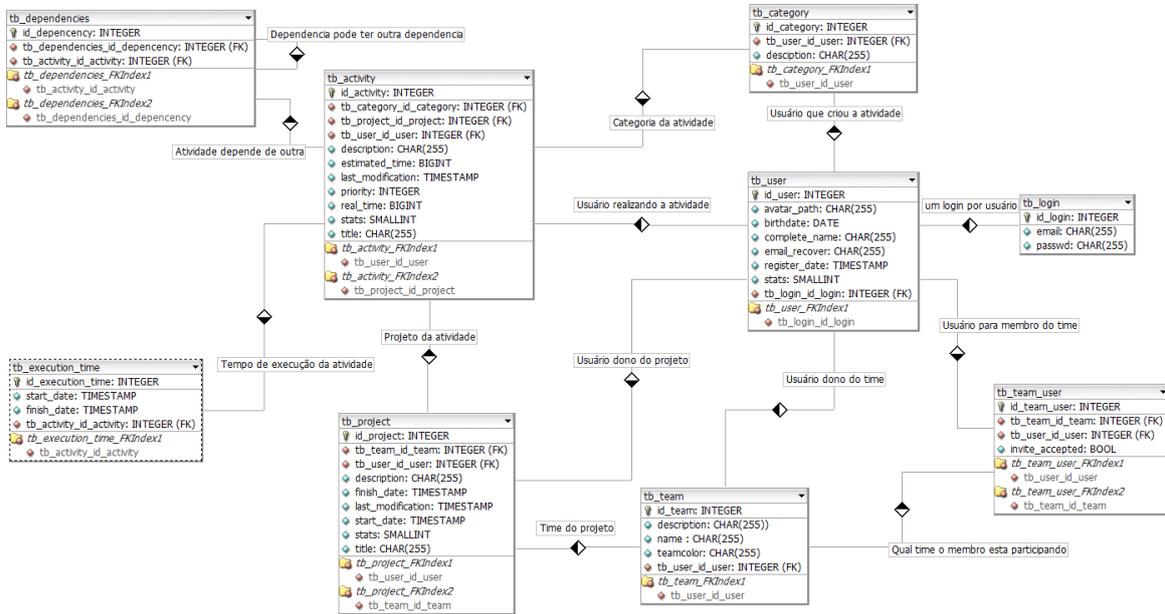
Fonte: Elaborado pelos autores

### 3.5 DIAGRAMA DE ENTIDADE RELACIONAMENTO

O diagrama de entidade relacionamento é uma representação gráfica do banco de dados do sistema, com ele, conseguimos uma visão dos relacionamento dos dados. Este tipo de diagrama só é possível ser realizado quando estamos tratando de bancos de dados relacionais.

A Figura 15 representa o diagrama de entidade relacionamento

Figura 15 – Diagrama de entidade relacionamento



Fonte: Elaborado pelos autores

## 4 DESENVOLVIMENTO

### 4.1 CRONOGRAMA DE DESENVOLVIMENTO

Este capítulo tem o propósito de apresentar o planejamento do projeto através da elaboração de um cronograma.

#### 4.1.1 PLANEJAMENTO

Foram estabelecidos marcos de início e conclusão das atividades, desenvolvendo uma cadeia sequencial e lógica. O principal objetivo era assegurar que as etapas fossem concluídas dentro do prazo definido.

A elaboração de um bom cronograma de atividades depende de importantes variáveis, a começar pela construção de um escopo bem traçado, passando pelo entendimento completo das fases requeridas para a conclusão da iniciativa. Devemos levar em conta que o projeto foi iniciado pelo desenvolvimento do sistema e não pela documentação.

Fizemos um mapeamento dos pontos de tensão da iniciativa, verificando os pontos que a equipe precisaria de atenção redobrada para não estourar os prazos e manter as entregas conforme os planos iniciais. Foram utilizadas ferramentas de gestão como Trello e Excel, afim de auxiliar a equipe medir seu próprio desempenho, encontrando alternativas ágeis e eficientes para o desenvolvimento de cada atividade.

#### 4.1.2 SPRINTS

Os sprints são ciclos de trabalho com uma duração média de 2 a 4 semanas, que são definidos de acordo com o Sprint Backlog. No início de cada Sprint, a equipe se reúne para priorizar as funcionalidades do projeto contidas no Product Backlog. De acordo com sua capacidade, o time define quais funcionalidades serão implementadas durante aquele ciclo. A estratégia de trabalho adotada pelo time de desenvolvimento do Cronote foi estabelecer um tempo médio de 4 horas que auxiliou na distribuição do tempo necessário às atividades (em dias).

Nas Sprints há 3 tipos de designações. Existe o Product Owner que fica responsável pela visão do produto; Scrum Master que tem o papel de orientador do time e a equipe de desenvolvimento que são responsáveis pela execução das atividades.

Uma vez na semana nos reuníamos brevemente para alinhar as atividades do projeto e identificar impedimentos para o trabalho. Essa reunião é feita a cada novo sprint para definir as atividades que serão executadas nesse ciclo. Levamos em conta o tempo de execução das atividades, a duração do ciclo e a prioridade de cada uma delas. Com tudo definido, criamos o Sprint Backlog, onde as tarefas a serem realizadas ficam visíveis para todos os membros da equipe.

A Tabela 13 representa a primeira *sprint*.

Tabela 13 – Sprint Planning: período do dia 23 de junho de 2018 a 7 de julho de 2018.

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Tela Login	1	1
Tela Login - view	2	2
LoginController	15	4
loginBO	2	2
loginVO	1	1
loginDAO	1	3
Tela cadastroPessoal	1	1
Tela cadastroPessoal - view	2	2
UserController	15	4
UserBO	2	2
UserVO	1	1
UserDAO	2	3
Tabela BD Login	1	1
Tabela BD User	1	1
Conexão BD	1	1

Fonte: Elaborado pelos autores.

A Tabela 14 representa a segunda *sprint*.

Tabela 14 – Sprint Planning: período do dia 08 de julho de 2018 a 21 de julho de 2018

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Classe Criptografia	1	4
Protótipo de tela Esqueceu a senha	2	2
Classe de geração aleatorio	15	1
View de tela Esqueceu a senha	2	2
Classe envio de email	1	5
Tela Login - View	2	2

Fonte: Elaborado pelos autores.

A Tabela 15 representa a terceira *sprint*.

Tabela 15 – Sprint Planning: período do dia 18 de agosto de 2018 a 01 de setembro de 2018.

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
EmailUtil	1	2
LoginController	2	2
ForgotPassword Controller	2	2
SignUpController	2	2
UserBO	2	2
UserVO	1	3
UserDAO	1	2
LoginBO	3	2
LoginVO	1	3
LogInDAO	1	3
Modificar Inserção de usuário; Criptografar Senha	1	1
Design Tela Login	2	2
Design Tela Cadastre-se	2	3
Design Tela Esqueceu a senha	2	2
ScreenUtil	1	2
GenericsDAO	2	4

Fonte: Elaborado pelos autores.

A Tabela 16 representa a quarta *sprint*.

Tabela 16 – Sprint Planning: período do dia 22 de setembro de 2018 a 6 de outubro de 2018

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Projetar tabelas necessárias	7	7
Tabela tempo de execução	1	1
Tabela de projeto	1	1
tabela de atividade	1	1
tabela de categoria	1	1
Atividade VO	1	1
Atividade BO	2	3
Atividade DAO	1	0.5

Projeto VO	1	2
Projeto BO	2	3
Projeto DAO	1	0.5
Categoria VO	1	2
Categoria BO	2	3
Categoria DAO	1	0.5
Prototipagem da tela home(atividades)	1	3
Ajustes na tela "Esqueceu senha"	3	3
View da tela home(atividades)	3	6
Controller da tela home(atividades)	5	10
Fluxo de login do facebook	3	5

Fonte: Elaborado pelos autores.

A Tabela 17 representa a quinta *sprint*.

Tabela 17 – Sprint Planning: período do dia 25 de Fevereiro de 2019 a 17 de março de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Refazer/reorganizar o excel de gerenciamento de sprint	3	2
Correções Funcionais	2	2
Correções de layout/Design	1	1
Home (Perspectiva de tarefas) View (atrasada)	5	5
Home controller (atrasada )	8	10
Fluxo de login do facebook (atrasada)	4	3
Fluxo de cadastro com facebook	4	4

Fonte: Elaborado pelos autores.

A Tabela 18 representa a sexta *sprint*.

Tabela 18 – Sprint Planning: período do dia 18 de março de 2019 a 07 de abril de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Gerar senha de app para o email	1	1
Incrementar as configurações do hibernate	1	1

Envio de email de forma assincrona	1	2
Explorar possibilidades de rootPane	3	3
Sidebar short	1	1
Sidebar large	1	2
Template Lista de Atividades	2	3
Validação para Apenas uma atividade em execução	1	2
Função de pausa de atividade	1	2
Função de inicio de atividade	1	2
Template Detalhes da Atividades (insert/consult/update)	3	4
Adicionar validações inline nos formulários já existentes	4	4
Adicionar botoes Trash   Edit   Save	1	1
Integração entre o back e front end	4	6
Preparar apresentação da sprint	2	2

Fonte: Elaborado pelos autores.

A Tabela 19 representa a sétima *sprint*.

Tabela 19 – Sprint Planning: período do dia 8 de abril de 2019 a 28 de abril de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Protótipo manter categoria	1	2
Adição dos botões Play/Pause/Stop	1	1
Desenvolvimento da tela manter categoria	2	3
Desenvolver progresso da atividade	1	2
Ajustes nos indicadores de progresso	2	1
Adicionar botão "+" em categorias	1	1
Campo switch em categoria	1	2
Adicionar contador de caracteres	1	1
Alterar forma de tempo estimado	1	1
Tela de projetos protótipo	5	4
Tela de projetos front-end	6	4
Tela de projetos back-end	6	4
Gerenciamento de dependências	3	3
REST- autenticação	7	4
REST- cadastro de usuário	4	5
Separar model em outro projeto	1	3

Aplicar efeito hover no menu	1	3
Ajustar o tamanho dos ícons	1	1
Adição dos botões Play/Pause/Stop	2	3
Preparar apresentação	2	1

Fonte: Elaborado pelos autores.

A Tabela 20 representa a oitava *sprint*.

Tabela 20 – Sprint Planning: período do dia 29 de abril de 2019 a 19 de maio de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Definir paleta de cores para repaginação do projeto	2	2
Aplicar nova paleta no sistema nos prototipos	1	2
Atualizar e sincronizar os prototipos no trello e repositório	1	2
Reformular o manipulador de projeto prototipo	3	3
Reformular o manipulador de projeto Front end	1	2
Reformular o manipulador de projeto back end	2	3
Trocar autenticação de acesso(login) por post	1	2
Trocar cadastro de usuario por post	1	2
Estender o REST cadastro de atividades	2	3
Estender o REST cadastro de categorias	2	3
Estender o REST cadastro de Projetos	2	3
Estender o REST manipulação de execução de atividades	2	2
Configurar e disponibilizar na nuvem	2	3
Revisar arquivos desnecessários e de backup	1	1
Reorganização de codigos	1	2
Consertar o dialog de gerenciar dependencias	1	1
Desenvolver o dialog de gerenciar dependencias Front end	1	2
Desenvolver o dialog de gerenciar dependencias Back end	1	2
Trocar Campo prioridade para rateInput(stars)	1	1
Paper layout Cadastro de time	1	4
Cadastro de time prototipo	1	2
Cadastro de time front end	1	2
Cadastro de time back end	1	2
Switch de conta ativa + logout + config e ajuda(icone)	1	4
Adicionar logout no menu	2	2

Preparar a apresentação	2	2
-------------------------	---	---

Fonte: Elaborado pelos autores.

A Tabela 21 representa a nona *sprint*.

Tabela 21 – Sprint Planning: período do dia 27 de maio de 2019 a 16 de junho de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Reformular manipulador de projeto	3	5
Estender o rest ao funcionamento	8	11
Configurar o servidor	2	3
Cadastro de time front-end e back-end	2	4
Correção de erros	4	9
Criação das validações	5	11

Fonte: Elaborado pelos autores.

A Tabela 22 representa a décima *sprint*.

Tabela 22 – Sprint Planning: período do dia 22 de junho de 2019 a 5 de julho de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Correção: Erro na troca de senha	1	2
Cadastro de time back end	2	3
Reformular o manipulador de projeto Front end	1	2
Reformular o manipulador de projeto back end	2	4
Correção: Front-end manipulador de projetos	2	4
Validação: Impedir usuário de deletar atividade	1	2
Listagem de atividades: Ordenação	2	4
Cadastro de times - API	1	1
Listagem de times - API	1	2
Criar relação entre projeto e time	1	1
Criar Checkbox "Lembrar de mim" na tela de login	1	1
Fazer destaque para projeto de time - Protótipo	1	1
Fazer destaque para projeto de time - Front-end	2	1

Card de time - protótipo	1	1
Card de time - front-end	2	1
listagem de time - Back-end	1	4
Criar chamada genérica de envio de email - API	2	2
Criar Model de E-mail	1	1
Função de aceitar convite para ingressar no time - API	1	3
Enviar email de convite para time	2	2

Fonte: Elaborado pelos autores.

A Tabela 23 representa a décima primeira *sprint*.

Tabela 23 – Sprint Planning: período do dia 6 de julho de 2019 a 19 de julho de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Cadastro de time back end	2	3
Reformular o manipulador de projeto front-end	1	2
Reformular o manipulador de projeto back-end	2	4
Correção no front-end do manipulador	2	4
Impedir que usuário que delete dependências	1	2
Ordernar listagem de atividades	2	4
Criar relação entre projeto e time	1	1
Criar checkbox na tela de login	1	1
Fazer de destaque para projeto de time - protótipo	1	1
Fazer de destaque para projeto de time front-end	2	1
Card de time - protótipo	1	1
Card de time front-end	1	1
Listagem de time back-end	1	4
Criar email generico para envios - Chamada de API	1	2
Função de aceitar convite de time - Chamda de API	1	3
Enviar email de convite para time	2	2

Fonte: Elaborado pelos autores.

A tabela 24 representa a décima segunda *sprint*.

Tabela 24 – Sprint Planning: período do dia 20 de julho de 2019 a 02 de agosto de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Reformular o manipulador de projeto Front end	1	2
Reformular o manipulador de projeto back end	2	1
Criar relação entre projeto e time	1	4
Impedir execução que possuam dependencias não executam	1	2
Fazer destaque para projeto de time - Front-end	2	1
Criar chamada genérica de envio de email - API	1	2
Função de aceitar convite para ingressar no time - API	1	3
Enviar email de convite para time	2	2
Validação: Impedir usuário de deletar atividade	1	2
Corrigir a geração de JSON: Loop infinito	1	2
Corrigir a geração de JSON: JSON's gigantes	5	5
Trocar função connection da API p	5	5
Validação: Uma atividade não pode depender de outra	2	2
Pesquisar possibilidade de fazer os cards	2	4
Placeholder nas listas	3	5
Ajustar a tela home para usar o jfxdialog	2	3

Fonte: Elaborado pelos autores.

A Tabela 25 representa a décima terceira *sprint*.

Tabela 25 – Sprint Planning: período do dia 3 de agosto de 2019 a 16 de agosto de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Fazer destaque para projeto de time front-end	2	1
Criar email generico para envios - Chamada de API	1	2
Função de aceitar convite de time	1	3
Enviar email de convite para time	2	1
Impedir que usuário que delete dependências	1	2
Trocar função de connection da api	3	2
Placeholder nas listas	1	2
Ajustar a tela home para usar jfxdialog com stackpane	2	3
Ajustes na tela de dependências	1	2
Adicionar campo para atribuir um time ao projeto	1	3

Diferenciação entre níveis de hierarquia dos membros	1	2
------------------------------------------------------	---	---

Fonte: Elaborado pelos autores.

A Tabela 26 representa a décima quarta *sprint*.

Tabela 26 – Sprint Planning: período do dia 17 de agosto de 2019 a 30 de agosto de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Função de aceitar convite para ingressar no time - API	2	3
Enviar email de convite para time	2	1
Esquema de redundância de membros de um time	1	2
Adicionar dialog para o tratamento de iniciar	1	2
Validação de código enviado por email	1	1
Erros de confirmação (contador)	2	4
Trocar snackbar por dialogs	1	4
Para voltar usando alt+<	1	2
Opção de arquivar e futuramente deletar atividades	2	4
Atribuição de time ao projeto	1	4
Esquema de permissões de membros do time	1	2
Diferenciação entre níveis de hierarquia dos membros	2	4
Definir um quarto tom pro painel de atividades	1	4
Ajuste de hierarquia dos cabeçalhos	1	1
Mudar o tom do roxo dos texto em destaque	2	4
Mudar o texto "LOGIN" para "ENTRAR"	1	3
esqueceu senha, pode ser "Recuperação de senha	1	1

Fonte: Elaborado pelos autores.

A Tabela 27 representa a décima quinta *sprint*.

Tabela 27 – Sprint Planning: período do dia 31 de agosto de 2019 a 13 de setembro de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Esquema de redundância de envio de email aos membros de um time	4	3
Atribuição de time ao projeto	1	2
Diferenciação entre níveis de hierarquia - Implementação	2	4
Definir quarto tom pro painel de atividades em projeto	3	1
Hierarquia do cabeçalho	2	4
Mudar tom de roxo dos textos em destaque	1	3
Título da tela de "Esqueceu a senha"	1	1

Fonte: Elaborado pelos autores.

A Tabela 28 representa a décima sexta *sprint*.

Tabela 28 – Sprint Planning: período do dia 14 de setembro de 2019 a 27 de setembro de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Cabeçalho e todas as telas do sistema	3	2
Atribuição de time ao projeto	3	2
Expirar invite de time	2	3
Protótipo card de memb. de time["convite enviado", "membro"]	2	2
Atualizar cabeçalho e todas as telas do sistema	4	2
Texto no card de memb. de time["convite enviado", "membro"]	2	1
Mudar o tom do roxo dos textos em destaques	2	3

Fonte: Elaborado pelos autores.

A Tabela 29 representa a décima sétima *sprint*.

Tabela 29 – Sprint Planning: período do dia 30 de agosto de 2019 a 13 de setembro de 2019

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Revisão da introdução	3	2
Revisão da tabela comparativa dos concorrentes	1	2
Documentação: Tutorial	6	3
Desenvolver os protótipos já feitos de outras telas	3	2
Ordenar as atividades	3	2
Recolocar sistema no heroku	2	4
Listagem de categorias	1	4
Assumir atividade	2	2
Atribuição de time ao projeto	3	2
Lista de atividades dentro do projeto	1	1
Estutura de auditoria	2	3

Fonte: Elaborado pelos autores.

A Tabela 30 representa a décima oitava *sprint*.

Tabela 30 – Sprint Planning: período do dia 30 setembro de 2019 a 13 de novembro de 2019.

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Documentacao : Revisão da introdução	2	2
Documentacao : Tabela comparativa dos concorrentes	3	2
Documentacao: Tutorial	5	3
Preparar as telas "Configuracoes"e "Ajuda"[Protótipo]	4	2
Documentacao: Diagramas	6	3
Auditoria: Estrutura de banco de dados dados	3	3

Fonte: Elaborado pelos autores.

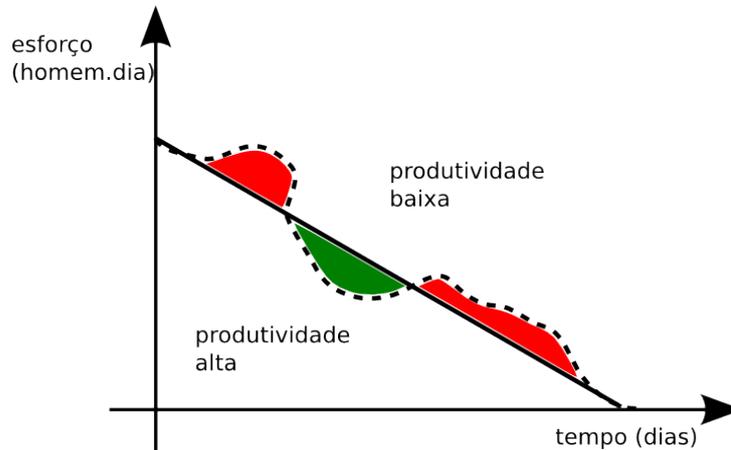
#### 4.1.3 BURNDOWN

O Gráfico de Burndown é um estilo de gráfico que explicita, no eixo vertical, o somatório de pontos de esforço necessários para realizar as atividades da iteração atual e, no eixo horizontal, os dias de trabalho da iteração. Idealmente, o gráfico deve apresentar uma reta que liga o topo do eixo vertical (sendo o total de pontos) ao limite do eixo horizontal (último dia de trabalho da iteração), uma vez que, caso a equipe seja mantida em trabalho durante toda a iteração, a produtividade deve se manter constante diariamente. Normalmente, porém, o gráfico apresenta uma curva ora crescente, ora decrescente. A curva pode crescer quando atividades não-planejadas são adicionadas a uma iteração.

O Gráfico de Burndown facilita o acompanhamento da produtividade da equipe durante a execução das atividades. Se um desvio acima da reta ideal ocorre, significa que a equipe está produzindo menos do que deveria e que, por isso, pode não acabar as tarefas designadas para a execução até o seu final. Por outro lado, se o desvio ocorre abaixo da reta ideal, a equipe está produzindo a mais do que o previsto, o que pode significar uma subestimação da produtividade da equipe ou uma superestimação de complexidade das tarefas designadas para a execução. [Campos \(2019\)](#)

A Figura 16 apresenta um exemplo de *burndown*

Figura 16 – Gráfico de Burndown

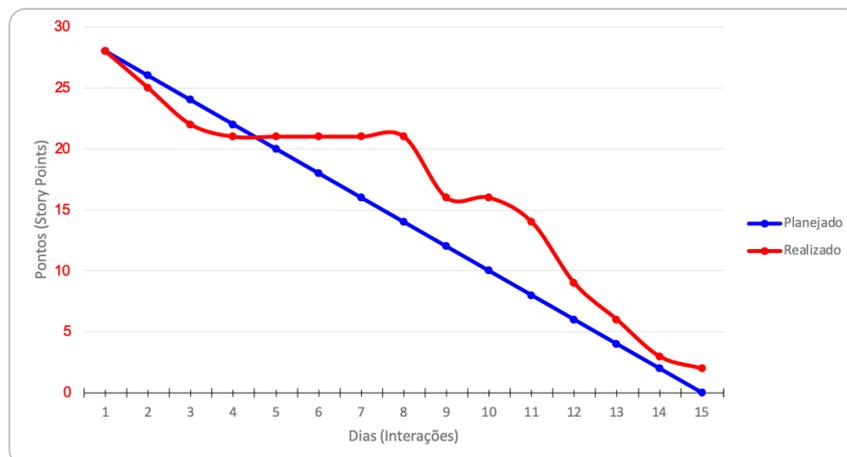


Fonte: Disponível em [http://demoiselle.sourceforge.net/process/ds/1.2.3-BETA1/ProcessoDemoisellePlugin/guidances/concepts/graficoBurndown\\_B9AC47B7.html](http://demoiselle.sourceforge.net/process/ds/1.2.3-BETA1/ProcessoDemoisellePlugin/guidances/concepts/graficoBurndown_B9AC47B7.html)

A equipe adotou o uso do gráfico de *burndown* pois revela não apenas se o prazo foi cumprido ou não, mas principalmente como a equipe lidou com o fluxo de atividades durante a *sprint*.

A Figura 17 representa o *burndown* da primeira semana de planejamento e desenvolvimento.

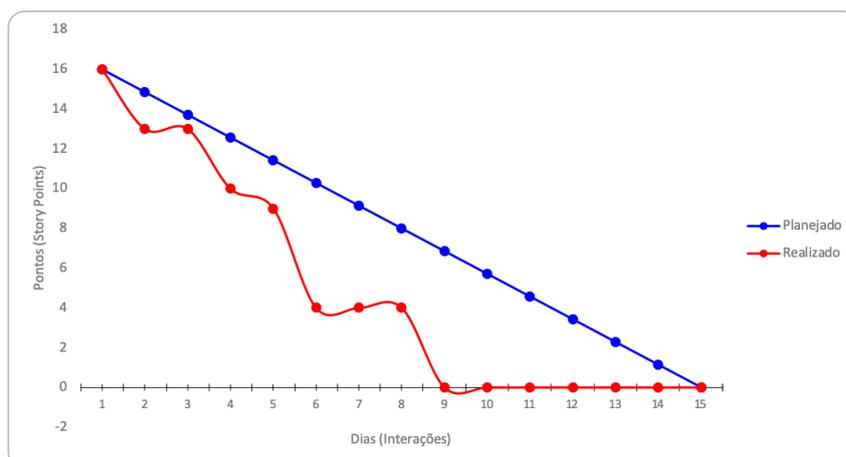
Figura 17 – Gráfico de Burndown - período do dia 23 de junho de 2018 a 7 de julho de 2018



Fonte: Elaborado pelos autores

A Figura 18 representa o *burndown* da segunda semana de planejamento e desenvolvimento.

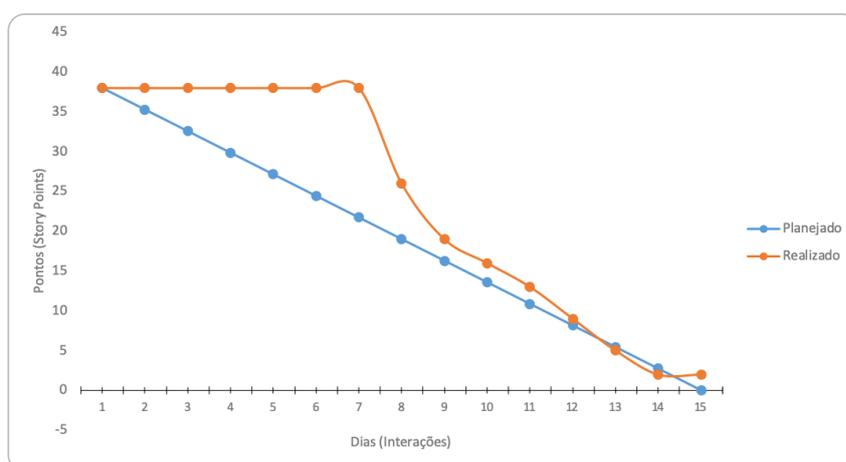
Figura 18 – Gráfico de Burndown - período do dia 08 de julho de 2018 a 21 de julho de 2018



Fonte: Elaborado pelos autores

A Figura 19 representa o *burndown* da terceira semana de planejamento e desenvolvimento.

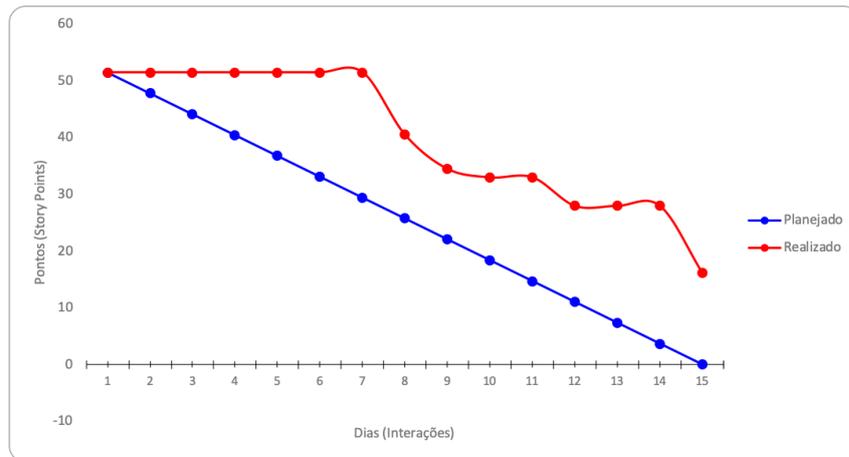
Figura 19 – Gráfico de Burndown - período do dia 18 de agosto de 2018 a 01 de setembro de 2018



Fonte: Elaborado pelos autores

A Figura 20 representa o *burndown* da quarta semana de planejamento e desenvolvimento.

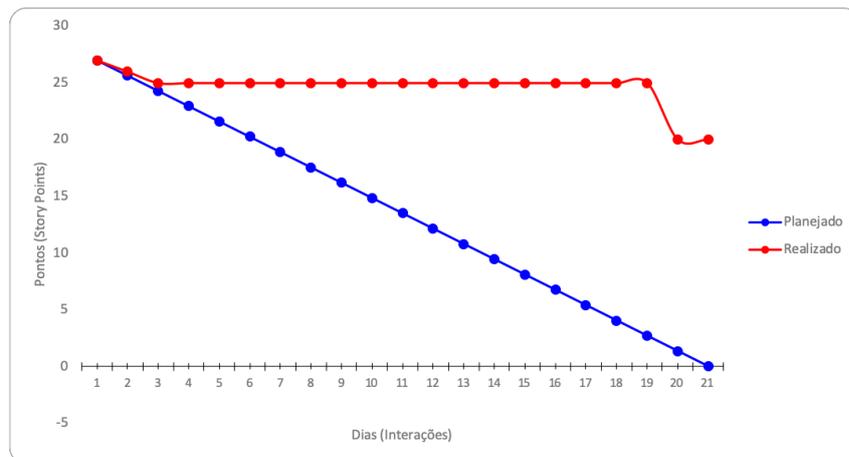
Figura 20 – Gráfico de Burndown - período do dia 22 de setembro de 2018 a 06 de outubro de 2018



Fonte: Elaborado pelos autores

A Figura 21 representa o *burndown* da quinta semana de planejamento e desenvolvimento.

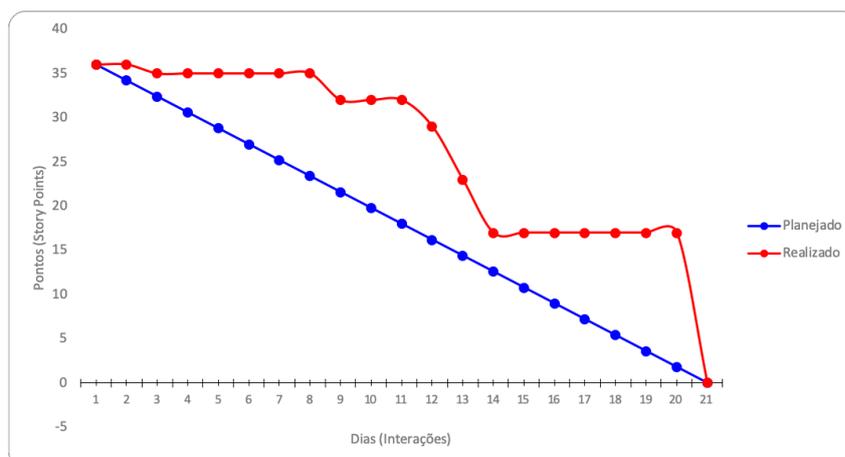
Figura 21 – Gráfico de Burndown - período do dia 25 de fevereiro de 2019 a 17 de março de 2019



Fonte: Elaborado pelos autores

A Figura 22 representa o *burndown* da sexta semana de planejamento e desenvolvimento.

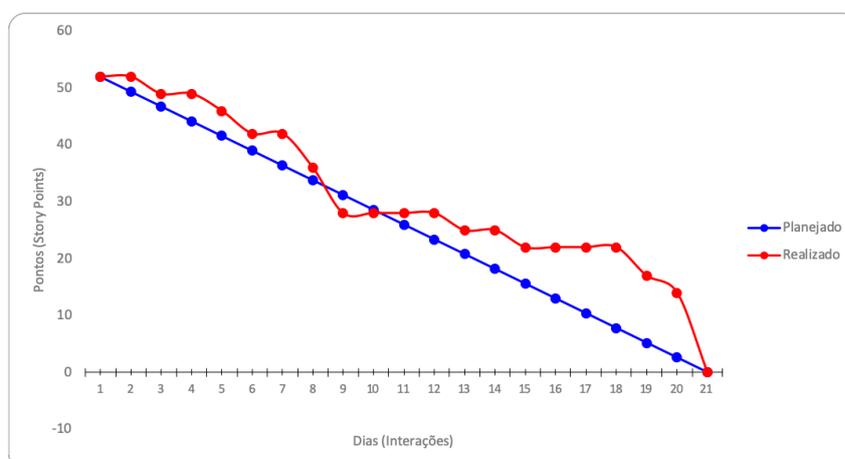
Figura 22 – Gráfico de Burndown - período do dia 18 de março de 2019 a 07 de abril de 2019



Fonte: Elaborado pelos autores

A Figura 23 representa o *burndown* da sétima semana de planejamento e desenvolvimento.

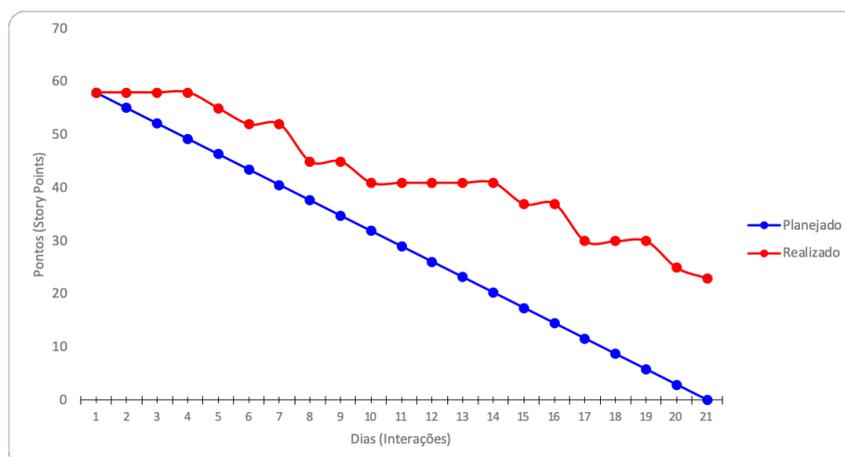
Figura 23 – Gráfico de Burndown - período do dia 08 de abril de 2019 a 28 de abril de 2019



Fonte: Elaborado pelos autores

A Figura 24 representa o *burndown* da oitava semana de planejamento e desenvolvimento.

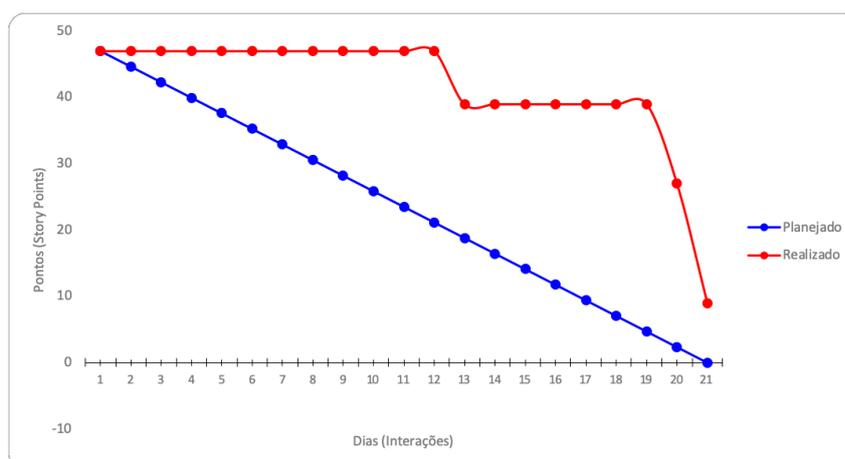
Figura 24 – Gráfico de Burndown - período do dia 29 de abril de 2019 a 19 de maio de 2019



Fonte: Elaborado pelos autores

A Figura 25 representa o *burndown* da nona semana de planejamento e desenvolvimento.

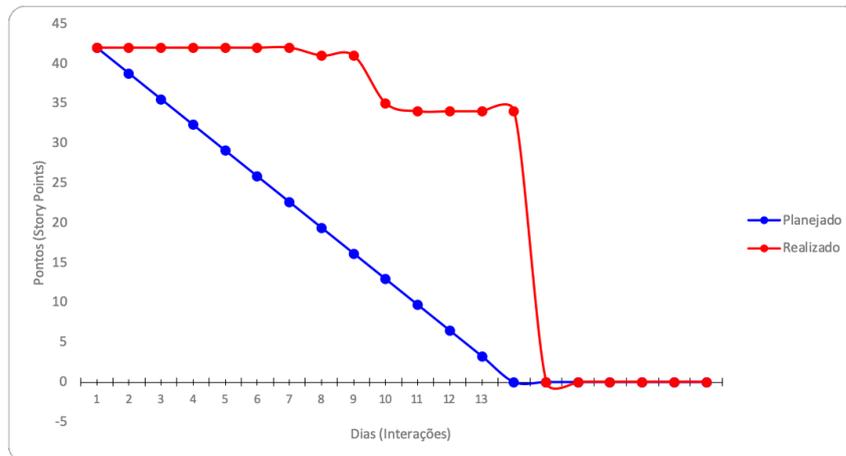
Figura 25 – Gráfico de Burndown - período do dia 27 de maio de 2019 a 16 de junho de 2019



Fonte: Elaborado pelos autores

A Figura 26 representa o *burndown* da décima semana de planejamento e desenvolvimento.

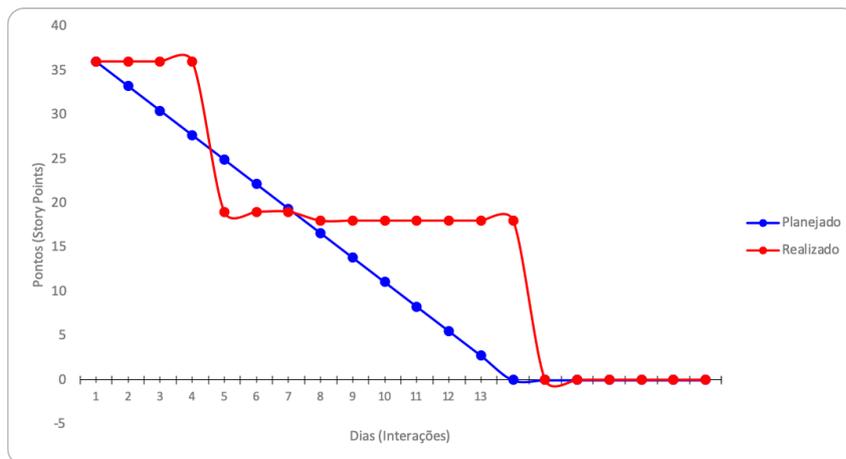
Figura 26 – Gráfico de Burndown - período do dia 22 de junho de 2019 a 05 de julho de 2019



Fonte: Elaborado pelos autores

A Figura 27 representa o *burndown* da décima primeira semana de planejamento e desenvolvimento.

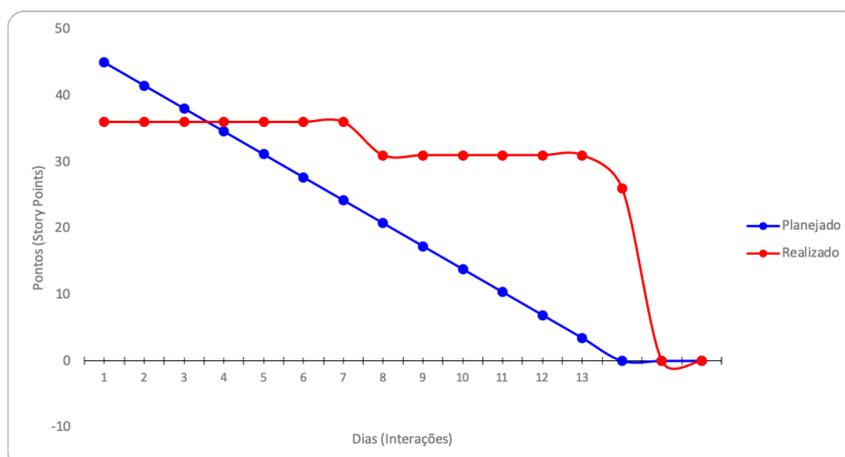
Figura 27 – Gráfico de Burndown - período do dia 06 de julho de 2019 a 19 de julho de 2019



Fonte: Elaborado pelos autores

A Figura 28 representa o *burndown* da décima segunda semana de planejamento e desenvolvimento.

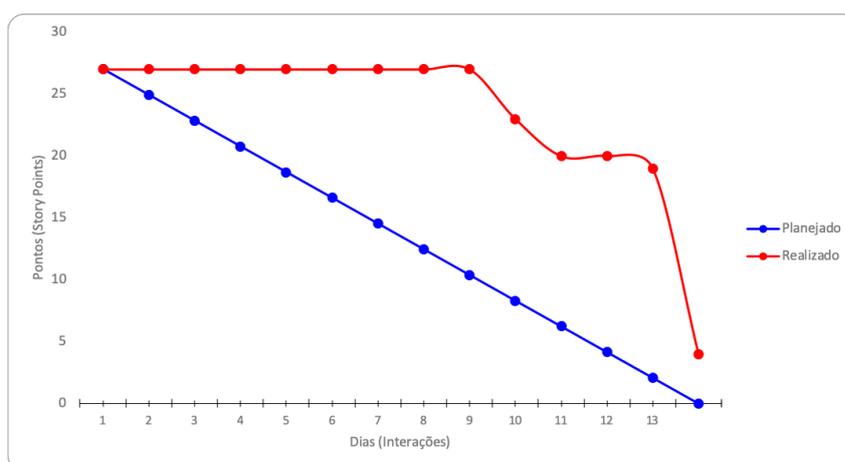
Figura 28 – Gráfico de Burndown - período do dia 20 de julho de 2019 a 02 de agosto de 2019



Fonte: Elaborado pelos autores

A Figura 29 representa o *burndown* da décima terceira semana de planejamento e desenvolvimento.

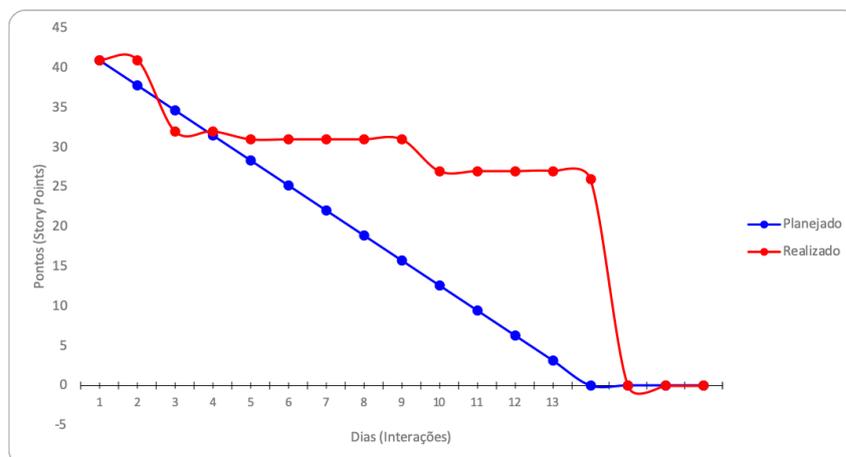
Figura 29 – Gráfico de Burndown - período do dia 03 de agosto de 2019 a 16 de agosto de 2019



Fonte: Elaborado pelos autores

A Figura 30 representa o *burndown* da décima quarta semana de planejamento e desenvolvimento.

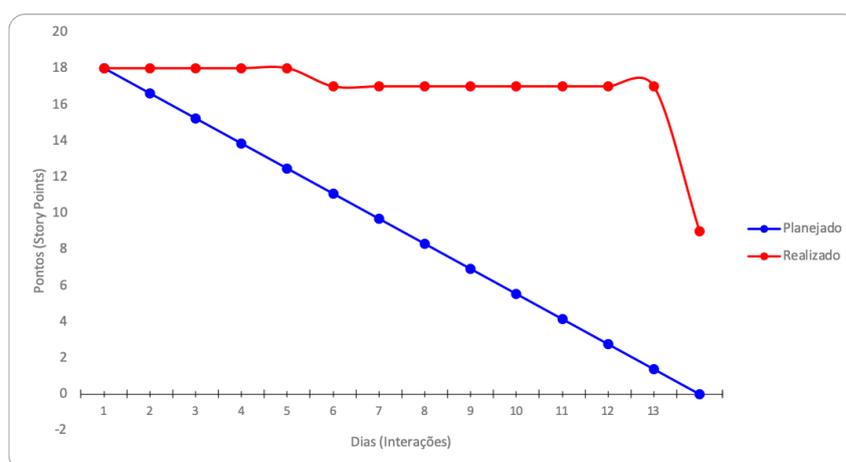
Figura 30 – Gráfico de Burndown - período do dia 17 de agosto de 2019 a 30 de agosto de 2019



Fonte: Elaborado pelos autores

A Figura 31 representa o *burndown* da décima quinta semana de planejamento e desenvolvimento.

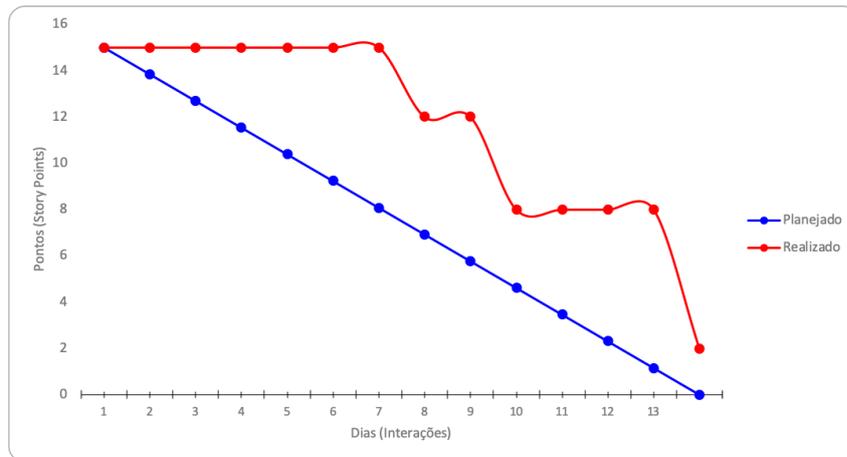
Figura 31 – Gráfico de Burndown - dia 31 de agosto de 2019 a 13 de setembro de 2019



Fonte: Elaborado pelos autores

A Figura 32 representa o *burndown* da décima sexta semana de planejamento e desenvolvimento.

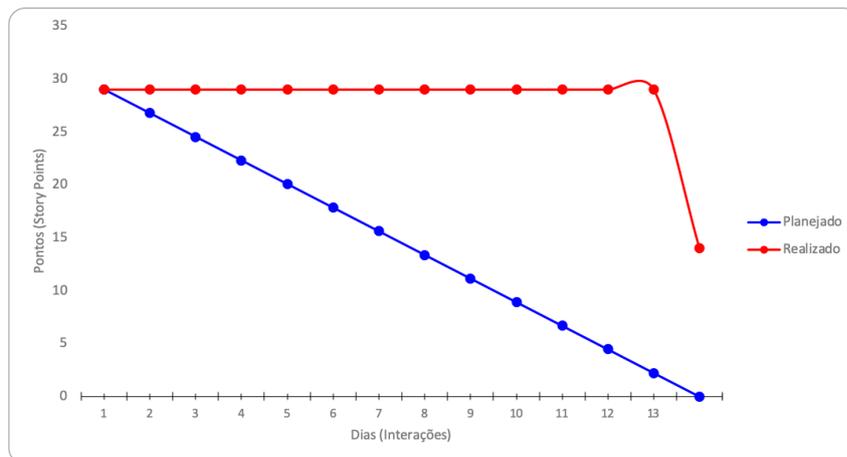
Figura 32 – Gráfico de Burndown - período do dia 14 de setembro de 2019 a 27 de setembro de 2019



Fonte: Elaborado pelos autores

A Figura 33 representa o *burndown* da décima sétima semana de planejamento e desenvolvimento.

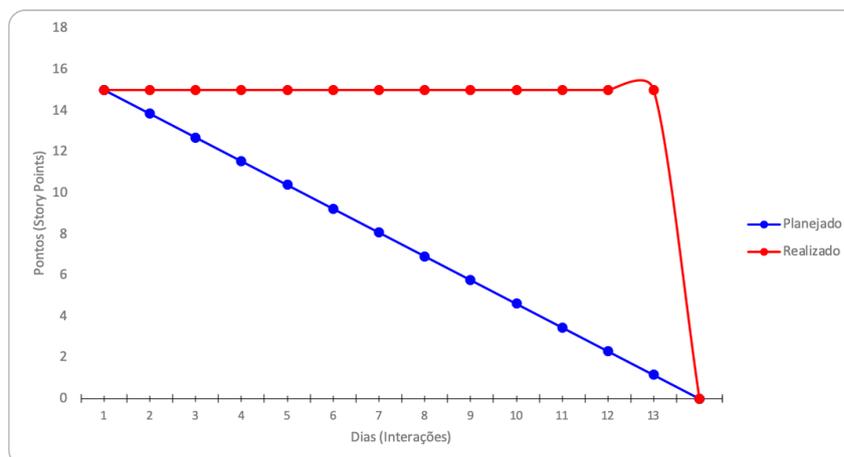
Figura 33 – Gráfico de Burndown - período do dia 30 de setembro de 2019 a 13 de outubro de 2019



Fonte: Elaborado pelos autores

A Figura 34 representa o *burndown* da décima oitava semana de planejamento e desenvolvimento.

Figura 34 – Gráfico de Burndown - período do dia 18 de outubro de 2019 a 07 de novembro de 2019



Fonte: Elaborado pelos autores

## 4.2 DESIGN

Entender os pilares do UI (User Interface) e UX (User Experience) Design é essencial para não desenvolver uma plataforma que é só tecnicamente bem feita e esteticamente interessante, mas também capaz de agradar o usuário. Sendo assim, é fundamental que uma interface digital permita a realização de todas as tarefas propostas de maneira simples e direta, sem frustrações ou a necessidade de recorrer a um “manual de instruções” para ações que deveriam ser intuitivas.

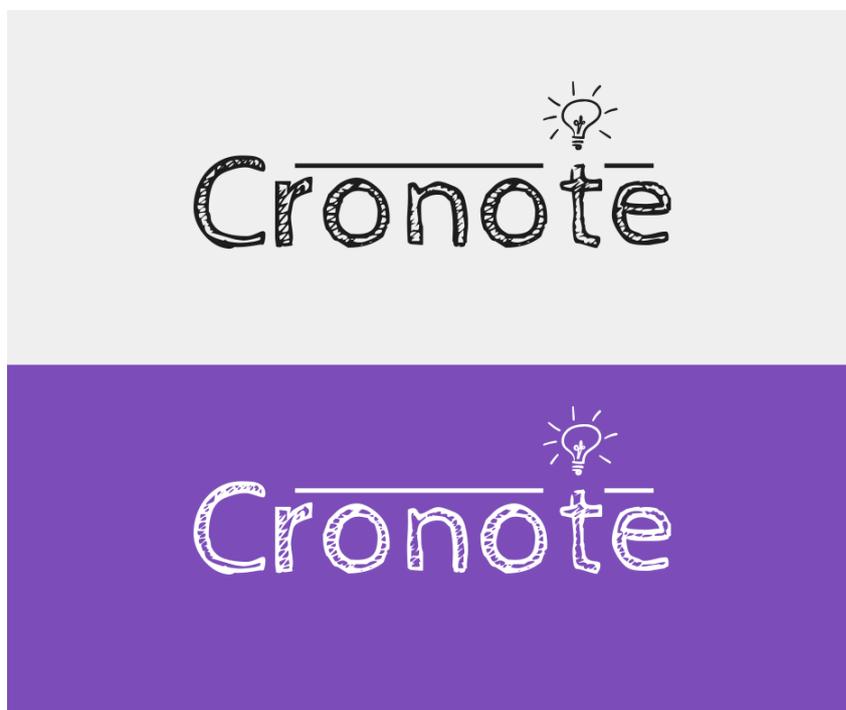
### 4.2.1 IDENTIDADE VISUAL

Todas as formas de representação visual de uma marca (logotipo, tipografia, fontes e cores) ajuda a criar uma identidade reconhecida pelo usuário.

O processo de construção de identidade visual foi documentado em cada uma das suas etapas para construir um verdadeiro padrão que represente o sistema. [Patel \(2019\)](#)

A Figura 35 representa o logotipo escolhido para o sistema.

Figura 35 – logotipo



Fonte: Elaborado pelos autores

A Figura 36 representa a paleta de cores escolhidas para o logotipo do sistema.

Figura 36 – Cores do logotipo



Fonte: Elaborado pelos autores

A Figura 37 representa a tipografia escolhida para o sistema.

Figura 37 – Tipografia do Cronote



Fonte: Elaborado pelos autores

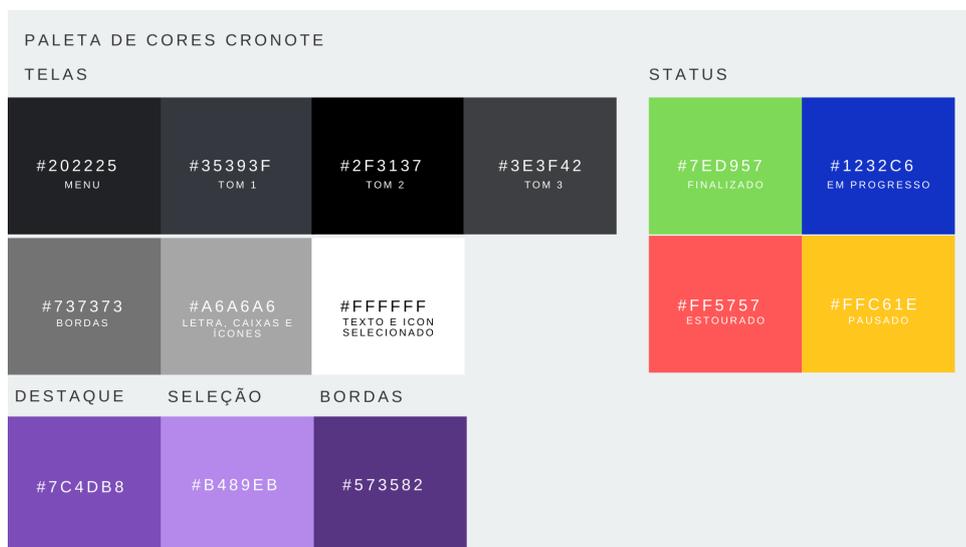
#### 4.2.2 PALETA DE CORES

Foi utilizado o UI, responsável pela parte visual do projeto, para garantir uma user-friendly, ou seja, uma experiência amigável.

A escolha das cores com o tema "dark". Por apresentar o fundo escuro, a tela emite menos luz, o que é ideal para quem faz o uso do sistema no período noturno ou por longas horas em frente ao monitor. Em telas OLED (em que o preto significa pixels desligados), o fundo preto pode render uma leve economia de bateria. [lutti \(2018\)](#)

A Figura 38 representa a paleta de cores escolhidas para serem usadas no sistema.

Figura 38 – Peleta de cores usadas no sistema



Fonte: Elaborado pelos autores

### 4.2.3 HEURÍSTICAS DE NIELSEN

O cientista da computação Jakob Nielsen, ao lado de Rolf Molich, desenvolveu dez heurísticas que são postulados muito respeitados sobre o design de interfaces digitais. Com o objetivo de garantir uma boa experiência ao usuário, o trabalho se transformou em um guia muito usado para a produção e o teste de plataformas digitais consistentes.

“Even the best designers produce successful products only if their designs solve the right problems. A wonderful interface to the wrong features will fail.” Nielsen (2018). Foi aplicado a heurísticas de Nielsen durante e após o desenvolvimento da nossa aplicação. Esses dez atributos também permitem a avaliação e o teste de usabilidade da nova plataforma. Ao conduzir uma boa análise heurística, foi possível direcionar os testes com os usuários para identificar e solucionar problemas, documentando todos os pontos que devem ser aperfeiçoados. Assim, garante-se de forma mais palpável que a nova plataforma terá um bom desenho de interface, navegabilidade apropriada e proporcionará boa interação e experiência ao usuário.

A Tabela 31 representa as heurísticas de Nielsen

A Tabela 32 representa as comparações do nosso sistema com as heurísticas de Nielsen

Tabela 31 – Heurísticas de Nielsen

Identificação	Heurística
01	Visibilidade do Status do Sistema
02	Correspondência entre o sistema e o mundo real
03	Controle do usuário e liberdade
04	Consistência e Uso de padrões
05	Prevenção de erros

06	Reconhecimento ao invés de memorização
07	Flexibilidade e eficiência de uso
08	Design estético e minimalista
09	Ajude os usuários a reconhecer, diagnosticar e recuperar erros
10	Ajuda e documentação

Fonte: Elaborado pelos autores.

Tabela 32 – Comparativo das telas do sistema com as heurísticas de Nielsen

Heurísticas	Telas						
	Login	Esqueceu Senha	Cadastro	Home	Atividades	Times	Projetos
01	X	-	X	X	-	X	X
02	-	X	X	X	X	X	X
03	X	X	X	X	X	X	X
04	-	X	X	X	-	X	X
05	X	-	X	X	-	-	-
06	X	X	X	X	X	X	X
07	X	X	X	-	-	-	-
08	X	X	X	X	X	X	X
09	X	X	X	X	-	X	-
10	X	X	X	X	X	X	X

Fonte: Elaborado pelos autores.

### 4.3 MANUAL

Este manual foi planejado para servir como um guia de utilização do Sistema Cronote. Ele pretende explicar funções básicas e comandos que se repetem em várias telas. A leitura é recomendada para quem deseja conhecer melhor o funcionamento geral do aplicativo.

A preocupação pode ser verificada na adoção de uma linguagem mais simples, telas que se adaptam a diferentes resoluções do monitor ou à liberdade que o usuário tem em escolher as preferências de visualização (cores para os times).

Uma característica importante do Cronote é que ele foi desenvolvido para ser um sistema colaborativo. O uso de tecnologia assistiva também guiou o desenvolvimento do sistema e o usuário não deverá encontrar problemas ao utilizá-lo.

A seguir você poderá entender melhor as divisões principais do e algumas funções básicas.

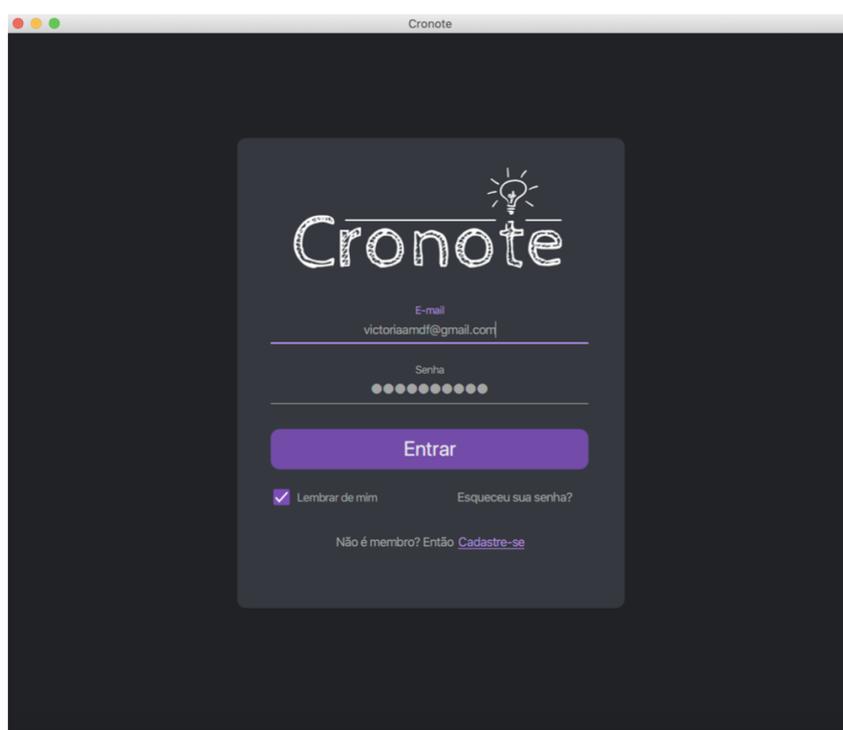
### 4.3.1 LOGIN

Na tela de login o usuário entrará com login e senha. Caso não possua um perfil ele poderá fazer o cadastro no CADASTRAR-SE (Ver 4.3.2)

Caso tenha se esquecido de sua senha, poderá recuperá-la em ESQUECEU SUA SENHA? (Ver 4.3.3)

A Figura 39 representa a tela de login do sistema.

Figura 39 – Tela Login



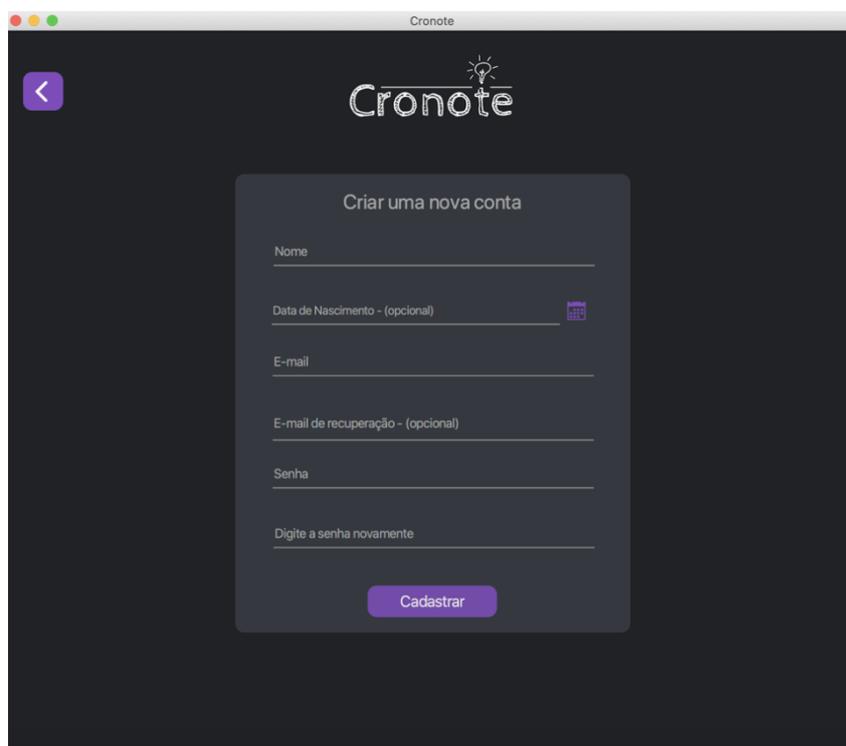
Fonte: Elaborado pelos autores

### 4.3.2 CADASTRO

Na tela de cadastro o usuário de entrará com os dados obrigatórios (aqueles não marcados com "opcional") e ao clicar em CADASTRAR efetuará o cadastro no sistema. Caso a tela não seja preenchida corretamente, alguns avisos irão surgindo para auxiliar.

A Figura 40 representa a tela de cadastro do usuário do sistema.

Figura 40 – Tela de cadastro do usuário



A imagem mostra a interface de usuário para criar uma nova conta no Cronote. O aplicativo é executado em um navegador, como indicado pelo título "Cronote" na barra superior. O layout é escuro, com o logotipo "Cronote" e um ícone de lâmpada no topo. Um botão de voltar (seta para trás) está no canto superior esquerdo. O formulário centralizado, intitulado "Criar uma nova conta", contém os seguintes campos de entrada:

- Nome
- Data de Nascimento - (opcional) com ícone de calendário
- E-mail
- E-mail de recuperação - (opcional)
- Senha
- Digite a senha novamente

Um botão "Cadastrar" em cor verde está localizado na base do formulário.

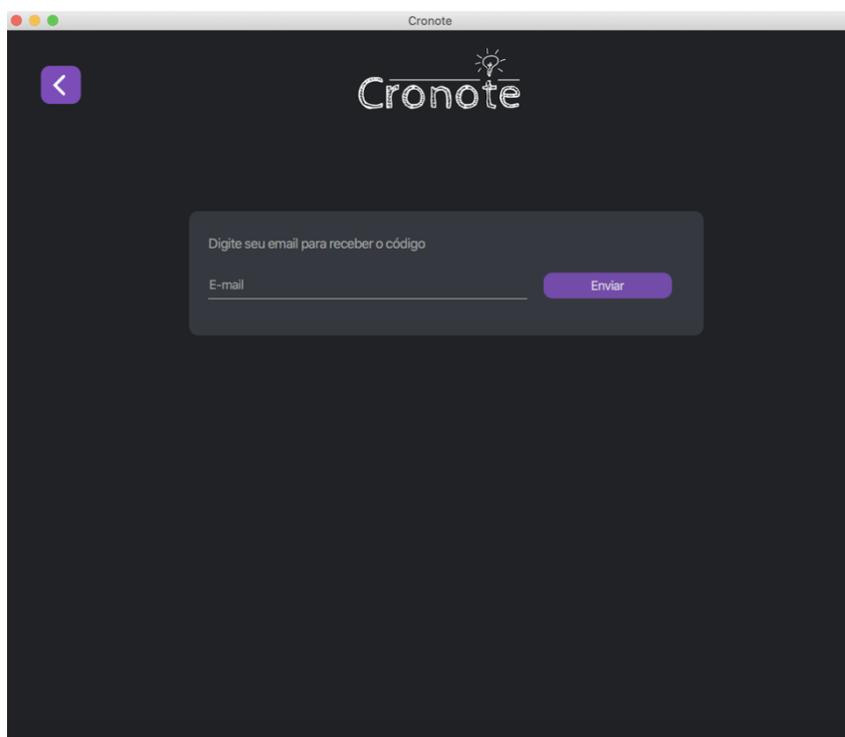
Fonte: Elaborado pelos autores

### 4.3.3 RECUPERAÇÃO DE SENHA

Na tela de recuperação de senha o usuário inicialmente deverá digitar o email cadastrado para receber o código pelo mesmo.

A Figura 41 representa a tela de recuperação de senha antes de se digitar o email.

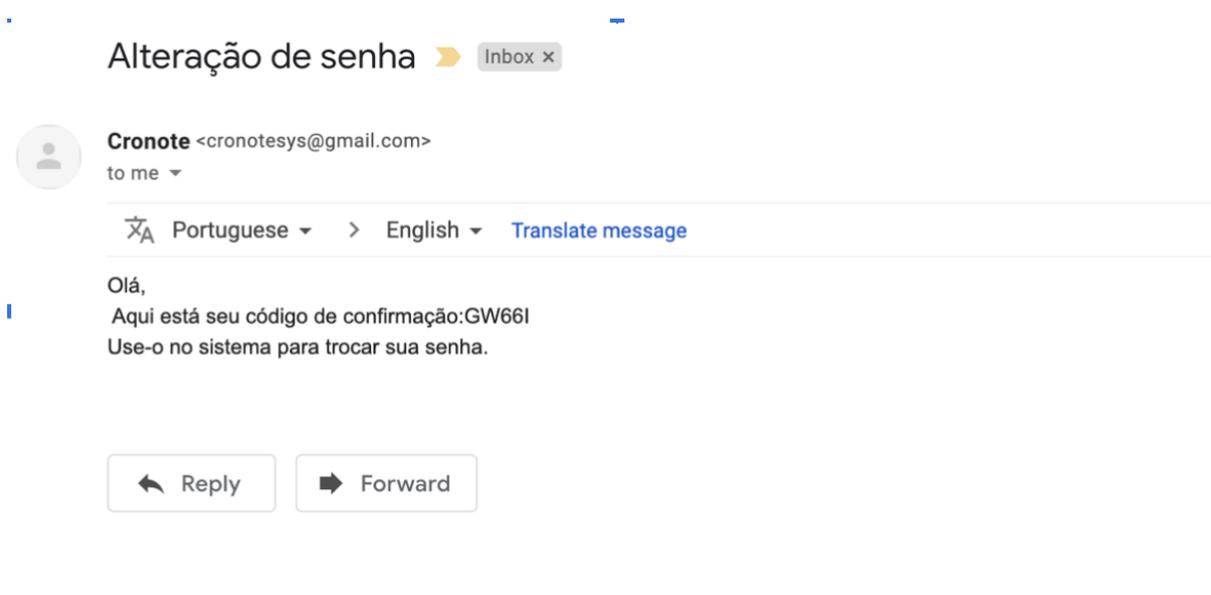
Figura 41 – Tela de recuperação de senha - Etapa 1



Fonte: Elaborado pelos autores

A Figura 42 representa o email que chegará ao usuário que se cadastrou.

Figura 42 – Exemplo de email recebido com código de verificação para a recuperação de senha.



Fonte: Elaborado pelos autores

Em seguida, usará o código recebido no campo correspondente e digitará a nova senha 2 vezes. Em seguida, clique em "Confirmar".

A Figura 43 representa a tela de recuperação de senha depois de se informar o email.

Figura 43 – Tela de recuperação de senha - Etapa 2

The screenshot shows a mobile application window titled "Cronote". At the top left is a back arrow icon. The "Cronote" logo is centered at the top. Below the logo is a form with the following elements:

- A text prompt: "Digite seu email para receber o código".
- An "E-mail" input field containing "victoriaamdf@gmail.com".
- An "Enviar" button.
- A "Digite o código de confirmação" input field.
- A "Nova senha" input field.
- A "Confirme a nova senha" input field.
- "Cancelar" and "Confirmar" buttons at the bottom.

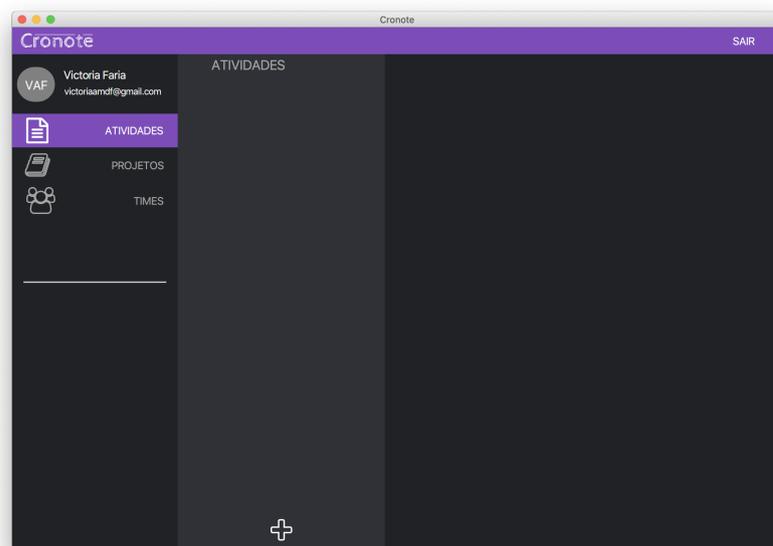
Fonte: Elaborado pelos autores

#### 4.3.4 ATIVIDADE

Para adicionar uma atividade, clique no item "Atividade" no menu lateral e clique no botão de "+" no inferior da aba.

A Figura 44 representa a tela de atividades sem nenhuma atividade cadastrada.

Figura 44 – Tela de atividades.

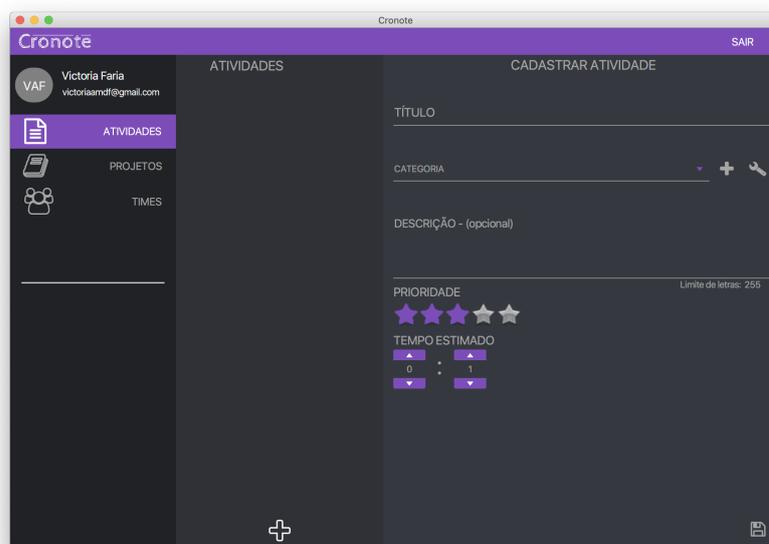


Fonte: Elaborado pelos autores

Ao clicar, abrirá uma aba para o cadastro da atividade. É obrigatório o uso da CATEGORIA, pois será necessário para a predição de tempo das atividades futuras.

A Figura 45 representa a tela de atividades na parte de cadastro de uma atividade.

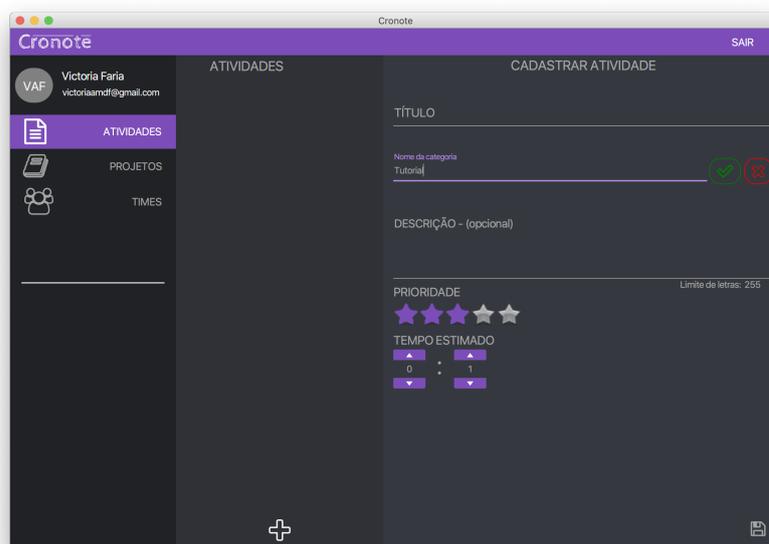
Figura 45 – Tela de cadastro de atividades.



Fonte: Elaborado pelos autores

O usuário pode adicionar novas categorias pelo botão "+" ao lado do campo. A Figura 46 representa a tela de atividades na parte de cadastro de uma categoria.

Figura 46 – Tela de cadastro de categorias.

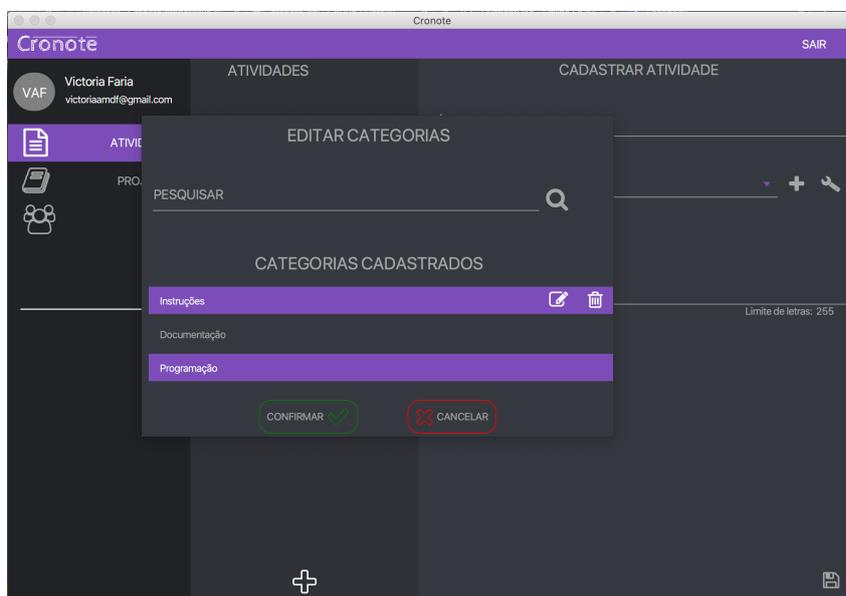


Fonte: Elaborado pelos autores

O usuário também poderá escolher, editar ou excluir outras categorias cadastradas anteriormente.

A Figura 47 representa a tela de atividades na parte de cadastro de edição ou exclusão de uma categoria.

Figura 47 – Tela de edição de categorias.

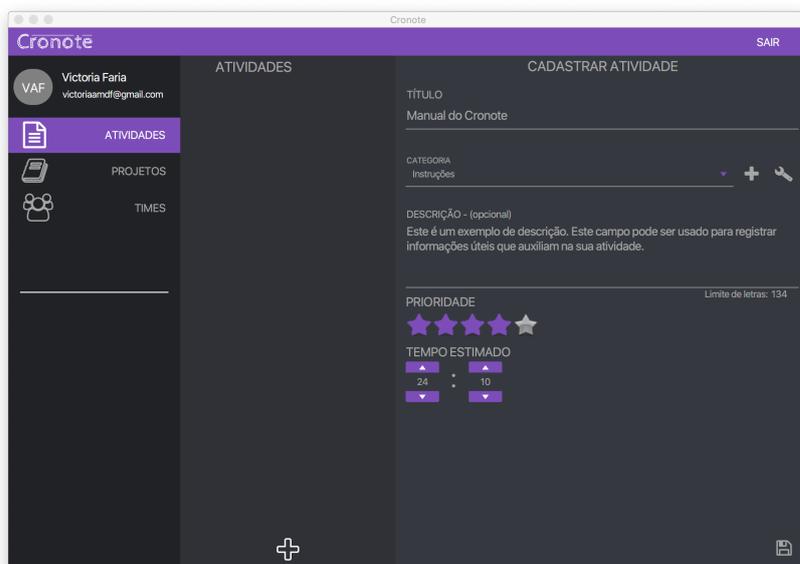


Fonte: Elaborado pelos autores

O usuário tem a possibilidade de adicionar uma descrição (opcional) e também uma prioridade para a atividade servindo para hierarquizar a listagem dos cartões. Sempre partindo da maior prioritária para menor. No TEMPO ESTIMADO o usuário deverá colocar o tempo que pretende para completar esta atividade.

A Figura 48 representa a tela de atividades com os campos sendo preenchidos.

Figura 48 – Tela de cadastro de atividade: descrição

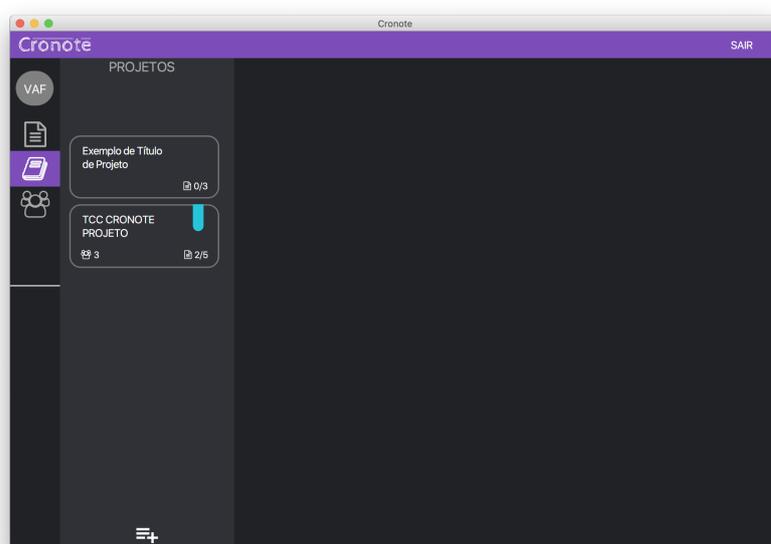


Fonte: Elaborado pelos autores

Para criar um projeto, clique no item "Projetos" no menu lateral e clique no botão de "="+ no inferior da aba.

A Figura 49 representa a tela de projetos antes de selecionar qualquer projeto.

Figura 49 – Tela de projetos.



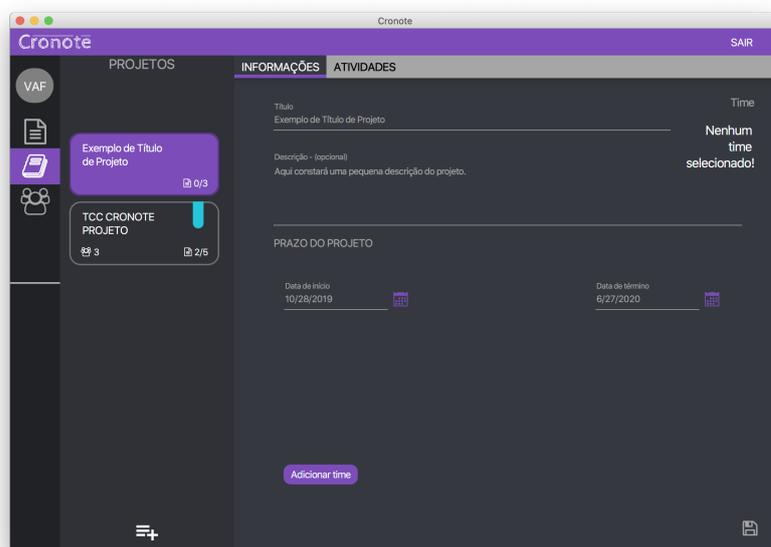
Fonte: Elaborado pelos autores

Ao clicar, irá aparecer uma aba de "informações". É aconselhável que o usuário complete os campos descritos como: título do projeto e descrição. No prazo do projeto o usuário definirá

uma data de início e fim, podendo também, adicionar um time ao projeto.

A Figura 50 representa a tela de projetos ao cadastrar um projeto.

Figura 50 – Tela cadastro de projetos.

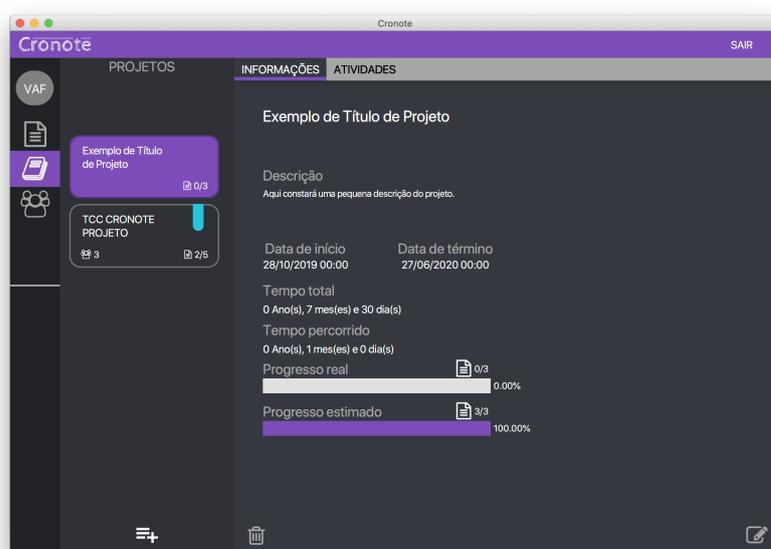


Fonte: Elaborado pelos autores

Para adicionar uma atividade, clique na aba "Atividade" na parte superior da tela.

A Figura 51 representa a tela de inicial de um projeto selecionado.

Figura 51 – Tela de visualização de projeto cadastrado.

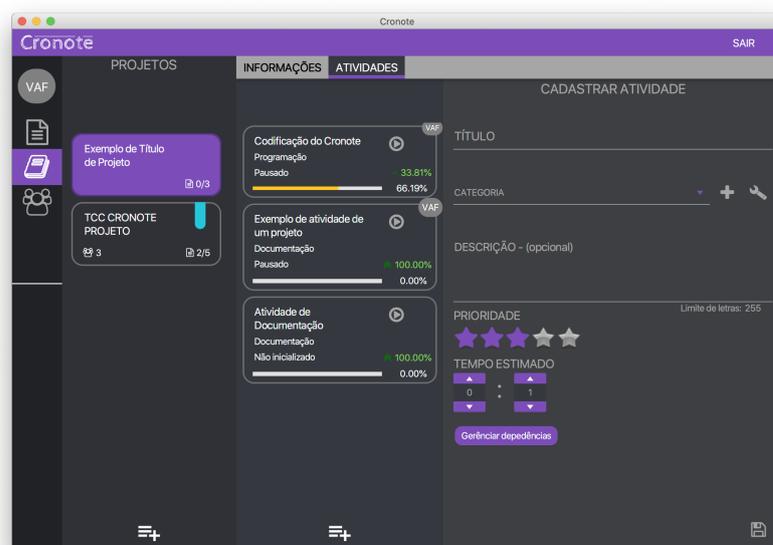


Fonte: Elaborado pelos autores

Clique no botão de "+" no inferior da aba. Para cadastrar uma atividade ao projeto, é necessário adicionar um título, uma descrição (opcional) e prioridade. No TEMPO ESTIMADO o usuário deverá colocar o tempo que pretende para completar esta atividade.

A Figura 52 representa a tela de cadastro de atividades dentro de um projeto, nesta tela, todas as atividades que serão cadastradas farão parte do projeto selecionado.

Figura 52 – Tela de projetos: cadastro de atividades.

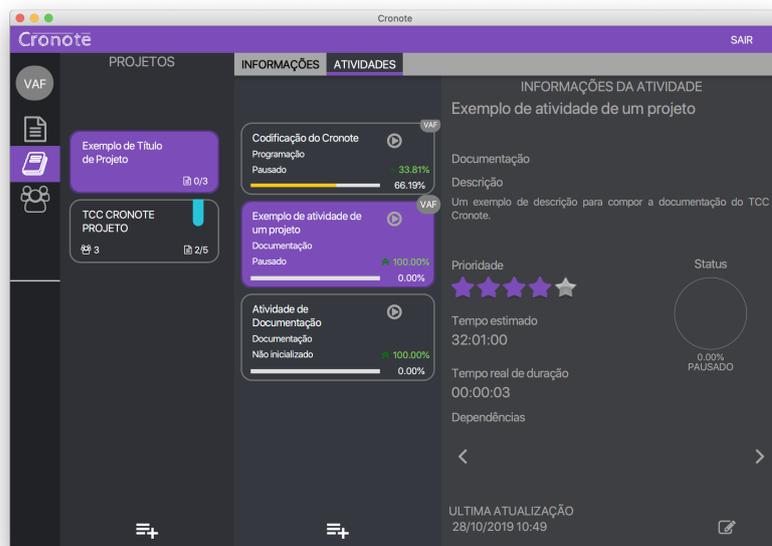


Fonte: Elaborado pelos autores

Ao salvar aparecerão as informações cadastradas.

A Figura 53 representa a tela de informações das atividades em um projeto selecionado.

Figura 53 – Tela de projetos: visualização de atividades.

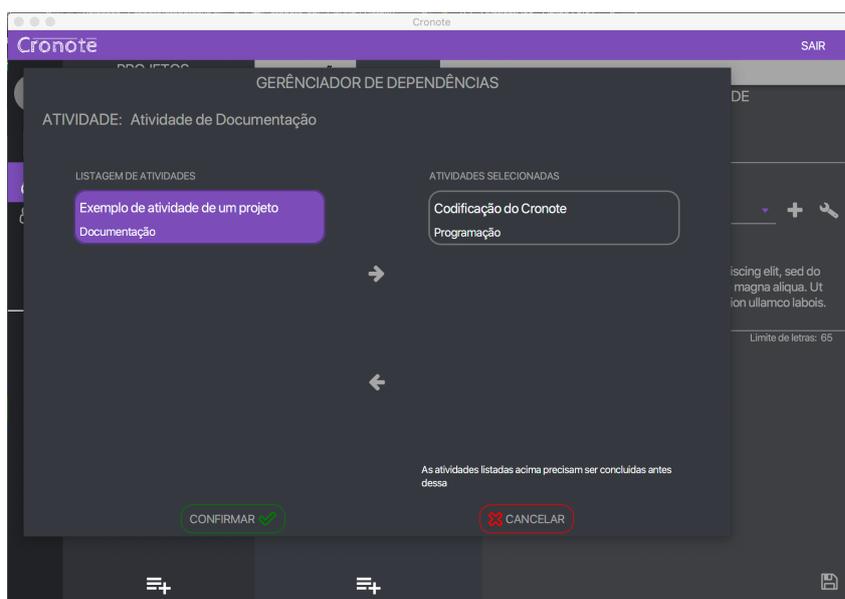


Fonte: Elaborado pelos autores

No gerenciador de dependência, o usuário poderá escolher uma ou mais atividades do projeto que dependerão de outra para ser executada. Isso ajudará o usuário no planejamento e na definição do tempo para cada uma das atividades. Clique em CONFIRMAR ou CANCELAR. Ao escolher, o usuário será obrigado a executar primeiro a atividade dependente.

A Figura 54 representa a tela de dependências de atividades.

Figura 54 – Tela de projetos: gerenciador de dependências.



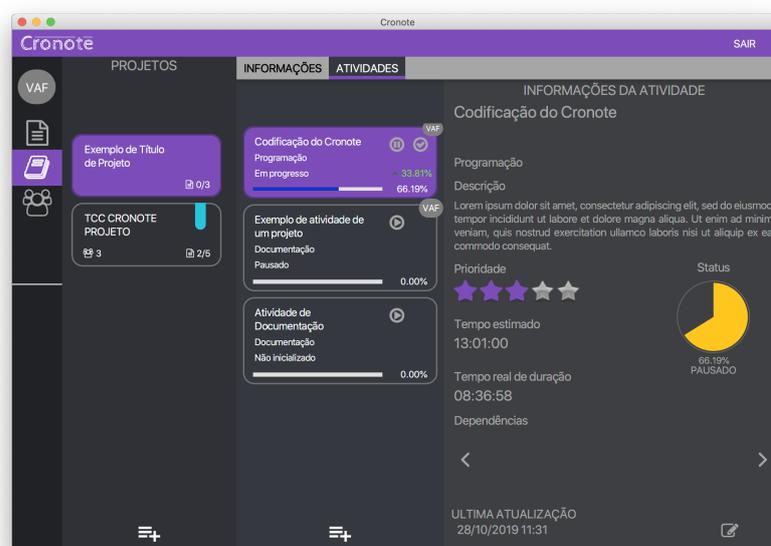
Fonte: Elaborado pelos autores

### 4.3.5 INICIANDO ATIVIDADE

Para inicializar uma atividade o usuário deverá clicar no botão de play. Em seguida, o tempo cadastrado começará a executar. No canto superior direito do cartão ficará as iniciais do usuário, responsável pela atividade. Importante: você poderá executar apenas UMA atividade por vez.

A Figura 55 representa a tela de atividades com uma delas em execução e outra sendo mostrada que está pausada.

Figura 55 – Tela de projeto: iniciando atividade de um projeto

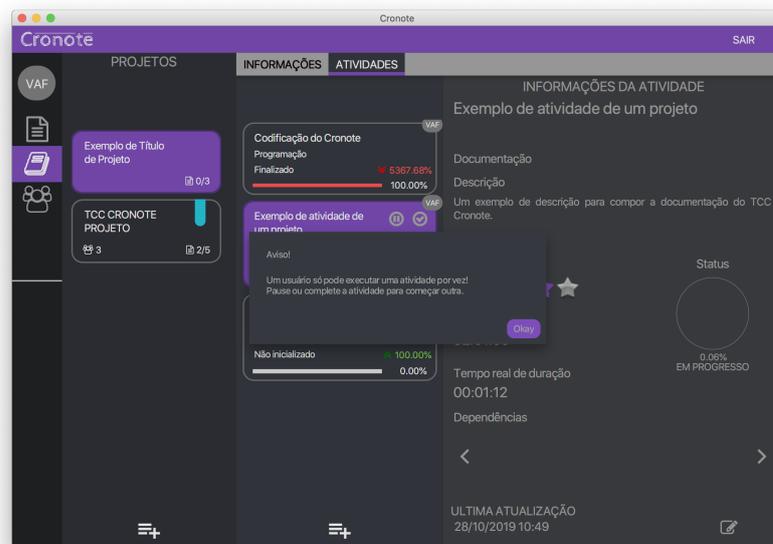


Fonte: Elaborado pelos autores

Para pausar a atividade o usuário poderá clicar no botão de pause que o tempo será congelado. Ao concluir uma atividade o usuário deverá finalizar a atividade para finalizar a contagem do tempo. Importante: o usuário poderá editar a atividade (nome e descrição) apenas quando a atividade estiver pausada!

A Figura 56 representa a tela de atividades com uma delas sendo pausada e mostrando o aviso.

Figura 56 – Tela de projeto: apenas uma atividade inicializada por vez

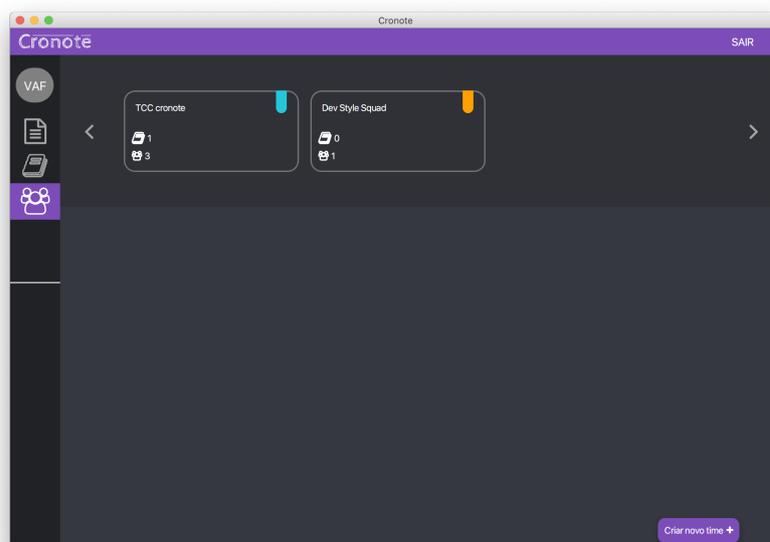


Fonte: Elaborado pelos autores

#### 4.3.6 TIME

O usuário tem a possibilidade de criar um time. Basta clicar em "Criar novo time". A Figura 57 representa a tela de times.

Figura 57 – Tela de time



Fonte: Elaborado pelos autores

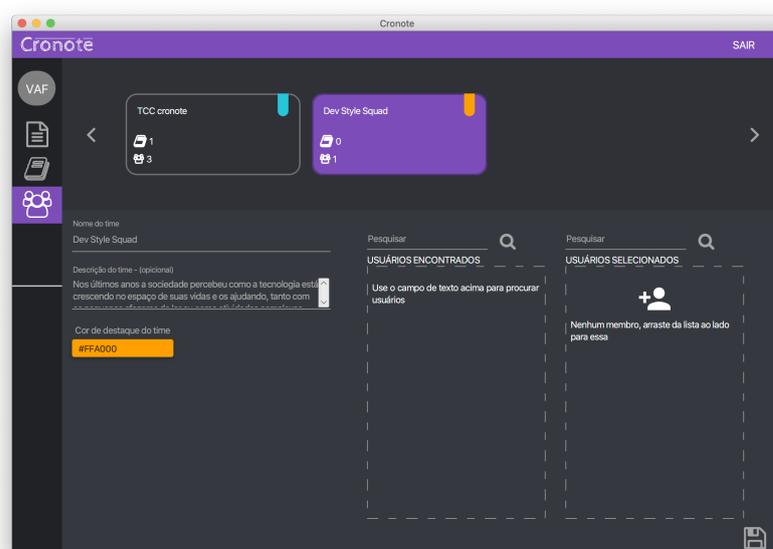
No cadastro é aconselhável que sejam adicionados o nome do time, a descrição e uma

cor de destaque que se refere ao selo do cartão.

Para adicionar os membros do time o usuário deverá pesquisar por nome no campo de busca e arrasta-lo para a área de "usuários selecionados". Ao salvar, os membros selecionados para participar do time irão receber um link de convite pelo email cadastrado no Cronote.

A Figura 58 representa a tela de times na parte de cadastro e que o usuário poderá adicionar um membro na hora do registro.

Figura 58 – Tela cadastro de time

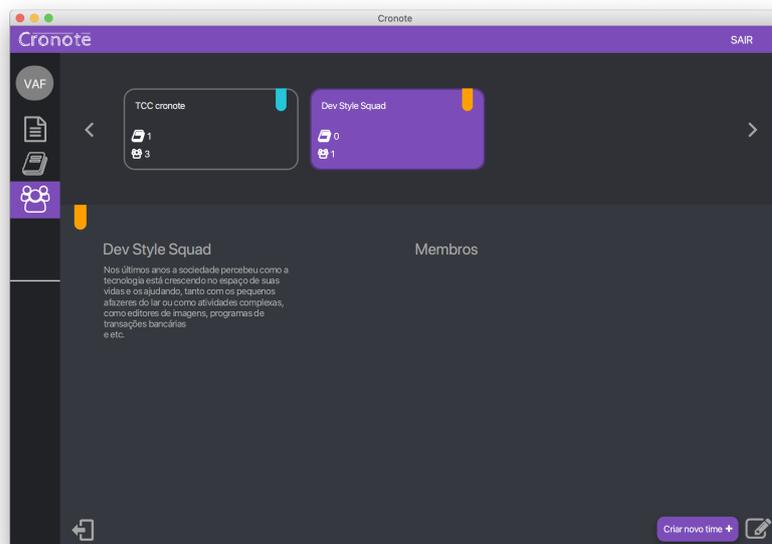


Fonte: Elaborado pelos autores

Ao salvar aparecerão as informações cadastradas. O líder do time não é exibido na listagem de membros.

A Figura 59 representa a tela de times com o time cadastrado e selecionado, mostrando algumas informações.

Figura 59 – Tela cadastro de time: visualização

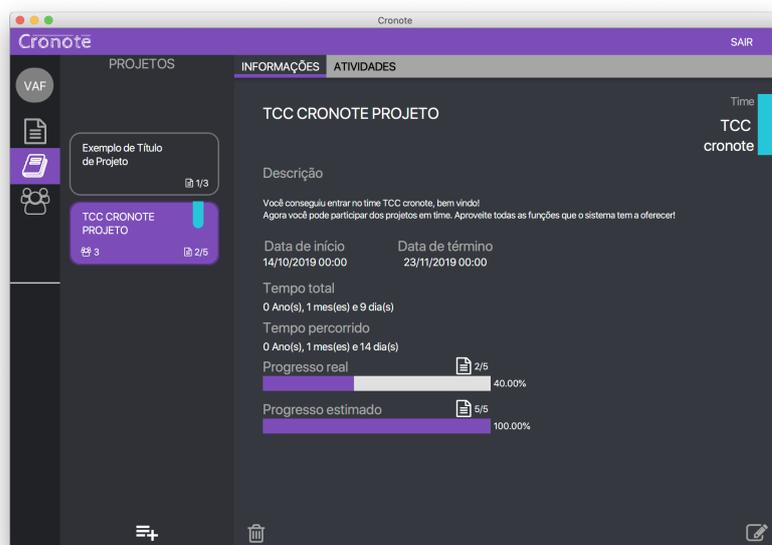


Fonte: Elaborado pelos autores

Existe a possibilidade de adicionar um time ao projeto. Com o projeto selecionado na aba INFORMAÇÕES o usuário além de poder editar as informações básicas do projeto poderá adicionar um time ao projeto, que terá total acesso às atividades do mesmo.

A Figura 60 representa a tela de projeto, com um deles selecionado mostrando que já há um time vinculado a este projeto.

Figura 60 – Tela de informações do projeto: Visualização

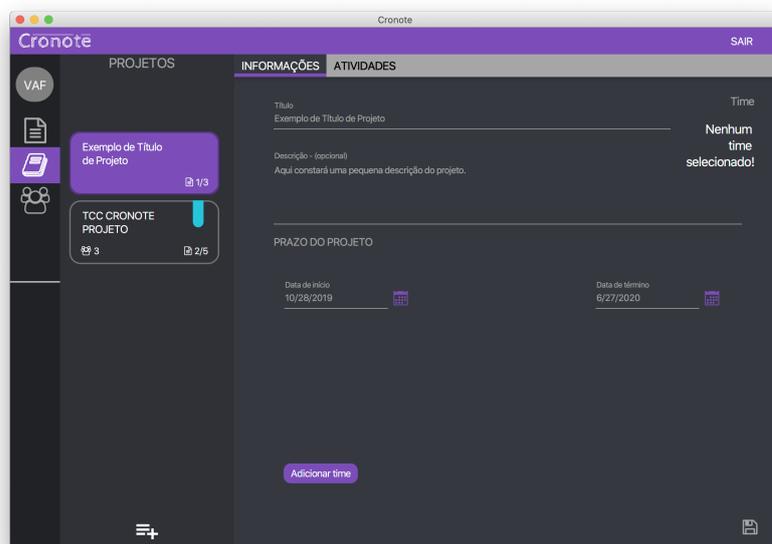


Fonte: Elaborado pelos autores

Basta clicar no botão "adicionar time", conforme a figura 61.

A Figura 61 representa a tela de projeto e mostrando o botão para se atribuir um time ao projeto.

Figura 61 – Tela de informações do projeto: Edição

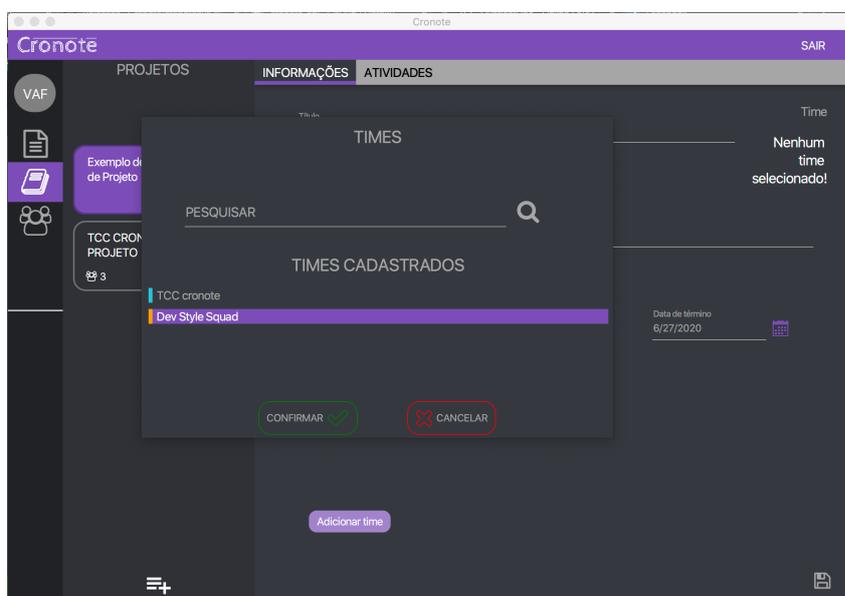


Fonte: Elaborado pelos autores

Abrirá uma tela para a seleção do time ao projeto. Caso não tenha a opção desejada, o usuário deverá verificar o seu cadastro na tela de times conforme a figura 57.

A Figura 62 representa a tela de atribuição de time ao projeto.

Figura 62 – Tela de informações do projeto: Adicionar Time

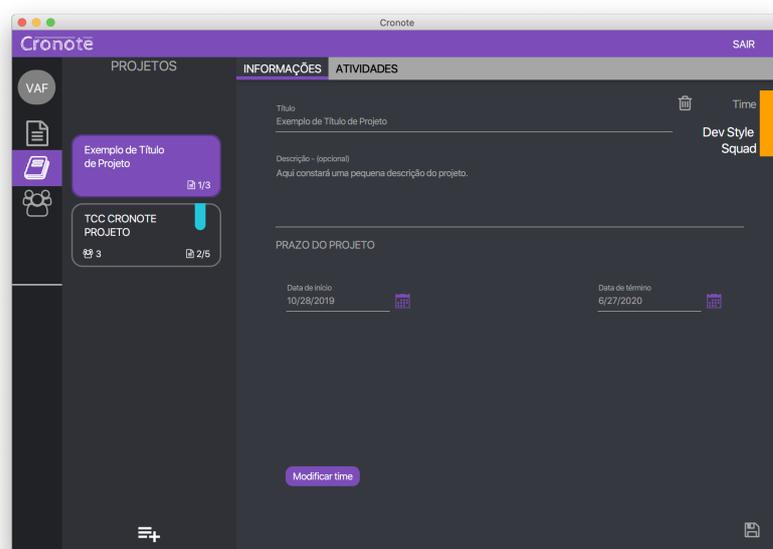


Fonte: Elaborado pelos autores

Após confirmar o time já está adicionado ao projeto e aparecerá no canto superior direito.

A Figura 63 representa a tela de projeto e mostrando o botão para se atribuir um time ao projeto.

Figura 63 – Tela de informações do projeto: Visualização com time

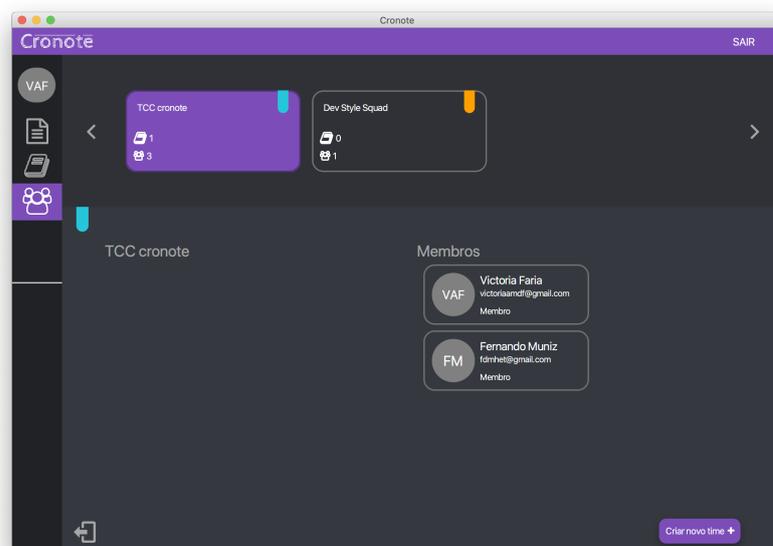


Fonte: Elaborado pelos autores

O usuário tem a opção de deixar de participar de um time a qualquer momento. Para isso, basta ir na visualização do time e clicar no ícone localizado no canto inferior esquerdo.

A Figura 64 representa a tela de time, colocando em destaque os membros participantes.

Figura 64 – Tela de time: Visualização do time



Fonte: Elaborado pelos autores

## 5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo final, desenvolver um aplicativo para planejamento e execução de atividades, sendo de modo individual ou em time. O então Cronote, foi desenvolvido usando a linguagem Java para todo o sistema e JavaFx como biblioteca de front-end. Graças à utilização de Java como linguagem de programação se torna possível a utilização deste sistema nos principais sistemas operacionais, sendo: Windows, MacOS e Linux.

No desenvolvimento houveram dificuldades, com destaque para a complexidade de se trabalhar com MVC, dado a inexperiência do grupo ao começo do desenvolvimento, mas com uma grande ajuda do orientador, esta dificuldade foi superada e conforme foi-se adquirindo familiaridade com a arquitetura foi-se tornando algo mais natural.

Em relação ao JavaFx, haviam diversas bibliotecas desconhecidas ao primeiro contato que foram de muita ajuda, principalmente por fornecer componentes já customizados, esta dificuldade foi superada com uma pesquisa mais a fundo sobre as possibilidades que envolvem a biblioteca.

Quanto à API Rest, além da falta de conhecimento a principal dificuldade encontrada na verdade foi a falta de ferramentas de hospedagem gratuitas que suportem a linguagem Java.

Dividindo o trabalho em três projetos, e trabalhando com MVC, foi encontrada uma dificuldade extra, pois tanto o servidor java como a aplicação no computador do usuário final teriam que usar o mesmo tipo de *model*, fazendo com que fosse necessário criar um *jitpack*, que se equivale ao *maven*, para que o *model* se tornasse uma biblioteca do sistema para os outros dois projetos, evitando assim código duplicados.

Como trabalhos futuros, as possibilidades são:

- Novas perspectivas para as diferentes necessidades, que se adequam a metodologia utilizada pelo usuário ou grupo de usuários, caso estejam utilizando Scrum, ou cascata, ou incremental, o sistema forneceria uma interface mais aprimorada para trabalhar com a metodologia utilizada, auxiliando-os para o desenvolvimento do projeto, deixando o planejamento mais pratico, fácil e eficiente.
- Um modulo relatórios com possibilidade de haver filtros, para fornecer aos usuários uma ferramenta de diagnostico de perfomance, mostrando suas horas trabalhadas; o projeto que está sendo mais desenvolvido; qual categoria tem mais atividades.
- Expandir o Cronote para as plataformas mobile, tanto em *android* quanto em iOS, criando uma facilidade para que o usuário consiga acessar o sistema de qualquer lugar e offline.
- Outros temas no sistema, com a popularidade dos temas *dark* o Cronote foi desenvolvido utilizando este tema, mas é ideal que atenda também a *light*, para que ao menos os dois tipos mais populares sejam contemplados.
- Chat dentro do sistema, para uma rápida e fácil comunicação entre os usuários do sistema.

## Referências

ASANA. **Asana, Inc.** 2008. Disponível em: <<https://asana.com/pt>>. Acesso em: 09 set. 2018. Citado na página 17.

CAMPOS, E. L. de. **Burndown chart – Mede o progresso da sprint e dá indicativos do processo de trabalho da equipe.** 2019. Disponível em: <<https://blog.myscrumhalf.com/burndown-chart-medindo-o-progresso-de-sua-sprint-e-trazendo-indicativos-do-processo-de-trabalho-da-equipe>>. Acesso em: 28 set. 2018. Citado na página 58.

CELESTINO, A. L. **O conceito e as dúvidas sobre o MVC.** 2014. Disponível em: <<https://www.profissionaisiti.com.br/2014/10/o-conceito-e-as-duvidas-sobre-o-mvc/>>. Acesso em: 28 set. 2018. Citado na página 22.

CHANGEVISION. **Site oficial do Astah.** 2019. Disponível em: <<http://astah.net/download>>. Acesso em: 28 set. 2018. Citado na página 23.

CODACY. **Site oficial da codacy.** 2018. Disponível em: <<https://www.codacy.com/product>>. Acesso em: 28 set. 2018. Citado na página 22.

COMMONS, A. **Commons Email.** 2018. Disponível em: <<https://commons.apache.org/proper/commons-email/>>. Acesso em: 28 set. 2018. Citado na página 23.

CONSERVANCY, S. F. **Site oficial do git.** 2016. Disponível em: <<https://git-scm.com/>>. Acesso em: 28 set. 2018. Citado na página 23.

DEVMEDIA. **Conheça o Apache Tomcat.** 2007. Disponível em: <<https://www.devmedia.com.br/conheca-o-apache-tomcat/4546#modulo-mvp>>. Acesso em: 28 set. 2018. Citado na página 22.

DEVMEDIA. **Guia Completo de Java.** 2017. Disponível em: <<https://www.devmedia.com.br/guia/linguagem-java/38169>>. Acesso em: 28 set. 2018. Citado na página 21.

DIAS, E. **4 Conceitos sobre REST que Qualquer Desenvolvedor Precisa Conhecer.** 2016. Disponível em: <<https://blog.algaworks.com/4-conceitos-sobre-rest-que-qualquer-desenvolvedor-precisa-conhecer/>>. Acesso em: 28 set. 2018. Citado na página 22.

GANTTER. **Ganttter.** 2011. Disponível em: <<https://www.ganttter.com>>. Acesso em: 09 set. 2018. Citado na página 17.

GUEDES, G. T. A. **UML 2 uma abordagem prática.** 1. ed. São Paulo: Novatec, 2009. Citado 4 vezes nas páginas 23, 24, 35 e 40.

HAT, R. **Site oficial do hibernate.** 2016. Disponível em: <<https://hibernate.org/>>. Acesso em: 28 set. 2018. Citado na página 23.

IDE, E. **Site oficial do Eclipse IDE.** 2019. Disponível em: <<https://www.eclipse.org/eclipseide/>>. Acesso em: 28 set. 2018. Citado na página 22.

LUTTI. **Design Patterns – O que são e quais os benefícios?** 2018. Disponível em: <<https://www.opus-software.com.br/design-patterns/>>. Acesso em: 28 set. 2018. Citado na página 70.

- MAVEN. **Maven**. 2018. Disponível em: <<https://pt.wikiversity.org/wiki/Maven>>. Acesso em: 28 set. 2018. Citado na página 21.
- NIELSEN, W. S. J. **As 10 Heurísticas de Nielsen**. 2018. Disponível em: <<https://eusoudev.com.br/heurísticas-de-nielsen/>>. Acesso em: 28 de novembro de 2019. Citado na página 71.
- PATEL, N. **Identidade Visual: Passo a Passo Para Criar a Identidade Visual Perfeita**. 2019. Disponível em: <<https://neilpatel.com/br/blog/identidade-visual/>>. Acesso em: 28 set. 2018. Citado na página 68.
- POSTGRESQL. **Site oficial do PostgreSQL**. 2017. Disponível em: <<https://www.postgresql.org>>. Acesso em: 28 set. 2018. Citado na página 22.
- RUNRUN.IT. **RunRun.it**. 2007. Disponível em: <<https://runrun.it/pt-BR>>. Acesso em: 09 set. 2018. Citado na página 16.
- SOMMERVILLE. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison-Wesley, 2007. Citado 2 vezes nas páginas 19 e 20.
- SUTHERLAND, J. **Scrum: A arte de fazer o dobro do trabalho na metade do tempo**. São Paulo: LeYa, 2014. Disponível em: <<http://leyaprimeiro.com.br/anteriores/jun-jul-16/pdf/jun-jul/scrum.pdf>>. Acesso em: 25 de fevereiro de 2019. Citado na página 18.
- TRELLO. **ATLASSIAN**. 2011. Disponível em: <<https://trello.com>>. Acesso em: 09 set. 2018. Citado na página 16.
- WIKIBOOKS. **Introdução a JavaFX**. 2016. Disponível em: <<https://pt.wikibooks.org/wiki/JavaFx/Introdu%C3%A7%C3%A3o>>. Acesso em: 28 set. 2018. Citado na página 21.