

UNIVERSIDADE FEDERAL DO PARANÁ

GUILHERME MARTINS WEINHARDT DA SILVEIRA

**AGILE MANAGER: GERENCIADOR DE PROJETOS ORIENTADO À
METODOLOGIA ÁGIL**

CURITIBA

2017

GUILHERME MARTINS WEINHARDT DA SILVEIRA

**AGILE MANAGER: GERENCIADOR DE PROJETOS ORIENTADO À
METODOLOGIA ÁGIL**

Monografia apresentada como requisito parcial à obtenção do grau de Especialista em Engenharia de Software, no Curso de Pós-Graduação em Engenharia de Software, Setor de Educação Profissional e Tecnológica, da Universidade Federal do Paraná.

Orientadora: Profa. Dra. Rafaela Montovani Fontana

CURITIBA

2017



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor SEPT
Curso de Pós-Graduação ENGENHARIA DE SOFTWARE

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE SOFTWARE da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **GUILHERME MARTINS WEINHARDT DA SILVEIRA** intitulada: **Gerenciador de Projetos Orientado à Metodologia Ágil**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 06 de Novembro de 2017.

RAFAELA MANTOVANI FONTANA
Presidente da Banca Examinadora (UFPR)

JAIME WOJCIECHOWSKI
Avaliador Interno (UFPR)

A todos os familiares, amigos e professores que contribuíram para a realização desse trabalho.

A persistência é o menor caminho do êxito.

Charles Chaplin

RESUMO

O presente trabalho refere-se a um sistema de gerenciamento de projetos de software, orientado à metodologia ágil, denominado Agile Manager. Dentre os diversos softwares de gerenciamento de projetos disponíveis no mercado, existe a carência de sistemas voltados a equipes que utilizam a metodologia ágil no desenvolvimento de seus trabalhos, público este que se encontra em ascensão. Visando suprir a necessidade de uma ferramenta de gerenciamento de projetos para pequenos times, este trabalho de conclusão de curso apresenta uma proposta que, a partir do gerenciamento de tarefas, os usuários possam acompanhar o desenvolvimento, progresso e possíveis impedimentos do projeto. As funcionalidades da aplicação foram desenvolvidas visando a execução dos princípios presentes na metodologia ágil de desenvolvimento de software (iteração, papéis, medição de progresso através de software funcional, efetividade, entre outros). O Scrum é a metodologia base utilizada para orientar o desenvolvimento da ferramenta Agile Manager. A fundamentação deste trabalho faz menção as camadas presentes na Engenharia de Software, abrangendo projeto, gerenciamento de projetos, apresentação e comparação entre metodologias de desenvolvimento e, por fim, o desenvolvimento do software proposto neste trabalho.

Palavras-chave: Metodologia Ágil. Gerência de Projetos de Software.
Engenharia de Software.

ABSTRACT

The present work refers to a project management software, oriented to agile methodology, called Agile Manager. Among the various project management softwares available in the market, there is the lack of systems oriented to teams that use the agile methodology in the development of their work. And this public is increasing. Aiming to supply the need for a project management tool for small teams, this final project shows a proposal that, from the task management, users can follow the development, progress and possible project impediments. The functionalities of the application were developed aiming the principles presented in the agile software development methodology (iteration, roles, measurement of progress through functional software, effectiveness, among others). Scrum is the basic methodology used to guide Agile Manager development. The foundation of this work refers to the layers presented in Software Engineering, covering project, project management, presentation and comparison between different development methodologies and the developed software proposed in this work.

Key-words: Agile Methodology. Software Project Management. Software Engineering.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| FIGURA 1 - CICLO DE VIDA DO MODELO CASCATA..... | 19 |
| FIGURA 2 - CICLO DE VIDA DO SCRUM..... | 21 |
| FIGURA 3 - FASES DO PROCESSO UNIFICADO..... | 27 |
| FIGURA 4 - GRÁFICO DE GANTT | 29 |
| FIGURA 5 - LOGIN NO SISTEMA | 33 |
| FIGURA 6 - CADASTRO DE USUÁRIO | 34 |
| FIGURA 7 - CADASTRO DE STATUS..... | 35 |
| FIGURA 8 - CADASTRO DE CATEGORIAS | 36 |
| FIGURA 9 - ATUALIZAR CADASTRO | 37 |
| FIGURA 10 - VISUALIZAR PROJETOS | 38 |
| FIGURA 11 - CADASTRO DE PROJETO..... | 39 |
| FIGURA 12 - CADASTRO DE SPRINT..... | 40 |
| FIGURA 13 - CADASTRO DE DEMANDA..... | 41 |
| FIGURA 14 - CADASTRO DE HISTÓRIA..... | 42 |
| FIGURA 15 - CADASTRO DE TAREFA..... | 43 |
| FIGURA 16 - CADASTRO DE APONTAMENTO | 44 |
| FIGURA 17 - MINHAS TAREFAS | 45 |
| FIGURA 18 - RELATÓRIO DE TAREFAS | 46 |
| FIGURA 19 - BURNDOWN | 47 |
| FIGURA 20 - LOGOUT | 48 |
| FIGURA 21 - RECUPERAR SENHA..... | 49 |
| FIGURA 22 - CASOS DE USO NEGOCIAIS..... | 54 |
| FIGURA 23 - CADASTRO DE PROJETO (PROTÓTIPO)..... | 58 |
| FIGURA 24 - CADASTRO DE SPRINT (PROTÓTIPO) | 59 |
| FIGURA 25 - CADASTRO DE DEMANDA (PROTÓTIPO)..... | 60 |
| FIGURA 26 - CADASTRO DE TAREFA (PROTÓTIPO) | 61 |
| FIGURA 27 - RELATÓRIO DE TAREFAS (PROTÓTIPO) | 62 |
| FIGURA 28 - GRÁFICO DE BURNDOWN (PROTÓTIPO) | 63 |
| FIGURA 29 - DIAGRAMA DE CLASSES NEGOCIAIS | 64 |
| FIGURA 30 - CASOS DE USO | 65 |
| FIGURA 31 - DIAGRAMA DE CLASSES (COM ATRIBUTOS) | 81 |

| | |
|---|-----|
| FIGURA 32 - EFETUAR LOGIN..... | 82 |
| FIGURA 33 - RECUPERAR SENHA..... | 83 |
| FIGURA 34 - EFETUAR LOGOUT..... | 84 |
| FIGURA 35 - MANTER DADOS PESSOAIS..... | 85 |
| FIGURA 36 - MANTER USUÁRIO..... | 86 |
| FIGURA 37 - MANTER STATUS..... | 87 |
| FIGURA 38 - MANTER CATEGORIA..... | 88 |
| FIGURA 39 - MANTER PROJETO..... | 89 |
| FIGURA 40 - MANTER SPRINT..... | 90 |
| FIGURA 41 - MANTER DEMANDA..... | 91 |
| FIGURA 42 - MANTER HISTÓRIA..... | 92 |
| FIGURA 43 - MANTER TEMPO GASTO..... | 94 |
| FIGURA 44 - GERAR RELATÓRIO DE TAREFAS..... | 95 |
| FIGURA 45 - GERAR BURNDOWN..... | 96 |
| FIGURA 46 - DIAGRAMA DE CLASSES (COMPLETO)..... | 97 |
| FIGURA 47 - DIAGRAMA DE CLASSES VO..... | 98 |
| FIGURA 48 - DIAGRAMA DE CLASSES PAGES..... | 99 |
| FIGURA 49 - DIAGRAMA ENTIDADE RELACIONAMENTO..... | 100 |
| FIGURA 50 - CAMPO OBRIGATÓRIO..... | 102 |
| FIGURA 51 - MENSAGEM DE SUCESSO..... | 103 |
| FIGURA 52 - BOTÃO CANCELAR..... | 104 |
| FIGURA 53 - CAMPO OBRIGATÓRIO ALTERAÇÃO..... | 106 |
| FIGURA 54 - MENSAGEM SUCESSO ALTERAÇÃO..... | 107 |
| FIGURA 55 - BOTÃO CANCELAR ALTERAÇÃO..... | 108 |
| FIGURA 56 - LOGIN INVÁLIDO..... | 110 |
| FIGURA 57 - LOGIN VÁLIDO..... | 111 |
| FIGURA 58 - HOME PAGE..... | 112 |
| FIGURA 59 - HOME PAGE DESENVOLVEDOR..... | 113 |

LISTA DE TABELAS

| | |
|--|-----|
| Tabela 1 - TECNOLOGIAS UTILIZADAS..... | 30 |
| Tabela 2 - CASO DE TESTE INSERT | 101 |
| Tabela 3 - CASO DE TESTE UPDATE | 105 |
| Tabela 4 - TESTES DE PERMISSÃO | 109 |

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 13 |
| 1.1 | PROBLEMA..... | 14 |
| 1.2 | OBJETIVOS | 15 |
| 1.2.1 | Objetivo Geral..... | 15 |
| 1.2.2 | Objetivos Específicos | 15 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 16 |
| 2.1 | PROJETO..... | 16 |
| 2.2 | GERÊNCIA DE PROJETOS..... | 17 |
| 2.3 | MÉTODOS TRADICIONAIS | 18 |
| 2.3.1 | Modelo Cascata..... | 19 |
| 2.4 | METODOLOGIA ÁGIL..... | 19 |
| 2.4.1 | Metodologia Scrum..... | 21 |
| 2.5 | METODOLOGIA TRADICIONAL OU ÁGIL? | 22 |
| 2.6 | SOFTWARES DE GESTÃO DE PROJETOS..... | 23 |
| 2.6.1 | Microsoft Project..... | 23 |
| 2.6.2 | Microsoft TFS | 24 |
| 2.6.3 | Jira..... | 24 |
| 2.6.4 | Trello..... | 25 |
| 3 | MATERIAIS E MÉTODOS | 26 |
| 3.1 | METODOLOGIA UTILIZADA..... | 26 |
| 3.1.1 | Processo Unificado..... | 26 |
| 3.1.2 | Gráfico de Gantt | 28 |
| 3.2 | MATERIAIS | 30 |
| 4 | APRESENTAÇÃO DO SISTEMA | 33 |
| 4.1 | CADASTROS BÁSICOS | 33 |
| 4.1.1 | Login..... | 33 |
| 4.1.2 | Cadastro de usuários..... | 33 |
| 4.1.3 | Cadastro de status..... | 34 |
| 4.1.4 | Cadastro de categorias..... | 35 |
| 4.1.5 | Atualizar cadastro | 36 |

| | | |
|----------|--|------------|
| 4.2 | GERENCIAMENTO DE PROJETOS..... | 37 |
| 4.2.1 | Visualizar projetos | 37 |
| 4.2.2 | Projeto | 38 |
| 4.2.3 | Sprint | 39 |
| 4.2.4 | Demanda | 40 |
| 4.2.5 | História | 41 |
| 4.2.6 | Tarefa | 42 |
| 4.2.7 | Apontamentos (Tempo gasto) | 43 |
| 4.3 | RELATÓRIOS | 44 |
| 4.3.1 | Minhas atividades | 44 |
| 4.3.2 | Relatório de tarefas | 45 |
| 4.3.3 | Burndown | 46 |
| 4.4 | LOGOUT | 47 |
| 4.4.1 | Sair do sistema | 47 |
| 4.4.2 | Recuperar senha | 48 |
| 5 | CONSIDERAÇÕES FINAIS | 50 |
| 5.1 | RECOMENDAÇÕES PARA TRABALHOS FUTUROS..... | 50 |
| | REFERÊNCIAS..... | 52 |
| | APÊNDICE A – VISÃO | 53 |
| | APÊNDICE B – CASOS DE USO NEGOCIAIS..... | 54 |
| | APÊNDICE C – GLOSSÁRIO..... | 56 |
| | APÊNDICE D – REGRAS DE NEGÓCIO | 57 |
| | APÊNDICE E – PROTÓTIPO DE TELAS..... | 58 |
| | APÊNDICE F – MODELO DE OBJETOS NEGOCIAIS..... | 64 |
| | APÊNDICE G – CASOS DE USO..... | 65 |
| | APÊNDICE H – MODELO DE OBJETOS (COM ATRIBUTOS)..... | 81 |
| | APÊNDICE I – DIAGRAMAS DE SEQUÊNCIA..... | 82 |
| | APÊNDICE J – MODELO DE OBJETOS (COMPLETO)..... | 97 |
| | APÊNDICE K – MODELO FÍSICO DE DADOS..... | 100 |
| | APÊNDICE L – CASOS DE TESTE | 101 |

1 INTRODUÇÃO

Segundo Pressman (2016), a engenharia de software contempla métodos, processos e ferramentas que auxiliam desenvolvedores a entregar software com altíssima qualidade. Dentre todos os métodos aderentes a engenharia de software quanto a desenvolvimento de projetos, a gestão de projetos vem sendo estudada e aperfeiçoada no decorrer dos anos.

Segundo o PMI (2017), “o gerenciamento de projetos é a aplicação de conhecimentos, habilidades e técnicas para a execução de projetos de forma efetiva e eficaz”. Diversas diretrizes de gestão de projetos voltadas para desenvolvimento de software são difundidas, tanto no meio corporativo quanto no meio acadêmico. Duas vertentes são praticadas: as metodologias tradicionais e as metodologias ágeis.

Os métodos tradicionais são baseados em processos bem definidos e bem documentados em todas as etapas do projeto com grande geração de artefatos, apostando nesta abordagem para garantir a qualidade do produto final e satisfação do cliente. Os métodos tradicionais mais conhecidos são: Modelo Cascata, Modelo Espiral e Processo Unificado (PU).

Com a ideia de valorizar os pontos mais importantes de todo o ciclo de desenvolvimento de software, excluir a burocracia desnecessária e visando uma geração de artefatos minimalista, os métodos ágeis estão ganhando cada vez mais adeptos. As metodologias de desenvolvimento ágil mais conhecidas são: Scrum, Kanban e Extreme Programming.

Segundo Pressman (2011, p. 81), “os princípios ‘[ágeis]’ de desenvolvimento priorizam a entrega mais que a análise e projeto (embora essas atividades não sejam desencorajadas)”. Diferentemente do gerenciamento de projeto tradicional, o ágil é baseado em um planejamento menos detalhado, apostando mais em uma abordagem compreendendo as mudanças e adaptações de escopo que ocorrem no decorrer do desenvolvimento do projeto.

Independente de metodologia aplicada, um projeto de software devidamente gerenciado possui benefícios. Segundo Vargas (2009), entre os benefícios podemos citar a melhora no cumprimento do prazo e custo; a flexibilidade para as alterações de escopo ocorridas durante o desenvolvimento do projeto; a antecipação de possíveis problemas que possam ocorrer no decorrer do projeto; facilita a

comunicação entre os envolvidos; maior qualidade entregue; e decisões mais assertivas.

1.1 PROBLEMA

Segundo Beck, Extreme Programming (também conhecido como XP) “é uma metodologia ágil para equipes pequenas e médias desenvolvendo software com requisitos vagos e em constante mudança”. Essa não é uma característica exclusiva do XP, mas comum de muitas equipes de desenvolvimento que optam por utilizar alguma metodologia ágil. Os frameworks de desenvolvimento ágil se encaixam mais facilmente em equipes pequenas e médias, visando melhor gerenciamento e acompanhamento do projeto.

Softwares para gestão de projetos auxiliam o time no controle do andamento de suas tarefas, acompanhamento das atividades dos outros membros da equipe e da evolução do projeto como um todo. Tais sistemas agem como uma ferramenta que complementa a comunicação frequente (reuniões diárias, revisões, entre outros) pregada nas metodologias ágeis, bem como auxiliar a equipe na visualização se o projeto está de acordo com o esperado em termos tempo e escopo, possibilitando uma visão macro do status do projeto.

Dado que a metodologia tradicional foi a precursora no gerenciamento de projetos de software, bem como ainda é a mais utilizada, existem diversas ferramentas, softwares e frameworks disponíveis visando este público (exemplos e comparação de mercado serão citados no capítulo seguinte). Em contrapartida, os métodos ágeis não possuem tanta diversidade de ferramentas para auxiliar o gerenciamento do projeto, por serem uma abordagem mais recente, flexível e ainda menos utilizada, se comparada a metodologia tradicional.

Apresentado isso, encontra-se a necessidade de um software de gerenciamento de projetos voltado para metodologia ágil em pequenas e médias equipes.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver um sistema, denominado Agile Manager, que permita o gerenciamento de projetos de desenvolvimento de software que utilizam métodos ágeis. O gerenciamento será dado através de cadastros de projetos, sprints, itens de backlog, demandas, histórias e tarefas. Além disso, gerar relatórios que auxiliem o time a identificar o progresso do projeto. Também é contemplado o gerenciamento de papéis, entre Scrum Master, Product Owner e Desenvolvedor/Tester, para fins de permissão de acessos no sistema.

1.2.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

- a) Permitir cadastro de projetos.
- b) Permitir que os usuários cadastrem sprints, itens de backlog, demandas, histórias e tarefas de determinado projeto, para fins de gerenciamento.
- c) Apontar tempo gasto nas atividades desenvolvidas.
- d) Gerar relatórios para medição de progresso do projeto.
- e) Permitir o gerenciamento de papéis dos membros do time.

Nos próximos capítulos deste trabalho, tem-se a fundamentação teórica, na qual será explicado os conceitos de projeto, metodologia tradicional, metodologia ágil e a comparação de softwares similares no mercado, apresentados no capítulo 2. No capítulo 3 é explicado a metodologia de desenvolvimento utilizada neste projeto (Processo Unificado) e as tecnologias utilizadas. No capítulo 4 é explicado a arquitetura do software, bem como é apresentado um manual detalhado do funcionamento do sistema. O capítulo 5 é dedicado as considerações finais, resumindo o trabalho realizado e possíveis implementações futuras do software. Por último os apêndices, contemplando os seguintes itens: Diagrama de classes, Diagrama de Entidade-Relacionamento, Diagramas de Caso de Uso, Especificações de Caso de Uso e os Diagramas de Sequência.

2 FUNDAMENTAÇÃO TEÓRICA

O objeto deste capítulo é apresentar conceitos que fundamentam o software desenvolvido. Serão abordados os conceitos de projeto, gerência de projetos, metodologias tradicionais de desenvolvimento de software, metodologias ágeis e a comparação entre métodos ágeis e tradicionais. Também será apresentado um estudo de softwares existentes no mercado com propostas similares ao sistema Agile Manager.

2.1 PROJETO

Projetos podem ter os mais variados tamanhos, complexidade, áreas afetadas, número de envolvidos, estratégias e recursos utilizados. Um projeto pode depender de apenas um indivíduo e ser realizado em menos de um dia, como pode também envolver centenas de pessoas e ser executado ao longo de vários anos.

Projeto é um empreendimento não repetitivo, caracterizado por uma sequência clara e lógica de eventos, com início, meio e fim, que se destina a atingir um objetivo claro e definido, sendo conduzido por pessoas dentro parâmetros pré-definidos de tempo, custo, recursos envolvidos e qualidade. (VARGAS, 2003, p. 84).

Como exemplo de projetos podemos citar:

- Construção de uma casa
- Pavimentação de uma rodovia
- Instalação de uma nova malha elétrica
- Realização de uma viagem
- Elaboração de um plano de vendas
- Elaboração de um plano de marketing
- Desenvolvimento de um software

O processo de desenvolvimento de software contém complexidade, fatores e muitas variáveis envolvidas, sendo necessário um levantamento e detalhamento do

que deve ser implementado. Aplicação de gerenciamento de projeto é altamente utilizada.

2.2 GERÊNCIA DE PROJETOS

Considerando as mais variadas categorias de projetos e as particularidades que cada um pode envolver, a gerência de projetos torna-se uma ciência. Vários conhecimentos da área são adquiridos através da observação, identificação, pesquisa e explicação dos eventos envolvidos na gerência de um projeto.

Com o passar dos anos, novos tipos de projetos vieram a existir, e analogamente, outros caíram em desuso. Dos mais variados exemplos, temos construção das pirâmides do antigo Egito, feitas anteriormente a 2600 a.C., até a construção do maior acelerador de partículas do mundo em 2008, e outros exemplos surgem a cada dia. Sendo assim, é possível concluir que diferentes técnicas de gerenciamento de projetos foram construídas ao longo dos anos. Segundo o PMI, temos a seguinte definição de gerência de projetos:

O Gerenciamento de Projetos, portanto, é a aplicação de conhecimentos, habilidades e técnicas para a execução de projetos de forma efetiva e eficaz. Trata-se de uma competência estratégica para organizações, permitindo com que elas unam os resultados dos projetos com os objetivos do negócio – e, assim, melhor competir em seus mercados. (PMI Brasil, 2017)

Ricardo Vargas também cita a definição de gerência de projetos como:

O gerenciamento de projetos é um conjunto de ferramentas e técnicas que permitem que a empresa desenvolva um conjunto de habilidades, incluindo conhecimento e capacidades individuais, destinado ao controle de eventos não repetitivos, únicos e muitas vezes complexos, dentro de um cenário de tempo, custo e qualidade predeterminados. (VARGAS, 2016, p.7).

Em outras palavras, gerenciamento de projetos envolve uma série de técnicas e conhecimentos adquiridos empiricamente, visando o desenvolver de um projeto da

melhor forma possível, em busca de alcançar os objetivos para a organização que o executa.

2.3 MÉTODOS TRADICIONAIS

Os métodos tradicionais de gerenciamento de projetos de software foram os primeiros a serem implementados e até hoje tem muitos adeptos ao redor do mundo, sendo ainda metodologia mais utilizada. Os métodos tradicionais buscam a qualidade e satisfação do cliente apostando em processos bem definidos e documentados. Segundo Arakaki e Ribeiro (2006, p. 1595)

O planejamento detalhado e o processo disciplinado que orientam o gerenciamento de projetos tradicional na Engenharia do Software permitem a medição e o controle de todas as etapas do desenvolvimento de software e da equipe do projeto, onde cada membro tem o seu papel claramente definido e os artefatos gerados, em cada fase, são os registros da evolução do projeto. (ARAKAKI e RIBEIRO, 2006, p. 1595).

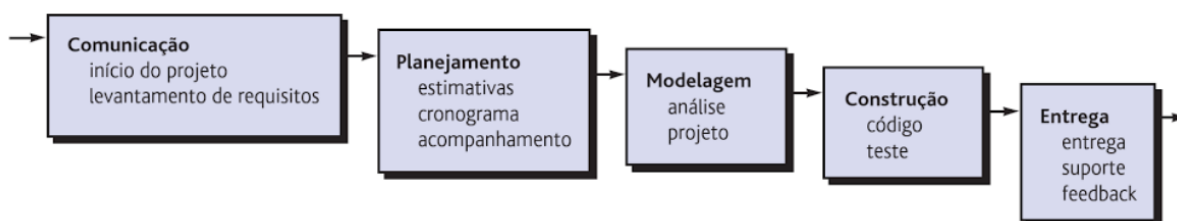
As metodologias tradicionais, também conhecidas como orientadas a documentação, métodos “pesados” ou até mesmo como métodos “prescritivos”, surgiram em um contexto bem diferente do atual, onde o desenvolvimento computacional era destinado em sua maioria a mainframes e terminais burros. Neste contexto, correções e alterações no projeto eram muito custosas, devido à falta de ferramentas de auxílio ao desenvolvimento e a limitação de acesso aos computadores. Sendo assim, a metodologia tradicional é baseada em um planejamento extenso e detalhado do projeto, com o intuito de levantar todos os requisitos, adiantar possíveis problemas e dificuldades e ser o mais assertivo em termos de prazo e custo, já na fase inicial do projeto. Cada etapa do projeto é detalhada e planejada minuciosamente, no intuito de evitar mudanças no decorrer do desenvolvimento.

Dentre os modelos tradicionais de desenvolvimento, o modelo cascata será utilizado para exemplificar o *mindset* tradicional.

2.3.1 Modelo Cascata

A metodologia de desenvolvimento cascata é o método mais antigo proposto na Engenharia de Software. A nomenclatura é dada devido aos processos serem propostos de maneira linear, sequencial e faseados, desenvolvendo o projeto de modo cascadeado, fase a fase. O projeto se inicia na fase de comunicação com o cliente, passando pelas fases de planejamento do projeto, modelagem do sistema, desenvolvimento e então a entrega do software concluído. Cada fase do projeto implica em suas devidas documentações e artefatos. Existe a dependência da conclusão da fase anterior para a iniciação da próxima, conforme (FIGURA 1).

FIGURA 1 - CICLO DE VIDA DO MODELO CASCATA



FONTE: PRESSMAN (2016).

Essa metodologia é melhor aplicada em cenários onde os requisitos não são variáveis, o cliente conhece sua real necessidade e a equipe de desenvolvimento compreende bem os problemas de cada requisito. Bem como é aplicável em alterações pontuais bem-definidas em um sistema já desenvolvido (Pressman, 2016).

Embora o modelo cascata proposto por Winston Royce em 1970 também preveja os chamados “feedback loops”, onde é possível a transição de mão dupla entre uma fase e outra, a grande maioria das organizações aderentes a esta metodologia a aplicaram de forma linear e contínua, assim evidenciando o *mindset* sequencial da época.

2.4 METODOLOGIA ÁGIL

Foi identificado na engenharia de software, que na grande maioria dos casos, não é possível levantar os requisitos em sua totalidade na fase de início do projeto.

Similarmente, nem sempre o cliente tem total ciência de suas necessidades, bem como a ordem de prioridades são adaptadas no decorrer do projeto. Essa instabilidade e variação dos requisitos, é uma característica bem comum no ramo de desenvolvimento de software. A metodologia ágil carrega em si um conjunto de valores e princípios, com o intuito de assegurar a satisfação do cliente adaptando o modo de desenvolvimento com este real cenário, gerando os artefatos realmente necessários para o projeto e mantendo o time de desenvolvimento motivado. Segundo Pressman (2016, p. 66)

A engenharia de software ágil combina filosofia com um conjunto de princípios de desenvolvimento. A filosofia defende a satisfação do cliente e a entrega incremental antecipada; equipes de projeto pequenas e altamente motivadas; métodos informais; artefatos de engenharia de software mínimos; e, acima de tudo, simplicidade no desenvolvimento geral. (PRESSMAN, 2016, p. 66).

Com o intuito de desenvolver software de maneira mais adaptada as realidades do mercado e necessidades dos clientes, a engenharia de software buscou se moldar a tais paradigmas e se aperfeiçoar para atendê-los. Foi então que em fevereiro de 2001 um grupo de 17 desenvolvedores levantaram o que se tornou o “grito de guerra” da metodologia ágil de desenvolvimento de software: o manifesto ágil. Neste está escrito:

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

Embora o manifesto ágil identifique valor nos itens a direita, valoriza ainda mais os itens a esquerda. O manifesto ágil também carrega consigo um conjunto de princípios que sustentam os valores presentes no manifesto. Toda metodologia ágil de desenvolvimento é criada baseada em tais valores e princípios.

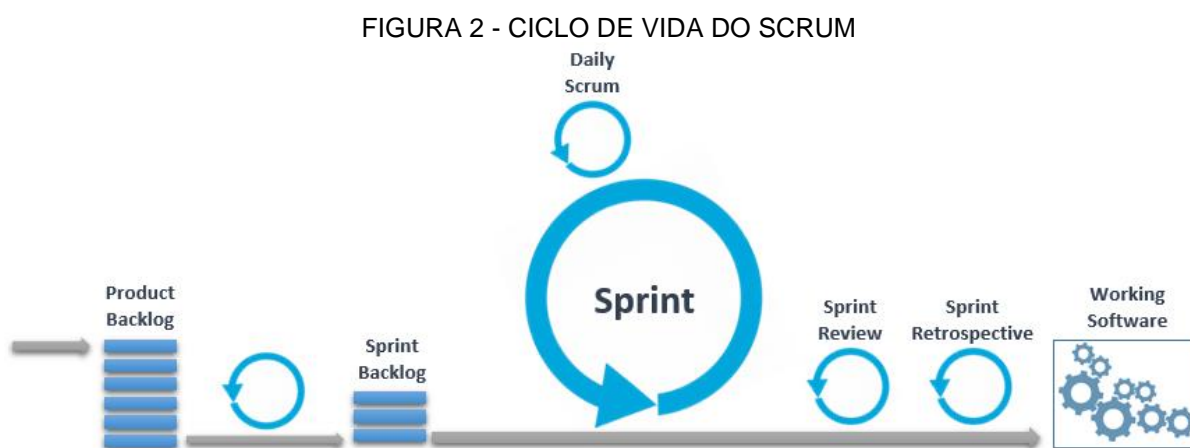
Dentre os modelos ágeis de desenvolvimento, utilizarei a metodologia Scrum para exemplificar o *mindset* ágil. O Scrum também foi utilizado como base para o desenvolvimento do software proposto neste trabalho.

2.4.1 Metodologia Scrum

O Scrum é um *framework* que adota os princípios ágeis de desenvolvimento e os incorpora no gerenciamento do projeto de maneira iterativa. O Scrum não tem como primícia levantar os requisitos de maneira clara e objetiva logo no início do projeto, mas sim, se adaptar as mudanças necessárias que surgem durante o desenvolvimento. Também, a equipe de desenvolvimento é treinada para absorver tais mudanças de modo positivo, pois é graças as mudanças que o software se adapta as reais necessidades do cliente, gerando um produto final de qualidade.

Visando a geração de artefatos minimalista presente na metodologia ágil, o Scrum exige somente o *backlog*. “O *backlog* é o principal artefato do Scrum [...]. É uma lista de todas as características, funções, tecnologias, melhorias e correções que constituem o produto a ser entregue” (CRUZ, 2015, p.61). Outros artefatos não são desprezados, todavia, devem ser desenvolvidos somente se agregam valor ao time ou ao cliente.

O Scrum tem um ciclo de vida bem dinâmico e adaptativo, contemplando eventos e cerimônias que auxiliam no desenvolvimento, conforme (FIGURA 2).



FONTE: SITECAMPUS (2017).

Dentro da metodologia Scrum, existem alguns papéis e responsabilidades atribuídos, sendo eles:

- **Scrum Master:** é o responsável por orientar e garantir que o time esteja aplicando a metodologia e aderido aos valores do Scrum. Também ajuda o time a se auto organizar, trabalhar a multidisciplinaridade e elimina

impedimentos que estejam desviando o foco do time. O scrum master **não** tem como papel gerenciar o time de desenvolvimento.

- Product Owner (PO): é o dono do produto, ou seja, aquele que compreende as regras de negócio, mantém os itens de *backlog* do produto e os prioriza. Também auxilia o time de desenvolvimento a compreender o *backlog* do produto, garantindo que a entrega seja de acordo com as expectativas do cliente. O PO representa o cliente nas decisões que dizem respeito ao negócio.
- Time de desenvolvimento: é o responsável por desenvolver os itens de backlog do produto e transformá-los em incrementos de funcionalidades, agregando ao produto final entregue ao cliente. Tem como característica ser multidisciplinar, multifuncional e auto organizável.

Além dos papéis, também é previsto na metodologia Scrum alguns eventos/cerimônias definidos para auxiliar o time, sendo eles:

- Sprint: é uma iteração do projeto, evento com duração fixa. O Scrum sugere duração de duas a quatro semanas. Tem objetivo e meta especificada, com o intuito de realizar o trabalho proposto. Na Sprint se realizam as outras cerimônias do Scrum.
- Planejamento: é a cerimônia onde é definido “o que será feito” na Sprint e “como será feito”.
- Revisão da Sprint: seu objetivo é a validação do PO, ou do cliente, dos itens de backlog desenvolvidos pelo time na Sprint em questão.
- Retrospectiva da Sprint: são levantados os pontos positivos e negativos notados na Sprint, e definidas ações de melhoria para a próxima. Essa cerimônia tem grande influência na melhoria constante do time.

2.5 METODOLOGIA TRADICIONAL OU ÁGIL?

Considerando os métodos tradicionais de desenvolvimento e também os métodos ágeis, existem diversas opções de *frameworks* de gerenciamento de projetos de software. Efetivamente, os engenheiros de software possuem várias abordagens disponíveis para gerenciar seus projetos, e essas possibilidades aumentam ainda mais se considerarmos modelos híbridos (unir um ou mais

frameworks, prática mais comum nos métodos ágeis) para o gerenciamento de projetos. Mas afinal de contas, qual é o melhor?

Um modelo de processo tradicional concentra-se em estruturar e ordenar o desenvolvimento de software. [...], mas os modelos tradicionais são adequados para o mundo do software que se alimenta de mudanças? Se rejeitarmos os modelos tradicionais e os substituímos por algo menos estruturado, tornaremos impossível atingir a coordenação e a coerência no trabalho de software? Não há respostas fáceis para essas questões. (PRESSMAN, 2016, p. 41).

De fato, não existe resposta única a essa pergunta, nem mesmo uma metodologia que sane todas as dificuldades presentes na engenharia de software. Todavia, dentre os modelos disponíveis para gerenciamento de projetos de software, existe aquele que mais se adequa às necessidades de determinado projeto, cumpra com as expectativas de determinado cliente ou se adapte melhor à cultura de determinada organização. Porém, é de consenso de vários autores (alguns citados neste trabalho) que os requisitos de um projeto de software tendem a ser instáveis e imprecisos, e os métodos ágeis tem se mostrado mais adaptativos nestes casos.

2.6 SOFTWARES DE GESTÃO DE PROJETOS

2.6.1 Microsoft Project

O software desenvolvido pela Microsoft, o Microsoft Project, promete recursos flexíveis para auxiliar no gerenciamento, eficiência e produtividade em projetos. A primeira versão do software foi lançada no ano de 1984, e desde então novas versões vem sendo lançadas no mercado, mostrando a maturidade que o produto tem para atender a necessidade de seu público. Dentre os recursos disponíveis no Project podemos citar o gráfico de Gantt, WBS, relatórios de medição de progresso, apresentações, multiusuários, integração com outros produtos Microsoft, entre outros.

A desvantagem para equipes ágeis é que o software não tem como proposta atender a tais metodologias, sendo assim seu uso limitado e não adequado para

esse público. O preço também pode não ser uma realidade para muitas equipes, variando de R\$ 2.999,00 na versão Standard a R\$ 4.999,00 na versão Professional.

2.6.2 Microsoft TFS

Outra ferramenta também desenvolvida pela empresa do Vale do Silício é o Microsoft Team Foundation Service (TFS). Este software possui diversos meios para auxiliar no gerenciamento do projeto, plug-ins e ferramentas para complementar o esforço do time de desenvolvimento. Ao contrário do Project, o Microsoft TFS tem vários princípios de metodologias ágeis incorporados na ferramenta. Dentre as funcionalidades deste software, podemos destacar o gerenciamento de tarefas (com recursos de metodologias ágeis), versionamento de código (integração com GIT), integração com IDEs de desenvolvimento, testes automatizados, build, relatórios e estatísticas, entre outros.

Assim como o Microsoft Project, a desvantagem do TFS também se refere ao custo, que pode não ser acessível para alguns times de desenvolvimento: US\$ 499,00 para até cinco usuários. Embora ferramentas pagas tenham suas vantagens (que em muitos casos fazem valer o investimento), muitos times ágeis optam por ferramentas gratuitas ou com preços mais acessíveis.

2.6.3 Jira

O Jira Software foi desenvolvido pela Atlassian, focado especificamente para equipes de desenvolvimento que utilizam metodologias ágeis. É o software de gestão de projetos mais utilizado por este público. Conta com recursos de quadros do Scrum, quadros do Kanban, relatórios ágeis, filtros personalizados, integração com Bitbucket ou Github, entre outros.

A vantagem deste software em relação as demais anteriormente apresentados, é o fato de ter sido desenvolvido especificamente para equipes de desenvolvimento que utilizam metodologias ágeis, o público alvo deste trabalho. Também apresenta custo acessível, a partir de R\$ 10,00 mensais para até 10 usuários.

2.6.4 Trello

A ferramenta web Trello não é exatamente um gerenciador de projetos de software, mas sim possui a proposta de ser especificamente um gerenciador de tarefas. Porém, em alguns casos pode ser utilizada para gerenciar as tarefas de um projeto de software.

O Trello disponibiliza a criação de quadros, com as tarefas organizadas em colunas, bem similar ao quadro de Kanban. É possível também vincular usuários responsáveis as tarefas, criar alertas, montar checklists, anexar imagens, entre outros recursos.

A vantagem desta ferramenta se dá na simplicidade e rápido aprendizado, bem como por ser uma ferramenta gratuita.

Neste capítulo foram apresentados conceitos e definições que fundamentam o software proposto. Por se tratar de um projeto de engenharia de software, o tema escolhido foi um gerenciador de projetos, utilizando como fundamento entendimentos de projeto, gerência de projetos e metodologia tradicionais e ágeis de desenvolvimento, que são áreas de estudo da própria engenharia de software. Além disso, o tema proposto contempla a parte técnica de programação da engenharia de software. Neste capítulo também foram apresentados softwares similares existentes.

3 MATERIAIS E MÉTODOS

Este capítulo tem como objetivo explicar a metodologia utilizada (Processo Unificado) no desenvolvimento deste trabalho e como ela foi aplicada. Também há a apresentação das tecnologias empregadas no software proposto.

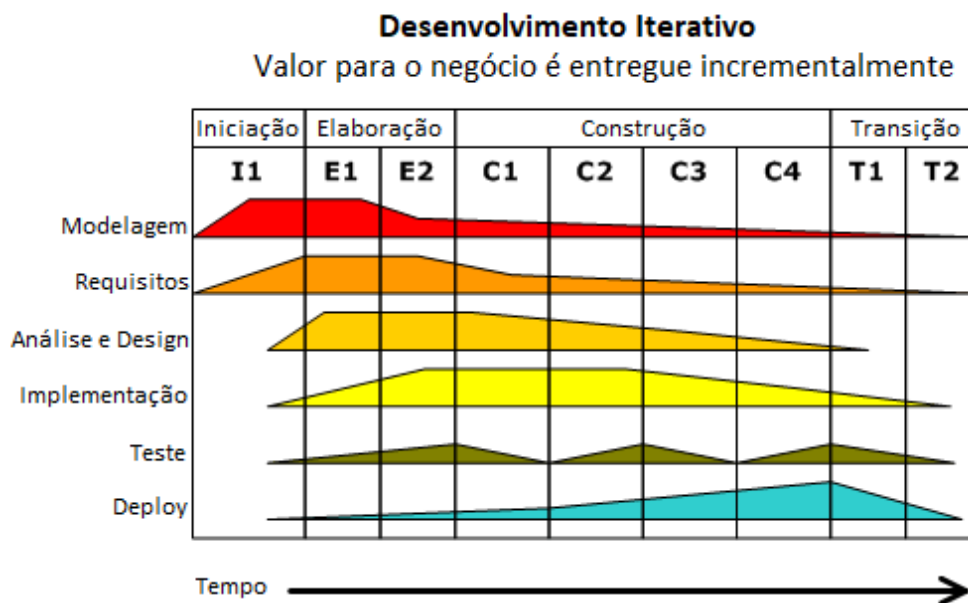
3.1 METODOLOGIA UTILIZADA

3.1.1 Processo Unificado

Como requisito parcial deste trabalho, a metodologia aplicada no desenvolvimento é o Processo Unificado (PU). Neste tópico explico cada etapa do PU, exemplificando cada fase com as ações tomadas no desenvolvimento deste trabalho.

O PU é o empreendimento de unir as melhores abordagens e práticas da metodologia tradicional, porém, aplica-os de modo semelhante a práticas presentes na metodologia ágil. O Processo Unificado utiliza um fluxo de processos evolucionário, iterativo e incremental. Também destaca a importância da comunicação com o cliente, além de enfatizar o desenvolvimento à arquitetura do software. O Processo Unificado utiliza quatro fases básicas para compor cada iteração: iniciação, elaboração, construção e transição (FIGURA 3).

FIGURA 3 - FASES DO PROCESSO UNIFICADO



FONTE: Adaptado de WIKIMEDIA (2017)

Iniciação: inclui a participação dos envolvidos e comunicação com o cliente, para fins de levantamento de requisitos, identificação de necessidades de negócio, definições de arquitetura e planejamento das atividades. Neste trabalho, na fase de iniciação, foram levantados os requisitos, desenvolvido a visão do sistema (APÊNDICE A) e a validação da proposta com a orientadora, os casos de uso negocias (APÊNDICE B), regras de negócio (APÊNDICE D) e o gráfico de Gantt referente ao planejamento do desenvolvimento (FIGURA 4).

Elaboração: engloba as atividades de modelagem, refina e expande levantamentos preliminares realizados na fase de iniciação. Neste trabalho, na fase de elaboração, o diagrama de caso de uso foi expandido e descrito (APÊNDICE G), elaborado o diagrama de classes negocial (APÊNDICE F) e o diagrama entidade-relacionamento (APÊNDICE K). A cada iteração, os artefatos gerados nas fases de iniciação e elaboração eram amadurecidos, atualizados e complementados conforme a necessidade. No decorrer das iterações, o diagrama de classes negocial era preenchido com atributos (APÊNDICE H) e com os métodos (APÊNDICE J).

Construção: nessa fase, a partir do modelo de arquitetura resultado das fases anteriores, são desenvolvidos os componentes do software. Implementa-se no código-fonte as funções e recursos necessários para gerar incremento ao software. Também é na fase de construção que os testes são executados. Neste trabalho, na fase de construção, foram desenvolvidas as classes levantadas na etapa anterior, de

acordo com a arquitetura proposta. Bem como a implementação dos métodos necessários para gerar incremento ao software e o desenvolvimento das telas. Eram executados os testes de validação do componente de software desenvolvido e testes de integração entre a nova funcionalidade com as outras desenvolvidas em iterações anteriores.

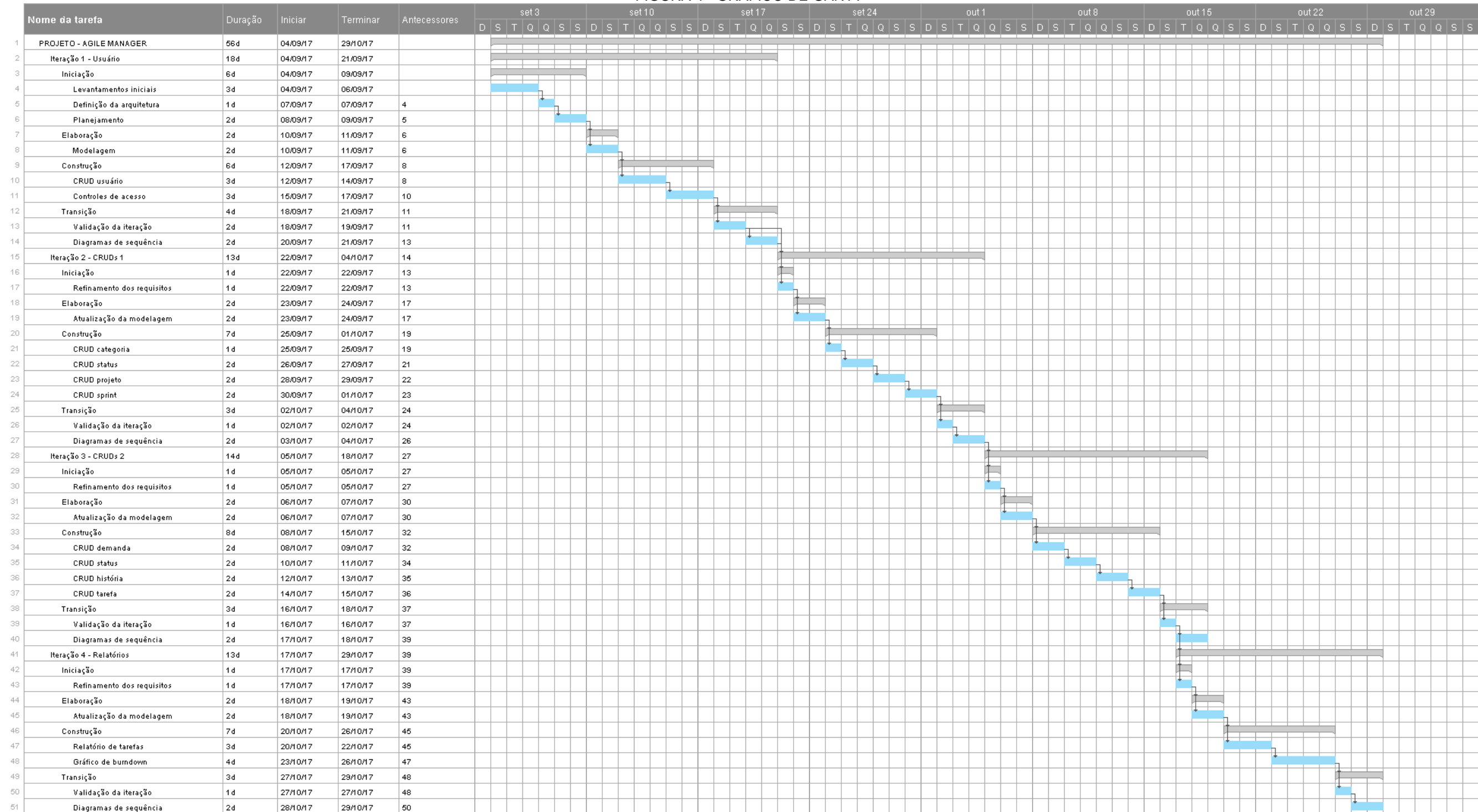
Transição: tem por objetivo a avaliação e feedback do usuário, para validar os componentes de software desenvolvidos e relatar possíveis mudanças. Além disso, é neste momento em que a equipe desenvolve manuais, arquivos de ajuda, tutorial de instalação, entre outros itens referentes. No fim da fase de transição, o incremento torna-se, de fato, uma versão utilizável do software. Neste trabalho, a fase de transição era validar com a orientadora o incremento desenvolvido e avaliar feedbacks recebidos de colegas. Também na fase de transição era gerado como artefato o diagrama de sequência referente a funcionalidade desenvolvida da UML (APÊNDICE H).

Cada iteração do PU contém as fases citadas acima. No desenvolvimento deste trabalho, cada iteração tinha por objetivo entregar um ou mais casos de uso em funcionamento, como um incremento em versão utilizável do software.

3.1.2 Gráfico de Gantt

Para fins de planejamento e acompanhamento de cronograma de entregas, foi utilizado o gráfico de Gantt (FIGURA 4). Este método permite listar as tarefas necessárias para a realização do projeto, definir tempo para realização de cada tarefa, dependências entre elas, recursos responsáveis por determinada tarefa, visualizar graficamente a evolução do projeto, entre outras facilidades.

FIGURA 4 - GRÁFICO DE GANTT



FONTE: O autor (2017)

3.2 MATERIAIS

Neste subcapítulo serão apresentadas todas as tecnologias utilizadas no desenvolvimento deste projeto.

Em termos de arquitetura, o sistema Agile Manager conta com três frameworks principais: Hibernate, Spring e Apache Click.

O Hibernate é responsável por mapear as entidades do sistema de modo a facilitar a interação com a camada de persistência. Tanto para métodos de consulta, inserção e *update* das informações no banco de dados, como também para geração de scripts de criação das tabelas e relacionamentos.

O Spring auxilia na injeção de dependência das classes, permitindo a construção de uma aplicação mais desacoplada.

O framework Apache Click é utilizado para construção das páginas web, *server-side*, e controle de acessos via herança de *pages* (FIGURA 48). O Click também permite a criação de componentes HTML, JavaScript e CSS através do Java. Os elementos e objetos criados, são acessíveis na view através do Apache Velocity, que fornece uma linguagem para objetos de referência definidos no código Java.

Também foi utilizado o padrão Visual Object (FIGURA 47) para manipulação de objetos com atributos do tipo primário, onde não havia possibilidade ou necessidade de utilizar objetos dependentes.

A tabela a seguir (TABELA 1) lista as tecnologias utilizadas, com uma breve descrição de sua utilidade.

Tabela 1 - TECNOLOGIAS UTILIZADAS

continua

| Tecnologia | Utilização no sistema |
|-------------------|---|
| JAVA | Linguagem de programação base do projeto |
| JAVASCRIPT | Criação de scripts no <i>front-end</i> do sistema web |
| CSS | Alteração dos estilos das páginas web |
| HTML | Estruturar conteúdo web |
| HIBERNATE | Interação mapeada com o banco de dados |
| APACHE CLICK | Construção das páginas web e controle de acessos |
| APACHE VELOCITY | Referenciar na view objetos definidos no código Java |

| Tecnologia | Utilização no sistema |
|-----------------|--|
| MYSQL | Banco de dados utilizado |
| TOMCAT | Servidor de aplicação |
| ECLIPSE | IDE utilizada para codificação do projeto |
| BOOTSTRAP | Framework web utilizado para facilitar o <i>front-end</i> do projeto |
| JQUERY | Utilizado para auxiliar na criação de <i>scripts</i> da parte web |
| AJAX | Atualização das páginas web sem a necessidade de carregá-las novamente |
| MAVEN | Ferramenta para o gerenciamento de dependências (bibliotecas externas) |
| MVC | Padrão de arquitetura utilizado para estruturar o projeto |
| DESIGN PATTERNS | Padrões de projeto, que visam estruturar melhor o projeto e evitar erros recorrentes no processo de desenvolvimento |
| VISUAL OBJECT | Objeto que contém apenas atributos com tipo primário, utilizado em ocasiões onde não é possível/desejável utilizar objetos dependentes |
| HQL | SQL orientado a objetos utilizado pelo framework Hibernate. |
| SQL | Linguagem de programação utilizada para criação de <i>queries</i> |
| GIT | Utilizado para o controle de versão do projeto |
| BITBUCKET | Software utilizado para o controle de versão do projeto |
| DATEPICKER | Otimizar a entrada de dados do tipo date |
| DATATABLES | Paginação e campo de busca para listas |
| SELECT2 | Otimização das <i>comboboxs</i> presentes em grande parte do sistema web |
| SMART SHEET | Desenvolvimento do gráfico de Gantt |
| OBJECT AID | Plugin para o Eclipse, utilizado na criação do diagrama de classes |
| LUCID CHART | Criação dos diagramas de casos de uso |
| STARUML | Criação dos diagramas de sequência |

No presente capítulo, foram apresentadas as metodologias utilizadas no desenvolvimento deste trabalho (Processo Unificado como modelo e Gantt como gerenciador de cronograma). Adicionalmente, a explicação da arquitetura do sistema e uma tabela com as principais tecnologias e frameworks utilizados na programação do software proposto. No próximo capítulo, é apresentado o sistema Agile Manager, com *print screens* e tutoriais de utilização do sistema.

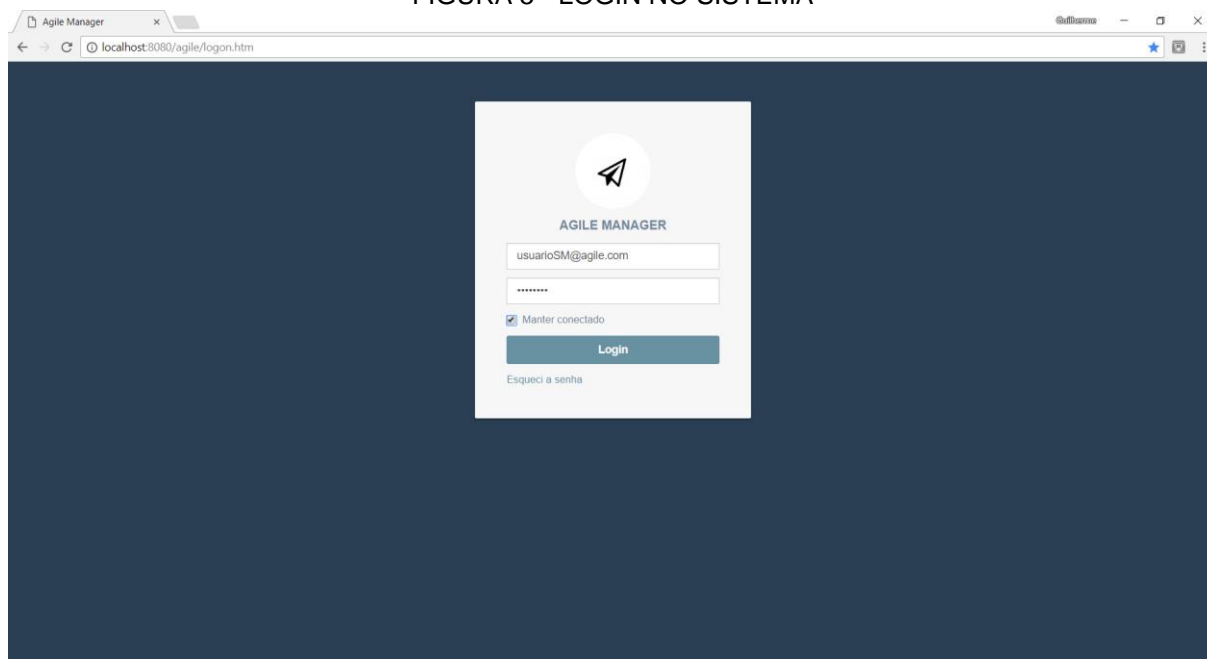
4 APRESENTAÇÃO DO SISTEMA

4.1 CADASTROS BÁSICOS

4.1.1 Login

O sistema Agile Manager conta com três visões: Scrum Master, Product Owner e Desenvolvedor/Tester. Para utilização do software, o usuário necessita ter uma conta cadastrada. O Agile Manager vem com uma conta pré-cadastrada padrão de Scrum Master (a única visão com permissão de cadastrar usuários), assim, o usuário responsável pode cadastrar os demais no sistema. Para realizar o login, o usuário deve inserir as informações de e-mail e senha, nos campos determinados, conforme (FIGURA 5). Clicar no botão “Login”.

FIGURA 5 - LOGIN NO SISTEMA



FONTE: O autor (2017)

4.1.2 Cadastro de usuários

A ferramenta conta com um dashboard, onde se encontram as principais ações e atalhos disponíveis para o usuário. Para cadastrar um novo usuário, deve-

se acessar o menu “Configurações”, submenu “Cadastro de usuários” (disponível somente para Scrum master), conforme (FIGURA 6). Preencher os campos destinados a “Nome completo”, “E-mail”, “Senha” e “Função” (relacionada ao nível de permissão do novo usuário). Clicar no botão “Gravar”.

FIGURA 6 - CADASTRO DE USUÁRIO

Dados do usuário

Nome completo *

E-mail *

Senha *

Função

Usuários cadastrados

Mostrar registros Buscar:

| Nome | E-mail | Função | Ação |
|----------------|-------------------------|---------------|----------------------------------|
| Usuario Dev | usuarioDev@agile.com | Desenvolvedor | <input type="button" value="✎"/> |
| Usuario Tester | usuarioTester@agile.com | Tester | <input type="button" value="✎"/> |
| Usuario PO | usuarioPO@agile.com | Product Owner | <input type="button" value="✎"/> |
| Usuario SM | usuarioSM@agile.com | Scrum Master | <input type="button" value="✎"/> |

FONTE: O autor (2017)

Esta tela também conta com a lista de usuários cadastrados, exibidos na grid na parte inferior da tela. Para editá-los, clicar no ícone de edição e preencher os campos citados anteriormente.

4.1.3 Cadastro de status

Para fins de acompanhamento das situações em que as tarefas, sprints, demandas, entre outras entidades do sistema se encontram, o Agile Manager conta com o cadastro de Status (FIGURA 7), presente no menu “Configurações” (disponível somente para Scrum master). Este parâmetro também é utilizado para fins de relatório (FIGURA 18) e burndown (FIGURA 19). Os status não são pré-definidos no sistema pois cada equipe tem seu próprio fluxo, partindo, por exemplo, de “to do, doing, done” até *workflows* mais complexos. O campo “Descrição” é o que será mostrado para o usuário em outros cadastros que utilizam status. O campo

“Conclui tarefa?” o sistema utiliza para identificar quais status tem o conceito de entrega, que encerram o ciclo de vida de determinada tarefa.

FIGURA 7 - CADASTRO DE STATUS

The screenshot shows the 'Status' registration page in Agile Manager. The page has a dark sidebar with navigation options like 'Home', 'Projetos', 'Relatórios', and 'Configurações'. The main content area is titled 'Status' and contains a form for 'Cadastro de status'. The form has a text input for 'Descrição *' and a checkbox for 'Conclui tarefa?'. Below the form are 'Gravar' and 'Cancelar' buttons. Underneath the form is a section for 'Status cadastrados' with a search bar and a table of existing statuses.

| Cód. | Descrição | Conclui tarefa | Ação |
|------|---------------------------|----------------|------|
| 1 | Novo | false | |
| 2 | Em andamento | false | |
| 3 | Em desenvolvimento | false | |
| 4 | Em testes | false | |
| 5 | Desenvolvimento concluído | true | |
| 6 | Teste concluído | true | |

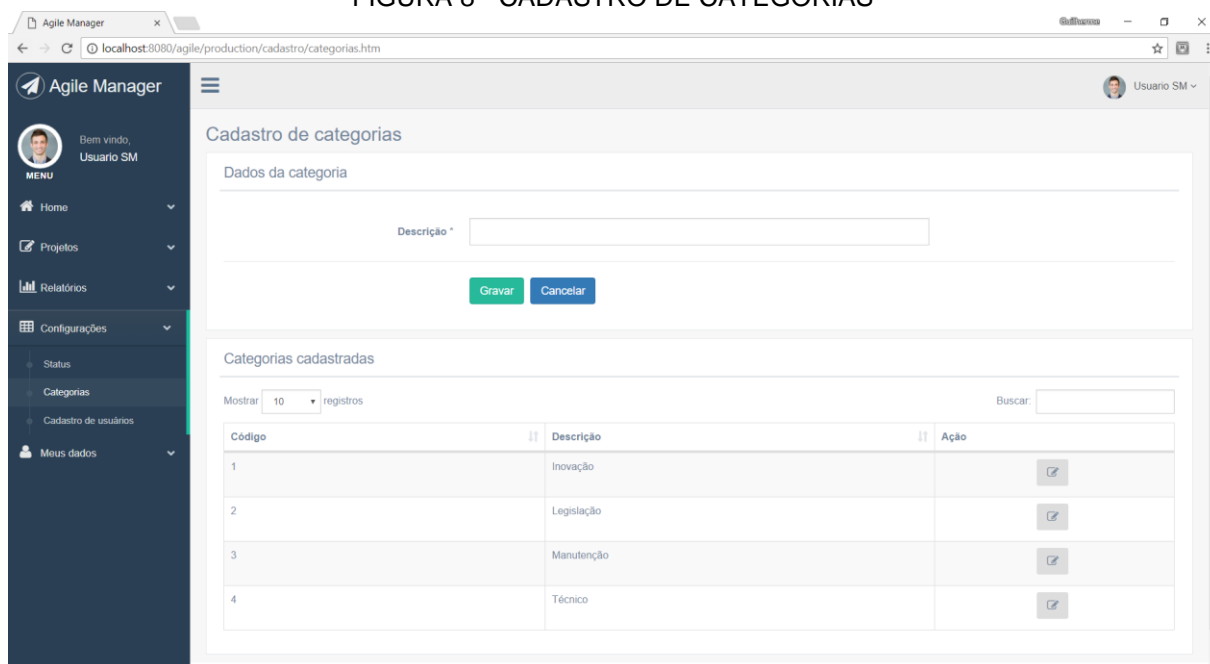
FONTE: O autor (2017)

Esta tela também conta com a lista de status cadastrados, exibidos na grid na parte inferior da tela. Para editá-los, clicar no ícone de edição e preencher os campos citados anteriormente.

4.1.4 Cadastro de categorias

Este cadastro é utilizado pelo sistema para categorizar os projetos do usuário. Cada equipe possui suas frentes de trabalho, categorias de projeto, divisão de clientes, entre outras formas de dispor em categorias seus projetos. Os cadastros de categorias têm a finalidade de se adequar aos moldes da equipe utilizadora do sistema. O usuário deve preencher o campo “Descrição” e clicar no botão “Gravar”, conforme exibido na (FIGURA 8). Disponibilizado no menu “Configurações” (disponível somente para Scrum master).

FIGURA 8 - CADASTRO DE CATEGORIAS



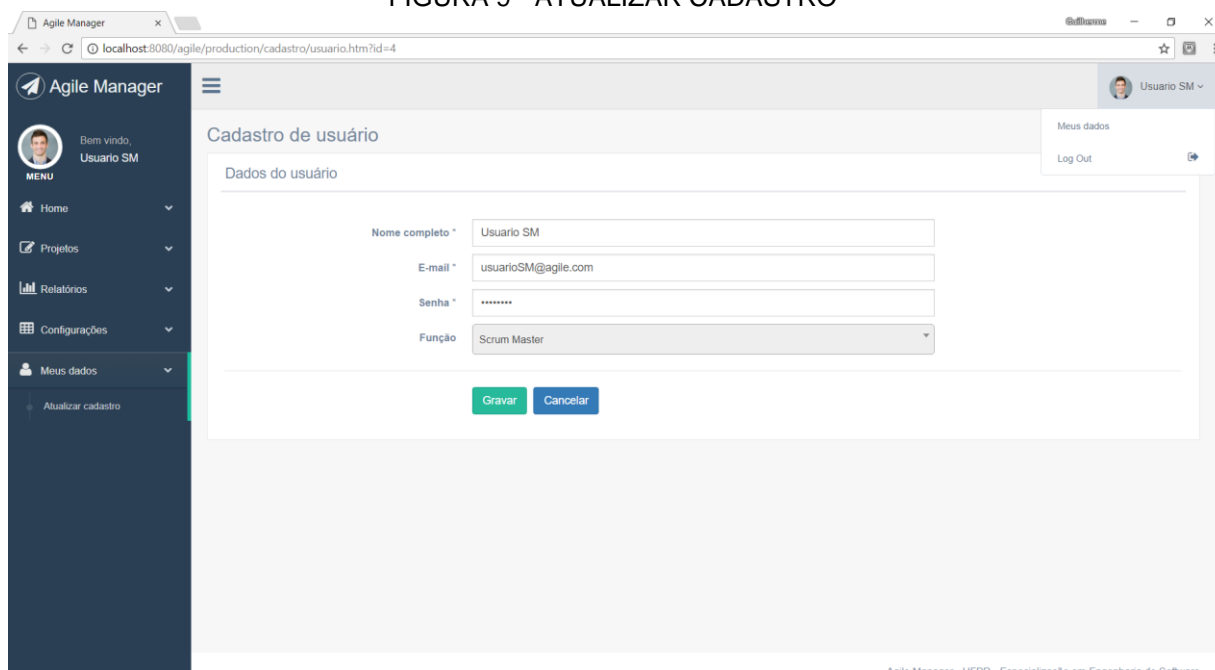
FONTE: O autor (2017)

Esta tela também conta com a lista de categorias cadastradas, exibidas na grid na parte inferior da tela. Para editá-los, clicar no ícone de edição e preencher os campos citados anteriormente.

4.1.5 Atualizar cadastro

Disponível no menu “Meus dados”, submenu “Atualizar cadastro”. Essa opção também se encontra localizada no canto superior direito da tela, “<Nome do usuário>”, “Meus dados”. Nesta seção (FIGURA 9), usuário pode atualizar seus dados cadastrais: “Nome completo”, “E-mail” e “Senha”. O campo de “Função” se encontra desabilitado. O usuário não pode alterar sua própria função, porém o Scrum master pode alterar funções de outros usuários (exibido no item 4.1.2 deste documento). Para efetivar as alterações, o usuário deve clicar no botão “Gravar”.

FIGURA 9 - ATUALIZAR CADASTRO



FONTE: O autor (2017)

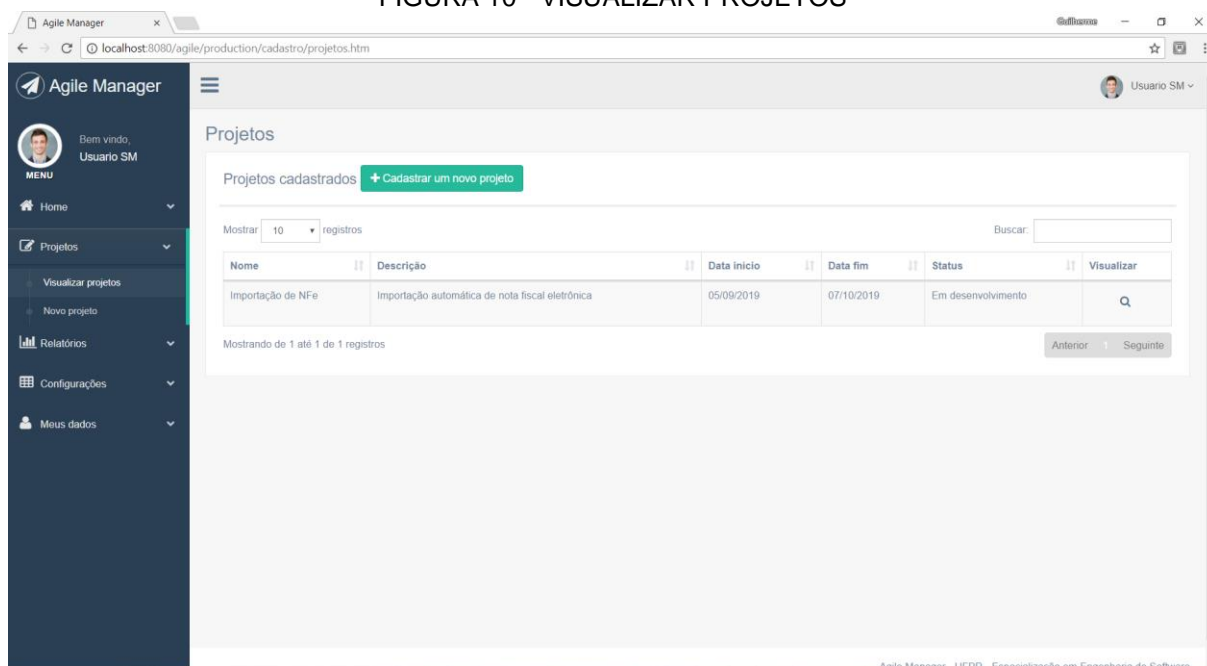
4.2 GERENCIAMENTO DE PROJETOS

O Agile Manager conta com a seguinte hierarquia de entidades para gerenciamento de projetos: projeto, sprint, demanda (também conhecido na metodologia ágil como épico), história, tarefa e apontamento de tarefa. Estes itens são explicados nos tópicos a seguir.

4.2.1 Visualizar projetos

No item de menu “Projetos” se encontram as ações disponíveis para projetos. No submenu “Visualizar projetos” o sistema mostra uma lista com os projetos cadastrados no sistema (FIGURA 10). Ao clicar no item “Visualizar”, o sistema direciona o usuário para a tela de informações/cadastro de projeto (FIGURA 11).

FIGURA 10 - VISUALIZAR PROJETOS



FONTE: O autor (2017)

4.2.2 Projeto

Para poder cadastrar um novo projeto no sistema (FIGURA 11), o usuário deve estar logado como Scrum master. A visualização dos dados está disponível para os outros usuários. O cadastro de projeto se encontra no menu “Projetos”, submenu “Novo projeto”. Também disponível no botão “Cadastrar um novo projeto”, conforme (FIGURA 10). O usuário deve preencher os campos “Nome do projeto”, “Descrição”, “Data início”, “Data término”, “Categoria” (conforme item 4.1.4 deste documento), “Status” (conforme item 4.1.3 deste documento), e “Observação”. Para gravar no banco de dados, clicar no botão “Gravar”. Na parte inferior desta tela, o sistema lista as sprints cadastradas vinculadas a este projeto.

FIGURA 11 - CADASTRO DE PROJETO

Dados do projeto

Nome do projeto *

Descrição

Data início * Data término

Categoria

Status

Observação

Sprints

Mostrar registros

| Cód. | Descrição | Data início | Data fim | Status | Ações |
|------|--------------------------------|-------------|------------|--------------|---|
| 1 | Sprint Burndown ideal | 01/09/2017 | 11/09/2017 | Concluído | <input type="button" value="Q"/> <input type="button" value="LMT"/> |
| 2 | Sprint Burndown com diferenças | 15/09/2017 | 25/09/2017 | Concluído | <input type="button" value="Q"/> <input type="button" value="LMT"/> |
| 3 | Sprint Burndown em andamento | 02/10/2017 | 14/10/2017 | Em andamento | <input type="button" value="Q"/> <input type="button" value="LMT"/> |

FONTE: O autor (2017)

4.2.3 Sprint

Para cadastrar uma sprint, o usuário deve estar logado como Scrum master. Para acessar a tela de cadastro de sprint, o usuário deve clicar no botão “Cadastrar uma nova sprint”, conforme (FIGURA 11). Para visualizar os dados de uma sprint, o usuário deve clicar no ícone de visualização, conforme (FIGURA 11). Ao acessar a tela de cadastro de sprint (FIGURA 12), o usuário deve preencher os campos “Descrição”, “Data início”, “Data término”, “Status” e “Observação”. Após os campos preenchidos, clicar no botão “Gravar”. Na parte inferior desta tela, o sistema lista as demandas cadastradas vinculadas a esta sprint.

FIGURA 12 - CADASTRO DE SPRINT

Agile Manager

Bem vindo
Usuário SM

MENU

- Home
- Projetos
- Relatórios
- Configurações
- Meus dados

Projeto: Importação de NFe

Sprint

Descrição * Sprint Burndown com diferenças

Data início * 15/09/2017 Data término 25/09/2017

Status Concluído

Observação

Gravar Cancelar Imt Burndown

Demandas + Cadastrar uma nova demanda

Mostrar 10 registros

Buscar:

| Cód. | Descrição | Status | Visualizar |
|------|-----------------------|-----------|------------|
| 2 | Demanda com histórias | Concluído | Q |

Mostrando de 1 até 1 de 1 registros

Anterior Seguinte

Agile Manager - IIFPR - Especialização em Engenharia de Software

FONTE: O autor (2017)

4.2.4 Demanda

Para cadastrar uma demanda, o usuário deve estar logado como Scrum master ou Product Owner. Para acessar a tela de cadastro de demanda, o usuário deve clicar no botão “Cadastrar uma nova demanda”, conforme (FIGURA 12). Para visualizar os dados de uma demanda, o usuário deve clicar no ícone de visualização, conforme (FIGURA 12). Ao acessar a tela de cadastro de demanda (FIGURA 13), o usuário deve preencher os campos “Descrição”, “Status” e “Observação”. Após os campos preenchidos, clicar no botão “Gravar”. Na parte inferior desta tela, o sistema lista as histórias cadastradas vinculadas a esta demanda.

FIGURA 13 - CADASTRO DE DEMANDA

Agile Manager

Bem vindo,
Usuário SM

MENU

- Home
- Projetos
- Relatórios
- Configurações
- Meus dados

Projeto: Importação de NFe
Sprint: Sprint Burndown com diferenças

Dados da demanda

Descrição * Demanda com histórias

Status Concluído

Observação

Gravar Cancelar

Histórias + Cadastrar uma nova história

Mostrar 10 registros

Buscar:

| Cód. | Descrição | Status | Visualizar |
|------|----------------------|-----------|------------|
| 3 | História com tarefas | Concluído | Q |

Mostrando de 1 até 1 de 1 registros

Anterior Seguinte

FONTE: O autor (2017)

4.2.5 História

Para cadastrar uma história, o usuário deve estar logado como Scrum master ou Product Owner. Para acessar a tela de cadastro de história, o usuário deve clicar no botão “Cadastrar uma nova história”, conforme (FIGURA 13). Para visualizar os dados de uma história, o usuário deve clicar no ícone de visualização, conforme (FIGURA 13). Ao acessar a tela de cadastro de história (FIGURA 14), o usuário deve preencher os campos “Descrição”, “Status” e “Observação”. Após os campos preenchidos, clicar no botão “Gravar”. Na parte inferior desta tela, o sistema lista as tarefas vinculadas a esta história.

FIGURA 14 - CADASTRO DE HISTÓRIA

Agile Manager

Bem vindo, Usuario SM

Projeto: Importação de NFe
Sprint: Sprint Burndown com diferenças
Demanda: Demanda com histórias

Dados da história

Descrição: História com tarefas

Status: Concluído

Observação:

Gravar Cancelar

Tarefas + Cadastrar uma nova tarefa

Mostrar 10 registros

Buscar:

| Cód. | Descrição | Observação | Status | Tempo estimado | Visualizar |
|------|-----------|------------|---------------------------|----------------|------------|
| 11 | Tarefa 01 | | Desenvolvimento concluído | 10 | Q |
| 12 | Tarefa 02 | | Teste concluído | 10 | Q |
| 13 | Tarefa 03 | | Concluído | 10 | Q |
| 14 | Tarefa 04 | | Desenvolvimento concluído | 10 | - |

FONTE: O autor (2017)

4.2.6 Tarefa

Esta funcionalidade está disponível para todas as visões do sistema, basta o usuário estar logado. É com o cadastro de tarefas que os usuários vão imputar no sistema quais as atividades necessárias para desenvolver os componentes do projeto, bem como tarefa de testes, análises técnicas, ou qualquer outra atividade que deva ser prevista, de acordo com o *workflow* do projeto. Para acessar a tela de cadastro de tarefa, o usuário deve clicar no botão “Cadastrar uma nova tarefa”, conforme (FIGURA 14). Para visualizar os dados de uma tarefa, o usuário deve clicar no ícone de visualização, conforme (FIGURA 14). Ao acessar a tela de cadastro de tarefa (FIGURA 15), o usuário deve preencher os campos “Descrição”, “Tempo estimado”, “Data prevista”, “Status”, “Usuário solicitante”, “Usuário atribuído” e “Observação”. Após os campos preenchidos, clicar no botão “Gravar”. Na parte inferior desta tela, o sistema lista os apontamentos vinculadas a esta tarefa.

FIGURA 15 - CADASTRO DE TAREFA

The screenshot displays the 'Cadastro de Tarefa' (Task Registration) interface in Agile Manager. The page title is 'Projeto: Importação de NFe'. The breadcrumb trail shows: 'Projeto: Importação de NFe' > 'Sprint: Sprint Burndown com diferenças' > 'Demanda: Demanda com histórias' > 'História: História com tarefas'.

The main form, titled 'Dados da tarefa', contains the following fields:

- Descrição:** Tarefa 01
- Tempo estimado:** 10.0
- Data prevista:** 03/10/2017
- Status:** Desenvolvimento concluído
- Solicitante:** Usuario Dev
- Atribuído:** Usuario Dev
- Observação:** (Empty text area)

Buttons for 'Gravar' (Save) and 'Cancelar' (Cancel) are located below the form.

Below the form is a section for 'Apontamentos' (Appointments) with a '+ Cadastrar apontamento' button. It includes a 'Mostrar 10 registros' dropdown and a search box. A table with the following columns is visible:

| Data | Usuario | Observação | Tempo Gasto | Visualizar |
|------|---------|------------|-------------|------------|
| | | | | |

FONTE: O autor (2017)

É com base no campo “Tempo estimado” (preenchido em horas) que o sistema irá gerar o gráfico de burndown (FIGURA 19). Quando o status da tarefa é alterado para um status que indica o fechamento desta atividade, o sistema guarda no banco de dados a data em que a tarefa foi finalizada. Esta informação também é utilizada no burndown. Os campos “Usuário solicitante” e “Usuário atribuído” são para fins de relatório (FIGURA 19).

4.2.7 Apontamentos (Tempo gasto)

No cadastro de apontamentos, os usuários inserem no sistema quais atividades trabalharam e quanto tempo foi dedicado. Para acessar a tela de cadastro de apontamentos, o usuário deve clicar no botão “Cadastrar apontamento”, conforme (FIGURA 15). Para visualizar os dados de um apontamento, o usuário deve clicar no ícone de visualização, conforme (FIGURA 15). Ao acessar a tela de cadastro de Tempo gasto (FIGURA 16), o usuário deve preencher os campos “Tempo gasto (horas)”, “Data” e “Observação”. Após os campos preenchidos, clicar no botão “Gravar”.

FIGURA 16 - CADASTRO DE APONTAMENTO

The screenshot displays the Agile Manager interface for recording a time log. The page title is 'Projeto: Importação de NFe'. The breadcrumb trail shows: 'Sprint: Sprint Burndown com diferenças' > 'Demanda: Demanda com histórias' > 'História: História com tarefas' > 'Tarefa: Tarefa 01'. The main form, titled 'Tempo gasto', contains the following fields and controls:

- Tempo gasto (horas):** A text input field containing the value '15.0'.
- Data:** A date picker field showing '03/10/2017'.
- Observação:** A text area containing the text 'Duração da tarefa além do estimado'.
- Buttons:** Two buttons at the bottom: 'Gravar' (Save) in blue and 'Cancelar' (Cancel) in green.

The left sidebar shows the user profile 'Usuario SM' and a menu with options: Home, Projetos, Relatórios, Configurações, and Meus dados. The browser address bar shows the URL: localhost:8080/agile/production/cadastro/tarefaApontamentos.htm?id=2&idTarefa=11.

FONTE: O autor (2017)

4.3 RELATÓRIOS

Neste subtópico, será mostrado os relatórios disponíveis no sistema.

4.3.1 Minhas atividades

A tela “Minhas tarefas” (FIGURA 17) é a *home page* dos usuários do sistema, após logado. Também disponível no menu “Home”, submenu “Minhas atividades”. Esta tela é responsável por listar as tarefas do usuário que estão em andamento. Conta com um botão de “Visualizar”, que redireciona o usuário à história onde a tarefa está alocada. Disponível para todas as visões do sistema, basta o usuário estar logado.

FIGURA 17 - MINHAS TAREFAS

The screenshot shows the 'Minhas tarefas' page in the Agile Manager application. The page features a table with the following data:

| Descrição | Status | Projeto | Demanda | História | Visualizar |
|-----------|--------|-------------------|-----------------------|----------------------|------------|
| Tarefa 03 | Novo | Importação de NFe | Demanda com histórias | História com tarefas | Q |
| Tarefa 05 | Novo | Importação de NFe | Demanda com histórias | História com tarefas | Q |
| Tarefa 06 | Novo | Importação de NFe | Demanda com histórias | História com tarefas | Q |

The interface also includes a sidebar menu with options like Home, Minhas atividades, Projetos, Relatórios, and Meus dados. A search bar is located at the top right of the table area, and pagination controls are at the bottom right, showing 'Mostrando de 1 até 3 de 3 registros'.

FONTE: O autor (2017)

4.3.2 Relatório de tarefas

Este relatório tem o intuito de ser dinâmico e adaptativo aos padrões da equipe. O usuário pode filtrar as tarefas de acordo com os filtros de: projeto, sprint, demanda, história, usuário atribuído e status. Para aplicar o filtro desejado, clicar no botão “Buscar”. O sistema exibe para o usuário as tarefas de acordo com o filtro desejado. Conta com um botão de “Visualizar”, que redireciona o usuário à história onde a tarefa está alocada. Disponível no menu “Relatórios”, submenu “Tarefas”. Disponível para todas as visões do sistema, basta o usuário estar logado.

FIGURA 18 - RELATÓRIO DE TAREFAS

Mostrar 10 registros

Buscar:

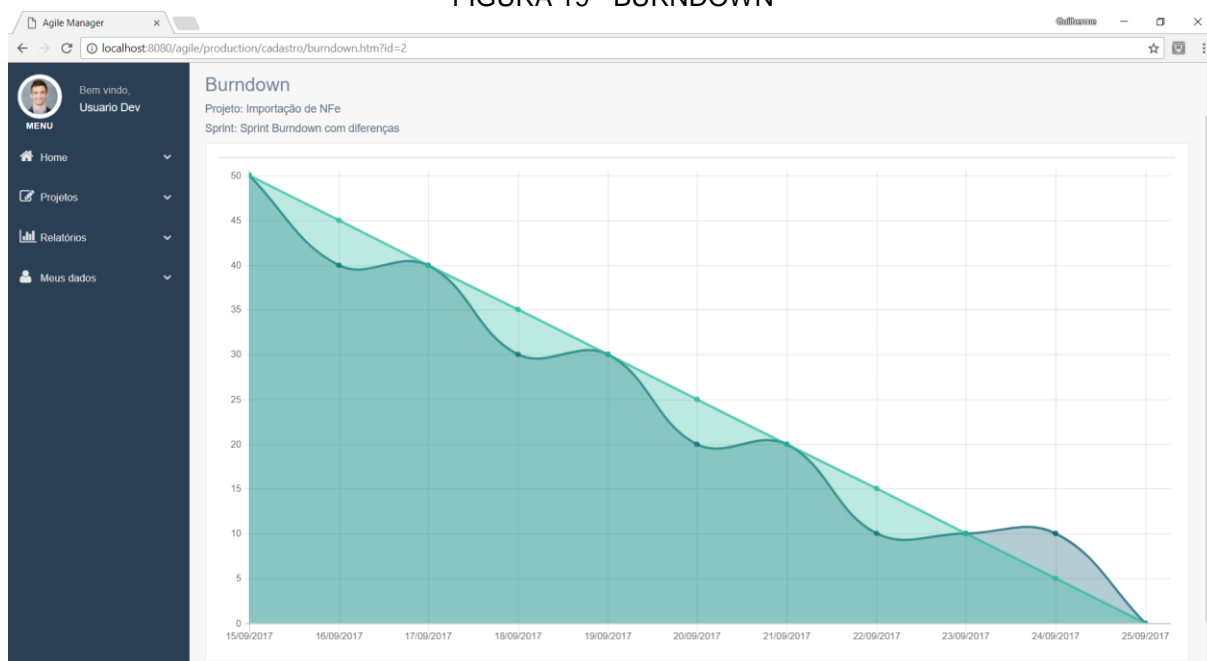
| História | Tarefa | Status | Data prevista | Tempo estimado | Usuário atribuído | Usuário solicitante | Visualizar |
|---|--|---------------------------|---------------|----------------|-------------------|---------------------|------------|
| Implementação da importação automática de NFe | Estudo da tecnologia | Concluído | 03/10/2017 | 10 | Usuario Dev | Usuario Dev | 🔍 |
| Implementação da importação automática de NFe | Criação das classes de negócio | Desenvolvimento concluído | 03/10/2017 | 10 | Usuario Dev | Usuario Dev | 🔍 |
| Implementação da importação automática de NFe | Implementação da importação automática | Desenvolvimento concluído | 03/10/2017 | 10 | Usuario Dev | Usuario Dev | 🔍 |

FONTE: O autor (2017)

4.3.3 Burndown

O gráfico de burndown (FIGURA 19) tem por objetivo auxiliar o time a ter uma visão macro do andamento da sprint. Facilita a visualização da entrega das tarefas em comparação com o planejado. Para exibir este relatório, o usuário deve clicar no ícone de gráfico, disponível na lista de sprints do projeto (FIGURA 11), ou clicar no botão “Burndown” na tela da sprint (FIGURA 12).

FIGURA 19 - BURNDOWN



FONTE: O autor (2017)

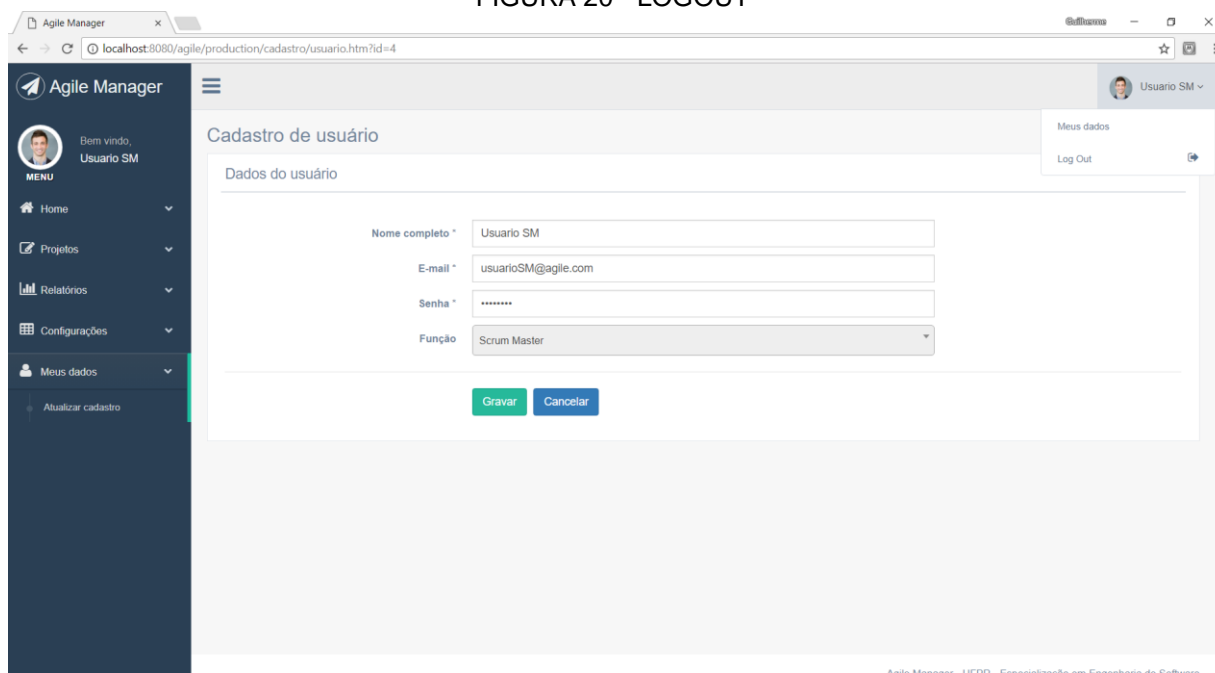
O eixo y do gráfico refere-se ao somatório de horas estimadas na sprint. O eixo x refere-se aos dias de sprint, listando-os a partir da data de início até a data de fechamento da sprint. O gráfico conta com o indicativo de entregas ideal da sprint, representado pela linha verde. A linha azul representa o andamento real da sprint, onde podem haver variações abaixo da linha ideal, indicando que a sprint está adiantada, ou variações acima da linha ideal, indicando que a sprint está entregando menos que o previsto. Cada ponto (x,y) da linha ideal indica quantas horas de trabalho ainda restam de acordo com a estimativa. Cada ponto (x,y) da linha real indica quantas horas de trabalho ainda restam, considerando o tempo estimado subtraído das horas das tarefas já entregues.

4.4 LOGOUT

4.4.1 Sair do sistema

Essa opção se encontra localizada no canto superior direito da tela, “<Nome do usuário>”, “Logout” (FIGURA 20). O usuário é redirecionado para a tela de login (FIGURA 5).

FIGURA 20 - LOGOUT

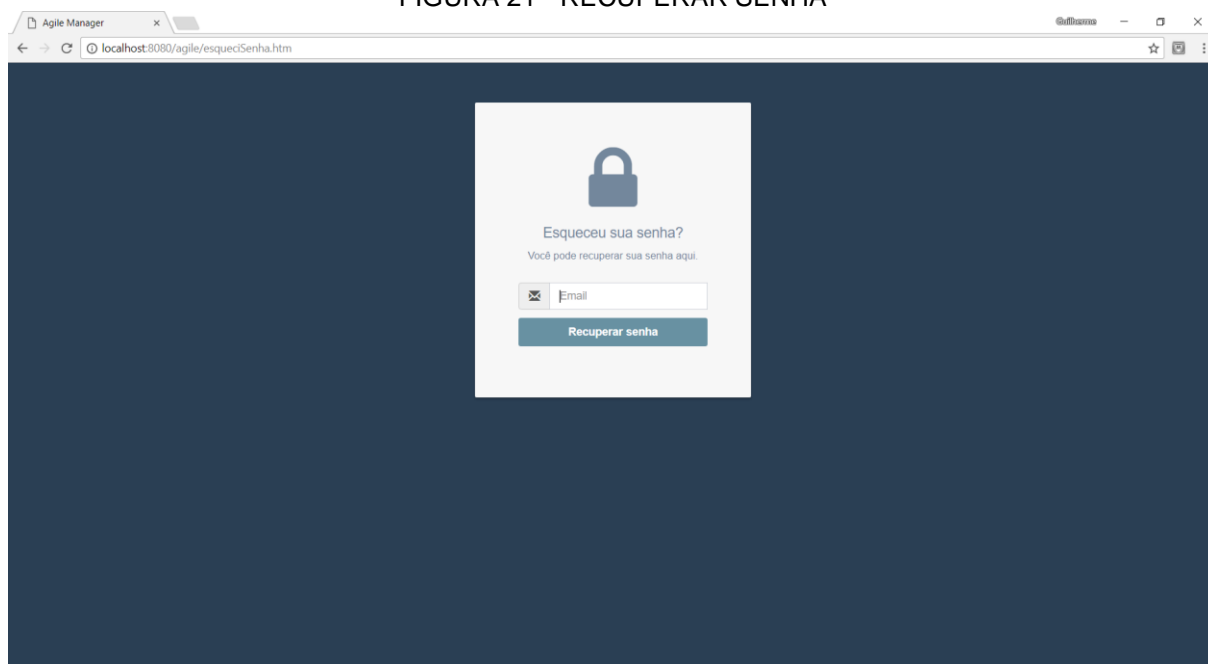


FONTE: O autor (2017)

4.4.2 Recuperar senha

O sistema Agile Manager conta com a opção de recuperar senha. Para utilizar essa opção, o usuário deve clicar no link “Esqueci a senha” na tela de login (FIGURA 5). O sistema redireciona o usuário para a tela de recuperar senha (FIGURA 21). Deve preencher o campo “E-mail” e clicar no botão “Recuperar senha”. O sistema enviará um e-mail com a senha, para assim, o usuário poder logar novamente no sistema.

FIGURA 21 - RECUPERAR SENHA



FONTE: O autor (2017)

Neste capítulo foi apresentado o sistema Agile Manager, seus conceitos, utilização e fluxo de trabalho. O Próximo capítulo contém as considerações finais, com um resumo do trabalho realizado, limitações do software final e implementações futuras.

5 CONSIDERAÇÕES FINAIS

Como trabalho de conclusão do curso de Engenharia de Software, o tema escolhido foi gerenciamento de projetos de software. Desta forma a aplicação dos conhecimentos obtidos durante o curso de especialização foram explorados e fundamentados neste projeto final.

O principal objetivo do sistema Agile Manager é auxiliar pequenas equipes de desenvolvimento ágil a gerenciarem seus projetos. Com base na metodologia Scrum, o sistema foi desenvolvido com o intuito de atender este público. Gerenciar um projeto não é uma tarefa trivial, que demanda tempo e dedicação. A ferramenta Agile Manager auxilia e facilita o gerenciamento do projeto, e a equipe de desenvolvimento pode se dedicar cada vez mais ao desenvolvimento do que ao gerenciamento.

O sistema não tem por objetivo gerenciar projetos de outros nichos de trabalho que não sejam da engenharia de software, porém, se aplicados os princípios ágeis de desenvolvimento, o gerenciamento se torna possível independentemente de área.

5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

O escopo inicial proposto para o Agile Manager foi implementado, porém, o sistema pode contar com alguns *upgrades* futuramente, dentre eles:

- Gerenciamento de permissão nos projetos;
- Disponibilizar as tarefas em quadro de Kanban;
- Manutenção de funções, não somente as estipuladas no sistema;
- Upload de arquivos que fazem parte da documentação dos itens de backlog, como por exemplo: análises de negócio, análises técnicas, planilhas para conferência de cálculos, entre outros;
- Espaço para registrar itens de interesse resultantes das cerimônias das sprints;

Estes são alguns itens que podem ser implementados com o intuito de manter o software atualizado e em constante melhoria. Mais itens ainda podem ser levantados a partir do *feedback* dos usuários do sistema.

REFERÊNCIAS

AGIL, Manifesto. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em 15 mai. 2017.

ARAKAKI, Reginaldo; RIBEIRO, André Luiz Dias. Gerenciamento de projetos tradicional x gerenciamento de projetos ágil: uma análise comparativa. Conferência proferida no 3 Congresso Internacional de Gestão da Tecnologia e Sistemas de Informação, São Paulo/SP, 31 maio a 2 jun. 2006.

BECK, Kent. Extreme Programming Explained: Embrace Change. AddisonWesley, 2000.

BRASIL, PMI. Disponível em: <<https://brasil.pmi.org/brazil/AboutUs/WhatIsProjectManagement.aspx>>. Acesso em 01 mai. 2017.

CRUZ, Fábio. Scrum e Agile em Projetos Guia Completo. Rio de Janeiro, Editora Brasport, 2015.

IEEE. IEEE Standard Glossary of Software Engineering Terminology/IEEE Std 610 12-1990. 1990.

PRESSMANN, Roger S. Engenharia de Software: Uma abordagem profissional, 7 ed. Bookman, Porto Alegre, 2011.

PRESSMANN, Roger S. Engenharia de Software: Uma abordagem profissional. 8 ed. AMGH Editora. 2016.

ROYCE, Winston Walker. Managing the development of large software systems: concepts and techniques. Proc. IEEE Westcon, Los Angeles, 1970.

VARGAS, Ricardo Viana. Gerenciamento de Projetos: estabelecendo diferencas competitivos. 8 ed. Rio de Janeiro, Editora Brasport, 2016.

WIKIMEDIA. Disponível em: <<https://upload.wikimedia.org/wikipedia/commons/1/19/Development-iterative.png>>. Acesso em 24 out. 2017.

APÊNDICE A – VISÃO

O sistema possui uma sequência de instruções para que o Scrum Master possa cadastrar os usuários e projetos de software a serem desenvolvidos. A equipe cadastra as atividades necessárias para desenvolver determinado projeto, bem como acompanhar a evolução do mesmo.

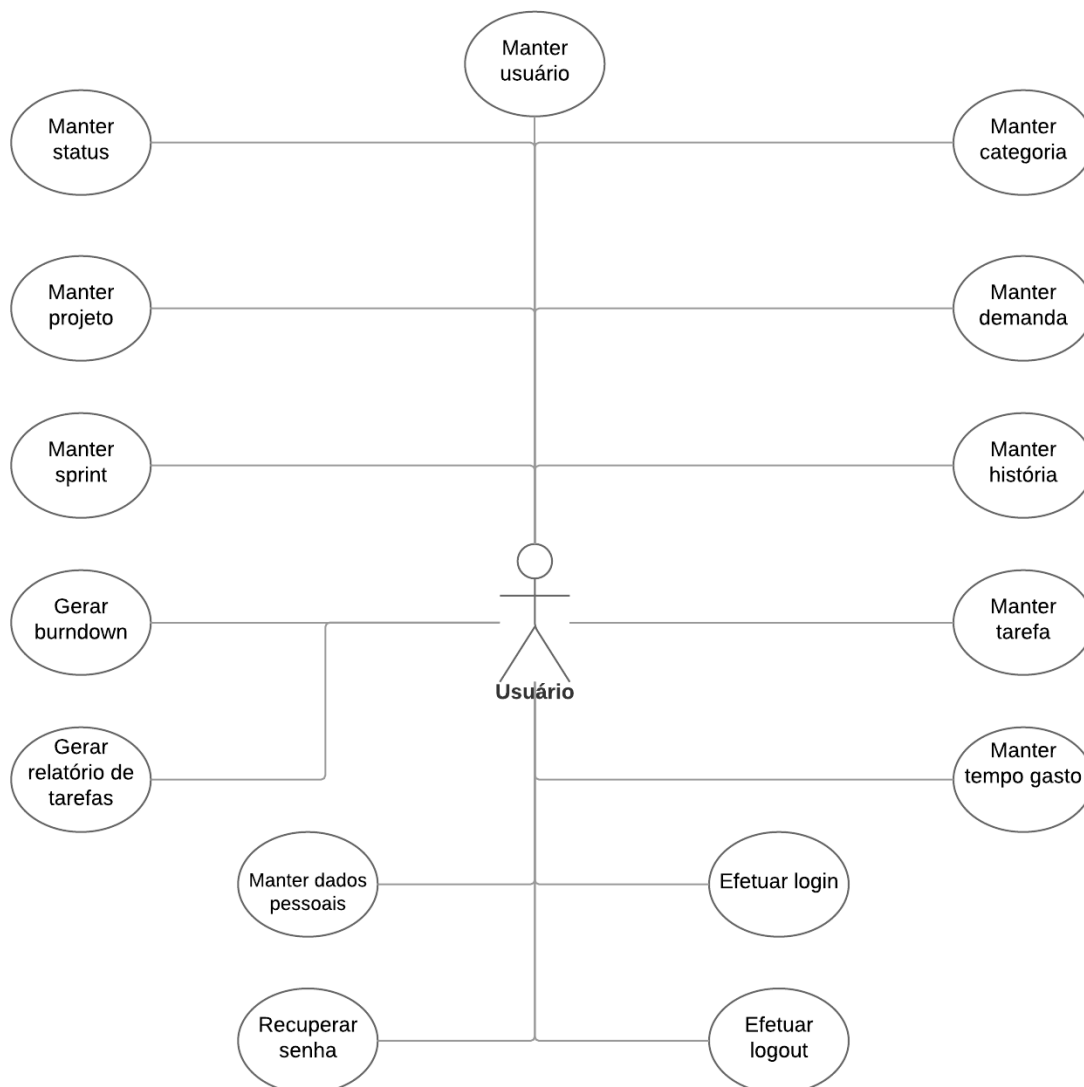
O sistema vincula as atividades aos respectivos usuários, a fim de monitoramento, e para isso é necessário que o usuário esteja previamente cadastrado no sistema.

Primeiramente, cabe ao Scrum Master cadastrar o projeto no sistema, para então cadastrar as sprints nos projetos. Após isso, cabe ao project owner (PO) cadastrar as demandas com suas respectivas histórias ao projeto (o Scrum Master também possui esta permissão), para então, os desenvolvedores e testers cadastrarem suas respectivas atividades com as estimativas.

De acordo com o andamento do projeto, os usuários podem atualizar os status de suas atividades. Com isso o sistema gera relatórios (tarefas por status, Burndown, entre outros) para auxiliar os envolvidos no projeto a escolherem as melhores estratégias no decorrer do desenvolvimento.

APÊNDICE B – CASOS DE USO NEGOCIAIS

FIGURA 22 - CASOS DE USO NEGOCIAIS



FONTE: O autor (2017)

Funcionalidades

- **Acesso:** o sistema possui controle de login, contendo o perfil de Scrum Master, Product Owner e desenvolvedor/tester. O acesso é feito por meio de e-mail e senha. Há também o controle de acesso as páginas de acordo com o perfil do usuário, seguindo as regras de negócio (APÊNDICE D).
- **Projeto:** é o core do software. Consiste em um grupo de demandas, histórias e tarefas e apontamento de tarefas (seguindo esta hierarquia) cadastrados pelos usuários. Contém as informações relevantes pertinentes ao projeto de

software a ser desenvolvido: descrição dos itens de backlog, estimativas, tempo gasto, tarefas concluídas, entre outros.

- Manutenção de usuários: somente o Scrum Master pode cadastrar usuários no sistema.
- Relatório de tarefas: o sistema possui uma ferramenta de relatório personalizado, permitindo ao usuário a escolha de filtros e as informações desejadas para visualização de tarefas.
- Burndown: para medir o progresso de entregas da sprint em relação ao prazo, o sistema gera o gráfico de burndown.

APÊNDICE C – GLOSSÁRIO

Burndown – Gráfico que mede o progresso da Sprint e dá indicativos do progresso do trabalho da equipe.

Backlog – É uma lista contendo todas as funcionalidades desejadas para o produto/Sprint.

Mindset – é uma palavra da língua inglesa que significa “Mentalidade” ou “Atitude Mental”.

Tester – Responsável por informar aos envolvidos no projeto se as implementações e alterações aplicadas no produto atendem aos requisitos de qualidade.

APÊNDICE D – REGRAS DE NEGÓCIO

- R1** – Somente o Scrum Master pode cadastrar projetos.
- R2** – Somente o Scrum Master pode cadastrar sprints.
- R3** – Somente o Scrum Master cadastrar usuários.
- R4** – Somente o PO e Scrum Master podem cadastrar demandas.
- R5** – Somente o PO e Scrum Master podem cadastrar histórias.
- R6** – Os dados não podem ser excluídos fisicamente no banco de dados.
- R7** – Data de início deve ser menor ou igual a data de término.
- R8** – Campo obrigatório.

APÊNDICE E – PROTÓTIPO DE TELAS

FIGURA 23 - CADASTRO DE PROJETO (PROTÓTIPO)

The screenshot displays the 'Agile Manager' interface. On the left is a dark blue sidebar with the 'Agile Manager' logo and a user profile for 'Guilherme'. The main content area is titled 'Novo projeto' and contains a form titled 'Dados do projeto'. The form fields are as follows:

- Nome do projeto ***: A text input field.
- Descrição**: A text input field.
- Data início ***: A date picker set to 04/04/2017.
- Data término**: A date picker set to 04/04/2017.
- Categoria**: A dropdown menu with 'Selecione' selected.
- Status**: A dropdown menu with 'Selecione' selected.
- Observações**: A large text area for notes.

At the bottom of the form are two buttons: 'Cancelar' (blue) and 'Gravar' (green).

FONTE: O autor (2017)

FIGURA 24 - CADASTRO DE SPRINT (PROTÓTIPO)

The image shows a web application interface for 'Agile Manager'. On the left is a dark blue sidebar with a user profile for 'Guilherme' and a menu with options: Home, Projetos, Tarefas, and Meus dados. The main content area is titled 'Projeto 123' and contains a form for creating a new sprint, labeled 'Nova Sprint'. The form includes a text input for 'Descrição *', date pickers for 'Data início *' (04/04/2017) and 'Data término *' (04/04/2017), a dropdown for 'Status', and a text area for 'Observações'. At the bottom of the form are two buttons: 'Cancelar' (blue) and 'Gravar' (green).

Agile Manager

Welcome, Guilherme

MENU

Home

Projetos

Tarefas

Meus dados

Projeto 123

Nova Sprint

Descrição *

Data início * 04/04/2017 Data término * 04/04/2017

Status

Observações

Cancelar Gravar

FONTE: O autor (2017)

FIGURA 25 - CADASTRO DE DEMANDA (PROTÓTIPO)

The image shows a web application interface for 'Agile Manager'. On the left is a dark blue sidebar with a user profile for 'Guilherme' and a menu with options: Home, Projetos, Tarefas, and Meus dados. The main content area is light gray and displays 'Projeto 123' and 'Sprint 123'. Below this is a form titled 'Nova Demanda' with the following fields: 'Descrição' (text input), 'Status' (dropdown menu), 'Sprint' (dropdown menu), and 'Observações' (text area). At the bottom of the form are two buttons: 'Cancelar' (blue) and 'Gravar' (green).

FONTE: O autor (2017)

FIGURA 26 - CADASTRO DE TAREFA (PROTÓTIPO)

The image shows a web application interface for 'Agile Manager'. On the left is a dark blue sidebar with a 'MENU' section containing icons and labels for 'Home', 'Projetos', 'Tarefas', and 'Meus dados'. The main content area has a header with 'Projeto 123' and 'Demanda 123 - História - 123'. Below this is a form titled 'Nova Tarefa'. The form contains several input fields: 'Descrição *' (text), 'Tempo estimado *' (text), 'Data prevista' (calendar icon and '04/04/2017'), 'Status' (dropdown), 'Sprint' (dropdown), and 'Observações' (text area). At the bottom of the form are two buttons: 'Cancelar' (blue) and 'Gravar' (green).

Agile Manager

Welcome,
Guilherme

MENU

Home

Projetos

Tarefas

Meus dados

Projeto 123

Demanda 123 - História - 123

Nova Tarefa

Descrição *

Tempo estimado *

Data prevista 04/04/2017

Status

Sprint

Observações

Cancelar Gravar

FONTE: O autor (2017)

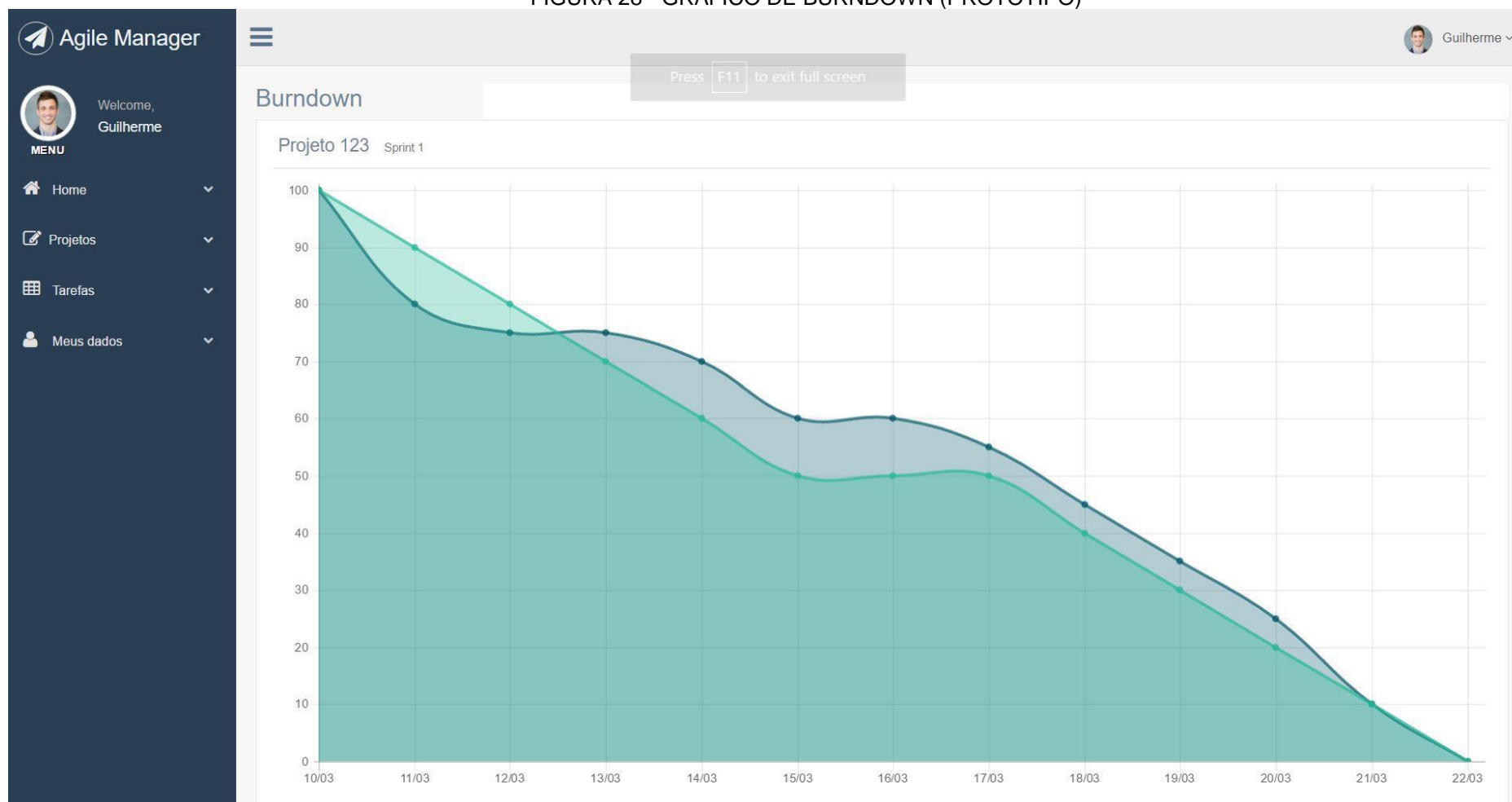
FIGURA 27 - RELATÓRIO DE TAREFAS (PROTÓTIPO)

The screenshot displays the Agile Manager interface. On the left is a dark sidebar with the 'Agile Manager' logo and a user profile for 'Guilherme' with a 'MENU' label. The main area shows 'Projeto 123 Inovação do sistema'. Below this, there are filter sections for 'Responsável' (Guilherme Weinhardt), 'Situação' (Desenvolvimento concluído), and 'Sprint' (Sprint 1). A 'Show 10 entries' dropdown and a search box are also present. A table lists three tasks with columns for 'Código', 'Descrição', 'Situação', and 'Tempo estimado'. At the bottom, it indicates 'Showing 1 to 3 of 3 entries' and includes 'Previous' and 'Next' navigation buttons.

| Código | Descrição | Situação | Tempo estimado |
|--------|-------------------------------------|---------------------------|----------------|
| 1 | Implementar nova funcionalidade | Desenvolvimento concluído | 6 |
| 2 | Criação do script do banco de dados | Desenvolvimento concluído | 4 |
| 3 | Implementar nova rotina | Desenvolvimento concluído | 16 |

FONTE: O autor (2017)

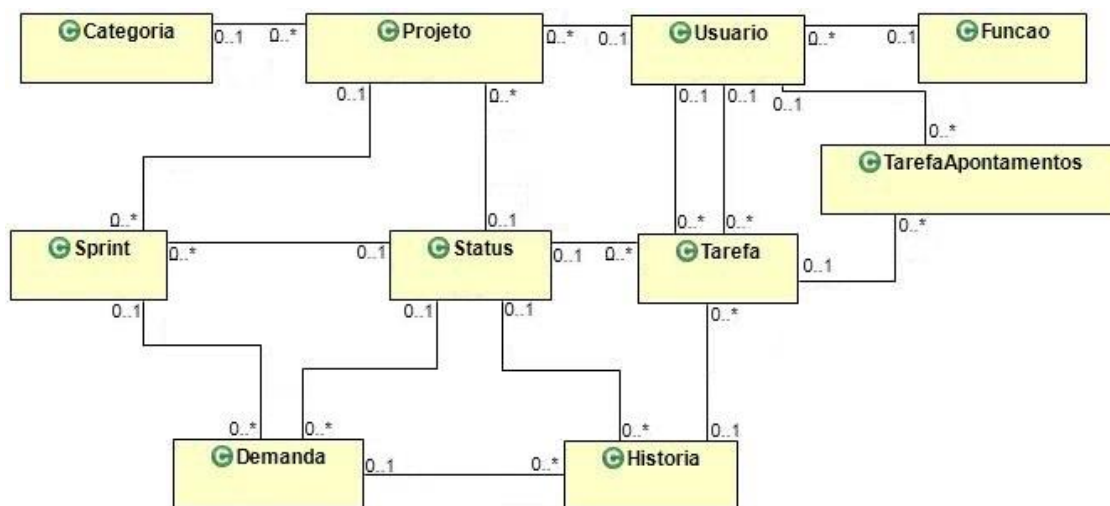
FIGURA 28 - GRÁFICO DE BURNDOWN (PROTÓTIPO)



FONTE: O autor (2017)

APÊNDICE F – MODELO DE OBJETOS NEGOCIAIS

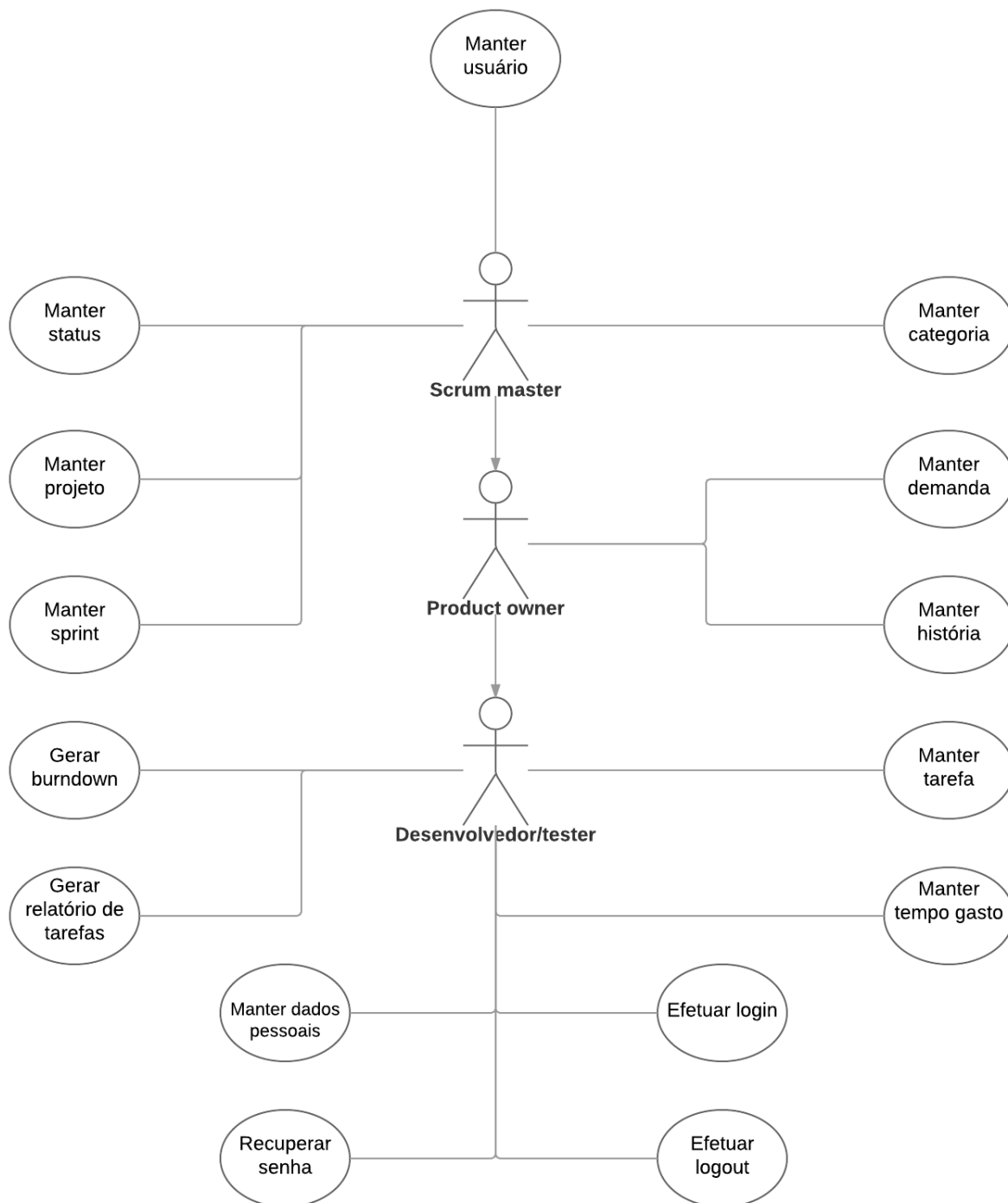
FIGURA 29 - DIAGRAMA DE CLASSES NEGOCIAIS



FONTE: O autor (2017)

APÊNDICE G – CASOS DE USO

FIGURA 30 - CASOS DE USO



FONTE: O autor (2017)

Especificação dos casos de uso

UC 001 - Efetuar login

Descrição

Este caso de uso descreve a tela de login do sistema.

Data view

FIGURA 5.

Pré-condições

Não há.

Pós-condições

Após o final deste caso de uso, o usuário estará com uma sessão válida no sistema.

Ator primário

Todos os atores podem executar este caso de uso.

Fluxo principal

- 1) O sistema exibe a tela de login (FIGURA 5)
- 2) O usuário preenche o campo E-mail
- 3) O usuário preenche o campo Senha
- 4) O usuário clica no botão Login
- 5) O sistema cria uma sessão para o usuário
- 6) O sistema redireciona o usuário para a home page (A1)
- 7) O caso de uso é encerrado

Fluxos alternativos

A1. O usuário preenche com dados de login incorretos

1. O sistema exibe a mensagem “Erro. Usuário e senha inválidos”.

UC 002 - Recuperar senha**Descrição**

Este caso de uso descreve a tela de recuperar senha do sistema.

Data view

FIGURA 21.

Pré-condições

Não há.

Pós-condições

Após o final deste caso de uso, o usuário poderá efetuar login novamente no sistema.

Ator primário

Todos os atores podem executar este caso de uso.

Fluxo principal

- 1) O sistema exibe a tela de login (FIGURA 5)
- 2) O usuário clica no link “Esqueci a senha”
- 3) O sistema redireciona o usuário para a tela de Recuperar senha
- 4) O usuário preenche o campo E-mail
- 5) O usuário clica no botão Recuperar senha
- 6) O sistema envia um e-mail para o usuário com a senha de acesso ao sistema
- 7) O caso de uso é encerrado

UC 003 - Efetuar logout**Descrição**

Este caso de uso descreve a ação de realizar logout no sistema.

Data view

FIGURA 20.

Pré-condições

O usuário deve estar logado no sistema.

Pós-condições

Após o final deste caso de uso, o sistema encerrará a sessão do usuário.

Ator primário

Todos os atores podem executar este caso de uso.

Fluxo principal

- 1) O sistema exibe a tela de logout (FIGURA 20)
- 2) O usuário clica no botão logout
- 3) O sistema invalida a sessão do usuário
- 4) O sistema redireciona o usuário para a tela de login (FIGURA 5)
- 5) O caso de uso é encerrado

UC 004 – Manter dados pessoais

Descrição

Este caso de uso descreve a tela de alterar os dados de usuário do sistema.

Data view

FIGURA 9.

Pré-condições

O usuário deve estar logado no sistema.

Pós-condições

Após o final deste caso de uso, o sistema atualizará os dados do usuário no banco de dados.

Ator primário

Todos os atores podem executar este caso de uso.

Fluxo principal

- 1) O sistema carrega os campos com os dados de usuário
- 2) O sistema exibe a tela de Atualizar cadastro (FIGURA 9)
- 3) O usuário preenche o campo Nome completo (R8)
- 4) O usuário preenche o campo E-mail (R8)
- 5) O usuário preenche o campo Senha (R8)
- 6) O usuário clica no botão Gravar (A1)
- 7) O sistema exibe a mensagem “Usuário atualizado com sucesso”
- 8) O caso de uso é encerrado

Fluxos alternativos

A1. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 005 – Manter usuário

Descrição

Este caso de uso descreve a tela de alterar os dados de usuário do sistema.

Data view

FIGURA 6.

Pré-condições

O usuário deve estar logado no sistema como Scrum master.

Pós-condições

Após o final deste caso de uso, o sistema atualizará os dados do usuário no banco de dados.

Ator primário

Scrum Master.

Fluxo principal

- 1) O sistema carrega o combo de Função
- 2) O sistema carrega a grid com os usuários cadastrados no sistema.
- 3) O sistema exibe a tela de Atualizar cadastro (FIGURA 6)
- 4) O usuário inicia o cadastro de um novo usuário (A1)
- 5) O usuário preenche o campo Nome completo (R8)
- 6) O usuário preenche o campo E-mail (R8)
- 7) O usuário preenche o campo Senha (R8)
- 8) O usuário preenche o campo Função (R8)
- 9) O usuário clica no botão Gravar (A2)
- 10) O sistema exibe a mensagem “Usuário atualizado com sucesso”
- 11) O sistema recarrega a grid de usuários
- 12) O caso de uso é encerrado

Fluxos alternativos

A1. Atualizar um cadastro já existente

1. O usuário clica no ícone de edição referente ao usuário que deseja atualizar
2. O sistema preenche os campos com os dados do usuário

A2. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 006 – Manter status

Descrição

Este caso de uso descreve a tela de cadastrar status no sistema.

Data view

FIGURA 7.

Pré-condições

O usuário deve estar logado no sistema como Scrum master.

Pós-condições

Após o final deste caso de uso, o sistema atualizará os dados de status no banco de dados.

Ator primário

Scrum Master.

Fluxo principal

- 1) O sistema carrega a grid com os status cadastrados no sistema.
- 2) O sistema exibe a tela de Cadastro de status (FIGURA 7)
- 3) O usuário inicia o cadastro de um novo status no sistema (A1)
- 4) O usuário preenche o campo Descrição (R8)
- 5) O usuário preenche o campo Conclui tarefa? (R8)
- 6) O usuário clica no botão Gravar (A2)
- 7) O sistema exibe a mensagem “Status gravado com sucesso”
- 8) O caso de uso é encerrado

Fluxos alternativos

A1. Atualizar um cadastro já existente

3. O usuário clica no ícone de edição referente ao status que deseja atualizar
4. O sistema preenche os campos com os dados do status

A2. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 007 – Manter categoria

Descrição

Este caso de uso descreve a tela de cadastrar categoria no sistema.

Data view

FIGURA 8.

Pré-condições

O usuário deve estar logado no sistema como Scrum master.

Pós-condições

Após o final deste caso de uso, o sistema atualizará os dados de categoria no banco de dados.

Ator primário

Scrum Master.

Fluxo principal

- 1) O sistema carrega a grid com as categorias cadastrados no sistema.
- 2) O sistema exibe a tela de Cadastro de categoria (FIGURA 8)
- 3) O usuário inicia o cadastro de uma nova categoria no sistema (A1)
- 4) O usuário preenche o campo Descrição (R8)
- 5) O usuário clica no botão Gravar (A2)
- 6) O sistema exibe a mensagem “Categoria gravada com sucesso”
- 7) O caso de uso é encerrado

Fluxos alternativos

A1. Atualizar um cadastro já existente

1. O usuário clica no ícone de edição referente a categoria que deseja atualizar
2. O sistema preenche os campos com os dados da categoria

A2. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 008 – Manter projeto

Descrição

Este caso de uso descreve a tela de cadastrar projeto no sistema.

Data view

FIGURA 11.

Pré-condições

O usuário deve estar logado no sistema como Scrum master.

Pós-condições

Após o final deste caso de uso, o sistema irá gravar os dados do projeto no banco de dados.

Ator primário

Scrum Master.

Fluxo principal

- 1) O sistema carrega os combos de Categoria e Status.
- 2) O sistema exibe a tela de Cadastro de projeto (FIGURA 11) (R1)
- 3) O usuário preenche o campo Nome do projeto (R8)
- 4) O usuário preenche o campo Descrição
- 5) O usuário preenche o campo Data início (R7) (R8)
- 6) O usuário preenche o campo Data término
- 7) O usuário preenche o campo Categoria
- 8) O usuário preenche o campo Status
- 9) O usuário preenche o campo Observação
- 10) O usuário clica no botão Gravar (A1)
- 11) O sistema exibe a mensagem “Projeto gravado com sucesso”
- 12) O caso de uso é encerrado

Fluxos alternativos

A1. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 009 – Manter sprint

Descrição

Este caso de uso descreve a tela de cadastrar sprint no sistema.

Data view

FIGURA 12.

Pré-condições

O usuário deve estar logado no sistema como Scrum master.

Pós-condições

Após o final deste caso de uso, o sistema irá gravar os dados da sprint no banco de dados.

Ator primário

Scrum Master.

Fluxo principal

- 1) O sistema carrega o combo de Status.
- 2) O sistema exibe a tela de Cadastro de sprint (FIGURA 12) (R2)
- 3) O usuário preenche o campo Descrição (R7)
- 4) O usuário preenche o campo Data início (R7) (R8)
- 5) O usuário preenche o campo Data término
- 6) O usuário preenche o campo Status
- 7) O usuário preenche o campo Observação
- 8) O usuário clica no botão Gravar (A1)
- 9) O sistema exibe a mensagem “Sprint gravada com sucesso”
- 10) O caso de uso é encerrado

Fluxos alternativos

A1. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 010 – Manter demanda

Descrição

Este caso de uso descreve a tela de cadastrar demanda no sistema.

Data view

FIGURA 13.

Pré-condições

O usuário deve estar logado no sistema como Scrum master ou PO.

Pós-condições

Após o final deste caso de uso, o sistema irá gravar os dados da demanda no banco de dados.

Ator primário

Scrum Master ou PO.

Fluxo principal

- 1) O sistema carrega o combo de Status.
- 2) O sistema exibe a tela de Cadastro de demanda (FIGURA 13) (R4)
- 3) O usuário preenche o campo Descrição (R7)
- 4) O usuário preenche o campo Status
- 5) O usuário preenche o campo Observação
- 6) O usuário clica no botão Gravar (A1)
- 7) O sistema exibe a mensagem “Demanda gravada com sucesso”
- 8) O caso de uso é encerrado

Fluxos alternativos

A1. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 011 – Manter história

Descrição

Este caso de uso descreve a tela de cadastrar história no sistema.

Data view

FIGURA 14.

Pré-condições

O usuário deve estar logado no sistema como Scrum master ou PO.

Pós-condições

Após o final deste caso de uso, o sistema irá gravar os dados da história no banco de dados.

Ator primário

Scrum Master ou PO.

Fluxo principal

- 1) O sistema carrega o combo de Status.
- 2) O sistema exibe a tela de Cadastro de história (FIGURA 14) (R5)
- 3) O usuário preenche o campo Descrição (R7)
- 4) O usuário preenche o campo Status
- 5) O usuário preenche o campo Observação
- 6) O usuário clica no botão Gravar (A1)
- 7) O sistema exibe a mensagem “História gravada com sucesso”
- 8) O caso de uso é encerrado

Fluxos alternativos

A1. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 012 – Manter tarefa

Descrição

Este caso de uso descreve a tela de cadastrar tarefa no sistema.

Data view

FIGURA 15.

Pré-condições

O usuário deve estar logado no sistema.

Pós-condições

Após o final deste caso de uso, o sistema irá gravar os dados da tarefa no banco de dados.

Ator primário

Todos os atores podem executar este caso de uso.

Fluxo principal

- 1) O sistema carrega os combos de Status, Solicitante e Atribuído.
- 2) O sistema exibe a tela de Cadastro de tarefa (FIGURA 15)
- 3) O usuário preenche o campo Descrição (R7)
- 4) O usuário preenche o campo Tempo estimado (R7)
- 5) O usuário preenche o campo Data prevista
- 6) O usuário preenche o campo Status
- 7) O usuário preenche o campo Solicitante
- 8) O usuário preenche o campo Atribuído
- 9) O usuário preenche o campo Observação
- 10) O usuário clica no botão Gravar (A1)
- 11) O sistema exibe a mensagem “Tarefa gravada com sucesso”
- 12) O caso de uso é encerrado

Fluxos alternativos

A1. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 013 – Manter tempo gasto

Descrição

Este caso de uso descreve a tela de cadastrar apontamento de tempo gasto no sistema.

Data view

FIGURA 16.

Pré-condições

O usuário deve estar logado no sistema.

Pós-condições

Após o final deste caso de uso, o sistema irá gravar os dados do apontamento no banco de dados.

Ator primário

Todos os atores podem executar este caso de uso.

Fluxo principal

- 1) O sistema exibe a tela de Cadastro de apontamento (FIGURA 16)
- 2) O usuário preenche o campo Tempo gasto (R7)
- 3) O usuário preenche o campo Data
- 4) O usuário preenche o campo Observação
- 5) O usuário clica no botão Gravar (A1)
- 6) O sistema exibe a mensagem “Tarefa apontamento gravada com sucesso”
- 7) O caso de uso é encerrado

Fluxos alternativos

A1. O usuário clica no botão Cancelar

1. O sistema limpa os campos.

UC 014 – Gerar relatório de tarefas

Descrição

Este caso de uso descreve a tela de relatório de tarefas do sistema.

Data view

FIGURA 18.

Pré-condições

O usuário deve estar logado no sistema.

Pós-condições

Após o final deste caso de uso, o sistema irá gerar na tela um relatório de tarefas.

Ator primário

Todos os atores podem executar este caso de uso.

Fluxo principal

- 1) O sistema carrega os combos de Projeto, Usuário atribuído e Status.
- 2) O sistema exibe a tela de Relatório de tarefas (FIGURA 18)
- 3) O usuário preenche o campo Projeto
- 4) O sistema carrega o combo Sprint
- 5) O usuário preenche o campo Sprint
- 6) O sistema carrega o combo Demanda
- 7) O usuário preenche o campo Demanda
- 8) O sistema carrega o combo História
- 9) O usuário preenche o campo História
- 10) O usuário clica no botão Buscar
- 11) O sistema lista na grid as tarefas que satisfazem o filtro da busca
- 12) O caso de uso é encerrado

UC 015 – Gerar burndown**Descrição**

Este caso de uso descreve a função de gerar burndown do sistema.

Data view

FIGURA 19.

Pré-condições

O usuário deve estar logado no sistema. O usuário deve estar na página Cadastro de sprint (FIGURA 12). A sprint deve estar previamente cadastrada no sistema.

Pós-condições

Após o final deste caso de uso, o sistema irá gerar na tela um gráfico de burndown.

Ator primário

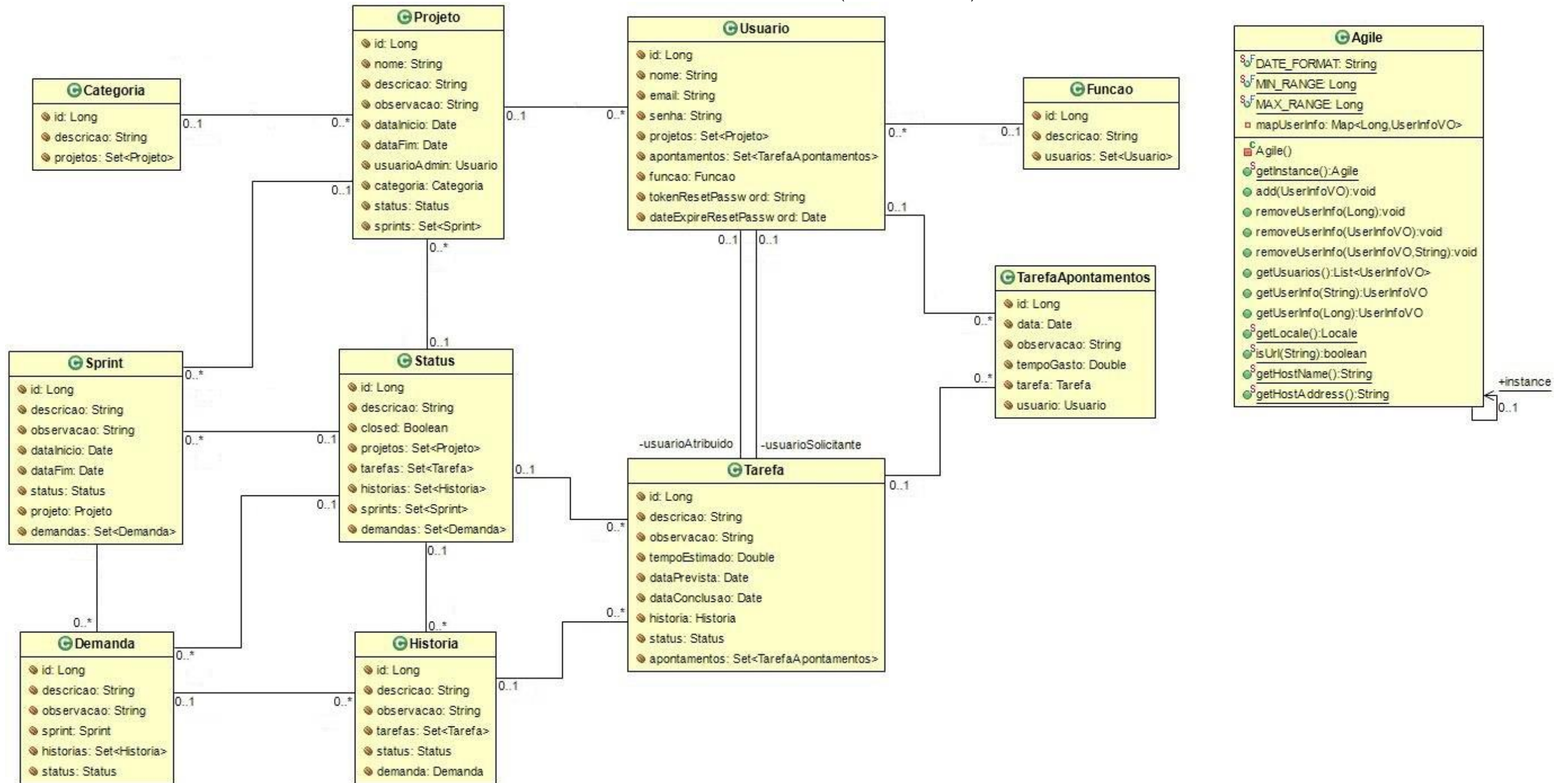
Todos os atores podem executar este caso de uso.

Fluxo principal

- 1) O sistema exibe a tela de cadastro da sprint (FIGURA 12)
- 2) O usuário clica no botão Burndown
- 3) O sistema redireciona o usuário para a tela do gráfico de burndown (FIGURA 19)
- 4) O caso de uso é encerrado

APÊNDICE H – MODELO DE OBJETOS (COM ATRIBUTOS)

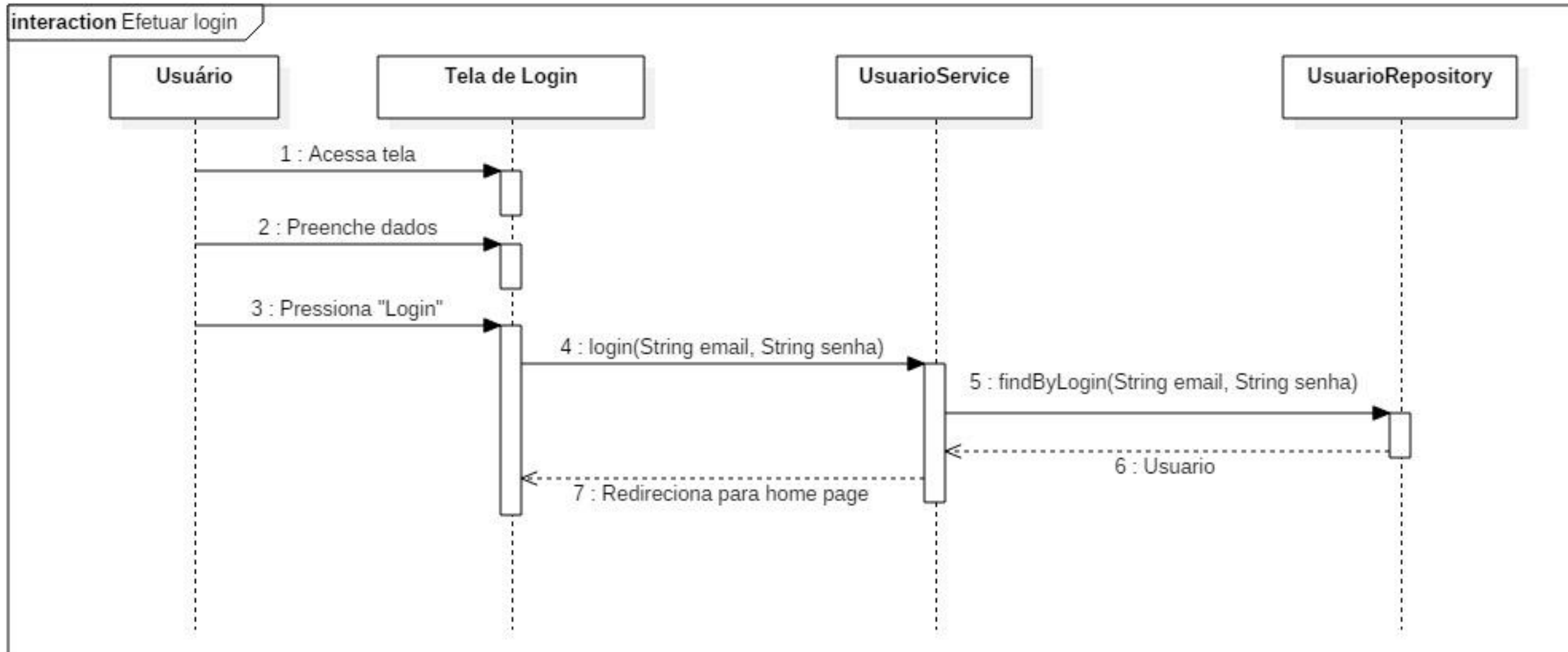
FIGURA 31 - DIAGRAMA DE CLASSES (COM ATRIBUTOS)



FONTE: O autor (2017)

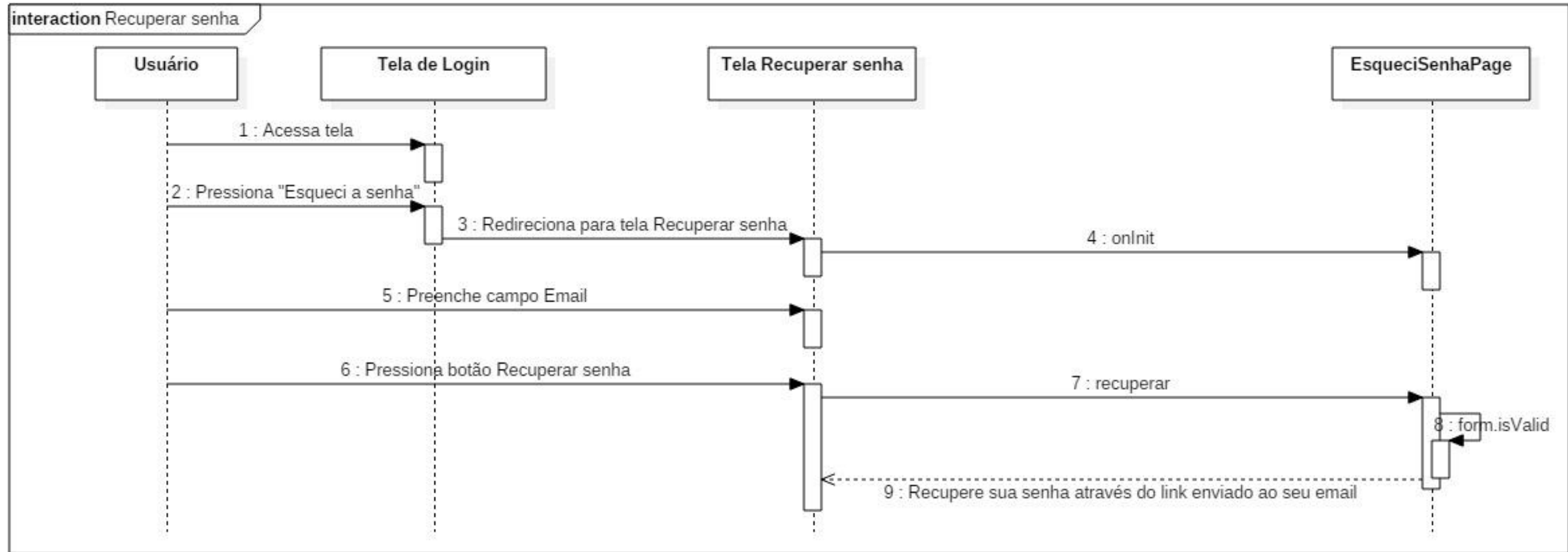
APÊNDICE I – DIAGRAMAS DE SEQUÊNCIA

FIGURA 32 - EFETUAR LOGIN



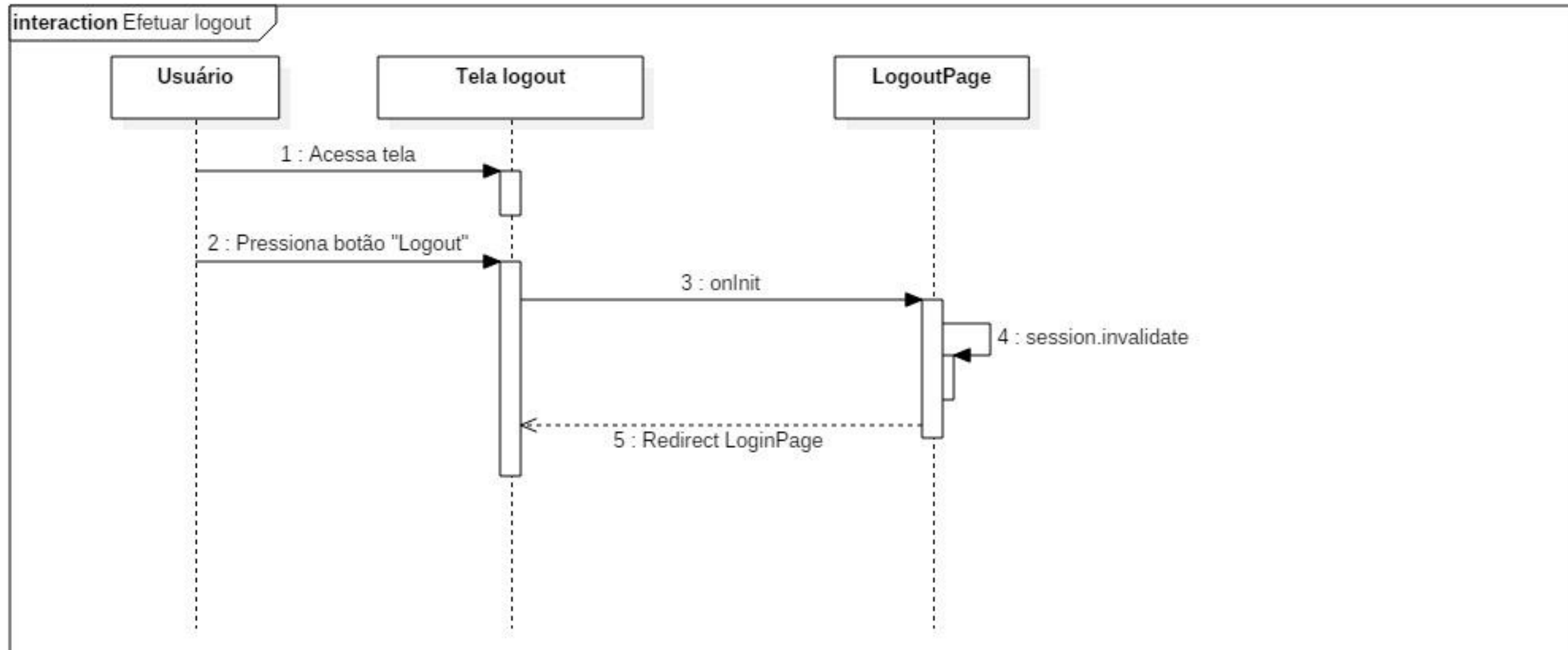
FONTE: O autor (2017)

FIGURA 33 - RECUPERAR SENHA



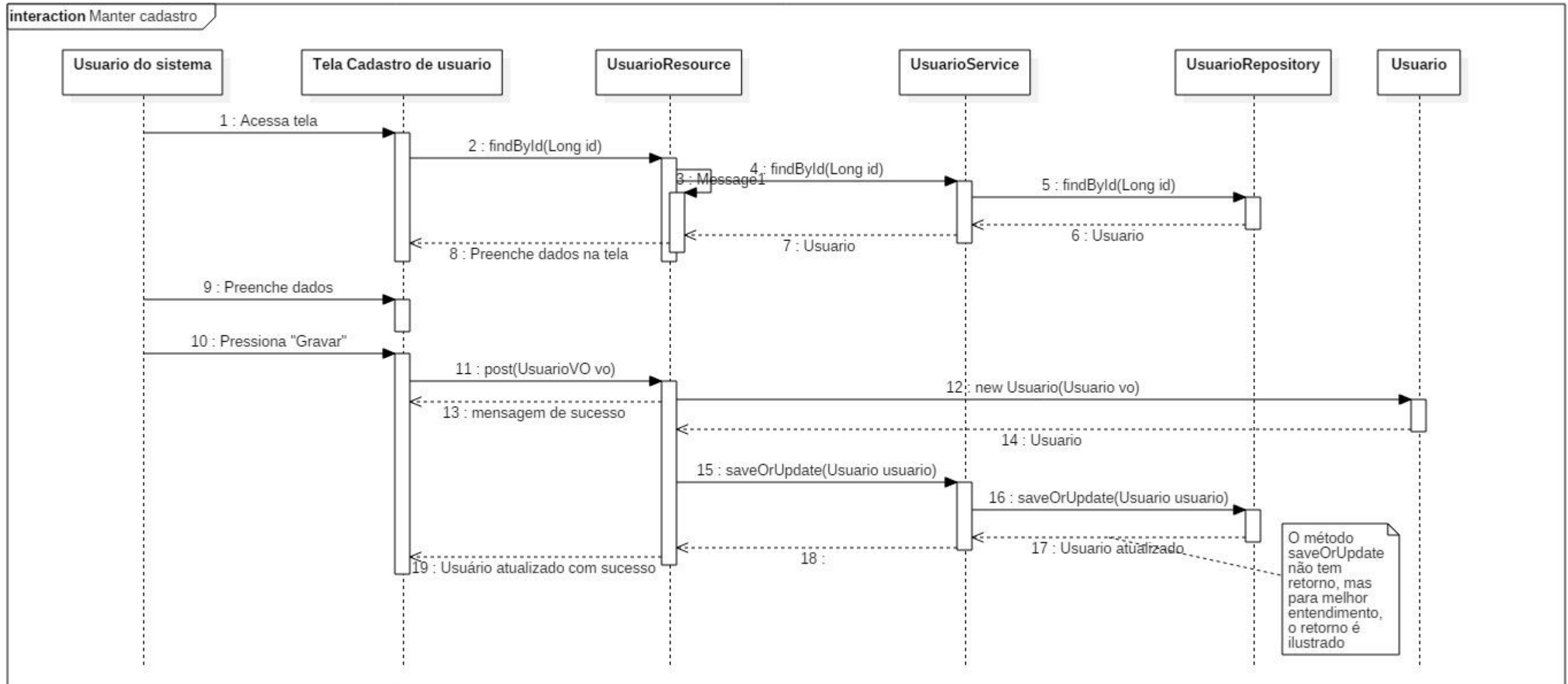
FONTE: O autor (2017)

FIGURA 34 - EFETUAR LOGOUT



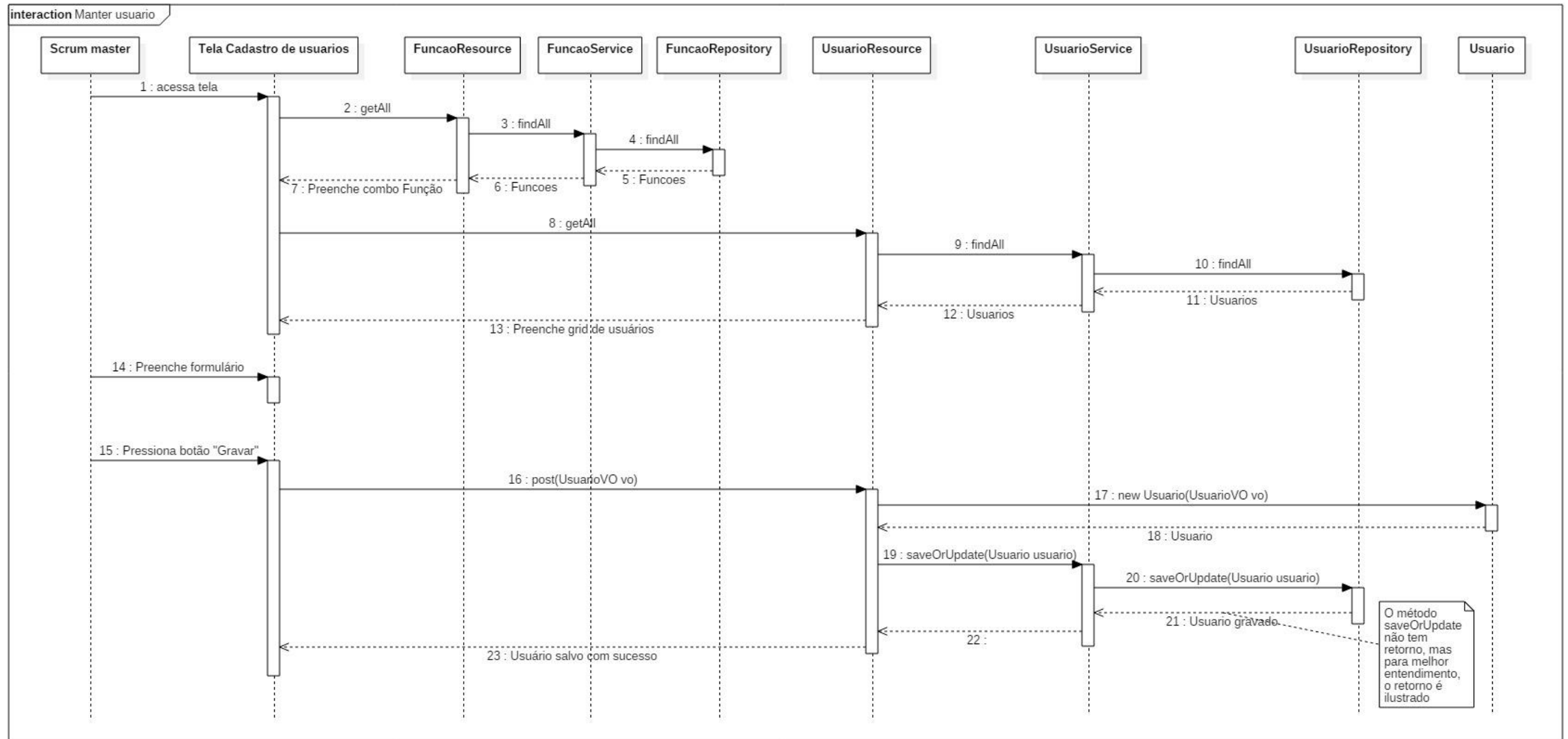
FONTE: O autor (2017)

FIGURA 35 - MANTER DADOS PESSOAIS



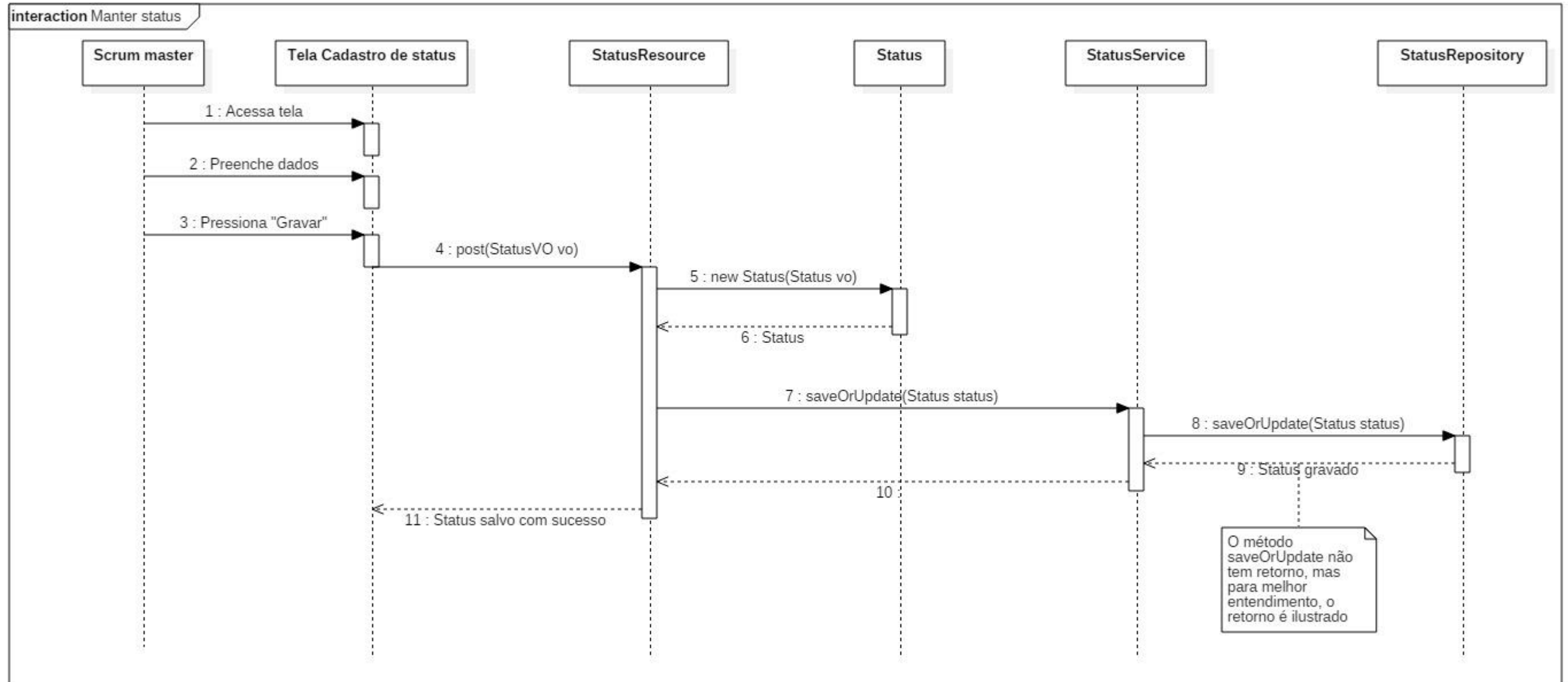
FONTE: O autor (2017)

FIGURA 36 - MANTER USUÁRIO



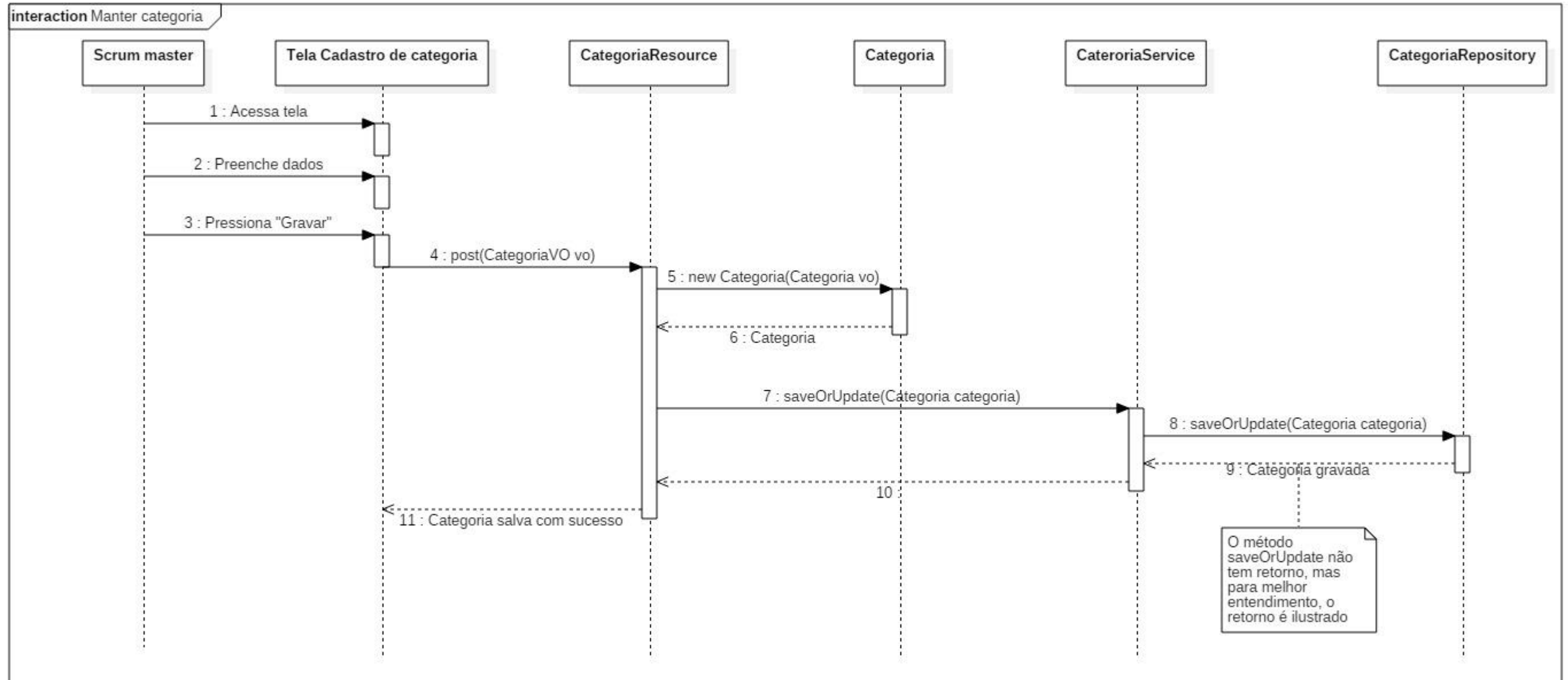
FONTE: O autor (2017)

FIGURA 37 - MANTER STATUS



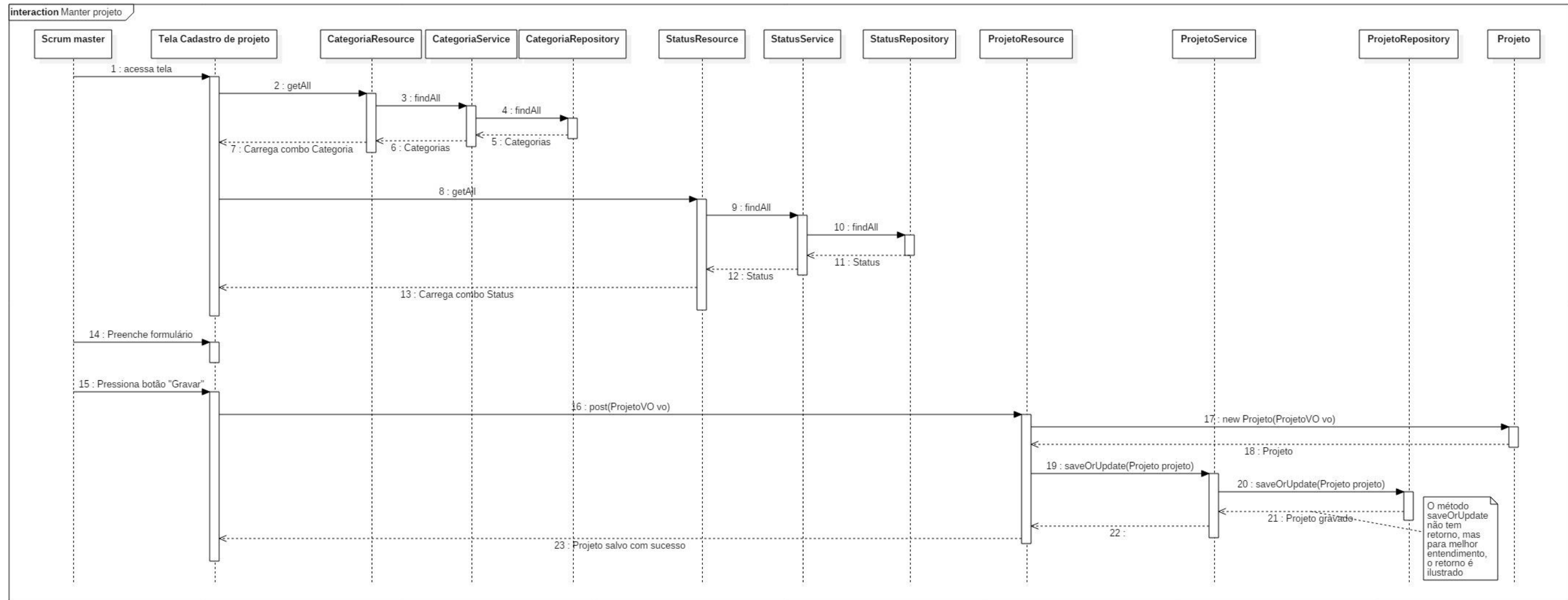
FONTE: O autor (2017)

FIGURA 38 - MANTER CATEGORIA



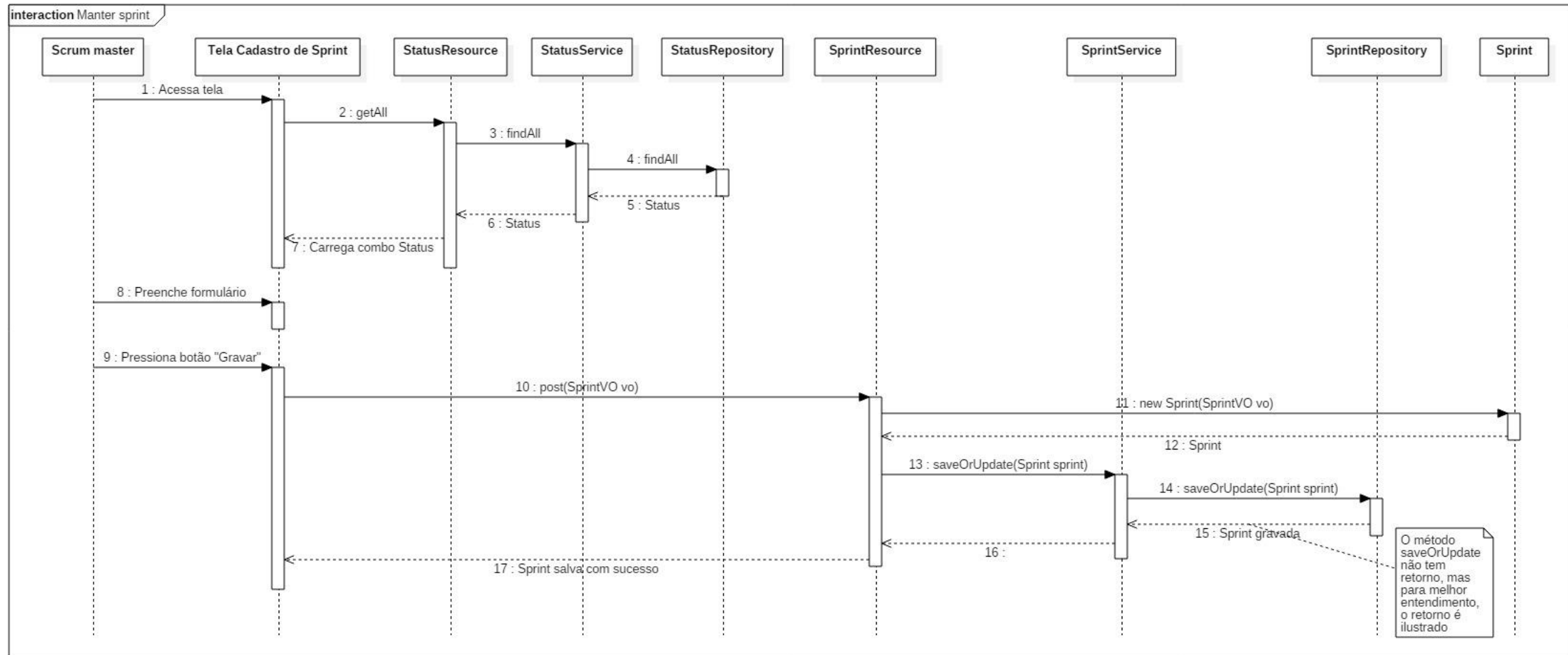
FONTE: O autor (2017)

FIGURA 39 - MANTER PROJETO



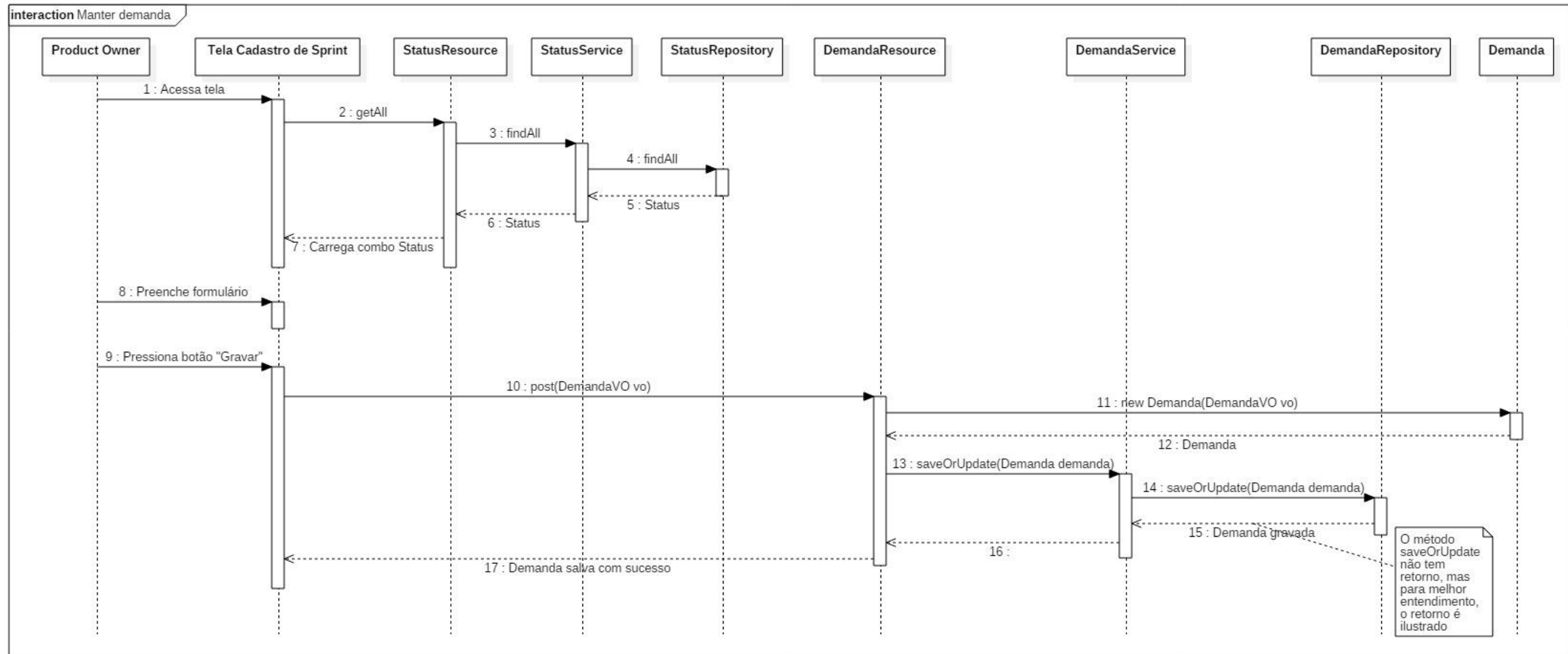
FONTE: O autor (2017)

FIGURA 40 - MANTER SPRINT



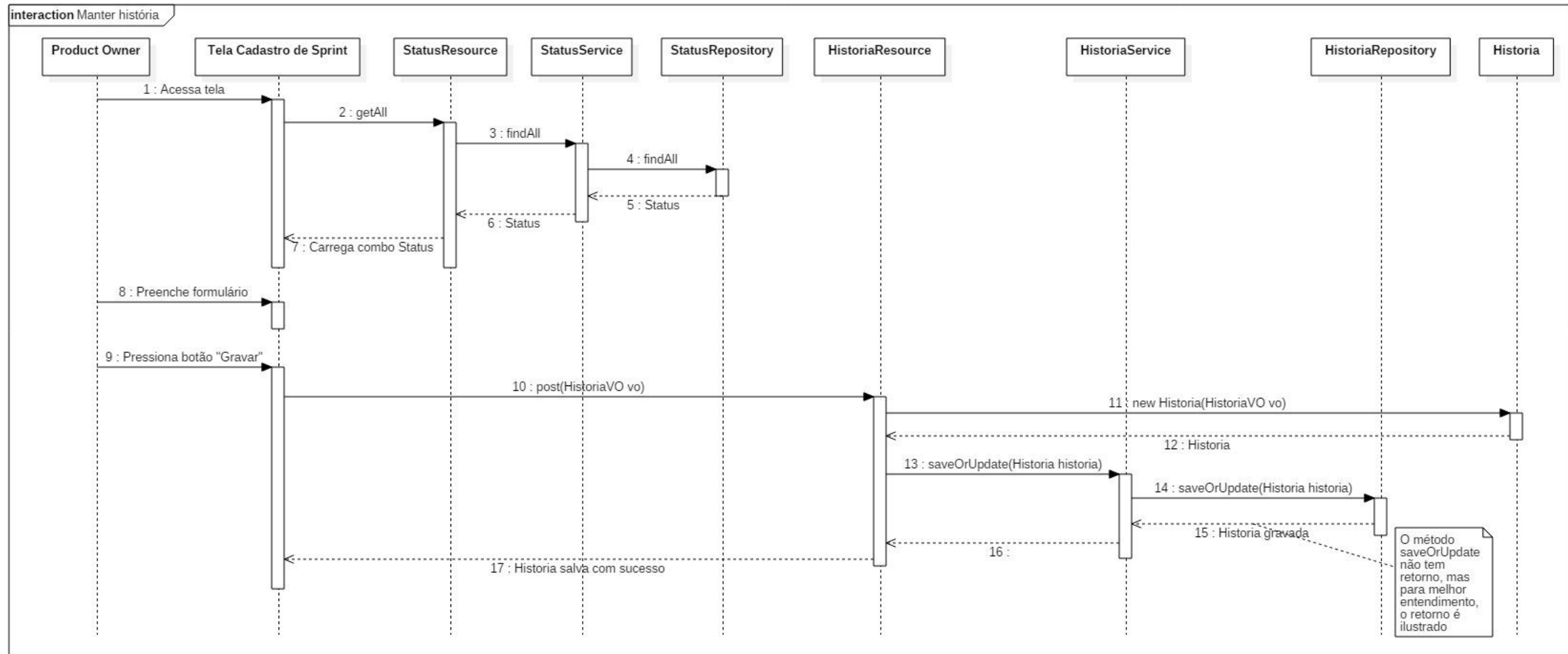
FONTE: O autor (2017)

FIGURA 41 - MANTER DEMANDA



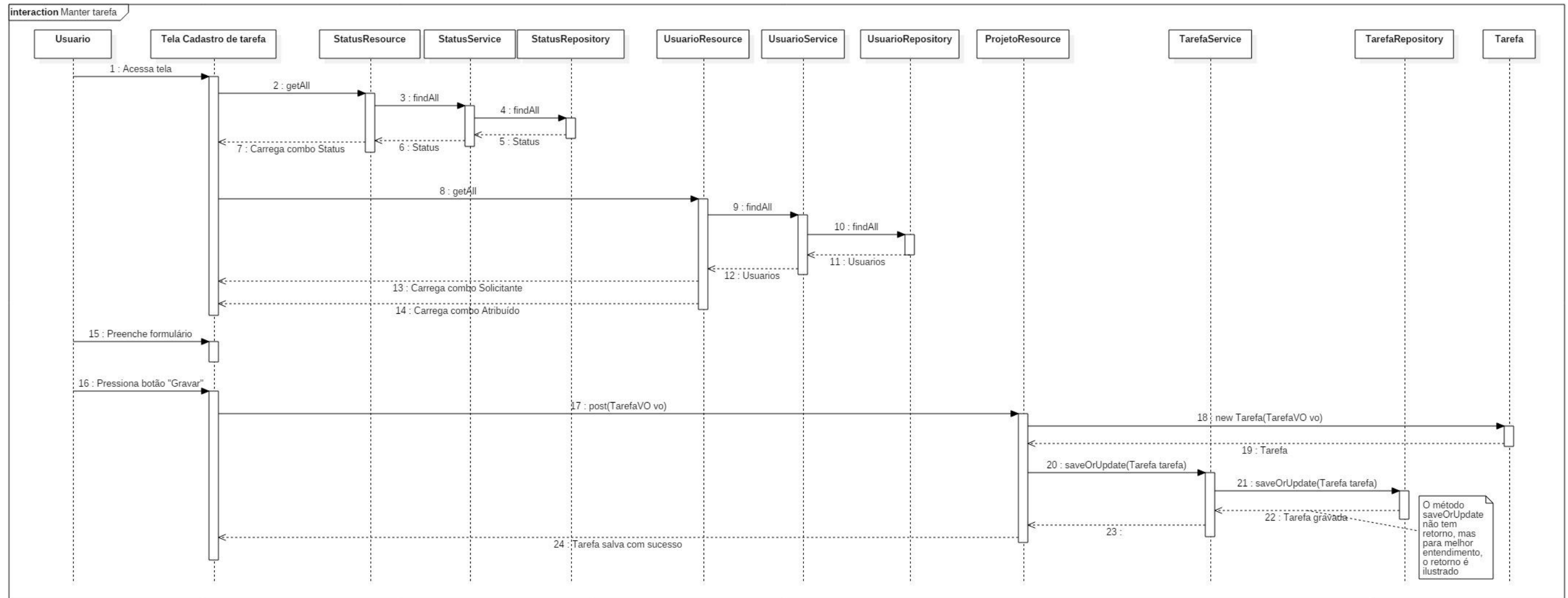
FONTE :O autor (2017)

FIGURA 42 - MANTER HISTÓRIA



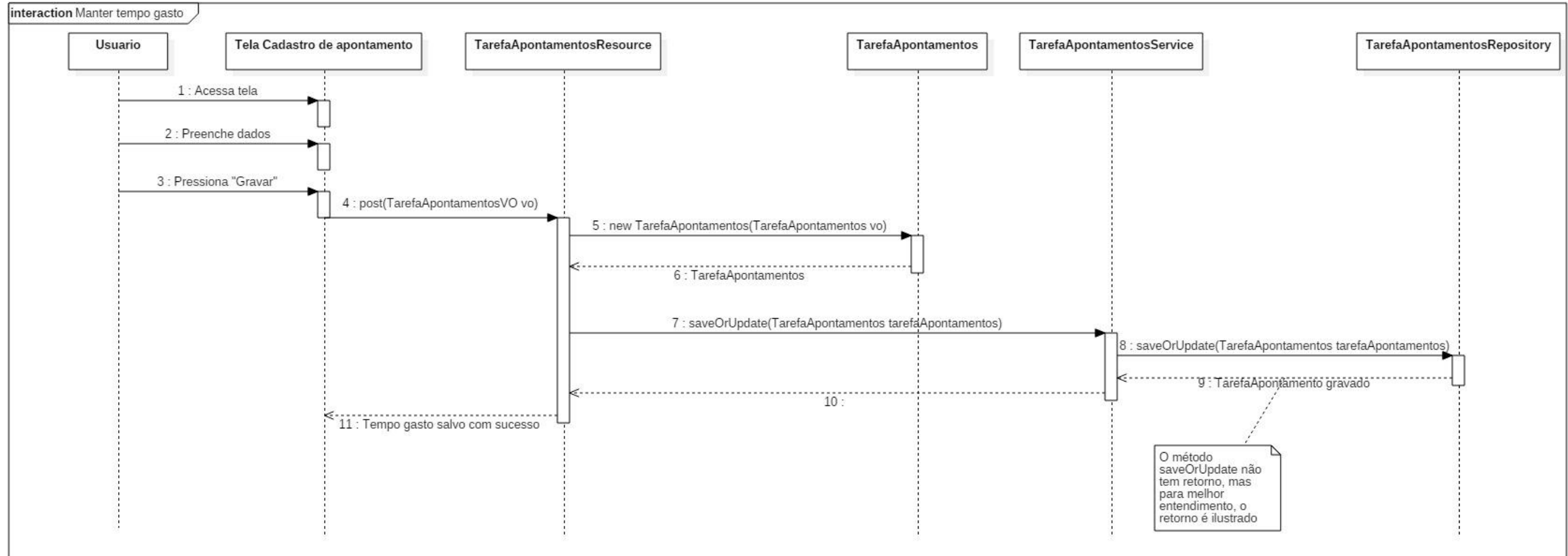
FONTE: O autor (2017)

FIGURA 42 – MANTER TAREFA



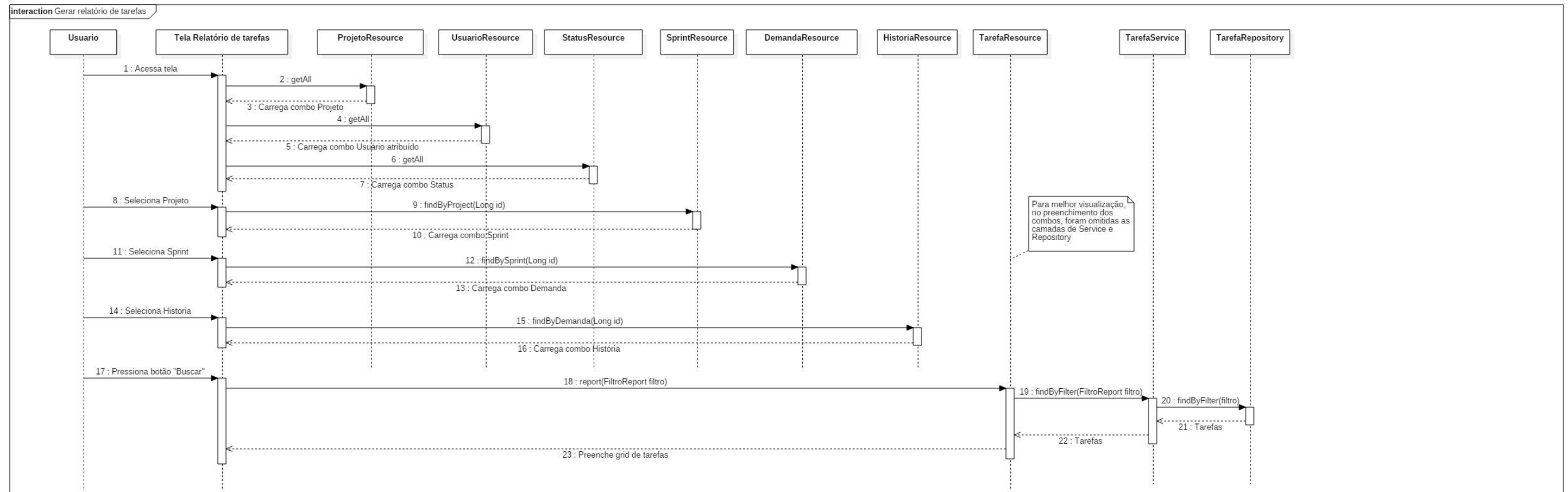
FONTE: O autor (2017)

FIGURA 43 - MANTER TEMPO GASTO



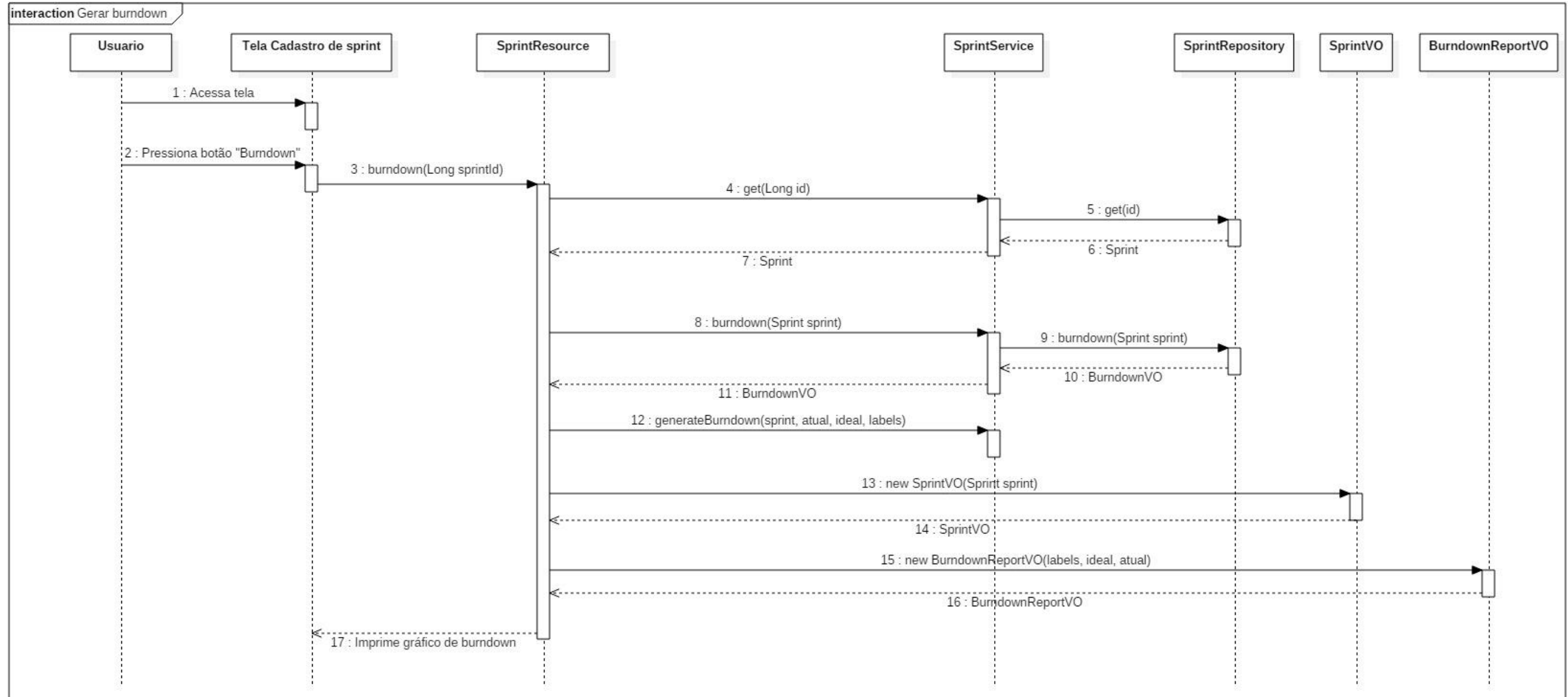
FONTE: O autor (2017)

FIGURA 44 - GERAR RELATÓRIO DE TAREFAS



FONTE: O autor (2017)

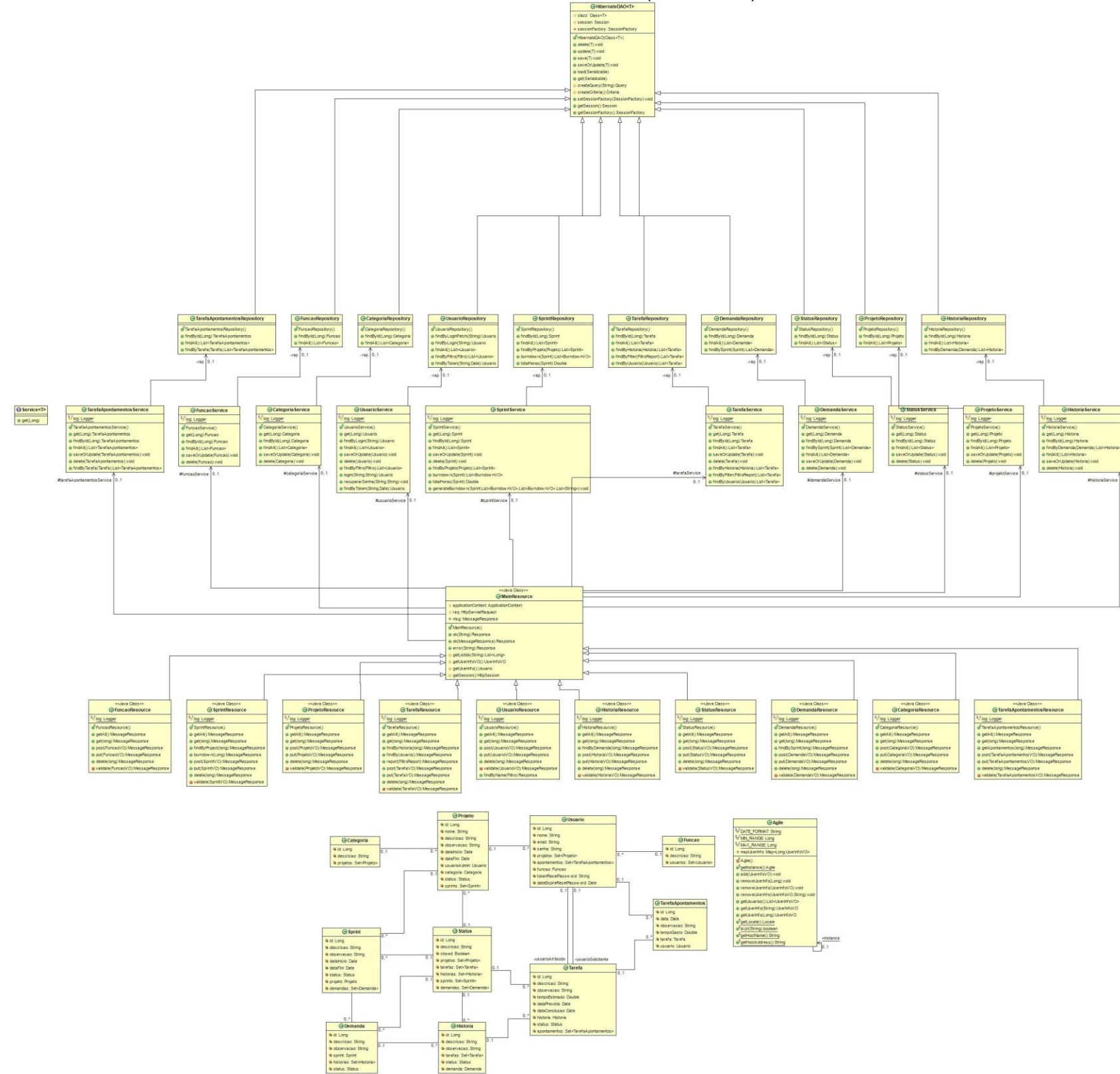
FIGURA 45 - GERAR BURNDOWN



FONTE: O autor (2017)

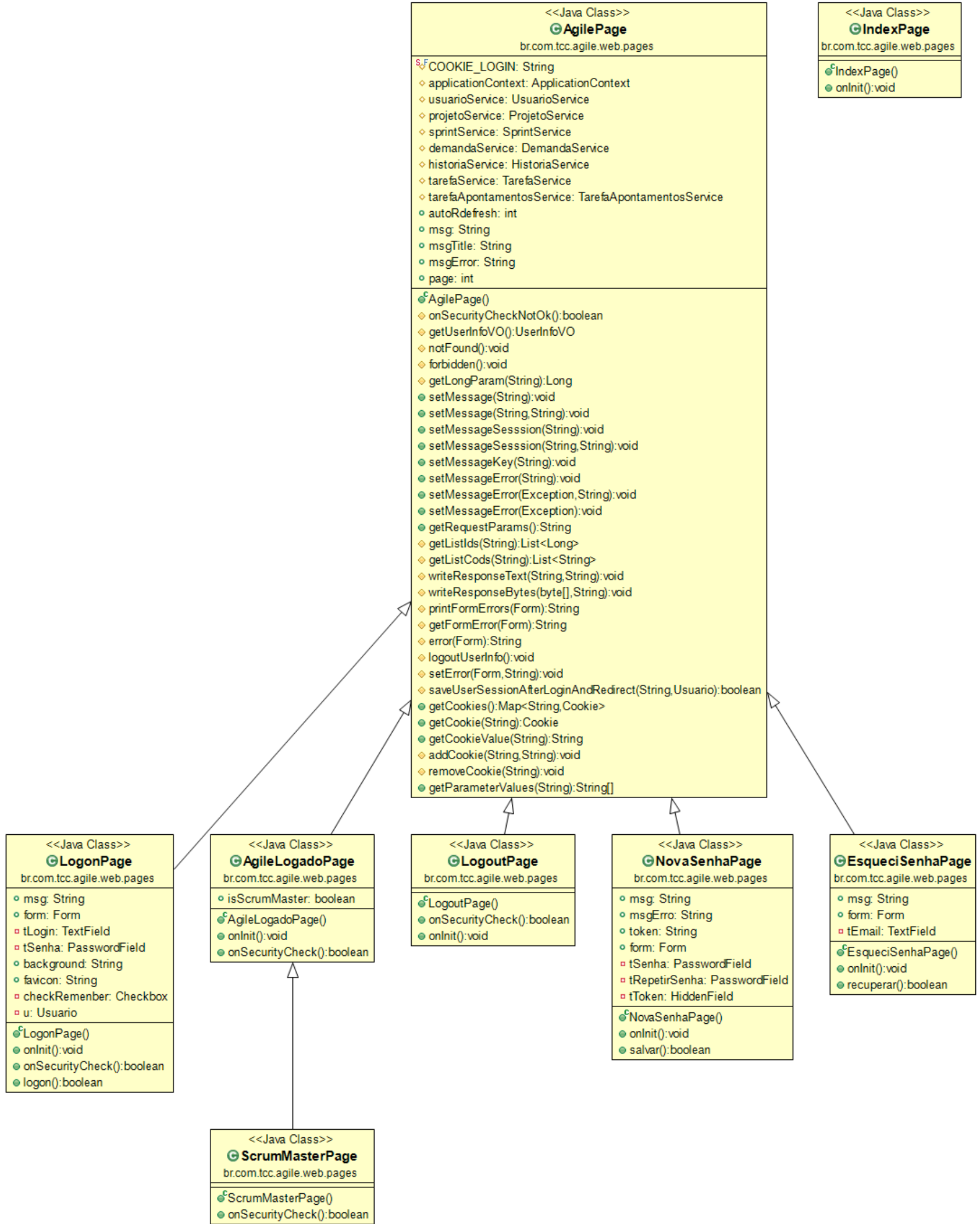
APÊNDICE J – MODELO DE OBJETOS (COMPLETO)

FIGURA 46 - DIAGRAMA DE CLASSES (COMPLETO)



FONTE: O autor (2017)

FIGURA 48 - DIAGRAMA DE CLASSES PAGES



FONTE: O autor (2017)

APÊNDICE L – CASOS DE TESTE

Tabela 2 - CASO DE TESTE INSERT

| | | | | |
|----------------------|----------------------|--|--------------------------------|---|
| Cenário | | Template de casos de testes INSERIR dados no sistema | | |
| Pré-condições | | O usuário estar logado com a permissão correspondente a rotina | | |
| Elaborador | | Guilherme | Data de elaboração | 29/11/2017 |
| Executor | | Guilherme | Data de execução | 29/11/2017 |
| Nº | Pré-condições | Entrada | Ação | Resultado |
| 1 | Nenhuma | Não preencher algum campo obrigatório (marcado com *) | Pressionar o botão Gravar | O sistema deve destacar o campo obrigatório não preenchido (FIGURA 50) |
| 2 | Nenhuma | Preencher todos os campos do formulário | Pressionar o botão Gravar | O sistema deve mostrar a mensagem de sucesso “<Entidade> gravado com sucesso” (FIGURA 51) |
| 3 | Nenhuma | Preencher todos os campos do formulário | Pressionar o botão Cancelar | O sistema deve limpar os campos do formulário (FIGURA 52) |

FONTE: O autor (2017)

FIGURA 50 - CAMPO OBRIGATÓRIO

Dados do projeto

Nome do projeto *

Descrição

Data início * **Data término**

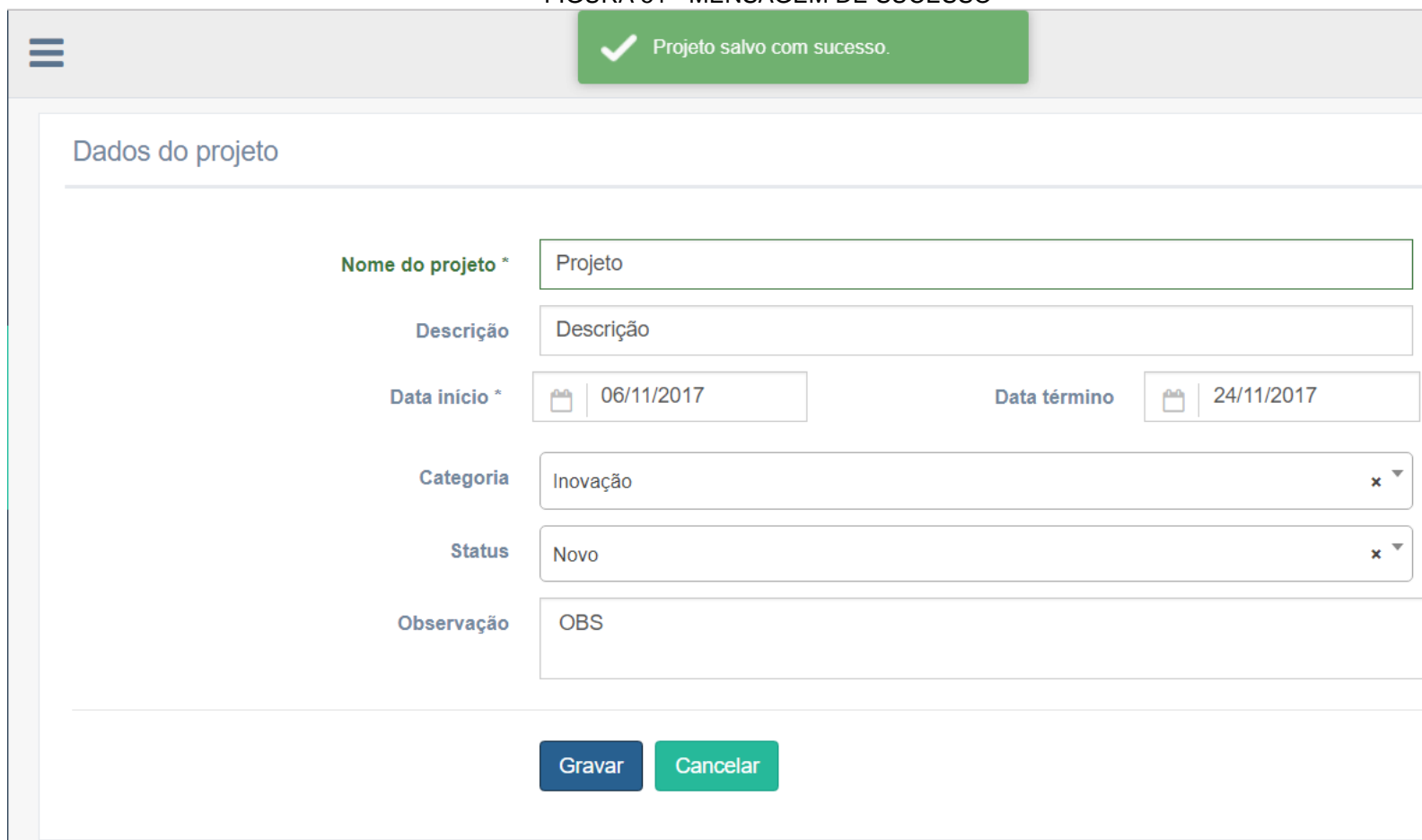
Categoria x ▾

Status x ▾

Observação

FONTE: O autor (2017)

FIGURA 51 - MENSAGEM DE SUCESSO



The image shows a web application interface. At the top, a green notification banner with a white checkmark icon contains the text "Projeto salvo com sucesso.". Below this, the main content area is titled "Dados do projeto". It contains a form with the following fields:

- Nome do projeto ***: Text input field containing "Projeto".
- Descrição**: Text input field containing "Descrição".
- Data início ***: Date picker field showing "06/11/2017".
- Data término**: Date picker field showing "24/11/2017".
- Categoria**: Dropdown menu with "Inovação" selected and a close icon (x).
- Status**: Dropdown menu with "Novo" selected and a close icon (x).
- Observação**: Text input field containing "OBS".

At the bottom of the form, there are two buttons: "Gravar" (dark blue) and "Cancelar" (teal).

FONTE: O autor (2017)

FIGURA 52 - BOTÃO CANCELAR

Dados do projeto

Nome do projeto *

Descrição

Data início *

Categoria x ▾

Status x ▾

Observação

FONTE: O autor (2017)

Tabela 3 - CASO DE TESTE UPDATE

| | | | | |
|----------------------|---|---|-----------------------------|--|
| Cenário | | Template de casos de testes ATUALIZAR dados no sistema | | |
| Pré-condições | | O usuário estar logado com a permissão correspondente a rotina | | |
| Elaborador | | Guilherme | Data de elaboração | 29/11/2017 |
| Executor | | Guilherme | Data de execução | 29/11/2017 |
| Nº | Pré-condições | Entrada | Ação | Resultado |
| 1 | Ter cadastros na base para serem listados na grid | Selecionar na grid o cadastro a ser atualizado. Não preencher algum campo obrigatório (marcado com *) | Pressionar o botão Gravar | O sistema deve destacar o campo obrigatório não preenchido (FIGURA 53) |
| 2 | Ter cadastros na base para serem listados na grid | Selecionar na grid o cadastro a ser atualizado. Preencher os campos a serem atualizados | Pressionar o botão Gravar | O sistema deve mostrar a mensagem de sucesso "<Entidade> atualizado com sucesso" (FIGURA 54) |
| 3 | Ter cadastros na base para serem listados na grid | Selecionar na grid o cadastro a ser atualizado | Pressionar o botão Cancelar | O sistema deve voltar os dados dos campos do formulário de acordo com o registro no banco de dados (FIGURA 55) |

FONTE: O autor (2017)

FIGURA 53 - CAMPO OBRIGATÓRIO ALTERAÇÃO

Dados do projeto

Nome do projeto *

Descrição

Data início * **Data término**

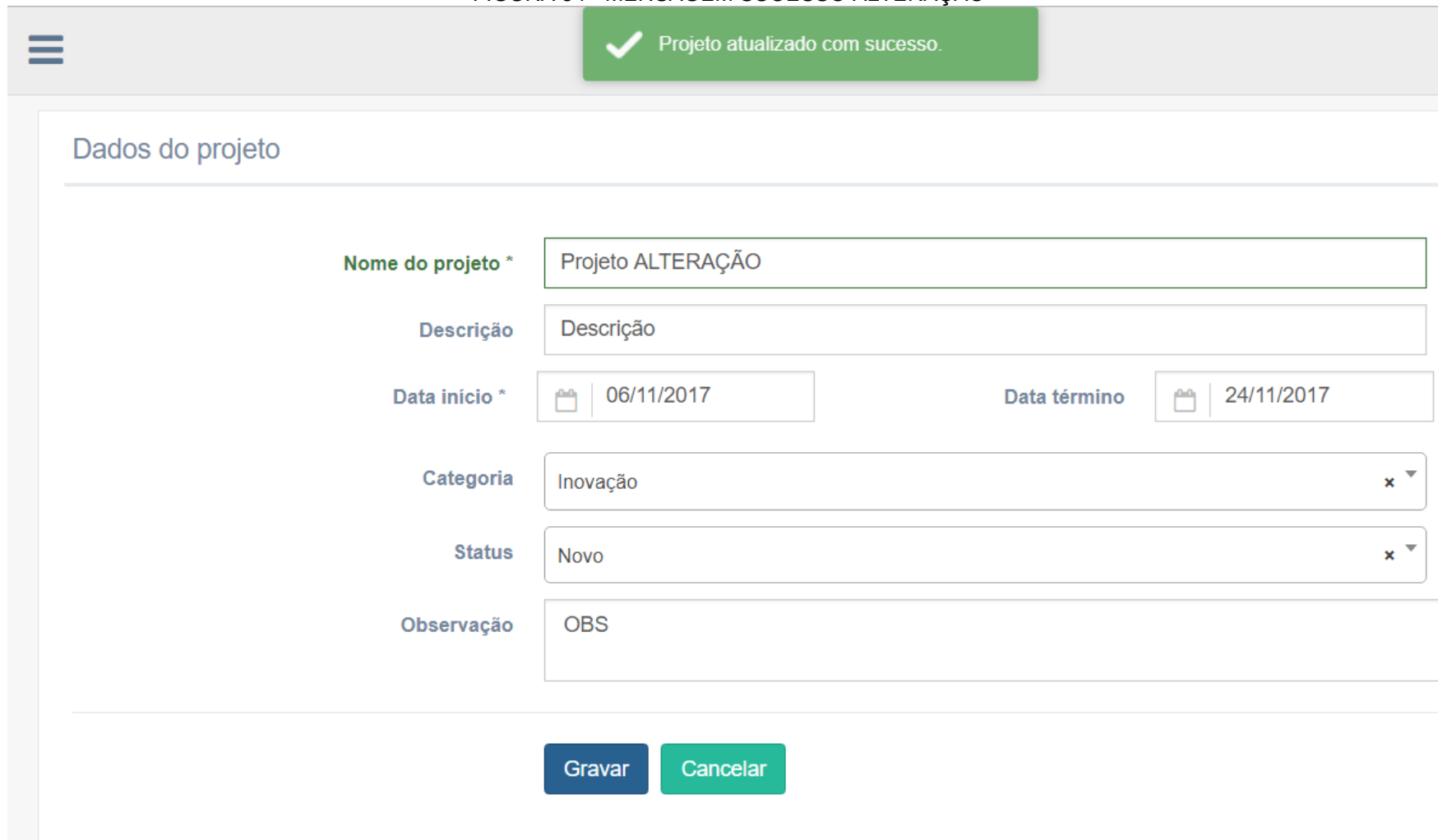
Categoria x ▾

Status x ▾

Observação

FONTE: O autor (2017)

FIGURA 54 - MENSAGEM SUCESSO ALTERAÇÃO



The image shows a web application interface. At the top, there is a green notification banner with a white checkmark icon and the text "Projeto atualizado com sucesso." To the left of the banner is a grey header bar containing a hamburger menu icon. Below the banner, the main content area is titled "Dados do projeto". It contains a form with the following fields:

- Nome do projeto ***: Text input field containing "Projeto ALTERAÇÃO".
- Descrição**: Text input field containing "Descrição".
- Data início ***: Date picker field showing "06/11/2017".
- Data término**: Date picker field showing "24/11/2017".
- Categoria**: Dropdown menu showing "Inovação" with a close icon (x) and a dropdown arrow.
- Status**: Dropdown menu showing "Novo" with a close icon (x) and a dropdown arrow.
- Observação**: Text input field containing "OBS".

At the bottom of the form, there are two buttons: "Gravar" (dark blue) and "Cancelar" (teal).

FONTE: O autor (2017)

FIGURA 55 - BOTÃO CANCELAR ALTERAÇÃO

Dados do projeto

Nome do projeto *

Descrição

Data início * **Data término**

Categoria x ▾

Status x ▾

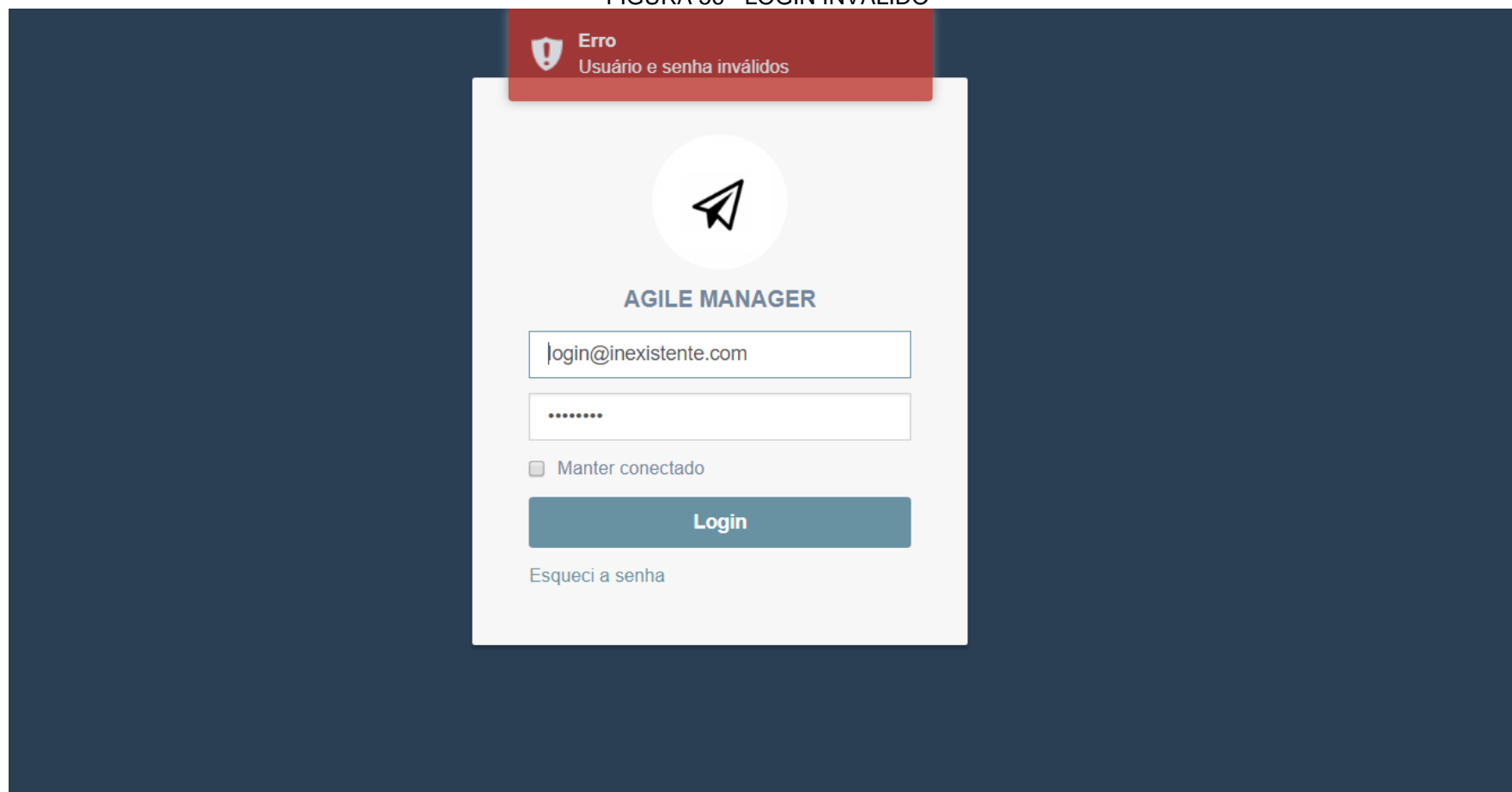
Observação

FONTE: O autor (2017)

Tabela 4 - TESTES DE PERMISSÃO

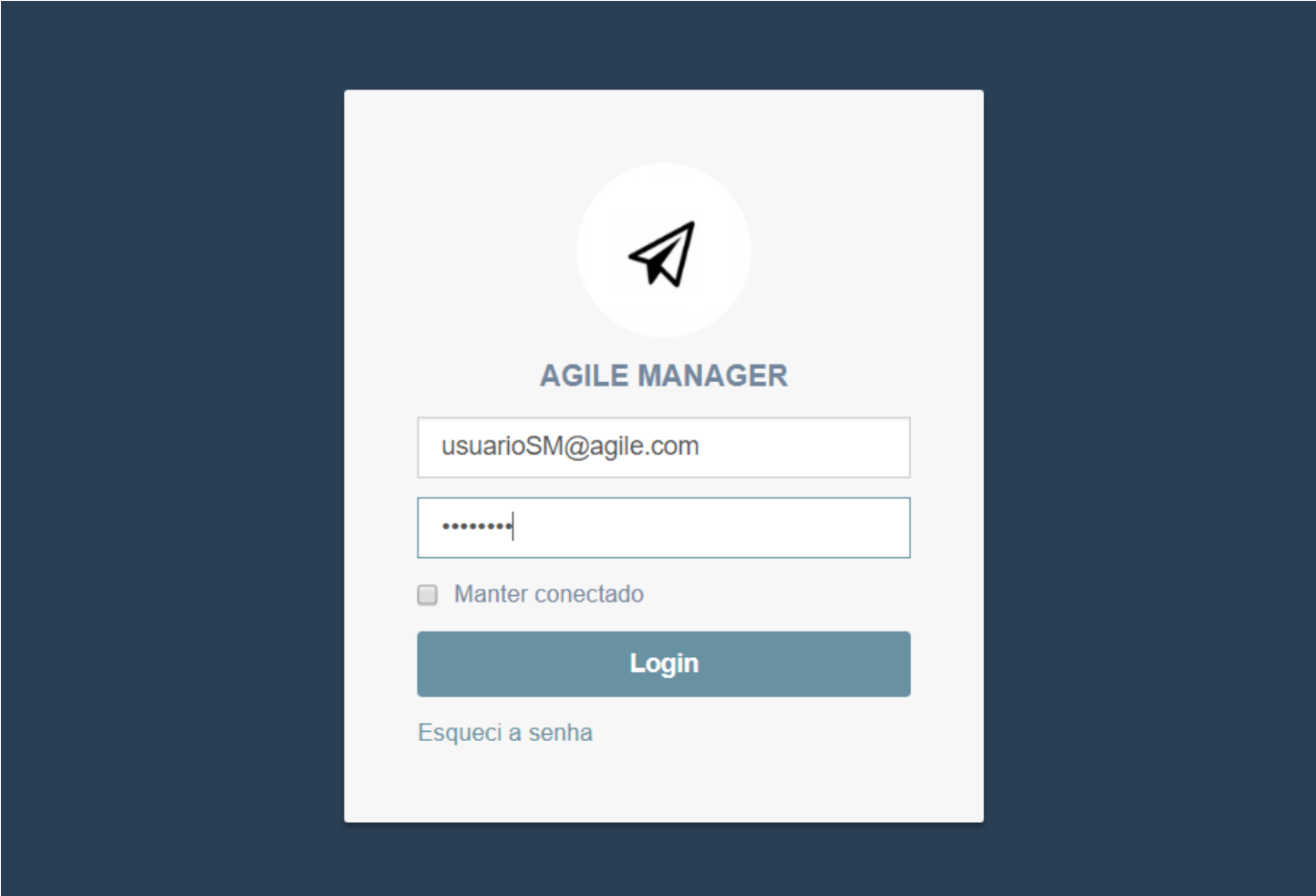
| Cenário | | Testes de permissão | | |
|------------|--------------------------------------|---|--------------------------|--|
| Elaborador | | Guilherme | Data de elaboração | 29/11/2017 |
| Executor | | Guilherme | Data de execução | 29/11/2017 |
| Nº | Pré-condições | Entrada | Ação | Resultado |
| 1 | Usuário não estar cadastrado na base | Usuário acessa página de login. Preenche com login inválido | Pressionar o botão Login | O sistema deve exibir a mensagem de erro “Erro. Usuário e senha inválidos” (FIGURA 56) |
| 2 | Usuário estar cadastrado na base | Usuário acessa página de login. Preenche com login válido | Pressionar o botão Login | O sistema deve redirecionar o usuário para a home page (FIGURA 57) e (FIGURA 58) |
| 3 | Usuário estar cadastrado na base | Usuário Desenvolvedor/Tester acessar a página de cadastro de status via barra de endereços do navegador | Acessar a página | O sistema deve redirecionar o usuário para a home page (FIGURA 59) |

FIGURA 56 - LOGIN INVÁLIDO



FONTE: O autor (2017)

FIGURA 57 - LOGIN VÁLIDO



The image shows a login form for 'AGILE MANAGER'. At the top is a circular logo with a paper airplane icon. Below the logo, the text 'AGILE MANAGER' is displayed. The form contains two input fields: the first contains the email 'usuarioSM@agile.com' and the second contains a password represented by seven dots. Below the password field is a checkbox labeled 'Manter conectado'. A blue 'Login' button is positioned below the checkbox. At the bottom of the form, there is a link that says 'Esqueci a senha'.

FONTE: O autor (2017)

FIGURA 58 - HOME PAGE

The screenshot displays the Agile Manager web application interface. The browser address bar shows the URL `localhost:8080/agile/production/cadastro/tarefasUsuario.htm`. The page title is "Agile Manager".

Left Sidebar (MENU):

- Home
- Minhas atividades
- Projetos
- Relatórios
- Configurações
- Meus dados

Main Content Area:

Minhas tarefas

Mostrar registros

Buscar:

| Descrição | Status | Projeto | Demanda | História | Visualizar |
|----------------------------------|--------|---------|---------|----------|------------|
| Não foram encontrados resultados | | | | | |

Mostrando de 0 até 0 de 0 registros

Anterior Seguinte

Footer: Agile Manager - UEPB - Especialização em Engenharia de Software

FONTE: O autor (2017)

FIGURA 59 - HOME PAGE DESENVOLVEDOR

The screenshot displays the Agile Manager web application interface. The browser address bar shows the URL `localhost:8080/agile/production/cadastro/tarefasUsuario.htm`. The page title is "Agile Manager". The user is logged in as "Usuario Dev".

The main content area is titled "Minhas tarefas" (My tasks). It features a search bar and a "Mostrar 10 registros" (Show 10 records) dropdown. Below this is a table with the following data:

| Descrição | Status | Projeto | Demanda | História | Visualizar |
|-----------|--------|-------------------|-----------------------|----------------------|------------|
| Tarefa 03 | Novo | Importação de NFe | Demanda com histórias | História com tarefas | 🔍 |
| Tarefa 05 | Novo | Importação de NFe | Demanda com histórias | História com tarefas | 🔍 |
| Tarefa 06 | Novo | Importação de NFe | Demanda com histórias | História com tarefas | 🔍 |

At the bottom of the table, it indicates "Mostrando de 1 até 3 de 3 registros" (Showing 1 to 3 of 3 records) and includes navigation buttons for "Anterior" (Previous) and "Seguinte" (Next).

The left sidebar contains a menu with the following items: Home, Minhas atividades (highlighted), Projetos, Relatórios, and Meus dados.

At the bottom right of the page, there is a footer: "Agile Manager - IIEPR - Especialização em Engenharia de Software".

FONTE: O autor (2017)