

UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL TECNOLÓGICA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

LAYLA CRISTINE DE OLIVEIRA CODOGNO

LEAF - Caderno de Receitas

CURITIBA

2017

LAYLA CRISTINE DE OLIVEIRA CODOGNO

LEAF - Caderno de Receitas

Trabalho de conclusão de curso para obtenção de grau do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná.

Orientadora: Profa. Me. Andreia de Jesus.

CURITIBA

2017

RESUMO

O presente trabalho visa apresentar à comunidade acadêmica de desenvolvedores a eficácia na utilização de metodologias ágeis para o desenvolvimento de softwares como Trabalhos de Conclusão de Curso. Além disso, apresenta-se uma análise sobre a problemática envolvida entre a formação de profissionais em um curso tecnólogo e a não compreensão dos movimentos ágeis presentes no âmbito profissional (mercado de trabalho). Para tanto, utilizou-se do *framework Scrum* e seus fundamentos emergentes do Manifesto Ágil nos quais são envolvidos sugestões e práticas destinadas a melhoria do desempenho das equipes de desenvolvimento, conseqüentemente um processo mais eficiente. Assim, é descrito o desenvolvimento do *software Leaf* - Caderno de Receitas o qual tem como finalidade o cadastro de receitas online, podendo gerenciar receitas, ingredientes e categorias, bem como pesquisar e listar receitas de outros usuários.

Palavras-Chave: Metodologia Ágil; Desenvolvimento de Sistemas; Scrum; Receitas.

ABSTRACT

This work aims to present to the academic community of developers the effectiveness in the use of agile methods for the development of softwares as academic purposes. In addition, an analysis is presented on the problems involved between the training of professionals in a technological course and the lack of understanding of the agile movements present in the professional context (job market). To do so, we used the Scrum framework and its emerging foundations of the Agile Manifesto in which suggestions and practices are designed to improve the performance of the development teams, consequently a more efficient process. It is described the development of the software Leaf - Cook Book which has the purpose of registering online recipes, being able to manage recipes, ingredients and categories, as well as list and search recipes of other users.

Key-Words: Methodology. Agile. Development. Software. Scrum. Recipes.

SUMÁRIO

RESUMO	2
ABSTRACT	3
SUMÁRIO	4
LISTA DE FIGURAS	6
LISTA DE TABELAS	7
INTRODUÇÃO	8
Problema	9
Objetivos	10
Objetivos Gerais	10
Objetivos Específicos	10
Justificativas	11
FUNDAMENTAÇÃO TEÓRICA	12
A Filosofia do Scrum	12
Gerência de Projetos com Scrum: principais fundamentos	13
METODOLOGIA DO TRABALHO	16
Descrição do Problema a ser solucionado	16
Gerência de Projeto com Scrum: aplicação da metodologia	17
Modelagem do Software	24
Requisitos Funcionais e Não Funcionais	24
Diagrama de Caso de Uso	24
Diagrama de Classes	25
Diagrama de Sequência	25
Diagrama Banco de Dados	25
APRESENTAÇÃO DO SISTEMA	27
O Sistema "Leaf - Caderno de Receitas"	27
CONSIDERAÇÕES FINAIS	35
REFERÊNCIAS	37
APÊNDICE A	38
APÊNDICE B	40
APÊNDICE C	42
APÊNDICE D	43

APÊNDICE E	48
APÊNDICE F	49
APÊNDICE G	55

LISTA DE FIGURAS

- Figura 1 - As etapas da metodologia *Scrum*.
- Figura 2 - Modelagem do Banco de Dados.
- Figura 3 - Gerência das etapas de desenvolvimento do projeto pelo *Trello*
- Figura 4 - Tela Inicial do Sistema / Feed de Receitas
- Figura 5 - Tela de Filtro de Receitas por Categorias
- Figura 6 - Tela de Filtro de Cadernos por Nome de Usuário
- Figura 7 - Tela de Login
- Figura 8 - Tela de Novo Usuário / Cadastro
- Figura 9 - Tela de Listagem de Categorias
- Figura 10 - Tela de Cadastro de Ingredientes
- Figura 11 - Tela de Listagem de Receitas
- Figura 12 - Tela de Cadastro de Receita
- Figura 13 - Tela de Visualização de Receita
- Figura 14 - Tela de Caderno do Usuário
- Figura 15 - Tela de Edição de Perfil Usuário
- Figura 16 - Diagrama de Caso de Uso Simplificado
- Figura 17 - Diagrama de Classe
- Figura 18 - Diagrama de Sequência UC001
- Figura 19 - Diagrama de Sequência UC002
- Figura 20 - Diagrama de Sequência UC003
- Figura 21 - Diagrama de Sequência UC004
- Figura 22 - Diagrama de Sequência UC005
- Figura 23 - Diagrama de Sequência UC006

LISTA DE TABELAS

Tabela 1: Equipe de Trabalho

Tabela 2 - *Backlog* inicial

Tabela 3 - *Sprint* 1

Tabela 4 - *Sprint* 2

Tabela 5 - *Sprint* 3

Tabela 6 - *Sprint* 4

Tabela 7 - *Sprint* 5

Tabela 8 - *Sprint* 6

Tabela 9 - *Sprint* 7

Tabela 10 - Análise comparativa entre metodologias

Tabela 11 - Caso de Uso UC001

Tabela 12 - Caso de Uso UC002

Tabela 13 - Caso de Uso UC003

Tabela 14 - Caso de Uso UC004

Tabela 15 - Caso de Uso UC005

Tabela 16 - Caso de Uso UC006

Tabela 17 - Caso de Uso UC007

1. INTRODUÇÃO

Desde o primeiro dia de aula na Universidade Federal do Paraná, no curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas, nos ensinam a "resolver problemas". A própria profissão de desenvolvedor de sistemas surgiu nessa crescente necessidade de automatizar, viabilizar e melhorar os processos do dia-a-dia. Diante dessa demanda, ao pensar no desenvolvimento de um sistema como produto final de um Trabalho de Conclusão de Curso, comecei a procurar os problemas a serem resolvidos ao redor; "O que pode ser melhorado?" ; "O que pode ser otimizado?".

Foi desta forma que surgiu a ideia de um sistema com o tema Caderno de Receitas. Se analisarmos, dentro da realidade de muitas cozinhas, sempre teve a presença de um caderno ou similar mas que atualmente, encontra-se esquecido dentro das gavetas, devido a grande quantidade de recursos tecnológicos disponíveis para pesquisar uma nova receita em meio a tantas opções.

Em consequência dessa variedade de recursos e suas constantes atualizações, o cidadão moderno passou a desenvolver novas habilidades para a execução de suas atividades no dia-a-dia, como: agilidade, autonomia e flexibilidade. E essa questão pode ser estendida para o processo de desenvolvimento de sistemas, pois as demandas da sociedade estão cada vez mais latentes no que diz respeito a agilidade com que esses recursos precisam ser disponibilizados. Logo, torna-se necessário incentivar na comunidade de desenvolvedores de *softwares* novas práticas de análise e desenvolvimento de sistemas que venham atender a essa demanda.

Considerando o que foi colocado, o objetivo principal deste trabalho é aplicar uma metodologia de desenvolvimento de *software* moderna para viabilizar um sistema que se encaixe nas necessidades temporais da sociedade atual, e também comprovar como essa metodologia atende essa característica e, com isso, instigar o seu uso na comunidade acadêmica de desenvolvedores de *software*.

Para tanto, o projeto nomeado como "*Leaf - Caderno de Receitas*", utiliza uma metodologia ágil de desenvolvimento, denominada *Scrum*, em conjunto com a linguagem de programação *Ruby*, para desenvolver um sistema web de gerenciamento de receitas.

1.1. Problema

As metodologias de desenvolvimento ágil começaram a evoluir a partir dos anos 1990, mas em 2001 foi consolidado o Manifesto Ágil que trouxe várias alternativas ao método tradicional de documentação e processos de desenvolvimento, considerados pesados e complicados (BARROS *et al*, 2010). Mesmo com dezesseis anos de Manifesto e a evolução das "novas" metodologias, ainda encontram-se muitas barreiras dentro dos meios tradicionais de desenvolvimento de sistemas.

Foi realizada uma pesquisa durante a execução desse projeto para identificar e compreender melhor a visão entre os ambientes acadêmico e profissional. Para isso foram realizadas perguntas sobre as metodologias utilizadas e os motivos pela escolha. A pesquisa encontra-se detalhada no Apêndice A.

Identificamos com a pesquisa dentro do ambiente acadêmico que 66% (sessenta e seis por cento) dos profissionais utiliza metodologias tradicionais para o desenvolvimento de sistemas. Dentro disso, foi evidenciado a documentação como ponto forte da utilização da metodologia tradicional. Em contrapartida dentro do ambiente profissional (mercado de trabalho) 100% (cem por cento) dos profissionais entrevistados alegaram utilizar metodologias ágeis e vários fatores foram evidenciados, desde agilidade e flexibilidade no processo de desenvolvimento, melhora no relacionamento com o cliente, simplicidade, dentre outros.

Além dessa pesquisa, também foi feito uma análise de Trabalhos de Conclusão do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Setor de Educação Profissional e Tecnológica da UFPR em busca das metodologias de desenvolvimento utilizadas. No ano de 2016 61% (sessenta e um por cento) dos 18 (dezoito) documentos analisados descreveram utilizar metodologias ágeis para o desenvolvimento do sistema, porém 72% (setenta e dois por cento) desses trabalhos apresentaram artefatos de modelagens tradicionais como itens necessários para o início do desenvolvimento. A exigência desses itens em um projeto de desenvolvimento de sistemas acaba por ir na contramão do que é colocado pelo Manifesto Ágil:

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.
Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda." (Manifesto Ágil, 2001)

É importante colocar que na comunidade de desenvolvedores, não há uma melhor ou uma pior forma de se desenvolver sistemas, o próprio *Scrum* pode ser adaptado e utilizado em conjunto a outras metodologias. Porém, analisando a forma com que essas novas metodologias estão sendo apresentadas no ambiente acadêmico, fecho esse capítulo com a pergunta: "O ambiente acadêmico está permitindo que seus egressos sejam capazes de produzir, de maneira eficaz, sistemas utilizando metodologias ágeis?".

1.2. Objetivos

Esta seção apresenta os objetivos do sistema, estabelecendo o que se espera conquistar ao final do desenvolvimento do projeto.

1.2.1. Objetivos Gerais

O objetivo geral deste projeto é desenvolver um estudo de caso que aplique a metodologia ágil – *Scrum* em todo o processo de desenvolvimento de um *software*, a fim de evidenciar a suas vantagens para a comunidade acadêmica.

1.2.2. Objetivos Específicos

Para atingir o objetivo geral deverão ser cumpridos os seguintes objetivos específicos:

- Definir e Analisar o escopo do projeto e as tecnologias a serem utilizadas.
- Realizar um comparativo entre as metodologias dentro da comunidade de desenvolvimento.
- Implementar um sistema dentro das diretrizes da metodologia ágil escolhida.

1.3. Justificativas

Viabilizar um projeto de conclusão de curso utilizando a metodologia de desenvolvimento ágil mais crescente do mercado motivou a construção desse projeto.

Além disso, o desenvolvimento de *software* envolve pessoas e mudanças, ou seja, ao fazer um planejamento completo com a totalidade dos itens a serem implementados e só depois dar início ao desenvolvimento, aumentam as chances de insucesso, se comparado a um projeto incremental. Segundo Sommerville (2010), ao seguir a forma incremental é possível entregar itens de alta prioridade ao cliente, mostrando rapidamente o valor do sistema. O que nas metodologias tradicionais demora meses para o usuário validar, na incremental em semanas o usuário já pode utilizar os requisitos na prática e verificar possíveis erros ou alterações.

Como dentro da comunidade de desenvolvedores há variedade na forma de desenvolver, bem como dentro do mercado de trabalho cada empresa tem sua cultura, considera-se importante dentro da Universidade o acesso a diferentes metodologias, bem como a prática de suas aplicações de forma concisa.

Logo, o propósito deste projeto é mostrar as vantagens e desvantagens do uso da metodologia ágil no desenvolvimento de uma aplicação web de complexidade real dentro do ambiente acadêmico. Com isso, poder de alguma forma contribuir para a formação de novos profissionais na área de TI, mais especificamente de formação tecnológica, a qual é voltada ao mercado de trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. A Filosofia do *Scrum*

Uma pesquisa feita pelo *The Standish Group* em 2015 concluiu que projetos realizados com o *Scrum* ou métodos similares (ágeis) são 3.5 vezes mais assertivos. Por ser um *framework* multidisciplinar, em que é trabalhado conceitos como auto-organização, trabalho em equipe, motivação, relacionamento, bem como vários outros conceitos já existentes no mercado, o *Scrum* tem se tornado o método mais eficaz no desenvolvimento de *softwares*. Pois, sua aplicação é feita de maneira simples e flexível, adaptando-se a realidade de cada projeto, priorizando a entrega rápida e efetiva.

Sabbagh (2014), precursor do *Scrum* no Brasil, sintetiza as principais características da metodologia: a redução dos riscos de insucesso, a entrega de valor mais rápido e, conseqüentemente, a melhor eficiência em lidar com as inevitáveis mudanças de escopo. Também menciona o aumento da qualidade do produto entregue e a melhor produtividade das equipes.

O primeiro fundamento, em contrapartida as metodologias tradicionais, surgiu com o manifesto ágil no quesito satisfação do cliente. Durante o desenvolvimento de um sistema, na forma tradicional, se tem um levantamento inicial de requisitos, alocação de recursos baseado nas métricas iniciais, somente então é dado início ao processo de desenvolvimento. Neste processo tradicional, o cliente só terá retorno após a conclusão de todos os requisitos inicialmente levantados o que pode levar meses e até anos. O *Scrum*, juntamente com as metodologias ágeis vieram para quebrar barreiras como essa.

"Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de *software* de valor." (Manifesto Ágil, 2001)

A criação do *Scrum* surgiu de uma necessidade real, após o *Federal Bureau of Investigation* (FBI) cancelar mais um de seus projetos de modernização de *software*. Jeff Sutherland decidiu reinventar a forma como as coisas eram. Em seu livro Sutherland (2014) conta como os projetos anteriores ao dele pareciam ser perfeitos no

papel e no planejamento, feito pelos melhores analistas, e como rapidamente se tornavam uma farsa.

No ano de 2010, o FBI já havia cancelado três projetos, gasto em torno de US\$ 400 milhões e ainda não tinham um *software* para automatizar os processos internos. Sutherland após avaliar toda a situação estimou a entrega do sistema em 20 meses, utilizando metade dos recursos disponíveis e isso só foi possível utilizando os preceitos do manifesto ágil.

Dentro do mercado atual de desenvolvimento de *softwares* temos inúmeros casos de utilização do *Scrum* para o desenvolvimento, um forte exemplo foi a construção da ferramenta *AdWords* do *Google*.

"Scrum é um framework para desenvolver e sustentar produtos complexos". (Scrum Guide – 2016, página 3)

Scrum é um *framework* que disponibiliza uma base de princípios, valores e práticas para as equipes com o foco de torná-las mais eficientes, com alto nível de envolvimento e comunicação. Alinhando-se ao mercado atual de desenvolvimento de *softwares* em que há necessidade de eficácia em prazos curtos deixando de lado as burocracias desnecessárias.

2.2. Gerência de Projetos com *Scrum*: principais fundamentos

De acordo com a comunidade de movimento ágil brasileira, metodologias ágeis de desenvolvimento de *software* são iterativas, ou seja, o trabalho é dividido em iterações, e no caso do *Scrum* são chamadas de *Sprints*.

A Figura 1 ilustra como se dá o desenvolvimento utilizando o *Scrum*. Temos a esquerda a visão inicial do que será o produto e a partir disso uma definição primária dos itens em *Backlog*. Desses itens são definidos quais serão priorizados para o desenvolvimento no *Sprint Backlog*. A partir dessa definição inicia-se uma nova *Sprint*, a qual se segue por um intervalo específico de tempo destinado a realização das tarefas previamente mapeadas.

O ciclo de vida de uma *Sprint* tem, aproximadamente, entre duas e quatro semanas (SCHWABER, et. Al. 2002; BISSI, 2007). Durante a *Sprint* são feitas várias

reuniões rápidas para compartilhar os avanços e as dificuldades de cada membro da equipe.

O resultado de cada *Sprint* são funcionalidades entregáveis e funcionais denominadas incremento. Ao se encerrar uma *Sprint*, inicia-se outra com um novo conjunto de funcionalidades e esse ciclo se repete até a finalização do sistema.

Scrum

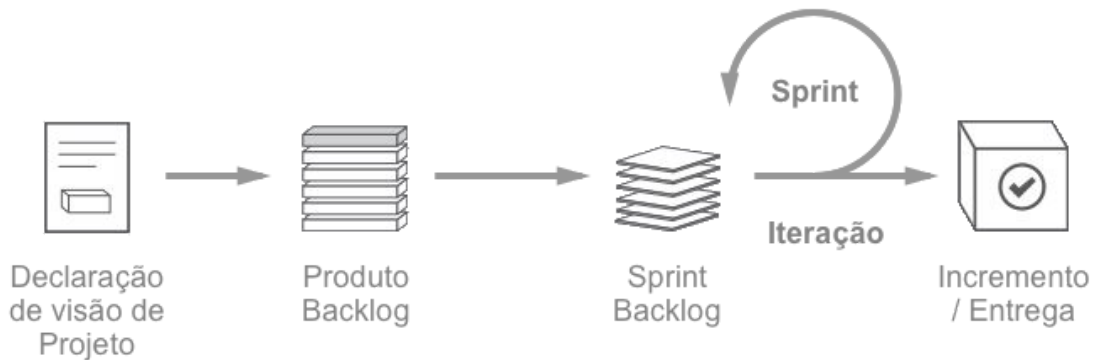


Figura 1: As etapas da metodologia *Scrum*.

Como mencionado acima, cada funcionalidade mapeada deve seguir um conjunto de etapas até a sua conclusão. Segue abaixo as definições de cada etapa:

- **Backlog:** Todo o *Roadmap* do sistema é mapeado nesta etapa de forma ordenada ou priorizada representando uma visão geral do Produto. Os itens desta etapa são todos aqueles que envolvem funcionalidades ou melhorias a serem desenvolvidas e podem ser reordenados e detalhados de acordo com o momento do projeto.
- **Sprint:** Nessa etapa é colocada a previsão dos itens a serem realizados no próximo ciclo (*sprint*). A escolha desses itens é feita de acordo com a somatização de seu grau de dificuldade dentro da capacidade de trabalho do time durante o ciclo. A ordenação desta etapa também é feita do mais ¹prioritário ao menos prioritário, ficando no topo da lista os itens a serem desenvolvidos por primeiro.
- **Done:** São os itens homologados e distribuídos em produção. Todos os itens priorizados na *Sprint* que efetivamente atingiram o status de "*pronto*" são efetivamente entregues para o cliente em produção. Os itens dessa etapa jamais

¹ Roadmap - uma espécie de "mapa" que visa organizar as metas de desenvolvimento de um software.

voltam para desenvolvimento, dessa forma, quando há necessidade de alterações ou mudanças para uma nova adequação do produto, outros itens são criados no *Backlog* e priorizados conforme necessidade.

Por se tratar de um *framework* e não um processo padronizado o *Scrum* pode assumir várias formas. Como citado no artigo publicado no site *Mindmaster* (MINDMASTER, 2017), o *Scrum* é como as fundações de uma construção, seriam os principais componentes estruturais. Dessa forma, não é possível alterar fundamentalmente seus princípios ou valores sem que haja o risco de queda, porém é possível personalizar seu interior, acrescentando recursos e personalizando artefatos para que se tenha um processo funcional dentro de uma realidade específica. A base dentro do *Scrum* é composta pelas práticas descritas abaixo.

- Gerente de Produto (*Product Owner*) responsável pela comunicação dos objetivos que devem ser alcançados pela equipe.
- *Scrum Master* tem o papel de facilitador, ele é responsável por ajudar a equipe a resolver problemas e realizar melhorias bem como filtrar as informações externas evitando interferências no processo.
- Time *Scrum* é a junção de todos os membros em uma equipe multidisciplinar.

As equipes dentro do *Scrum* são auto-gerenciáveis e com cinco a nove membros, caso haja dentro do projeto equipes maiores a sugestão é que seja subdividida em grupos menores para melhor eficácia.

3. METODOLOGIA DO TRABALHO

Para o desenvolvimento desse projeto, com a aplicação da metodologia *Scrum*, foi escolhido o tema "caderno de receitas online". Assim, um sistema web de complexidade real foi desenvolvido.

3.1. Descrição do Problema a ser solucionado

O tema escolhido para a aplicação foi Caderno de Receitas visando automatizar um processo tradicional dentro das cozinhas melhorando a visualização das receitas bem como facilitar o compartilhamento.

O sistema teve como principal característica o cadastro de receitas, e a partir disto criou-se uma visão de produto na qual foram definidos os elementos necessários para atingir essa meta. O texto sobre a visão do produto encontra-se no Apêndice B.

Dentro dessa visão de produto foram definidos requisitos para atender a demanda. Em uma receita tem-se alguns elementos básicos como a listagem dos ingredientes e suas respectivas quantidades. Em um caderno é simples a forma com que se descreve esses itens mas ao digitalizar esse processo e padronizá-lo é necessário construir itens auxiliares.

Para esse caso, decidiu-se realizar a conversão de unidades. Dessa forma, os itens líquidos seriam sempre inseridos como *mililitros* ou *Litros* e com esses dados normalizados outros processos, como por exemplo a formulação de uma lista de compras, seriam mais assertivos.

Exemplificando a complexidade do sistema, outro item é a modelagem das listas de ingredientes e categorias. Discutido com os membros da equipe, uma sistema no qual cada usuário possui sua base de dados própria para esses itens tornaria a aplicação densa e com repetições desnecessárias, já que muitos usuários teriam, por exemplo, itens como água, farinha de trigo e óleo vegetal em suas listas de ingredientes, por serem ingredientes comuns na composição de várias receitas.

Como solução a esse caso, decidiu-se sobre o conceito de Ingredientes Comuns e Categorias Comuns. Uma base de dados geral na qual os itens mais comuns seriam coletados e disponibilizados de forma geral, evitando uma base de dados inflada.

3.2. Gerência de Projeto com *Scrum*: aplicação da metodologia

O projeto foi desenvolvido utilizando 100% de metodologias ágeis. Essa escolha se deve pela agilidade e flexibilidade do processo de desenvolvimento de *software* e todos os outros benefícios descritos nos capítulos anteriores a esse.

O gerenciamento das etapas na ferramenta *Trello* facilitou a visão geral do projeto ao longo de seu desenvolvimento, como mostra a Figura 3. Nesta figura, tem-se como exemplo os itens do quarto ciclo (*sprint*) e nela podemos identificar uma visão *Burning Down*² do projeto como um todo.

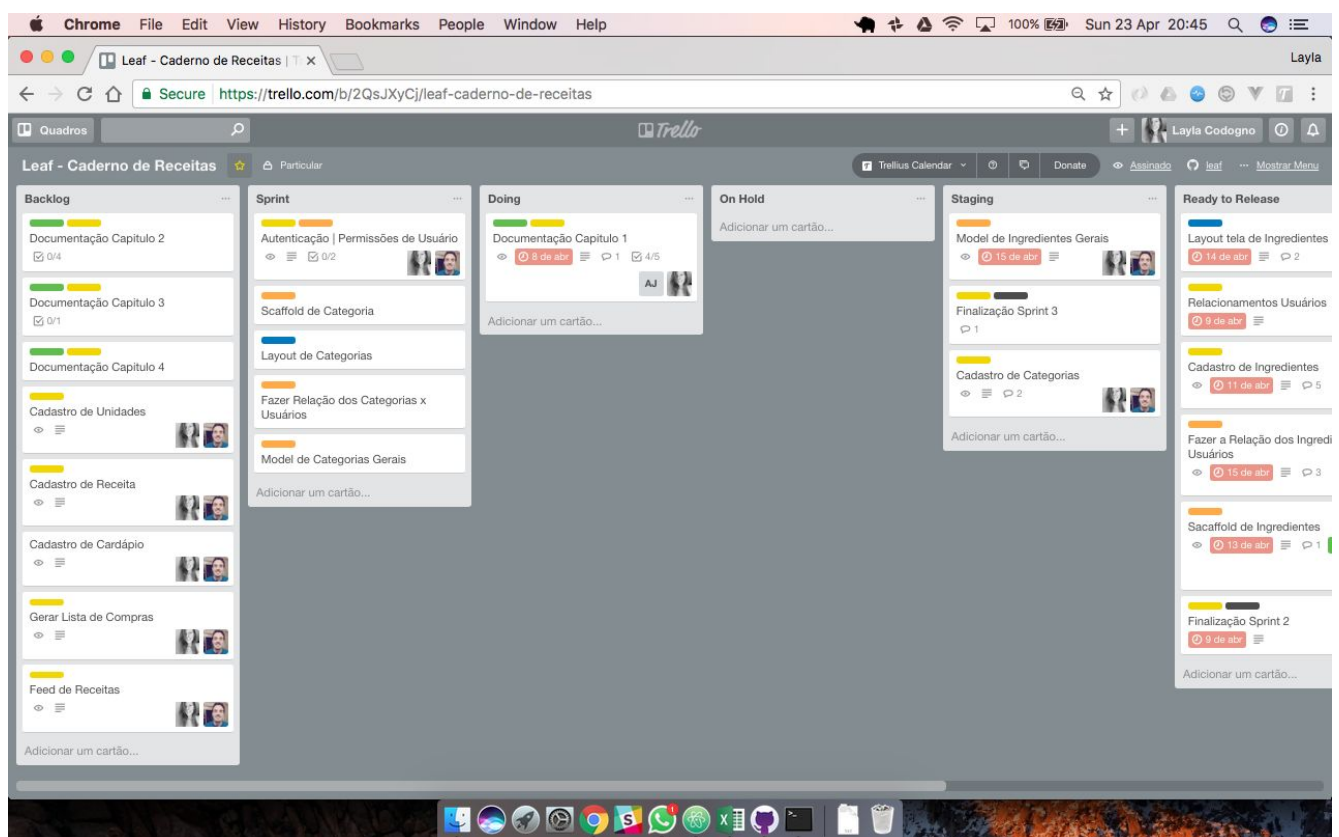


Figura 3: Gerência das etapas de desenvolvimento do projeto pelo *Trello*.

Para a melhor visualização do andamento do projeto, foram incluídas algumas sub-etapas no processo além das tradicionais descritas no *Scrum*. As sub-etapas inseridas foram:

² *Burning Down* é um gráfico utilizado para representar visualmente o progresso do trabalho em desenvolvimento.

- *Doing*: É a sub-etapa da *Sprint* que identifica os itens que estão sendo realizados naquele momento pelo desenvolvedor. Essa etapa reflete o andamento do ciclo de desenvolvimento, dessa forma é possível ter uma visão maximizada do progresso.
- *on Hold*: São os itens que começaram a ser construídos porém devido algum evento precisaram ser pausados. Essa etapa não está definida no modelo padrão do *framework*, porém devido a necessidade do time, foi implementada para controlar de forma mais detalhada o fluxo de cada item. Quando um item for movido para essa etapa, é definido o motivo da pausa e o evento esperado para retorná-lo à etapa de realização.
- *Staging*: São os itens que já foram construídos e precisam ser testados e validados. Essa etapa é crucial para a definição de "*pronto*" escolhida pela equipe. Nesse caso, os itens só são considerados prontos após passaram por todos os critérios de avaliação.
- *Ready to Release*: São os itens testados prontos para serem distribuídos em produção. Nesta etapa nenhum dos itens sofrerá alterações ou tarefas adicionais, dessa forma ao final do ciclo, apenas os itens dessa etapa são considerados aptos a entrega.

A equipe deste trabalho foi composta por quatro membros, cada um representando um papel específico no projeto, conforme descreve Tabela 1. A adaptabilidade e flexibilidade do *Scrum* trouxe benefícios para a equipe, evidenciando principalmente o fato de ser composta por apenas um membro no time de desenvolvimento.

Tabela 1: Equipe de Trabalho

Membro	Responsabilidade
Pedro Belli	Product Owner
Andreia de Jesus	Scrum Master
Alexander Kutzke	Product Owner / Tester

Layla Codogno	Desenvolvedora
---------------	----------------

Para o desenvolvimento do sistema teve-se um prazo de dezesseis semanas (quatro meses) e para essa realidade foram definidas *Sprints* com duração de duas semanas (quinze dias). Os itens em *Backlog* foram mapeados logo no início do projeto em conjunto com os orientadores e foram reavaliados ao longo dos ciclos.

As reuniões de definição de *Sprint* e *Sprint Review* foram condensadas em um mesmo dia, o qual foi definido para facilitar o encontro de todos os membros. Nessa reunião a desenvolvedora tinha como objetivo apresentar as funcionalidades finalizadas e em produção, coletar o *feedback* de todos e revisar em conjunto aos outros membros os aprendizados positivos obtidos durante a *Sprint* e os itens de melhoria. Em seguida já se dava início a próxima *Sprint* priorizando e detalhando os itens a serem desenvolvidos na sequência.

Nas tabelas abaixo é possível visualizar o comparativo entre o planejamento realizado inicialmente de forma enxuta (Tabela 2 - *Backlog* Inicial) e os itens executados efetivamente em cada *Sprint*.

Tabela 2 - *Backlog* inicial

Data de Início	Data de Entrega	Backlog Projeto Inicial
01/03/2017	20/06/2017	Configuração Inicial do Projeto
		Cadastro de Ingredientes
		Cadastro de Categorias
		Cadastro de Unidades
		Cadastro de Receitas
		Cadastro de Cardápios
		Gerar Lista de Compras
		Feed de Receitas

Tabela 3 - *Sprint* 1

Sprint 1	
-----------------	--

Data Inicio	Data Entrega	Cards Relacionados
01/03/2017	21/03/2017	Reunião Inicial Projeto
		Mapear Funcionalidades
		Documentação Capitulo 1
		Finalização Sprint 1

A primeira *Sprint* foi destinada ao entendimento do que seria desenvolvido como Trabalho de Conclusão de Curso (TCC). Dentro do mercado de *softwares* normalmente o cliente apresenta uma ideia inicial de produto, já nesse caso, aplicando a realidade para o ambiente acadêmico, iniciamos com uma conversa e definições do conteúdo deste trabalho. Como resultado obtivemos a Visão de Produto e o *Backlog* Inicial.

A entrega dessa *Sprint*, inicialmente prevista para 22/03/2017, se atrasou devido a inclusão do Documento de TCC como *feature*, porém após analisado em conjunto durante o *Sprint Review*, decidiu-se por não incluí-lo e desenvolvê-lo paralelamente.

Tabela 4 - *Sprint 2*

Sprint 2		
Data Inicio	Data Entrega	Cards Relacionados
21/03/2017	04/04/2017	Documentação Capitulo 1
		Iniciar Projeto
		Cadastro de Usuários
		Gerar Scaffold do Usuário
		Cadastro de Foto de Usuário com Gravatar*
		Layout Front-end do Usuário*

Com a Visão de Produto definida, pôde-se dar início oficialmente o desenvolvimento do *software*. Os itens previamente definidos do *Backlog* foram reavaliados, priorizados e detalhados nas atividades a serem executadas. Toda a parte inicial como configuração de ambiente de desenvolvimento, configuração das ferramentas definidas para a utilização do desenvolvimento e o início da aplicação foram executadas na *Sprint 2*.

Devido a problemática envolvida juntamente com a curva de aprendizagem da desenvolvedora, os itens "Cadastro de Foto de Usuário com Gravatar" e "Layout Front-end do Usuário" foram retirados dessa *Sprint* para que a entrega e o incremento fossem feitos na data estipulada. A decisão em remover os itens foi avaliada por toda equipe em reunião feita remotamente. Como resultado obtivemos o ambiente de desenvolvimento configurado e o primeiro CRUD³ do sistema funcionando.

Tabela 5 - *Sprint 3*

Sprint 3		
Data Inicio	Data Entrega	Cards Relacionados
04/04/2017	18/04/2017	Cadastro de Foto de Usuário com Gravatar
		Layout Front-end do Usuário
		Configuração do Servidor Deploy
		Relacionamentos Usuários
		Finalização Sprint 2
		Issues Sprint 2
		Cadastro de Ingredientes
		Scaffold de Ingredientes
		Fazer a relação de Ingredientes x Usuário
		Model de Ingredientes Gerais
		Layout tela de Ingredientes

A *Sprint 3* recebeu os itens retirados da *Sprint* anterior e novos itens priorizados. Dentro do *Scrum* o aprendizado da equipe com relação a sua capacidade de execução é essencial no momento de montar os itens da *Sprint*, e isso se dá ao longo do desenvolvimento. Nesse momento o conceito de "Ingredientes Gerais" surgiu em discussão para melhorar a usabilidade do produto, no qual o usuário não iria precisar cadastrar os ingredientes mais comuns (ex.: farinha, açúcar, água), eles já viriam pré-cadastrados nessa tabela geral.

³ CRUD - acrônimo (Create Read Update Delete) para as quatro operações básicas utilizadas em base de dados.

Como resultado obtivemos os itens da *Sprint* anterior concluídos e o cadastro de Ingredientes finalizados. Nessa *Sprint* o cliente já pôde iniciar os cadastros viabilizados.

Tabela 6 - *Sprint* 4

Sprint 4		
Data Inicio	Data Entrega	Cards Relacionados
18/04/2017	02/05/2017	Autenticação \ Permissões do Usuário
		Scaffold de Categoria
		Layout de Categoria
		Fazer relação de Categoria x Usuário
		Model de Categorias Gerais

Na *Sprint* 4 foi priorizada pelo cliente a autenticação do usuário bem como as permissões de acesso. Para isso foi feito uma análise conjunta dos itens que seriam públicos e os itens privados. Foi incluído também para essa *Sprint* o cadastro de Categorias.

Como resultado obtivemos a autenticação, bloqueando os acessos privados e o cadastro de categorias juntamente com o conceito de "Categoria Geral", implementado para Ingredientes anteriormente.

Tabela 7 - *Sprint* 5

Sprint 5		
Data Inicio	Data Entrega	Cards Relacionados
02/05/2017	16/05/2017	Autenticação / Redirecionamentos
		Cadastro de Unidades
		Scaffold Unidades
		Conversão de Unidades
		Relação de Unidades x Ingredientes x Receitas

Após os testes da *Sprint* anterior a equipe notou a falta de alguns itens não mapeados previamente. Então, para a *Sprint 5*, foram mapeados os redirecionamentos relacionados a autenticação. Desde a definição do tema tinha-se mapeado o tratamento de Unidades e suas conversões e foram incluídos nessa *Sprint*.

Devido ao fato da única desenvolvedora ser responsável também pela execução do Documento de TCC, houveram dificuldades e essa *Sprint* não foi entregue na data prevista. Dentro do *Scrum* prioriza-se a comunicação entre a equipe para que as falhas sejam notadas o quanto antes e que possam ser contornadas.

Nesse caso a desenvolvedora comunicou a equipe com relação as suas dificuldades e foi definido em conjunto adiar a data para que a *Sprint* fosse entregue em sua totalidade. Como resultado obtivemos os redirecionamentos e a conversão das unidades.

Tabela 8 - *Sprint 6*

Sprint 6		
Data Inicio	Data Entrega	Cards Relacionados
30/05/2017	13/06/2017	Scaffold de Receitas
		Layout de Receitas
		Relações Receitas

Finalmente durante a *Sprint 6* foi possível juntar todo o trabalho para o objetivo principal do sistema, o cadastro de receitas. Como resultado obtivemos o cadastro de receitas. Nesta *Sprint* o cliente já pôde utilizar o sistema em sua totalidade, desde a autenticação, o cadastro de ingredientes, categorias, bem como a receita em si.

Tabela 9 - *Sprint 7*

Sprint 7		
Data Inicio	Data Entrega	Cards Relacionados
13/06/2017	20/06/2017	Relações Receitas
		Issues Sprint 7
		Home da Aplicação

		Receitas por Usuário
		Testes Finais

Após os testes da *Sprint* anterior, notou-se algumas relações previamente não mapeadas. Dessa forma, para a *Sprint 7* foram mapeados os itens relacionais da receita, bem como os novos itens. Também foi incluído um item de correção, relacionado a um erro ocorrido em produção (detalhe encontrado pelo cliente durante o uso do sistema). Por se tratar da última *Sprint* foi mapeado um item para testes finais de qualidade, garantindo a entrega efetiva e o encerramento do desenvolvimento do sistema.

Em todas as *Sprints* as entregas eram feitas para o responsável pelos testes e para o cliente, para isso utilizou-se a ferramenta *Heroku* para viabilizar a hospedagem da aplicação. Foram criados dois ambientes, um para testes chamado *development* e outro para produção chamado *master*. Todo item concluído e testado foi integrado na aplicação *master* e disponibilizado para o cliente utilizar.

3.3. Modelagem do Software

Por se tratar de um desenvolvimento ágil de software, a modelagem do sistema se deu de forma incremental. Nesse item estão descritos os artefatos de modelagem demonstrando a estrutura final do software.

3.3.1. Requisitos Funcionais e Não Funcionais

Durante o desenvolvimento deste projeto foram identificados requisitos funcionais e não funcionais. Requisitos funcionais são as funcionalidades que o sistema necessita desempenhar e requisitos não-funcionais são relacionados às restrições, validações e condições realizadas sob os requisitos funcionais. Os requisitos estão listados no Apêndice C.

3.3.2. Diagrama de Caso de Uso

O Diagrama de Caso de Uso (DCU) é composto por atores e seus casos de uso. A definição de ator se dá por uma entidade a qual interage com o sistema e essas ações ou funcionalidades desempenhadas pelo sistema, são chamadas de casos de

uso. Os Diagramas de Caso de Uso citados neste trabalho podem ser encontrados no Apêndice D.

3.3.3. Diagrama de Classes

O Diagrama de Classes é responsável pela visualização da relação e estrutura das classes que servem o modelo para objetos. O Diagrama de Classes pode ser encontrado no Apêndice E.

3.3.4. Diagrama de Sequência

O Diagrama de Sequência é utilizado para demonstrar a sequência dos processos dentro de um software, de forma simples e lógica. No Diagrama de Sequência é descrito a maneira com que os objetos colaboram ao longo de um período de tempo. Os Diagramas de Sequência podem ser encontrados no Apêndice F.

3.3.5. Diagrama Banco de Dados

Um diagrama de banco de dados são detalhados os componentes da estrutura física como por exemplo tabelas, campos, tipos de valores e índice. O diagrama físico de banco de dados referente a esse trabalho encontra-se no Apêndice G.

3.4. Tecnologias, Ferramentas e Linguagens utilizadas

- Interfaces do Sistema: para o desenvolvimento das interfaces utilizou-se os padrões *Google Material* com o auxílio do *framework Materialize*.
- Tecnologias:
 - Ruby on Rails: Framework desenvolvido no padrão MVC (Model-View-Controller);
 - PostgreSQL: Sistema gerenciador de banco de dados;
 - Materialize: Framework front-end responsivo baseado em Material Design;
- Ferramentas e Linguagens: os recursos utilizados para o desenvolvimento deste sistema, bem como recursos para análise e documentação encontram-se listados abaixo.

- *Ruby*: Linguagem de programação interpretada, multiparadigma e de tipagem forte e dinâmica;
- *GitHub*: Plataforma online para hospedagem e versionamento de código;
- *Heroku*: Plataforma online (PaaS) que provê infraestrutura de máquinas virtuais para hospedagem de aplicações web.
- *Atom*: Editor de texto apropriado para codificação. Disponibiliza *snippets* para auto marcação das linguagens de programação;
- *Google Docs*: Editor de texto utilizado na documentação do projeto;
- *Trello*: Plataforma *online* para gerenciamento de atividades;
- *PSequel*: Interface gráfica para PostgreSQL;

4. APRESENTAÇÃO DO SISTEMA

Neste capítulo será apresentado os resultados conquistados durante o desenvolvimento do projeto bem como a apresentação das telas do sistema desenvolvido.

4.1. O Sistema "Leaf - Caderno de Receitas"

Ao acessar o sistema através da internet a primeira tela apresentará o *feed* de receitas como mostra a Figura 4. Essa tela é pública, ou seja, não há necessidade de permissões especiais ou autenticação do usuário para acessá-la.

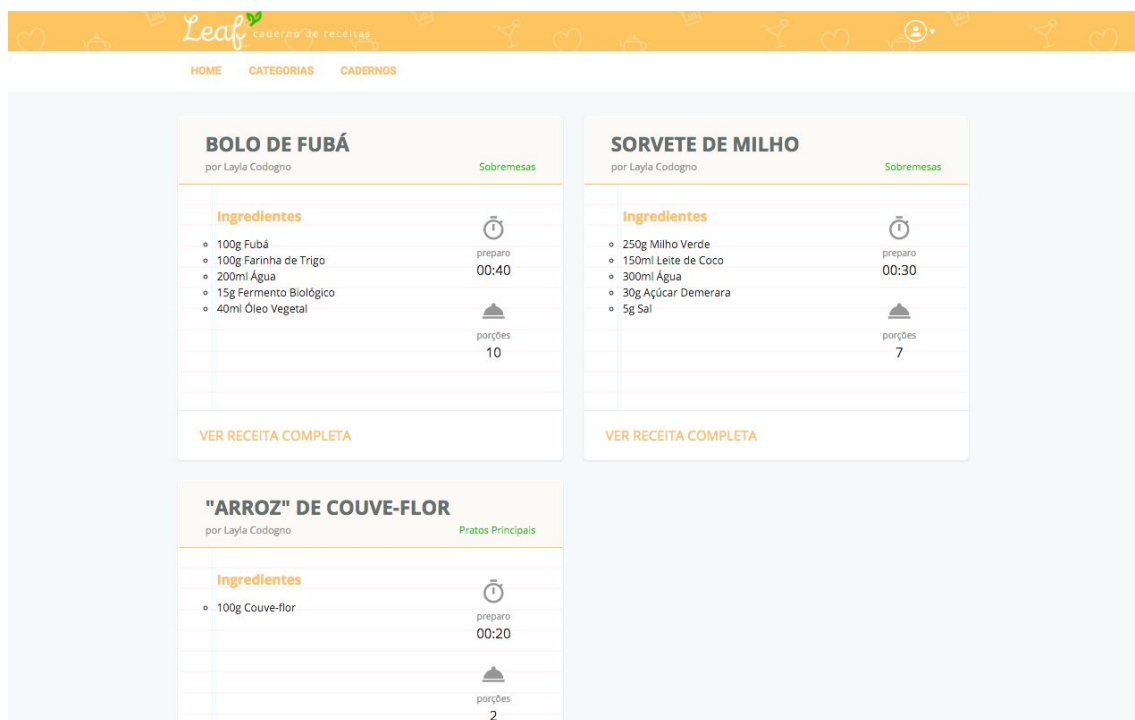


Figura 4 - Tela Inicial do Sistema / Feed de Receitas

Ainda de forma pública, é possível acessar a lista de receitas pesquisando através de categorias como mostra a Figura 5, para isso o usuário deve digitar a categoria desejada ou parte do nome da categoria e clicar em buscar.

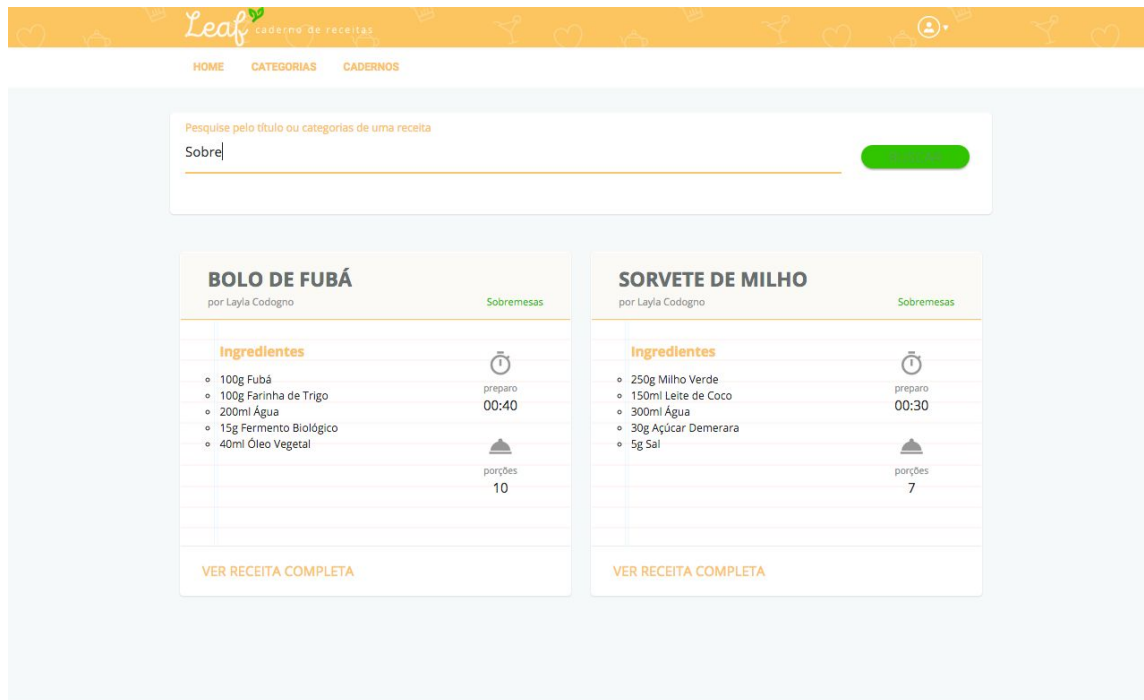


Figura 5 - Tela de Filtro de Receitas por Categorias

O usuário poderá também da mesma forma, pesquisar os cadernos de receitas da aplicação, como mostra a Figura 6. Ao clicar no caderno será redirecionado para a tela de perfil do usuário (Figura 13).

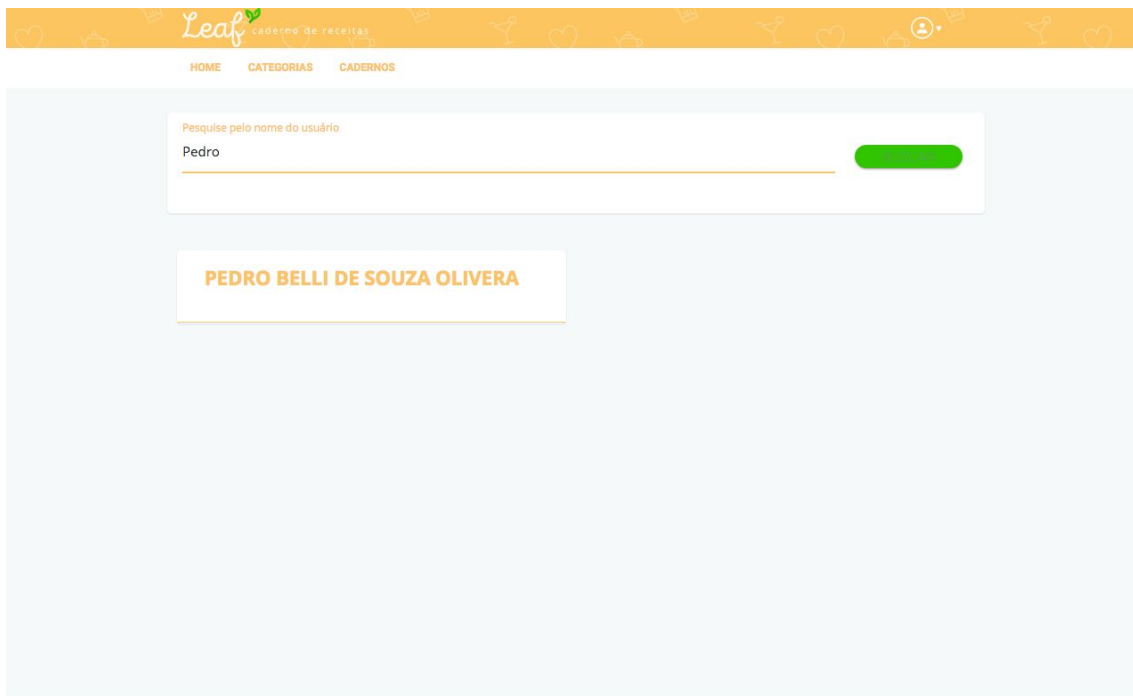


Figura 6 - Tela de Filtro de Cadernos por Nome de Usuário

Dentro do sistema o usuário poderá realizar seu login para acessar e gerenciar suas próprias receitas. Para isso basta clicar no menu superior e será redirecionado para a tela de Login como mostra a Figura 7.

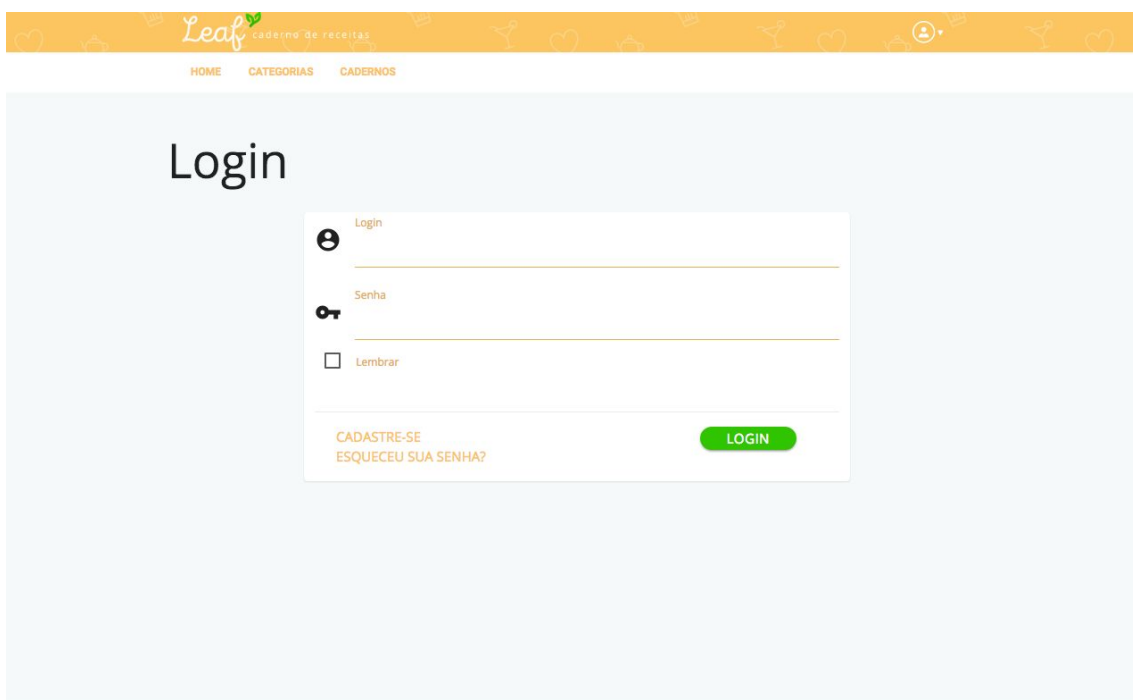
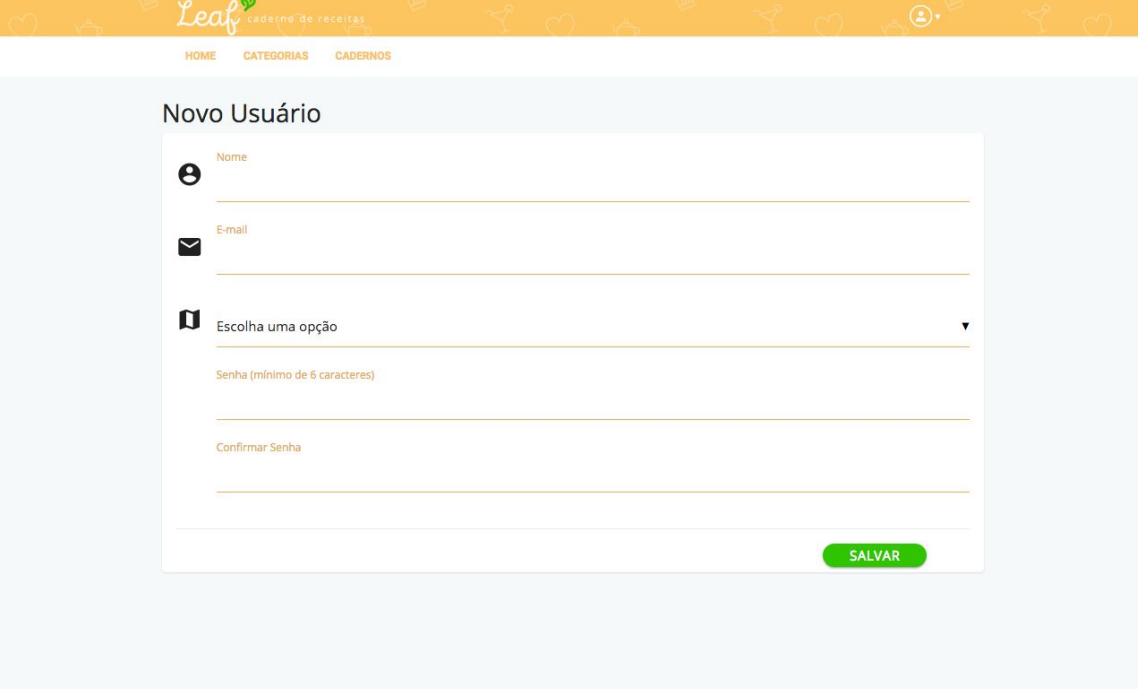


Figura 7 - Tela de Login

Caso o usuário ainda não tenha um cadastro, ele poderá realizar seu cadastro como mostra a Figura 8.



The image shows a web browser window displaying the registration page for 'Leaf - Caderno de receitas'. The page has a light blue background and a white registration form. The form is titled 'Novo Usuário' and contains the following fields: 'Nome' (with a person icon), 'E-mail' (with an envelope icon), 'Escolha uma opção' (with a dropdown arrow), 'Senha (mínimo de 6 caracteres)', and 'Confirmar Senha'. A green 'SALVAR' button is located at the bottom right of the form. The website's navigation bar at the top includes 'HOME', 'CATEGORIAS', and 'CADERNOS'.

Figura 8 - Tela de Novo Usuário / Cadastro

Feito o cadastro ou autenticado o usuário terá acesso às telas privadas para o gerenciamento de seu perfil. Dentro do sistema é possível listar e cadastrar Categorias (Figura 9), Ingredientes (Figura 10) e Receitas (Figura 11 e Figura 12).

A receita pode ser visualizada após cadastro como mostra Figura 13.

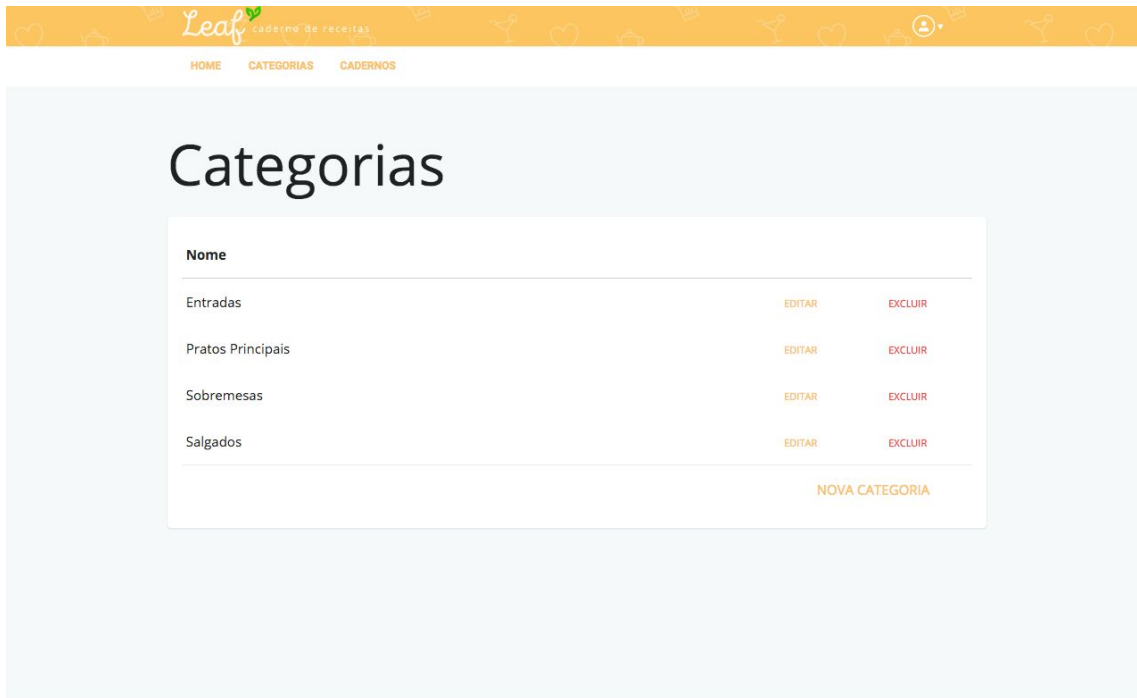


Figura 9 - Tela de Listagem de Categorias

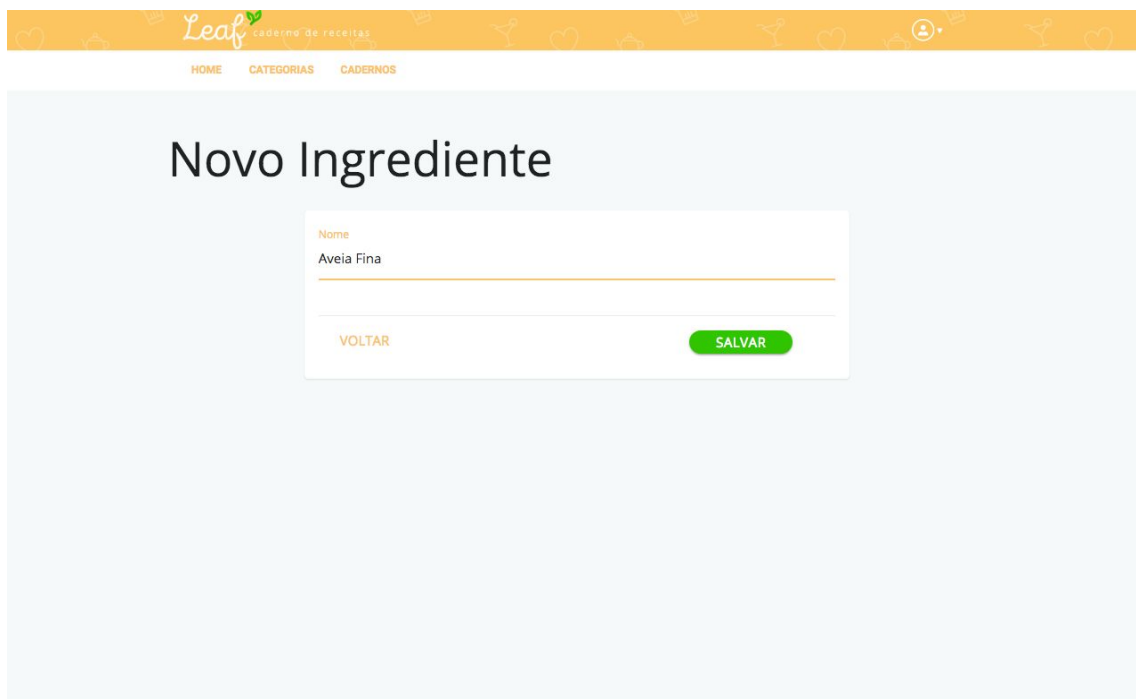


Figura 10 - Tela de Cadastro de Ingredientes

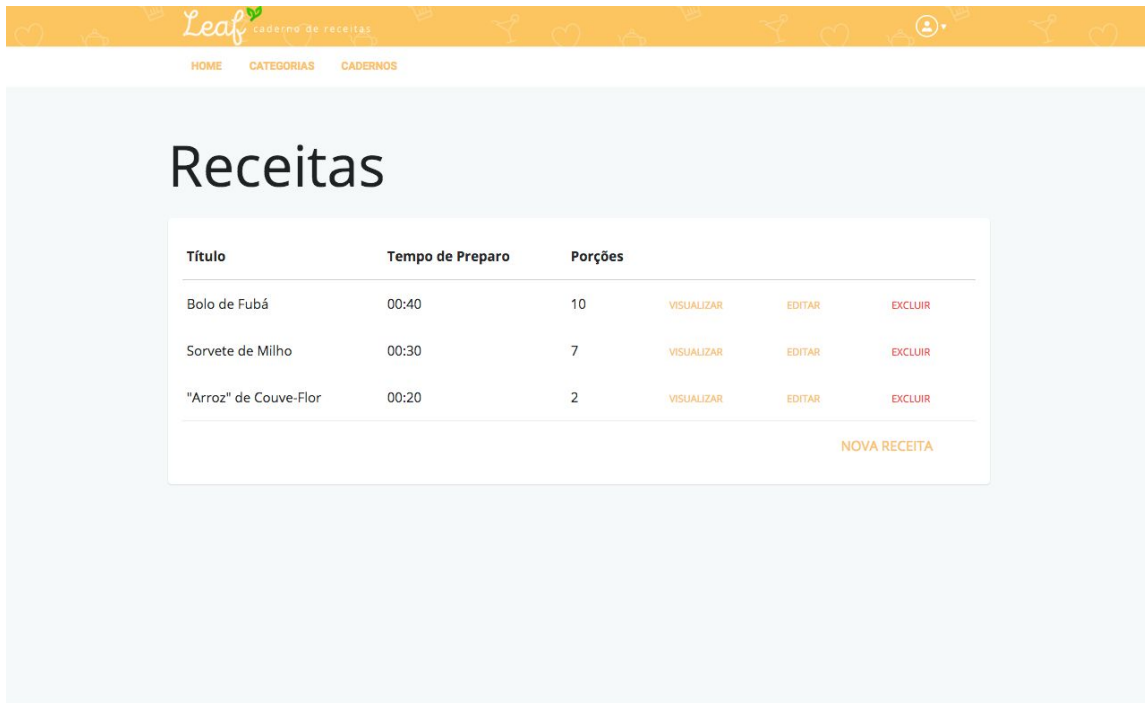


Figura 11 - Tela de Listagem de Receitas

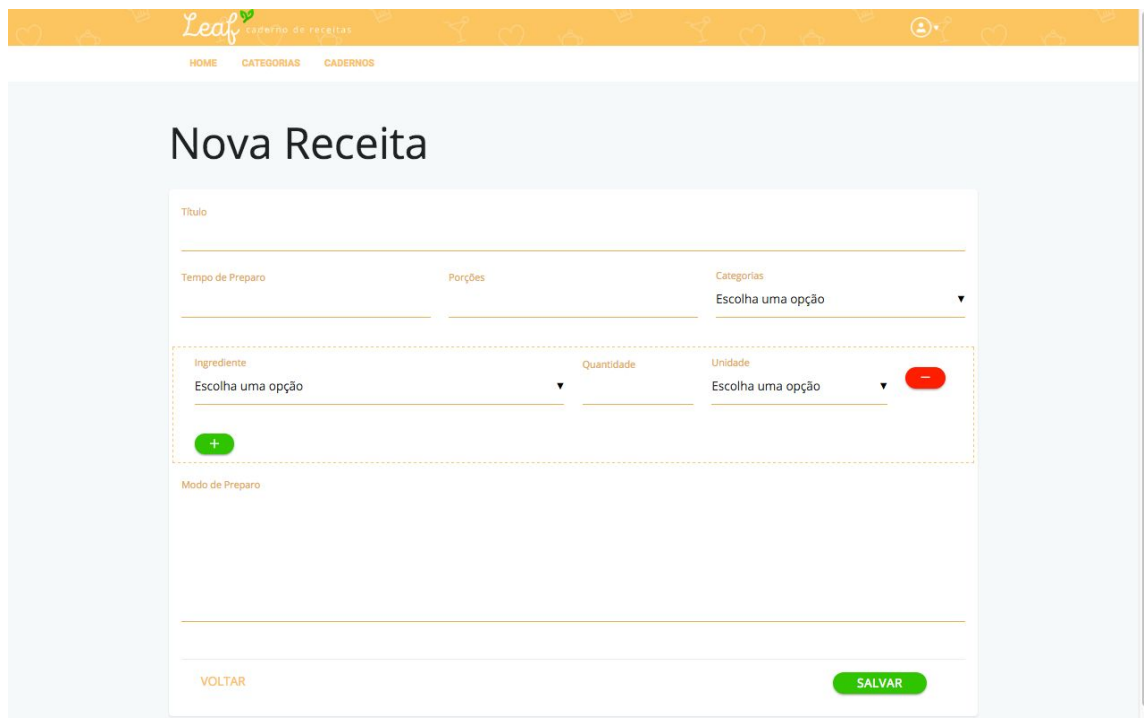


Figura 12 - Tela de Cadastro de Receita



Figura 13 - Tela de Visualização de Receita

A tela de Caderno do Usuário lista todas as receitas para aquele usuário. Para editar suas informações o usuário deve clicar em editar e será redirecionado para a tela de edição de perfil como mostra a Figura 15.

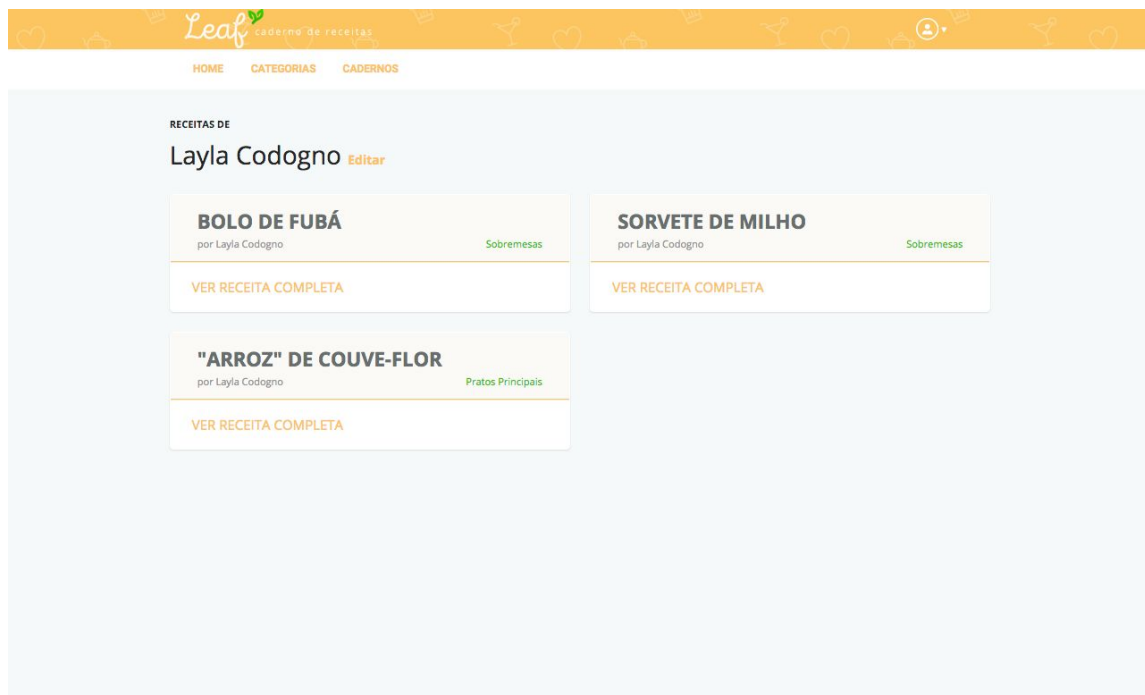


Figura 14 - Tela de Caderno do Usuário

Leaf caderno de receitas

HOME CATEGORIAS CADERNOS

Editar Usuário

Nome
Layla Codogno

E-mail
layla.codogno@

PR

VOLTAR SALVAR

Figura 15 - Tela de Edição de Perfil Usuário

5. CONSIDERAÇÕES FINAIS

A problemática em se aplicar apenas metodologias tradicionais no ambiente acadêmico, ou a não compreensão das metodologias ágeis de desenvolvimento, gera uma inconsistência e possivelmente, até, uma defasagem dentro de uma formação tecnológica que tenha como foco a formação de profissionais para o mercado de trabalho.

O planejamento do sistema se deu de forma flexível de maneira que possibilitou certa liberdade diante a necessidade de mudanças. Dentro de uma metodologia tradicional, mudanças ao longo do desenvolvimento poderiam sofrer grandes resistências, devido ao planejamento rígido construído.

Dentro da abordagem tradicional, o planejamento do desenvolvimento é executado em sua totalidade antes do início do desenvolvimento, utilizando métricas e metodologias nem sempre assertivas. A comunicação durante o desenvolvimento do projeto deu-se de forma fluida e sincera, a cada iteração o aprendizado sobre a capacidade de execução e sobre o projeto eram reavaliados, dessa forma tornando mais assertivo a definição de metas (no caso *sprints*). Durante o projeto desenvolvido houverem erros em prazos e medidas, porém todos foram resolvidos sem causar grandes problemas no processo.

As entregas contínuas foram consideradas o maior ganho desse projeto devido a rápida validação dos itens, pois foi possível analisar com rapidez itens que não se encaixaram conforme previsto, bem como a inclusão de novos itens para atender a demanda. Em metodologias tradicionais esses itens seriam identificados apenas ao final do projeto, podendo inviabilizar a utilização do *software*.

O projeto por sua vez alcançou os objetivos tirando proveito de vários pontos positivos do método ágil escolhido.

Com o presente trabalho, é possível demonstrar à comunidade acadêmica o potencial das metodologias ágeis no desenvolvimento de software como Trabalho de Conclusão de Curso. Os resultados alcançados com esse projeto exibem, também, a eficiência das metodologias ágeis, especificamente o *Scrum*, comprovando a maior agilidade e flexibilidade no processo, evitando procedimentos complexos e, por vezes burocráticos instituídos em metodologias mais rígidas.

O sistema "*Leaf* - caderno de receitas" contribuiu para que os membros da equipe compreendessem melhor a gestão de desenvolvimento de *software* utilizando o *framework Scrum*, bem como melhorassem seus conhecimentos em Ruby on Rails e Implementação de Interfaces utilizando Materialize.

Para trabalhos futuros será desenvolvido os itens que constituem o cadastro de cardápios e a formulação automatizada de uma lista de compras, somatizando os ingredientes de cada receita inserida no cardápio em questão. Também serão incluídos refinamentos nos processos já estabelecidos no sistema buscando melhor desempenho.

6. REFERÊNCIAS

- Cohn, Mike. Desenvolvimento de Software com Scrum: Aplicando métodos ágeis com sucesso, Bookman, Porto Alegre, 2011.
- Pressman, Roger S. Engenharia de Software: Uma abordagem profissional, Bookman, Porto Alegre, 2011;
- Sommerville, Ian. Engenharia de Software, Person, São Paulo, 2010.
- Sutherland, Jeff. Scrum: a arte de fazer o dobro do trabalho na metade do tempo, LeYa, 2014
- Sabbagh, Rafael. Scrum: Gestão Ágil para Projetos de Sucesso, Casa do Código, São Paulo, 2013
- The Standish Group, 2015 *CHAOS Report*, <<https://www.standishgroup.com/store/services/chaos-report-2015-blue-pm2go-membership.html>>, acessado em março de 2017.
- Manifesto Ágil, <<http://manifestoagil.com.br/>>, acessado em março de 2017.
- Desenvolvimento Ágil, <<http://www.desenvolvimentoagil.com.br/scrum/>>, acessado em maio de 2017
- Infoq, <<https://www.infoq.com/articles/standish-chaos-2015>>, acessado em abril de 2017
- MindMaster, <<http://www.mindmaster.com.br/scrum/>>, acessado junho de 2017

APÊNDICE A

QUESTIONÁRIO APLICADO NA PESQUISA

Questão	Tipo de Resposta	
Dados Iniciais		
Qual o âmbito de trabalho? (caso trabalhe nos dois meios, por gentileza escolha apenas um para responder este questionário)	Fechada	- Acadêmico - Profissional
Qual a sua idade?	Aberta	
Quando você começou a trabalhar com desenvolvimento de softwares?	Fechada	- <1980 - 1980 < 1990 - 1990 < 2000 - 2000 < 2010
Qual o tipo de metodologia você utiliza para o desenvolvimento?	Condicional	- Metodologias Tradicionais - Metodologias Ágeis
Metodologias Tradicionais		
Qual das Metodologias Tradicionais abaixo você utiliza/utilizou para o desenvolvimento de softwares?	Semi-Fechada	- Cascata - Prototipação - Espiral - RUP - Outras
Qual o principal motivo em desenvolver utilizando Metodologias Tradicionais?	Aberta	
Em sua opinião, quais os pontos positivos em se utilizar uma Metodologia Tradicional?	Aberta	
Você já utilizou alguma Metodologia Ágil para o desenvolvimento de software?	Condicional	- Sim - Não
Metodologias Tradicionais - Utilizou Metodologias Ágeis		
Em sua opinião, há alguma vantagem no uso da metodologia ágil? Descreva sobre	Aberta	
Metodologias Tradicionais - Não Utilizou Metodologias Ágeis		
Em sua opinião, o que te impede de utilizar alguma metodologia ágil?	Aberta	
Metodologias Ágeis		
Qual das metodologias ágeis abaixo você utiliza/utilizou para o desenvolvimento de softwares?	Semi-Fechada	- Scrum - Xp - Kanban - Outras
Qual o principal motivo em desenvolver utilizando metodologias ágeis?	Aberta	
Em sua opinião, quais os pontos positivos em se utilizar uma metodologia ágeis?	Aberta	
Você já utilizou alguma Metodologia Tradicional para o desenvolvimento de software?	Condicional	- Sim - Não
Metodologias Ágeis - Utilizou Metodologias Tradicionais		

Em sua opinião, há alguma vantagem no uso do método tradicional? Descreva sobre.	Aberta	
Em sua opinião, o que te impede de utilizar metodologia tradicional?	Aberta	

APÊNDICE B

VISÃO DE PRODUTO

"O produto a ser desenvolvido será um Caderno de Receitas online. Para tanto, será necessário realizar os seguintes itens:

- Cadastro de Usuários

Os usuários do sistema devem poder se cadastrar para ter acesso a gestão de suas receitas e cardápios. Apenas usuários logados poderão ter acesso a essas funcionalidades.
- Cadastro de Ingredientes

Para o cadastro deve-se ter um CRUD completo. O ingrediente é composto apenas por nome.
- Cadastro de Categorias

Para o cadastro deve-se ter um CRUD completo. A categoria é composta apenas por nome.
- Cadastro de Unidades

As unidades serão fixas, sendo unidade, grama, quilograma, mililitro e litro. Deverá conter uma função de conversão entre as unidades grama e quilograma, mililitro e litro para facilitar a montagem da lista de compras.
- Cadastro de Receitas

O cadastro de receitas será composto por título, tempo de preparo, porções, categorias, ingredientes e modo de preparo. Não será possível cadastrar uma receita sem categoria ou ingrediente.
- Feed de Receitas

O *feed* de Receitas será a página inicial do sistema e será aberta ao público (sem necessidade de autenticação). As receitas serão apresentadas por ordem de cadastro.
- Cadastro de Cardápio

O cardápio será cadastrado a partir das receitas escolhidas pelo usuário.

- Lista de Compras

A lista de compras será gerada a partir de um cardápio. O usuário deverá selecionar o cardápio que deseja e então o sistema irá somatizar todos os ingredientes gerando uma lista de compras.

O sistema deverá conter uma interface padronizada de fácil navegação."

APÊNDICE C

REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS

❖ Requisitos Funcionais

RF01 - Realizar Cadastro

Os usuários do sistema deverão poder se cadastrar.

RF02 - Gestão de Ingredientes

Os usuários do sistema deverão poder gerir os ingredientes.

RF03 - Gestão de Categorias

Os usuários do sistema deverão poder gerir as categorias.

RF04 - Gestão de Receitas

Os usuários do sistema deverão poder gerir as receitas.

RF05 - Listar Receitas de Todos os usuários

Os usuários deverão poder listar todas as receitas cadastradas.

RF06 - Conversão de Unidades

O sistema deverá conter uma função de conversão entre as unidades grama e quilograma, e entre mililitro e litro.

❖ Requisitos Não-Funcionais

RNF01 - Ergonomia de Interfaces

O sistema deverá possuir interface padronizada e amigável.

APÊNDICE D

DIAGRAMAS DE CASO DE USO

A Figura 16 demonstra o Diagrama de Caso de Uso Simplificado e suas respectivas funcionalidades. Abaixo temos a documentação dos casos de uso detalhados.

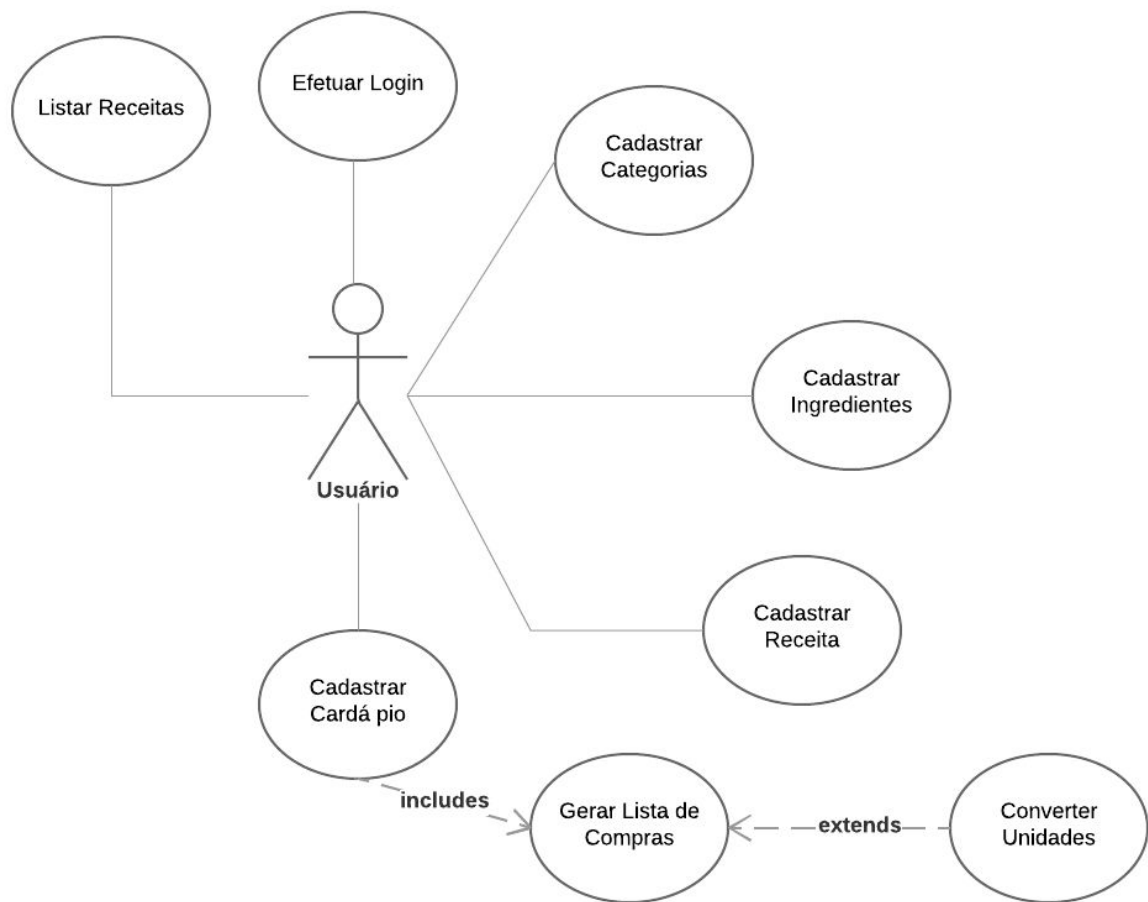


Figura 16 - Diagrama de Caso de Uso Simplificado

Tabela 11 - Caso de Uso UC001

Nome de Caso de Uso	UC001 - Efetuar Login
Ator Principal	Usuário
Atores Secundários	Não se Aplica
Resumo	Esse caso de uso descreve as etapas durante o processo de autenticação do usuário.

Pré-Condições	Usuário deve estar cadastrado no Banco de Dados.
Pós-Condições	Usuário deverá acessar sua conta.
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Clicar no botão Login	
	2. Exibir a Página de Login
3. Preencher os dados solicitados na tela	
	4. Validar os dados Inseridos
	5. Autenticar o Usuário
	6. Redirecionar a tela Inicial

Tabela 12 - Caso de Uso UC002

Nome de Caso de Uso	UC002 - Cadastrar Ingredientes
Ator Principal	Usuário
Atores Secundários	Não se Aplica
Resumo	Esse caso de uso descreve as etapas durante o processo de cadastro de ingredientes.
Pré-Condições	Usuário deve estar logado em sua conta.
Pós-Condições	O novo ingrediente deve ser cadastrado no sistema
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Clicar no botão Novo Ingrediente	
	2. Exibir a Página de Cadastro de Ingrediente
3. Preencher os dados solicitados na tela	
	4. Validar os dados Inseridos
5. Clicar em Salvar	
	6. Salvar o novo Ingrediente
	7. Redirecionar para Listagem de Ingredientes

Tabela 13 - Caso de Uso UC003

Nome de Caso de Uso	UC003 - Cadastrar Categorias
----------------------------	------------------------------

Ator Principal	Usuário
Atores Secundários	Não se Aplica
Resumo	Esse caso de uso descreve as etapas durante o processo de cadastro de categorias.
Pré-Condições	Usuário deve estar logado em sua conta.
Pós-Condições	A nova categoria deve ser cadastrado no sistema
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Clicar no botão Nova Categoria	
	2. Exibir a Página de Cadastro de Categoria
3. Preencher os dados solicitados na tela	
	4. Validar os dados Inseridos
5. Clicar em Salvar	
	6. Salvar o nova Categoria
	7. Redirecionar para Listagem de Categorias

Tabela 14 - Caso de Uso UC004

Nome de Caso de Uso	UC004 - Cadastrar Receita
Ator Principal	Usuário
Atores Secundários	Não se Aplica
Resumo	Esse caso de uso descreve as etapas durante o processo de cadastro de receita.
Pré-Condições	Usuário deve estar logado em sua conta. Usuário deve ter cadastrado pelo menos uma categoria. Usuário deve ter cadastrado pelo menos um ingrediente.
Pós-Condições	A nova receita deve ser cadastrado no sistema
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Clicar no botão Nova Receita	
	2. Exibir a Página de Cadastro de

	Receita
3. Preencher os dados solicitados na tela	
	4. Validar os dados Inseridos
5. Clicar em Salvar	
	6. Salvar o nova Receita
	7. Redirecionar para Listagem de Receitas

Tabela 15 - Caso de Uso UC005

Nome de Caso de Uso	UC005 - Listar Receitas
Ator Principal	Sistema
Atores Secundários	Usuário
Resumo	Esse caso de uso descreve as etapas durante o processo de listagem de todas as receitas.
Pré-Condições	
Pós-Condições	Listagem de todas as receitas cadastradas no sistema por todos os usuário.
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Acessar a tela inicial da aplicação	
	2. Exibir a página de feed de Receitas

Tabela 16 - Caso de Uso UC006

Nome de Caso de Uso	UC006 - Cadastrar Cardápio
Ator Principal	Usuário
Atores Secundários	Não se Aplica
Resumo	Esse caso de uso descreve as etapas durante o processo de cadastro de cardápio.
Pré-Condições	Usuário deve estar logado em sua conta. Usuário deve ter cadastrado pelo menos uma receita.
Pós-Condições	O novo cardápio deve ser cadastrado no sistema
Fluxo Principal	
Ações do Ator	Ações do Sistema

1. Clicar no botão Novo Cardápio	
	2. Exibir a Página de Cadastro de Cardápio
3. Preencher os dados solicitados na tela	
	4. Validar os dados Inseridos
5. Clicar em Salvar	
	6. Salvar o nova Cardápio
	7. Chamar o Caso de Uso UC007
	8. Redirecionar para Listagem de Cardápios

Tabela 17 - Caso de Uso UC007

Nome de Caso de Uso	UC007 - Gerar Lista de Compras
Ator Principal	Usuário
Atores Secundários	Não se Aplica
Resumo	Esse caso de uso descreve as etapas durante o processo de gerar lista de compras.
Pré-Condições	Usuário deve estar logado em sua conta. Usuário deve ter cadastrado pelo menos um cardápio
Pós-Condições	A nova lista de compras será gerada
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Clicar no botão Gerar Lista	
	2. Apresentar Lista de Compras

APÊNDICE E

DIAGRAMA DE CLASSE

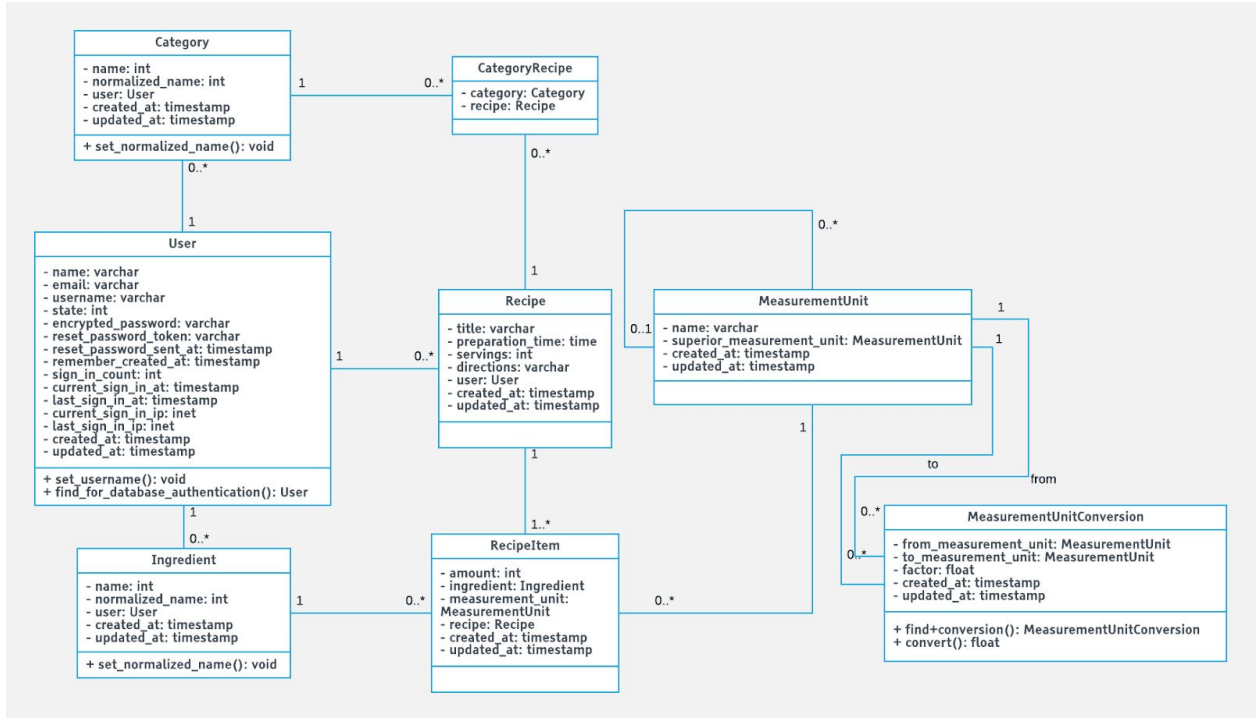


Figura 17 - Diagrama de Classe

APÊNDICE F

DIAGRAMA DE SEQUÊNCIA

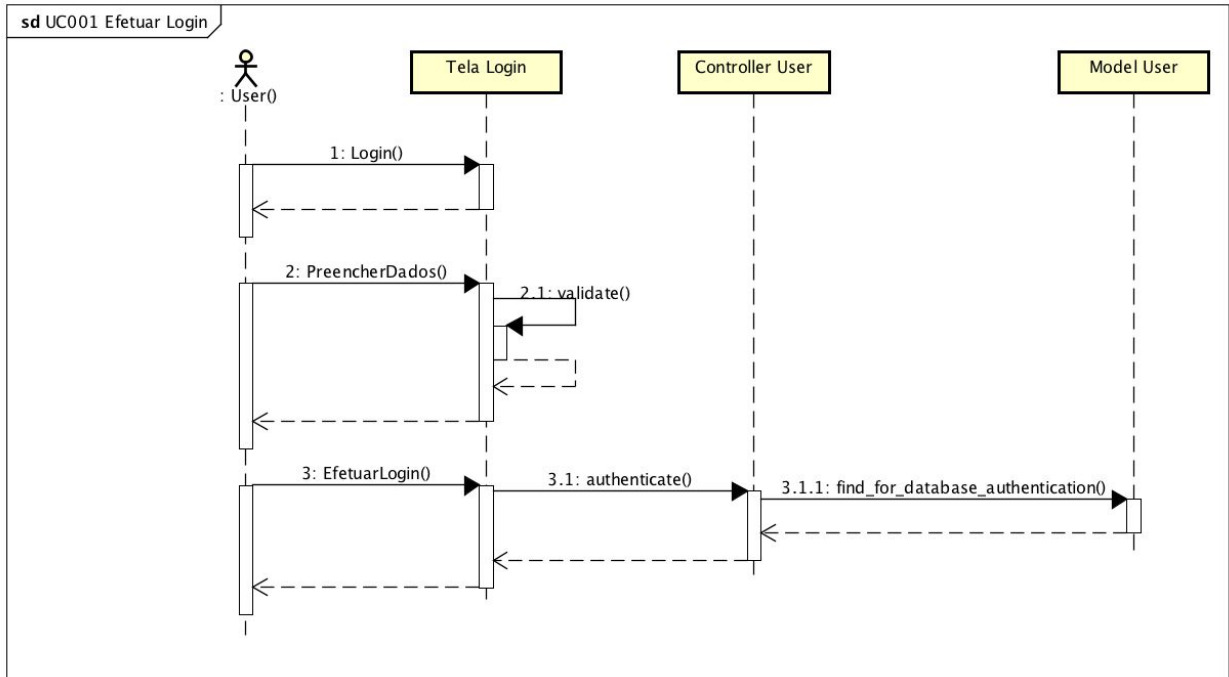


Figura 18 - Diagrama de Sequência UC001

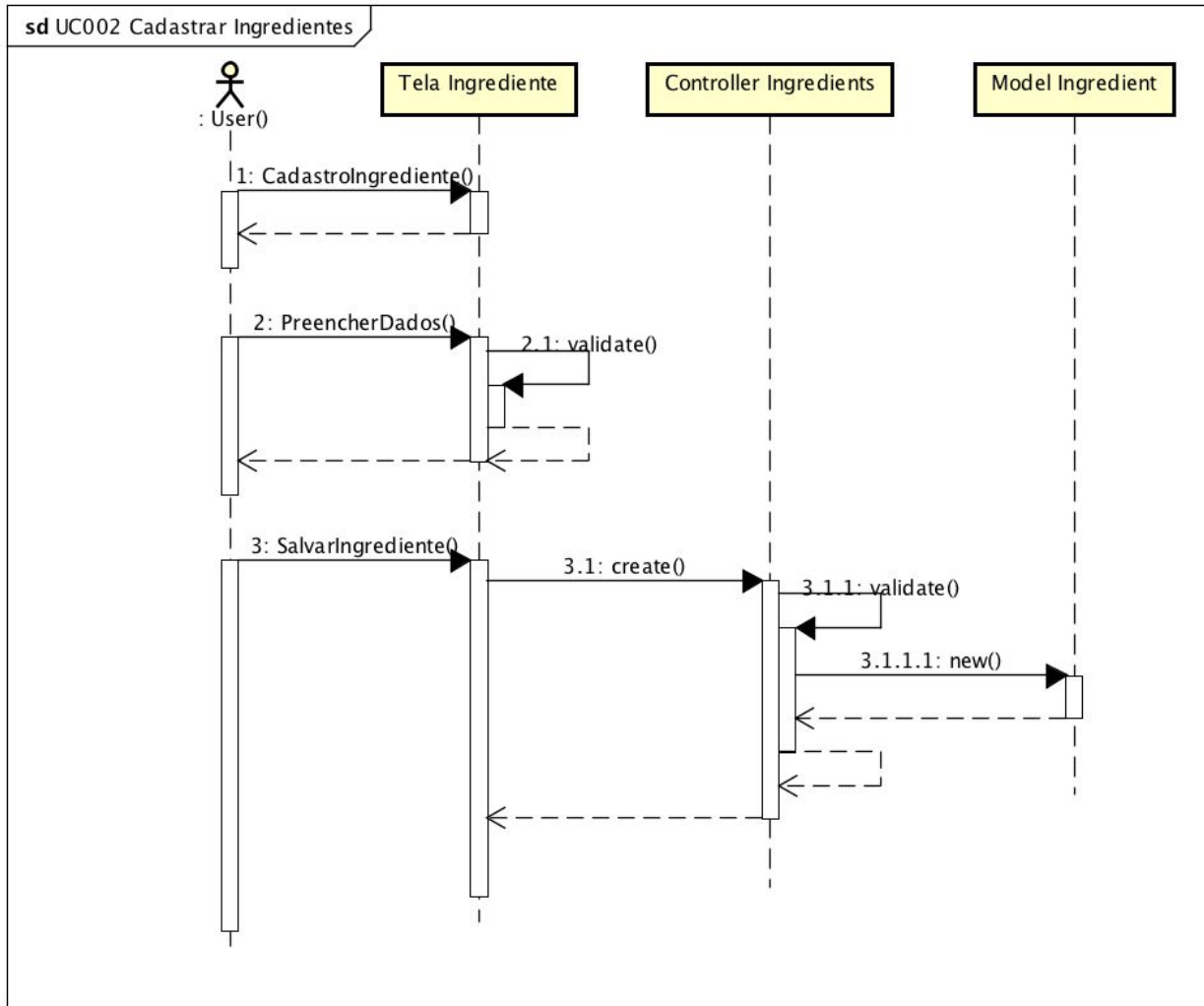


Figura 19 - Diagrama de Sequência UC002

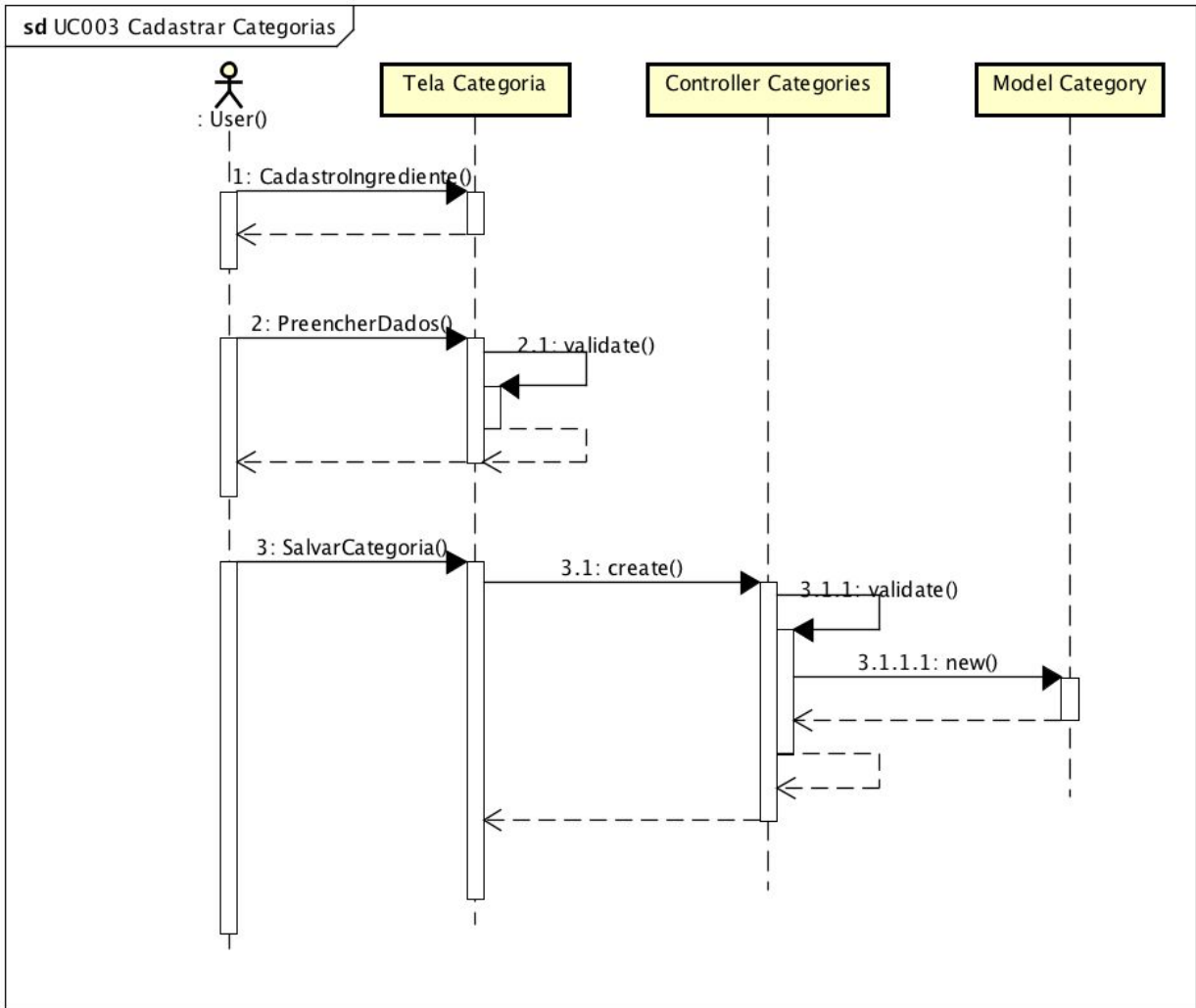


Figura 20 - Diagrama de Sequência UC003

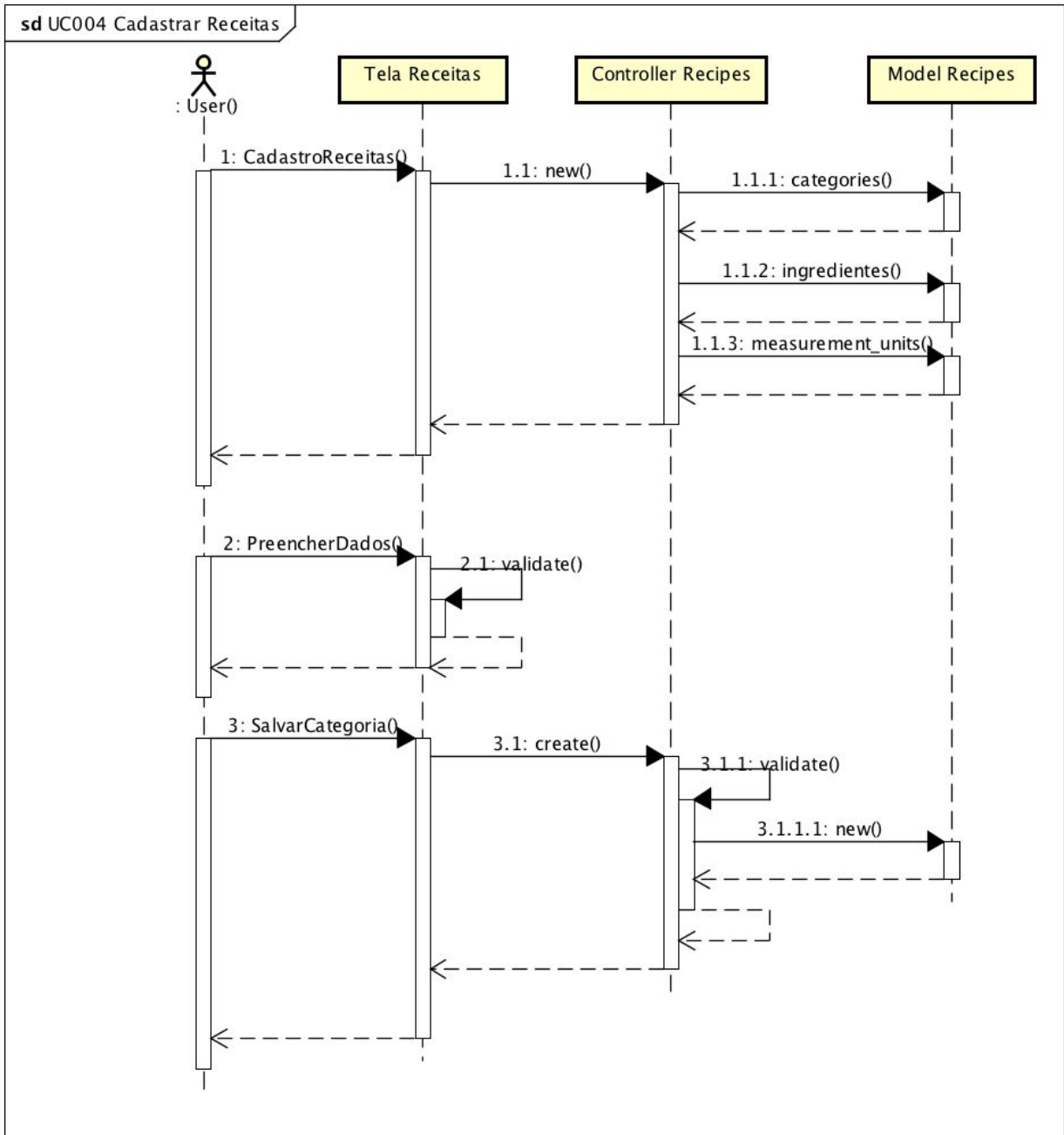


Figura 21 - Diagrama de Sequência UC004

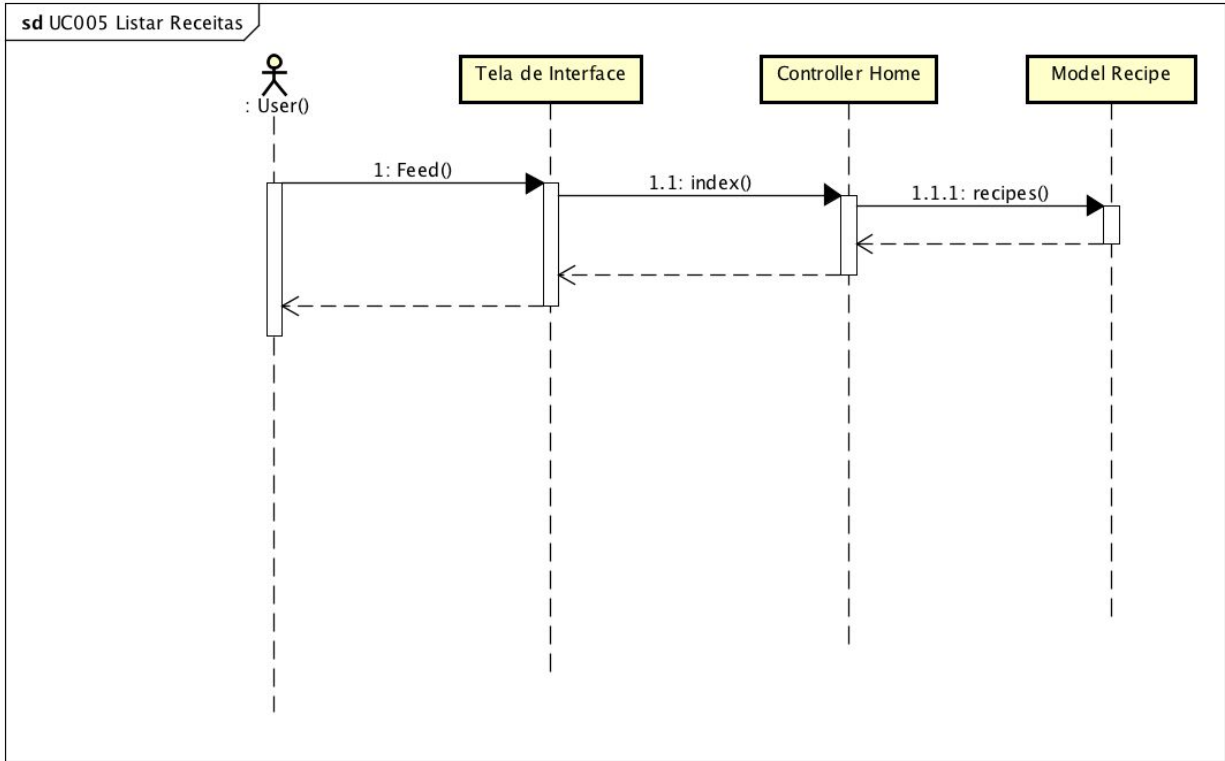


Figura 22 - Diagrama de Sequência UC005

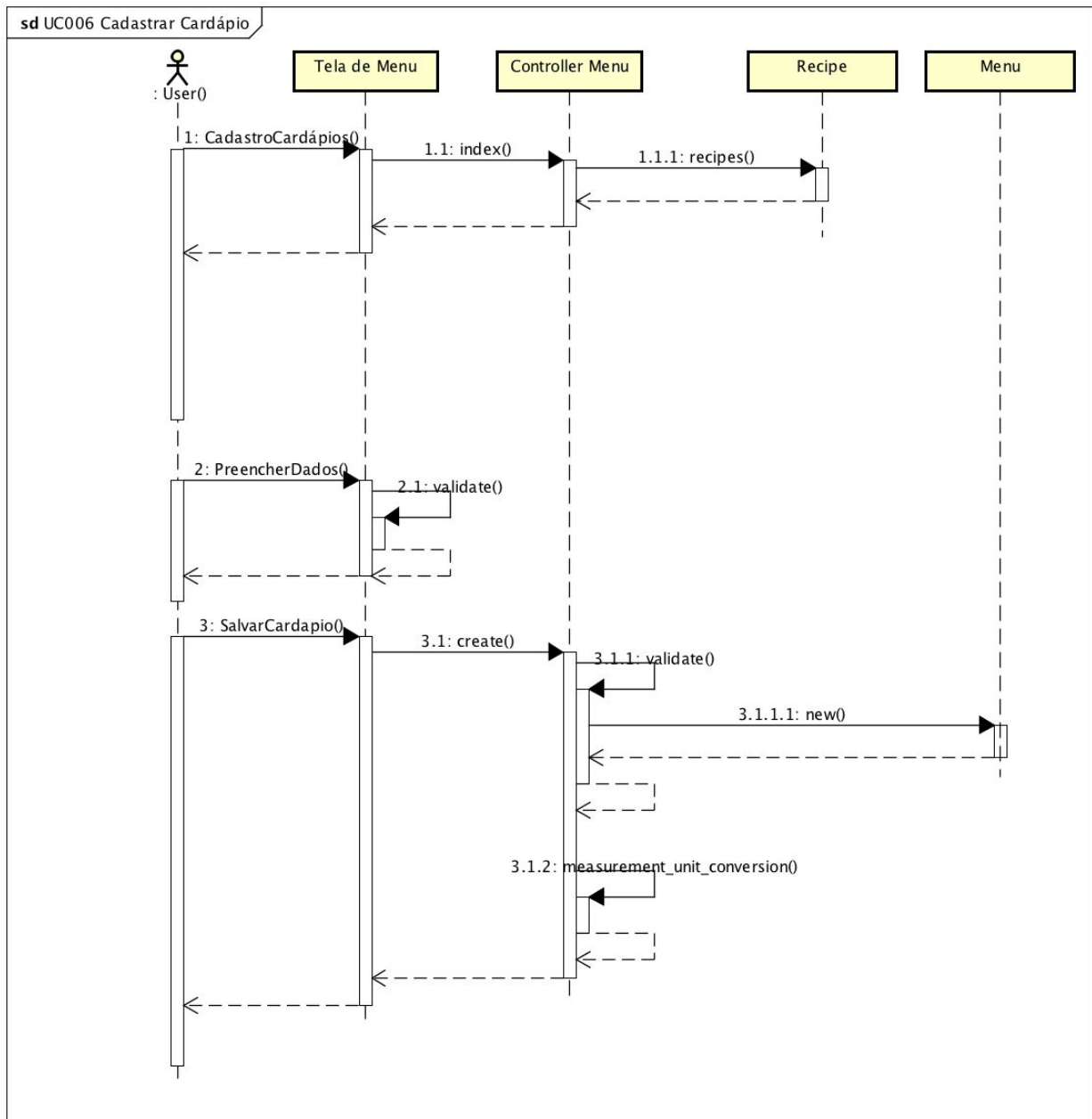


Figura 23 - Diagrama de Sequência UC006

APÊNDICE G

DIAGRAMA BANCO DE DADOS

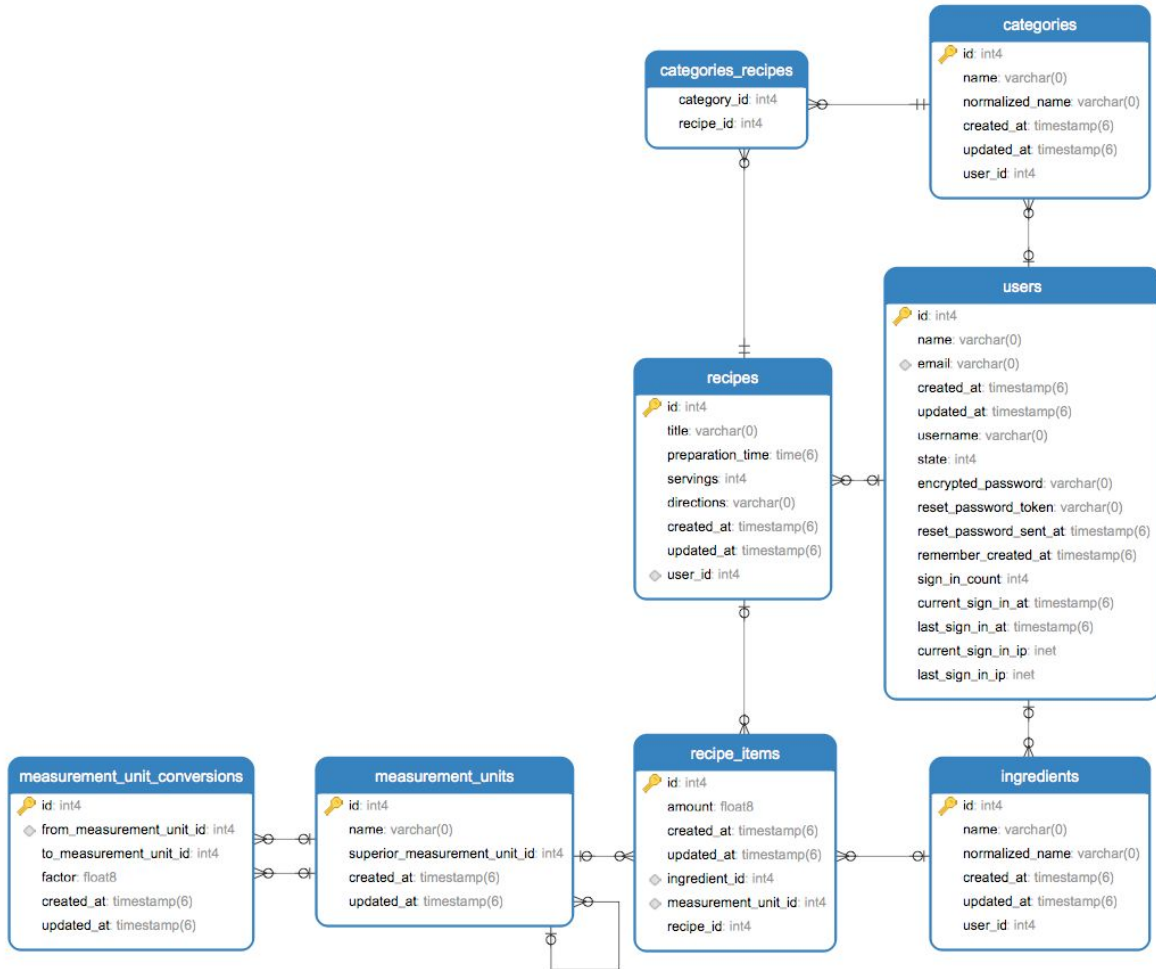


Figura 2 - Banco de Dados