

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COINT - TECNOLOGIA EM SISTEMAS PARA INTERNET
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

DONIZETY BALTOKOSKI

**SISTEMA WEB COLABORATIVO DE RECEITAS COM BUSCA
POR INGREDIENTES**

TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA
2018

DONIZETY BALTOKOSKI

SISTEMA WEB COLABORATIVO DE RECEITAS COM BUSCA POR INGREDIENTES

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Paulo André Filipak
Universidade Tecnológica Federal do Paraná

GUARAPUAVA
2018

The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; premature optimization is the root of all evil (or at least most of it) in programming. (KNUTH, Donald, 1974).

RESUMO

BALTOKOSKI, Donizety. Sistema web colaborativo de receitas com busca por ingredientes. 2018. 28 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2018.

O Brasil está entre os países que mais desperdiçam comida no mundo, perdendo aproximadamente 41 mil toneladas de alimento ao longo de toda cadeia alimentar que começa no plantio e termina no consumo final. Este trabalho propõe o desenvolvimento de um sistema *web* de receitas colaborativas onde todos podem compartilhar suas receitas favoritas, além de proporcionar um mecanismo de avaliação, área de comentários e busca de receitas possíveis de fazer a partir dos ingredientes informados. O sistema deve proporcionar ao consumidor a utilização de receitas a partir dos ingredientes que tem à disposição, evitando desperdício e contribuindo com o meio ambiente.

Palavras-chave: Receita. Sistema *Web*. Sustentabilidade. Desperdício de Alimentos.

ABSTRACT

BALTOKOSKI, Donizety. Collaborative recipe web system with a search for ingredients. 2018. 28 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2018.

Brazil is among the countries that waste most food in the world, losing approximately 41,000 tons of food throughout the food chain that begins at the planting and ends with the end customer. This paper proposes the development of a collaborative web system of recipes where everyone can share their favorite recipes, as well as providing an evaluation mechanism, comment area and search for recipes possible to cook from the informed ingredients. The system should provide consumers with recipes possibles to do from the ingredients they have available, avoiding waste and contributing to the environment.

Keywords: Recipe. Web System. Sustainability. Food Waste.

LISTA DE FIGURAS

Figura 1 – TudoGostoso	3
Figura 2 – Vovó Palmirinha	4
Figura 3 – Receitas CyberCook	4
Figura 4 – Receitas GSHOW	5
Figura 5 – Arquitetura Cliente-Servidor do SABRE	8
Figura 6 – Trello do projeto SABRE durante a Sprint 5	11
Figura 7 – Diagrama do Banco de Dados	12
Figura 8 – Página inicial	13
Figura 9 – Receitas do usuário	14
Figura 10 – Caixa de pesquisa	14
Figura 11 – Receita com pré-visualização de ingredientes	15
Figura 12 – Formulário de envio de receitas	16
Figura 13 – Receitas para aprovar (usuário administrativo)	17
Figura 14 – Editar receita para aprovar (usuário administrativo)	18
Figura 15 – Listagem de usuários (usuário administrativo)	19
Figura 16 – Formulário de cadastro	20
Figura 17 – Formulário de login	20
Figura 18 – Formulário de edição de usuário (usuário administrativo)	21
Figura 19 – Gráfico Burndown com a Sprint 8 concluída	25

LISTA DE QUADROS

Quadro 1 – Cronograma de Atividades.	22
--	----

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BDD	Behaviour Driven Development
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JS	JavaScript
JSON	JavaScript Object Notation
MVC	Model, View, Controller
REST	Representational State Transfer
SPA	Single-Page Applications
SQL	Structured Query Language
TDD	Test Drive Development
URI	Uniform Resource Identifier
XP	Extreme Programming

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 OBJETIVOS	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
2 – SISTEMAS CORRELATOS	3
2.1 TudoGostoso	3
2.2 Vovó Palmirinha	3
2.3 Receitas CyberCook	4
2.4 Receitas GSHOW	5
2.5 DIFERENCIAL TECNOLÓGICO	5
3 – METODOLOGIA DE DESENVOLVIMENTO DO SISTEMA	6
4 – ANÁLISE E PROJETO DO SISTEMA	7
4.1 USER STORIES	7
4.2 TECNOLOGIAS USADAS NO DESENVOLVIMENTO	8
4.2.1 REST e JSON	8
4.2.2 PostgreSQL	9
4.2.3 Ruby e Ruby on Rails	9
4.2.4 PgSearch	9
4.2.5 RSpec	9
4.2.6 RuboCop	9
4.2.7 HTML e CSS	9
4.2.8 JavaScript, Vue.js e Axios	10
4.2.9 Karma	10
4.2.10 ESLint	10
4.2.11 Git e GitHub	11
4.2.12 Trello	11
4.3 BANCO DE DADOS	11
4.4 PROTÓTIPO DE TELAS	12
4.5 PLANEJAMENTO DO TRABALHO	22
4.6 RECURSOS NECESSÁRIOS	22
5 – DESENVOLVIMENTO DO SISTEMA	23
5.1 SPRINTS	23
5.1.1 Sprint 1: Registro e Autenticação de Usuário	23

5.1.2	Sprint 2: Cadastro de Receitas	23
5.1.3	Sprint 3: Edição de Receitas	23
5.1.4	Sprint 4: Listagem de Receitas e Busca Simples	24
5.1.5	Sprint 5: Comentários em Receitas	24
5.1.6	Sprint 6: Sistema de Avaliação de Receitas	24
5.1.7	Sprint 7: Categorias	24
5.1.8	Sprint 8: Usuários Administrativos	24
5.2	GRÁFICO BURNDOWN	25
6	– CONSIDERAÇÕES FINAIS	26
	Referências	27

1 INTRODUÇÃO

Farinha de trigo, cevada selvagem, raízes de plantas trituradas e água são os ingredientes de uma das mais antigas receitas de pão que se tem conhecimento ([G1, 2018](#)). Esta receita tem aproximadamente 14 mil anos, situando-a antes do início da agricultura, podendo ser um dos motivos que levou a humanidade a cultivar cereais, dando início a agricultura. Outro aspecto desta receita é que ela data de antes da invenção da escrita, tornando impossível transmiti-la por pergaminhos ou livros, limitando-se apenas à transmissão oral, tradição que em menor grau perdura até os dias atuais. O advento da escrita e posteriormente da prensa tornou possível a disseminação de receitas em grande escala, capacidade rivalizada apenas pelo surgimento e popularização da internet, permitindo que uma receita recém elaborada possa ser compartilhada com pessoas em qualquer parte do mundo com um dispositivo com acesso à Internet.

O passar do tempo não mudou apenas a forma de transmitir receitas. O surgimento da agricultura e da industrialização massificou a produção de alimentos, tornando-os extremamente acessíveis, criando um novo problema: desperdício de alimentos. O artigo *Global food losses and food waste – Extent, causes and prevention* traz dados alarmantes: cerca de 1,3 bilhão de toneladas de alimento é descartado todo ano - um terço de toda produção; em países desenvolvidos, uma pessoa desperdiça entre 95Kg e 115Kg de alimento por ano ([FAO, 2011](#)). Este desperdício se estende por toda a cadeia de produção, e contribui para o aumento do aquecimento global. Se toda comida desperdiçada fosse um país ele ficaria atrás apenas da China e dos E.U.A. na emissão de CO₂ equivalente (o potencial de efeito estufa de outros gases são estimados em CO₂ e chamados de CO₂e ou CO₂ equivalente) ([WRI, 2015](#)). É estimado que em 2050 o planeta terá 9 bilhões de habitantes, e para alimentar todo mundo, será necessário produzir mais 60% de caloria alimentícia. Se o desperdício for reduzido pela metade será necessário produzir apenas mais 22% ([LIPINSKI et al., 2013](#)). O desperdício de alimentos afeta a todos, direta ou indiretamente, e não deve ser ignorado.

Este documento propõe o desenvolvimento do SABRE (Sistema Avançado de Busca de Receitas), sistema web de receitas colaborativo que visa facilitar o aproveitamento de ingredientes que o usuário já possui, colaborando com a redução de alimento que vira lixo no final da cadeia de produção, o consumidor. O sistema contará com uma funcionalidade de busca de receitas com base em ingredientes fornecidos: o usuário informará os ingredientes que tem à disposição e o sistema exibirá as receitas possíveis de fazer apenas com os ingredientes informados. Este recurso não está presente em aplicações similares, dificultando a busca por receitas que se possa fazer apenas com os itens que já possui, geralmente tendo que adquirir mais ingredientes para poder preparar uma receita, que pode desestimular o uso dos itens ou aumentar ainda mais a quantidade de ingredientes sem uso. O SABRE também deverá ter os recursos mais comuns entre os sistemas similares: busca por nome, área de comentários e

avaliação de receita.

O principal desafio deste projeto está em como relacionar os ingredientes e as receitas, selecionando apenas aquelas que contêm alguns ou todos os ingredientes da lista e não necessite de nenhum outro item. Durante o desenvolvimento será necessário analisar algoritmos que façam esta correlação e escolher o que melhor se adeque ao contexto. Este algoritmo deverá receber um conjunto de ingredientes e verificar em cada receita se o conjunto de ingredientes desta está contido no conjunto fornecido pelo usuário. Será verificado a possibilidade de automatizar este relacionamento direto via banco de dados, fazendo uso de técnicas como *Full Text Search*. Também será analisado a possibilidade de identificar os ingredientes previamente e fazer este relacionamento via comandos diretamente no banco de dados, relacionar diretamente na camada de aplicação ou desenvolver um terceiro aplicativo especializado para fazer esta tarefa.

Este projeto auxiliará com a otimização do uso de alimentos evitando desperdícios, resultando em economia para o consumidor e redução do impacto ambiental gerado durante todo o processo de produção do alimento descartado.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um sistema web que permita busca de receitas por meio de ingredientes disponíveis.

1.1.2 Objetivos Específicos

- Estudar e definir um algoritmo para selecionar receitas de acordo com conjunto de ingredientes.
- Desenvolver componentes para listar as receitas.
- Desenvolver um sistema de busca por palavra-chave.
- Desenvolver formulários de cadastro e autenticação de usuários.
- Desenvolver formulários de cadastro de receitas.
- Desenvolver sistema de categorias.
- Desenvolver sistema de avaliação de receitas.
- Desenvolver sistema de comentários.
- Desenvolver área administrativa.

2 SISTEMAS CORRELATOS

Este capítulo lista alguns sistemas existentes com funcionalidades similares ao SABRE.

2.1 TudoGostoso

TudoGostoso ([Figura 1](#)) é um site colaborativo de receitas pioneiro na área com um grande acervo. Permite encontrá-las por palavra-chave no nome ou ingrediente. Possui lista de favoritos, área de comentários, categorias, envio de receitas, envio de fotos, avaliação de receitas e *ranking* de usuários que mais postam receitas durante a semana.

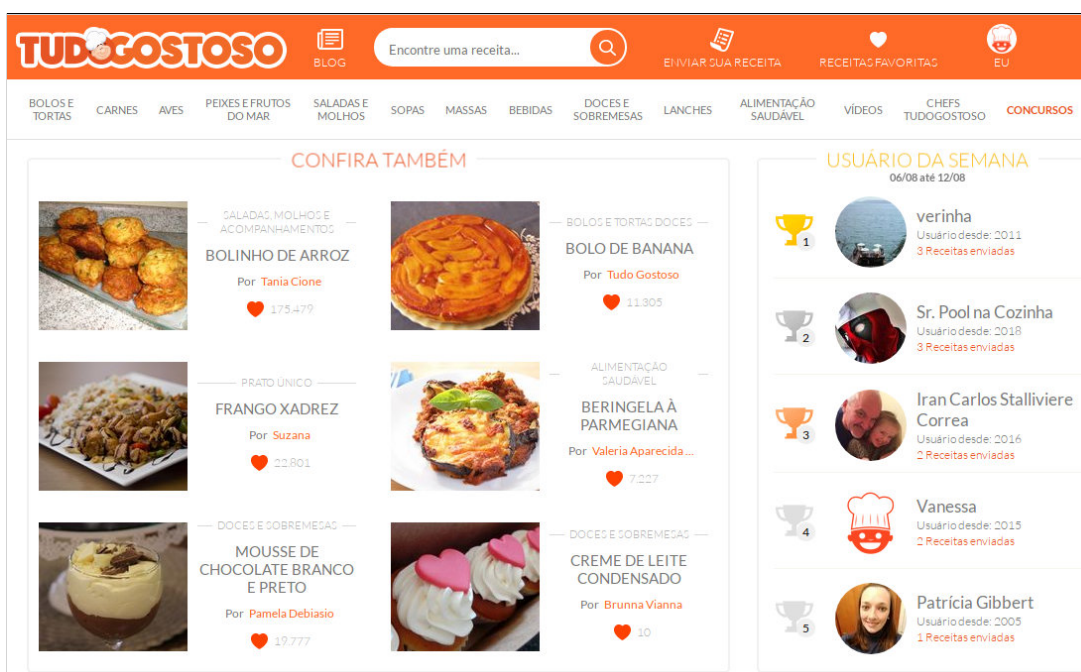


Figura 1 – TudoGostoso

Fonte: [TudoGostoso](#) (2018)

2.2 Vovó Palmirinha

Assim como TudoGostoso, Vovó Palmirinha ([Figura 2](#)) é um site colaborativo de receitas com um grande acervo. Permite encontrá-las por nome, além de possuir área de comentários, categorias e envio de fotos. Não possui sistema de qualificação para avaliar receitas, favoritos ou busca por ingredientes.

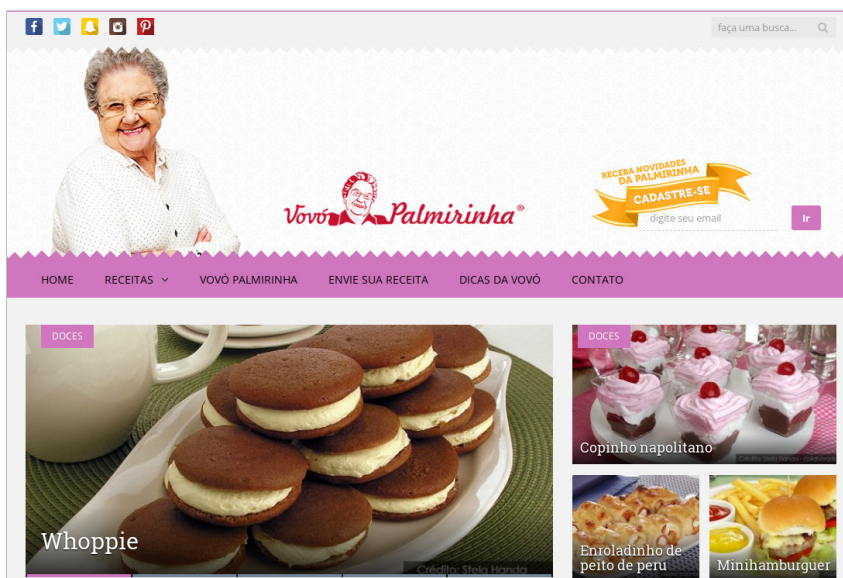


Figura 2 – Vovó Palmirinha

Fonte: [Palmirinha](#) (2018)

2.3 Receitas CyberCook

Assim como TudoGostoso e Vovó Palmirinha, Receitas CyberCook ([Figura 3](#)) é um site colaborativo de receitas. Possui área de comentários, categorias, envio de receitas, fotos, sistema de qualificação para avaliar das mesmas, favoritos e busca por voz e texto - apenas nome. Não possui busca por ingredientes.



Figura 3 – Receitas CyberCook

Fonte: [CyberCook](#) (2018)

2.4 Receitas GSHOW

Diferente dos anteriores, Receitas GSHOW não é um site colaborativo de receitas, pois seu foco é disponibilizar as que aparecem em programas de TV da Rede Globo, sendo exibidas como texto ou por meio de vídeo. Permite encontrá-las por nome, contudo não contém um mecanismo de avaliação, favoritos, área de comentários, envio de receitas e fotos, categorias ou busca por ingredientes.



Figura 4 – Receitas GSHOW

Fonte: [Gshow](#) (2018)

2.5 DIFERENCIAL TECNOLÓGICO

O projeto apresentado neste documento contém a característica única de encontrar receitas possíveis de preparar usando apenas os ingredientes disponíveis. Dos sistemas similares existentes apresentados nesta seção apenas o site TudoGostoso chega a ter um mecanismo que leva os ingredientes em consideração, mas diferente do SABRE sua busca contempla outras receitas que exigem ingredientes não disponíveis, não tendo um filtro semelhante do sistema proposto. Usar estes livros digitais de receitas para aproveitar o que já tem pode demandar um tempo desnecessário para o usuário, levando-o a desistir de aproveitar os recursos que tem mãos.

3 METODOLOGIA DE DESENVOLVIMENTO DO SISTEMA

O desenvolvimento do SABRE faz-se de maneira incremental e iterativa. A cada iteração são entregues novas funcionalidades até que no último ciclo o sistema proposto esteja completamente desenvolvido. Para o desenvolvimento são selecionados elementos dos modelos de desenvolvimento TDD (*Test Drive Development*) e *Scrum*. O aluno tem os papéis de usuário e desenvolvedor.

Os requisitos do sistema são extraídos do item 1.1.2 deste documento e convertido em *User Story* (História do Usuário, em tradução livre) e devem responder as perguntas "Quem?", "Por quê?" e "O Quê?". Por exemplo, para o módulo de cadastro de receitas: "Como usuário eu posso cadastrar receitas para que outras pessoas possam fazê-las" (ver 4.1).

O fluxo de trabalho é organizado em quadros e cartões, que são categorizados em *Product Backlog*, *Current Sprint*, *To do*, *In Progress*, *Review* e *Done*. *Product Backlog* contém todas as *User Story* ordenadas por prioridade e movidas para *Current Sprint* quando seu desenvolvimento é iniciado. As histórias marcadas para desenvolver têm suas tarefas definidas em *To Do* e movidas para *In Progress* quando em desenvolvimento, seguindo para *Review* e *Done* após revisão. Uma tarefa marcada como *Done* deve passar pelo ciclo: Requisito, *Product Backlog*, *Current Sprint*, *To Do*, *In Progress*, Teste e Revisão. Após a revisão e verificação de conformidade com requisito solicitado a tarefa passa para o estado de *Done*.

No método *Scrum*, uma *Sprint* é uma unidade de tempo definido para ter uma entrega de requisitos convertidos em produto (SCHWABER; SUTHERLAND, 2017), cada *Sprint* tem duração de uma semana durante o projeto. Este método também define várias reuniões que são simplificadas neste projeto. Há apenas uma reunião semanal onde é discutido o que foi feito na última *Sprint*, o que será feito na próxima e os possíveis problemas encontrados durante o último ciclo. Nesta reunião também pode ser realizado uma *Sprint Retrospective*, evento onde a metodologia é discutida e adaptada visando aumentar a produtividade.

No modelo TDD o desenvolvedor primeiro escreve um teste automatizado para a funcionalidade que irá desenvolver, este teste deve falhar enquanto a funcionalidade não estiver funcionando propriamente. Em seguida é escrito o código da funcionalidade para que passe no teste. Por último o código é refatorado seguindo boas práticas como simplicidade e não duplicidade (AGILEALLIANCE, 2015). A cada nova funcionalidade o sistema acumula mais códigos de teste que verificam o funcionamento de todas as funcionalidades implementadas. Este procedimento é adotado durante todo o desenvolvimento do projeto onde as tarefas são facilmente definidas antes do desenvolvimento. Em partes onde mostre-se necessário prototipar para verificar o caminho a seguir, os testes são escritos depois das funções.

4 ANÁLISE E PROJETO DO SISTEMA

Este projeto é desenvolvido seguindo fortemente os preceitos de desenvolvimento ágil definido por [Fowler \(2005\)](#) como: adaptativo ao invés de preditivo e orientado a pessoas ao invés de processos. Neste tipo de projeto evita-se a produção de artefatos que representam a visão final do produto visto que o desejo e necessidade do cliente é mais importante que os processos, e este pode mudar de ideia ou perceber que a funcionalidade pedida não corresponde as suas necessidades. Sua percepção pode mudar a cada entrega de funcionalidades e todo planejamento pode ser descartado para dar origem a um projeto diferente. Para fins didáticos e de cumprimento de requisitos da disciplina de TCC esta seção contém alguns artefatos que ilustram o resultado final do projeto de acordo com a percepção do sistema durante a modelagem inicial. O sistema finalizado pode variar em alguns aspectos deste projeto. Este capítulo também apresenta ferramentas tecnológicas selecionadas para usar no projeto e histórias de usuário (*User Stories*).

4.1 USER STORIES

Uma *User Story* é um artefato de alto nível em metodologias ágeis como *Scrum* e *Extreme Programming* (XP), ela deve conter informação suficiente para o desenvolvedor estimar um prazo de desenvolvimento ([AGILEMODELING, 2013](#)). As *User Stories* definidos até o momento são:

- Como usuário eu posso cadastrar receitas para que outras pessoas possam fazê-las.
- Como administrador eu posso liberar receitas cadastradas pelos usuários para que outros usuários possam prepará-las.
- Como visitante eu posso ver receitas cadastradas para prepará-las.
- Como visitante eu posso procurar receitas cadastradas para prepará-las.
- Como usuário eu posso comentar receitas para compartilhar minha experiência com outras pessoas.
- Como visitante eu posso ver os comentários nas receitas para me ajudar a decidir se faço ou não a receita.
- Como usuário eu posso avaliar receitas para expressar o quanto gostei da mesma.
- Como visitante eu posso ver as avaliações das receitas para me ajudar a decidir qual receita fazer.
- Como visitante eu posso ver receitas por categoria para facilitar minha busca.
- Como administrador eu posso cadastrar outro administrador para gerenciar receitas e administradores.
- Como administrador eu posso bloquear receitas de usuários para não exibir conteúdo inadequado ou ilegal.

- Como administrador eu posso bloquear usuários para que não comentem ou cadastrem receitas.
- Como administrador eu posso excluir comentários para remover conteúdo inadequado ou ilegal.
- Como visitante eu posso ver receitas possíveis de fazer com um conjunto de ingredientes especificados para aproveitar o que já tenho.

No decorrer do desenvolvimento *User Stories* podem ser adicionadas ou removidas de acordo necessidades percebidas.

4.2 TECNOLOGIAS USADAS NO DESENVOLVIMENTO

Esta seção apresenta as principais tecnologias selecionadas para serem usadas no desenvolvimento.

4.2.1 REST e JSON

A arquitetura REST (*Representational State Transfer*) foi descrita pela primeira vez por Roy Thomas Fielding em sua tese de doutorado. É uma arquitetura que faz uso dos padrões URI e HTTP e define uma série de restrições e propriedades que visam performance de interação, simplicidade, modificabilidade, portabilidade e confiabilidade (FIELDING, 2000, cap. 2 e 5). As definições mais significativas para este projeto são: arquitetura cliente-servidor, *stateless* (cada requisição deve conter toda informação necessária, não fazendo uso de nenhum contexto salvo previamente no servidor) e interface uniforme.

JSON (*JavaScript Object Notation*) é um formato leve para troca de dados, fácil para humanos lerem e para máquinas gerarem e interpretarem. Embora seja baseado em um subconjunto da linguagem JavaScript seu formato é independente de linguagem (JSON.ORG, 2018). Será usado nas trocas de informação entre o servidor e cliente.

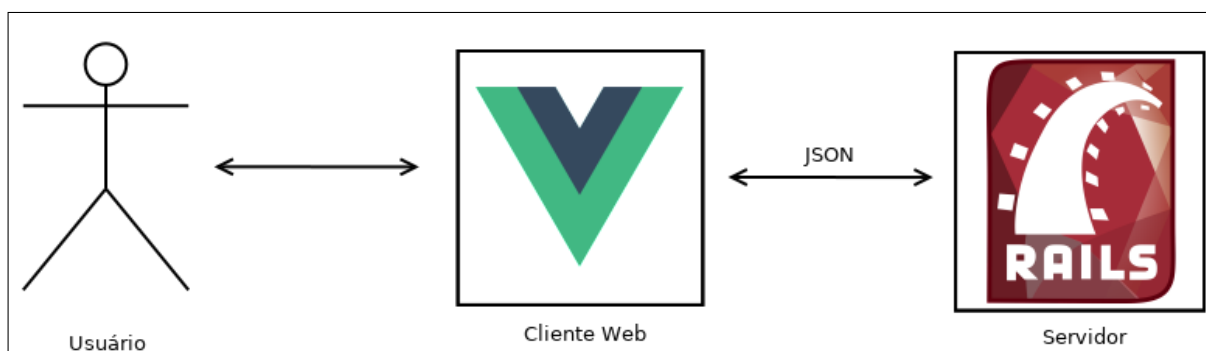


Figura 5 – Arquitetura Cliente-Servidor do SABRE

4.2.2 PostgreSQL

É um sistema de banco de dados objeto-relacional de código aberto, que usa e estende a linguagem SQL combinando com vários recursos para armazenar os dados com segurança ([POSTGRESQL, 2018](#)). Terá a função de armazenar usuários, receitas, comentários e qualquer outra informação que precise ser persistida.

4.2.3 Ruby e Ruby on Rails

Ruby é uma linguagem de programação criada por Yukihiro “Matz” Matsumoto que mistura partes de suas linguagens favoritas (Perl, Smalltalk, Eiffel, Ada, e Lisp) para formar uma nova linguagem que balance os paradigmas funcionais e imperativos ([RUBY, 2018](#)).

Ruby on Rails (também chamado apenas de Rails) é um *framework* de desenvolvimento *web* feito em Ruby. Rails segue o padrão MVC (*Model, View, Controller*) e tem suporte nativo a banco de dados relacional, APIs REST e outros recursos comuns em sistemas *web* ([RAILSGUIDES, 2018](#)).

Estas tecnologias serão usadas no servidor do SABRE.

4.2.4 PgSearch

PgSearch é um *plugin* Rails que adiciona a funcionalidade de busca *Full Text Search* do PostgreSQL, recurso não presente nativamente no Rails. Com ele é possível usar o *Full Text Search* com as extensões *Trigram*, *Unaccent* e *Fuzzystrmatch*, que enriquecem a busca usando similaridades em sequencias de caracteres, ignorando acentos e por aproximação fonética ([PGSEARCH, 2018](#)). Será usado, ao menos no princípio, pela busca comum de palavras chaves nas receitas.

4.2.5 RSpec

RSpec é uma ferramenta de testes para aplicativos criados na linguagem Ruby feito para facilitar o desenvolvimento em BDD (*Behaviour Driven Development*) e TDD ([RSPEC, 2018](#)). Será usado essencialmente para testes unitários e de requisição no servidor do projeto.

4.2.6 RuboCop

RuboCop é um analisador de código fonte Ruby. Ele verifica se o código segue o [Ruby Style Guide](#), procura por erros, consegue corrigir alguns problemas automaticamente e é altamente personalizável. ([RUBOCOP, 2018](#)).

4.2.7 HTML e CSS

HTML (*Hypertext Markup Language*) é a linguagem que descreve a estrutura de páginas *web*. Com ela é possível publicar documentos com cabeçalhos, texto, tabelas, fotos, etc.

A estrutura das páginas é definida usando marcações que rotulam conteúdos como "formulário", "paragrafo", etc ([W3C, 2018](#)).

CSS (*Cascading Style Sheets*) é a linguagem que define a apresentação de páginas *web*, tais como cores, posição de conteúdo e fontes tipográficas. Com ela é possível adaptar o conteúdo em diferentes dispositivos, como telas grandes, pequenas ou para impressoras ([W3C, 2018](#)).

Estas duas linguagens são fundamentais no desenvolvimento do cliente *web* do SABRE.

4.2.8 JavaScript, Vue.js e Axios

JavaScript[®] (comumente abreviado para JS) é uma linguagem interpretada, multi-paradigma, baseada em protótipos, com suporte a orientação a objetos, programação imperativa e declarativa. Inicialmente criada para proporcionar dinamismo as páginas *web*, agora ela pode ser usada em outras áreas como servidores e aplicativos não *web* ([MOZILLA, 2018](#)).

Vue.js é um *framweork* JavaScript progressivo para criar interfaces de usuário. Seu núcleo é focado na parte visual, enquanto outras bibliotecas oficiais e não oficiais são fornecidas para adicionar outras funcionalidades, como temas, rotas, gerenciamento de estado, etc. Em conjunto com outras ferramentas e bibliotecas Vue tem a capacidade de criar SPA (*Single-Page Applications*) ([VUE, 2018](#)).

Axios é um cliente HTTP feito em JavaScript. Suporta requisições GET, POST, DELETE, PATCH e PUT, além de converter automaticamente os dados em JSON ([AXIOS, 2018](#)).

O cliente *web* do projeto será uma SPA com o conteúdo gerenciado dinamicamente pelo Vue.js e as trocas de informação com o servidor será feito através do axios.

4.2.9 Karma

É uma ferramenta que testa código JavaScript em diversos navegadores. Ele funciona carregando um servidor e um ou mais navegadores, em seguida executa os testes em cada navegador. O Resultado é exibido no terminal. Extensível com *plugins* que adicionam suporte a mais navegadores e recursos extras de teste ([KARMA, 2018](#)).

4.2.10 ESLint

ESLint é um *linter* JavaScript de código aberto. Um *linter* é uma ferramenta que faz análise estática de código fonte, procurando por padrões problemáticos ou código que não respeitam as regras de estilo. ESLint é um utilitário flexível, podendo ativar, desativar ou alterar regras pré-definidas, além da possibilidade de adicionar novas regras personalizadas ([ESLINT, 2018](#)).

4.2.11 Git e GitHub

Git é um sistema livre e gratuito de versionamento de código projetado para lidar com projetos pequenos ou grandes com velocidade e eficiência (GIT, 2018).

GitHub é uma plataforma de desenvolvimento onde é possível hospedar e revisar código, gerenciar projetos e construir aplicativos de forma colaborativa (GITHUB, 2018).

Os códigos do projeto estão nos repositórios <https://github.com/DoniB/sabre_server> e <https://github.com/DoniB/sabre_web> do GitHub. Tanto o módulo servidor quanto o cliente estão disponíveis publicamente sob a licença livre *BSD 3-Clause*

4.2.12 Trello

Trello é uma ferramenta *web* para organizar projetos e ideias. Nele é possível ter listas e cartões organizados facilmente e visualmente. Os projetos podem ter vários membros, que podem interagir com as listas e cartões, além de comentar nos mesmos (TRELLO, 2018).

Seu principal papel será de organizar o projeto e registrar o progresso do desenvolvimento do SABRE.

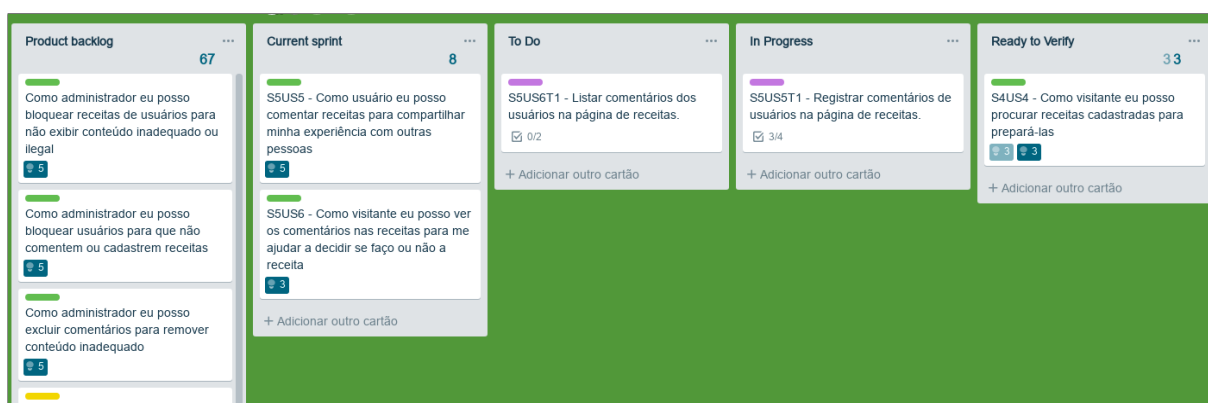


Figura 6 – Trello do projeto SABRE durante a Sprint 5

4.3 BANCO DE DADOS

O SABRE usa o PostgreSQL para gerenciar o banco de dados, contando com os recursos do Ruby on Rails para acessar e manipular os registros.

A Figura 7 ¹ é uma representação do banco de dados considerando a visão atual do sistema, além de não contar com as especificidades e limitações do framework que gerencia os dados. O resultado final pode ter alterações pequenas ou significativas.

¹Este modelo foi desenvolvido com dbdiagram.io e está disponível publicamente em <<https://dbdiagram.io/d/5be4b302cf7cf50014f2a984>>

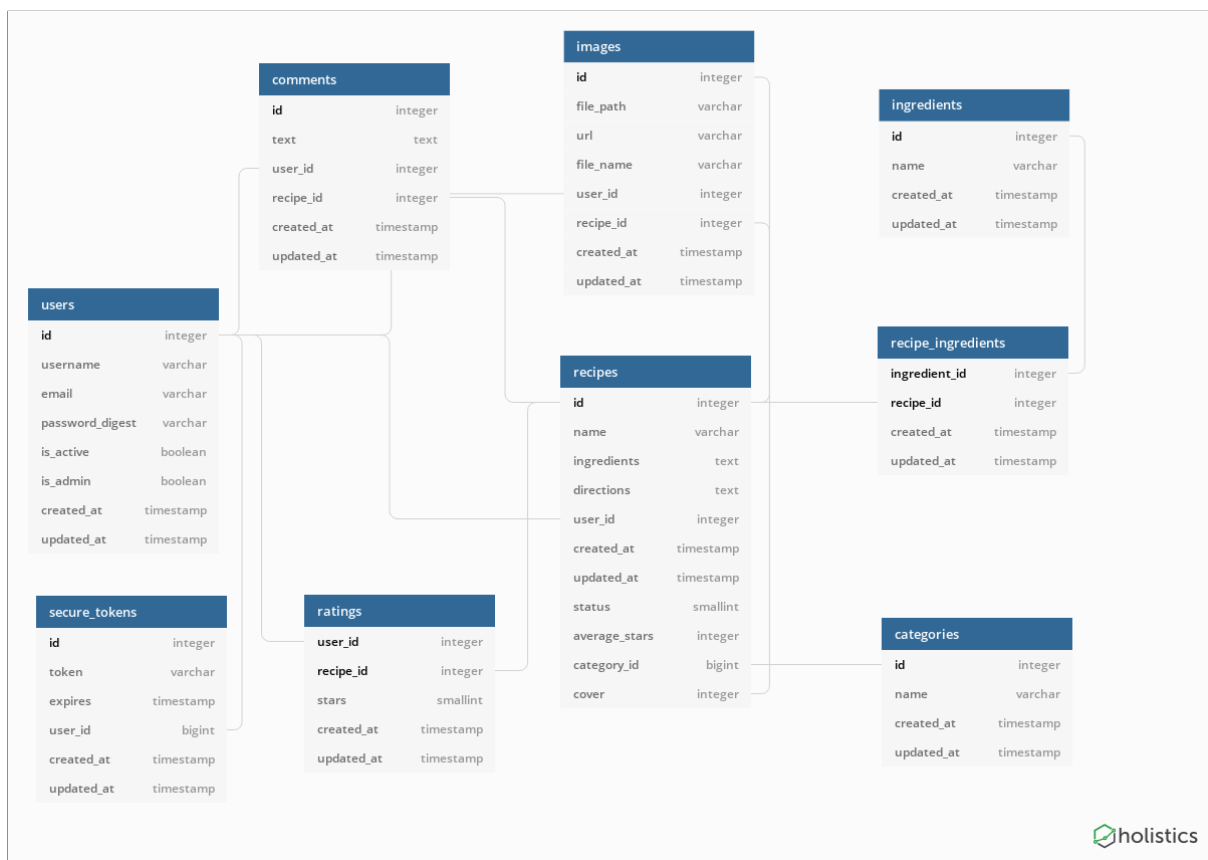


Figura 7 – Diagrama do Banco de Dados

4.4 PROTÓTIPO DE TELAS

Esta seção apresenta alguns protótipos de telas do SABRE de acordo com a perspectiva atual do sistema.

A página inicial ([Figura 8](#)) contém uma barra com caixa de pesquisa, uma área com as categorias e listagem de receitas.

Cada receita na página inicial ([Figura 11](#)) exibe o nome da receita, nome de quem enviou, foto, média de avaliações, botão para adicionar aos favoritos, botão para ver a receita completa e botão para visualizar os ingredientes sem precisar abrir a receita.

A [Figura 10](#) exibe a caixa de pesquisa em três estados: não ativa; ativa; ativa com itens separados por vírgula. Quando ativa aparece a sugestão para separar os ingredientes com virgula. Se uma vírgula é detectada aparece a opção para procurar por receitas que possam ser feitas apenas com aqueles ingredientes.

A página de receitas de usuário ([Figura 9](#)) exibe as receitas usando a mesma aparência familiar da página inicial. A página de favoritos é similar, mas exibe as receitas que foram marcadas como favoritas.

Um usuário administrativo pode acessar todas as receitas, podendo filtrar por suas receitas e receitas pendentes de aprovação ([Figura 13](#)).

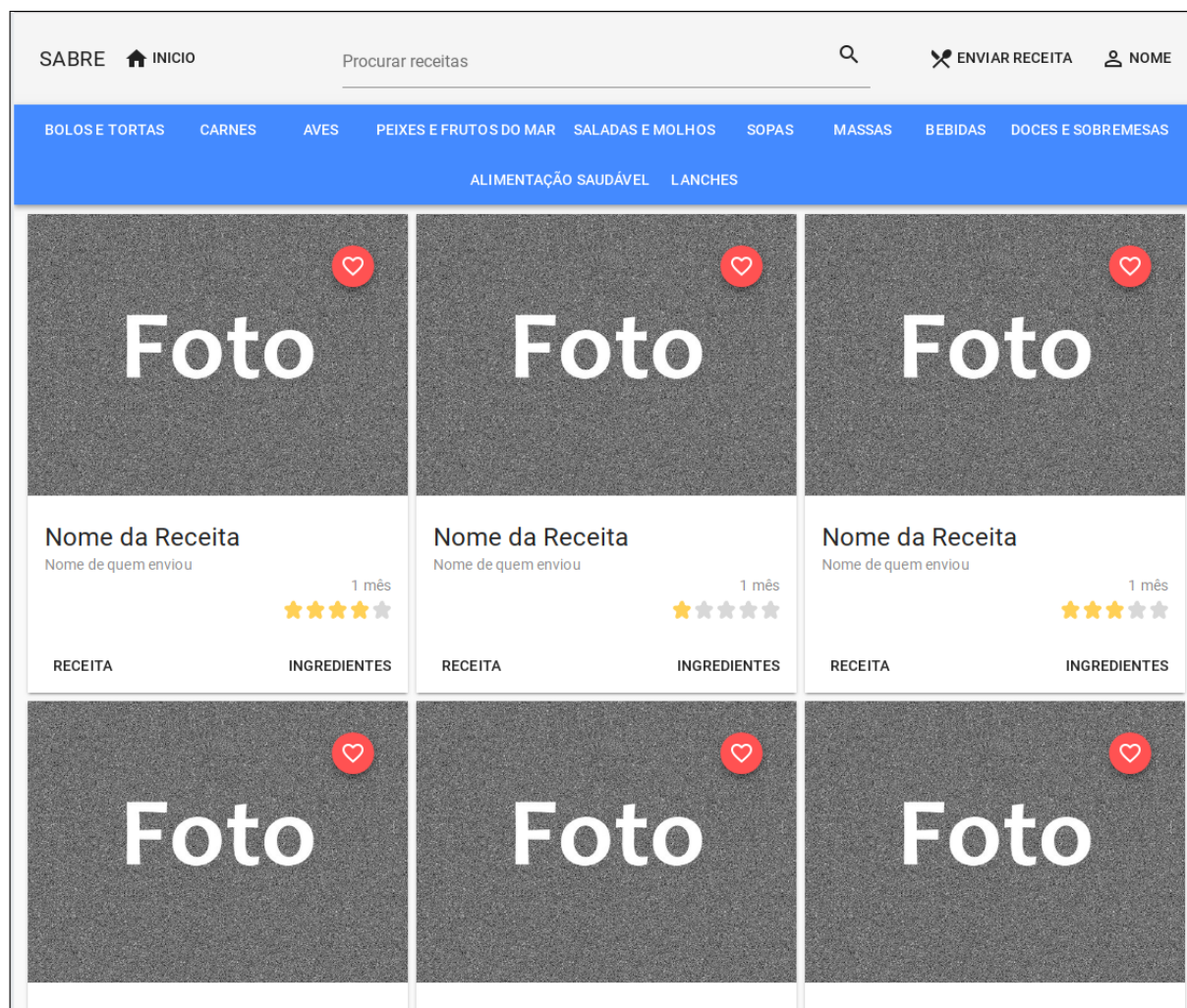


Figura 8 – Página inicial

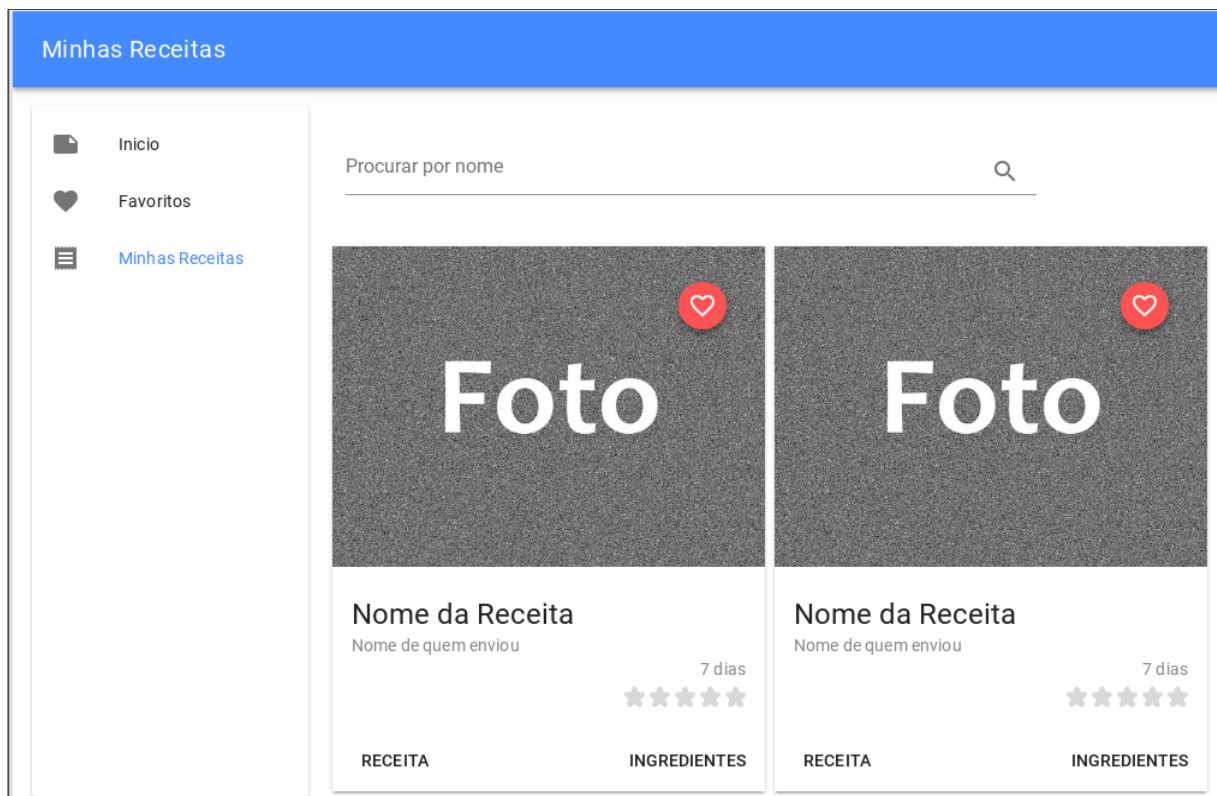


Figura 9 – Receitas do usuário

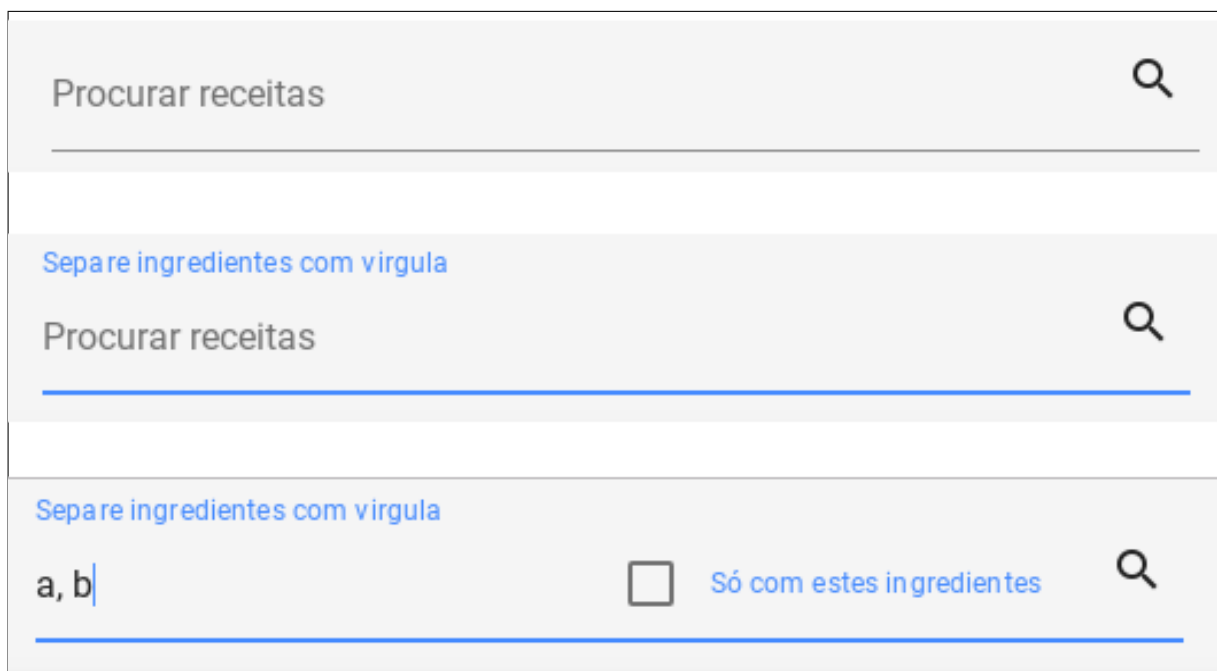


Figura 10 – Caixa de pesquisa



Figura 11 – Receita com pré-visualização de ingredientes

Cadastro de Receitas

Foto

 Foto (opcional)

Nome da receita *

Categoria

Ingredientes *

Modo de preparo *

ENVIAR

Figura 12 – Formulário de envio de receitas

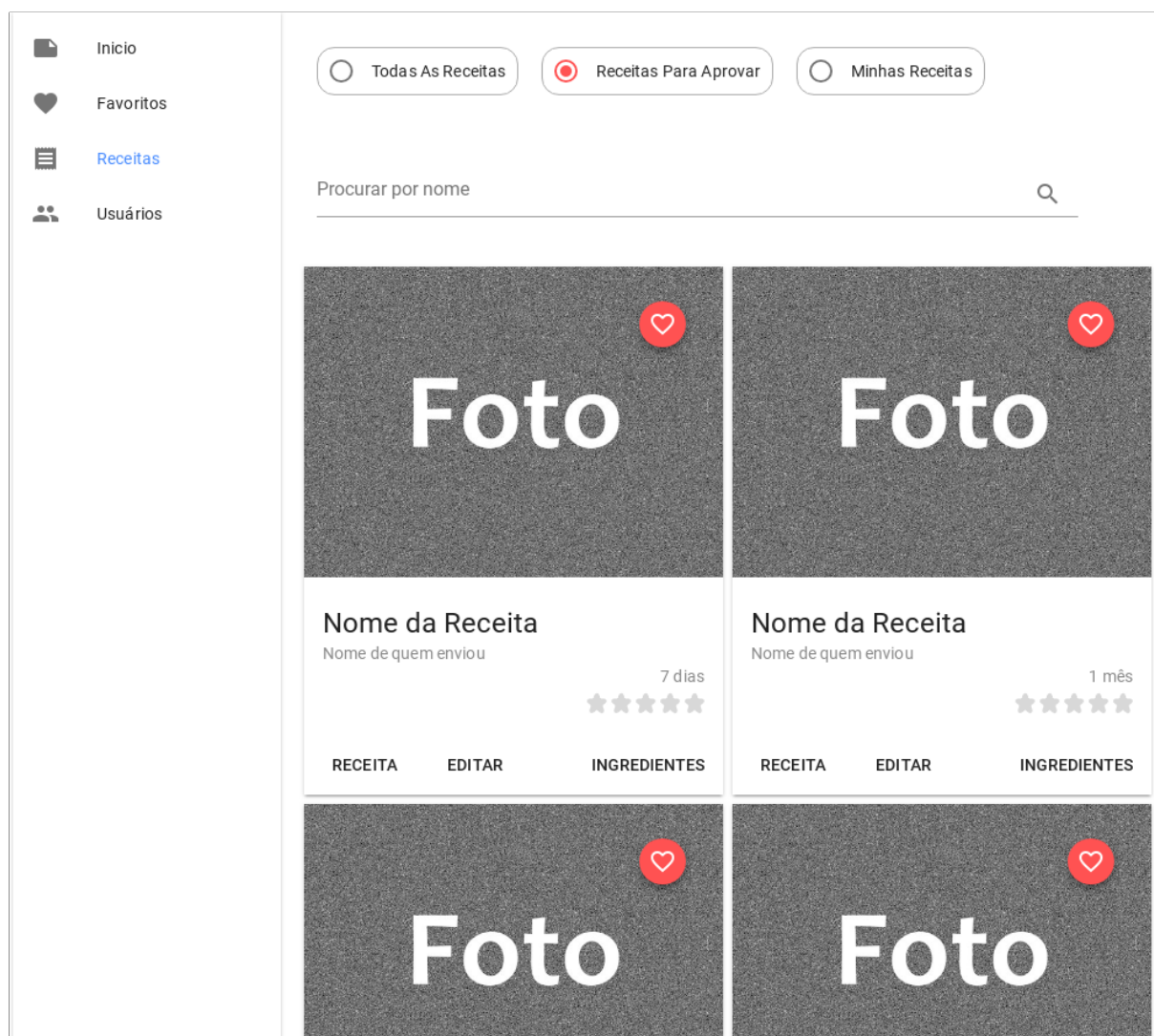


Figura 13 – Receitas para aprovar (usuário administrativo)

Receita Para Liberar

 foto.png

Nome da receita *

Nome da Receita

Ingredientes *

Ingrediente 1
Ingrediente 2
Ingrediente 3
Ingrediente 4
Ingrediente 5

Modo de preparo *

Passo 1
Passo 2
Passo 3
Passo 4
Passo 5
Passo 6
Passo 7

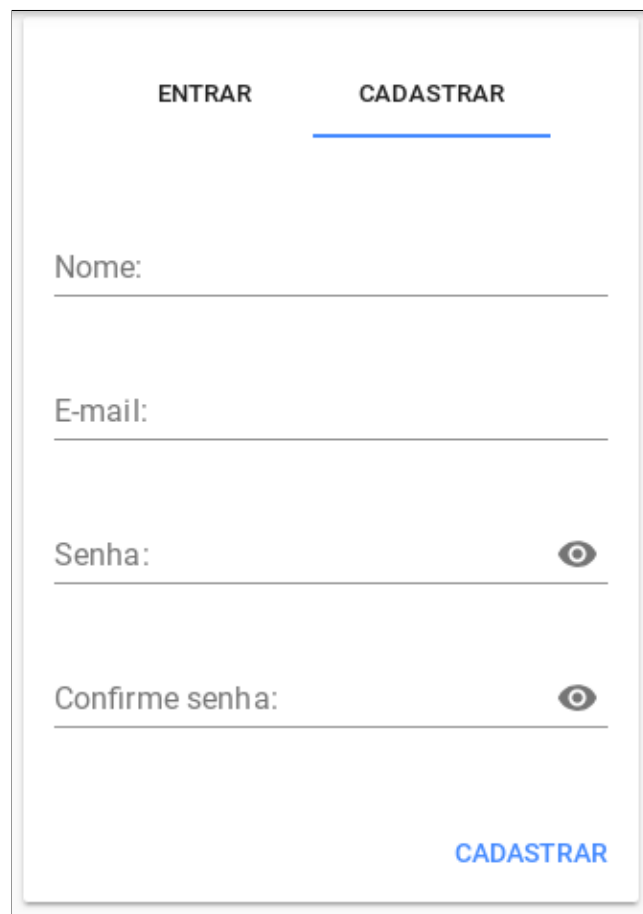
☐ Aguardando Ativação ☒ Ativo ☐ Pausado ☐ Rejeitado

ENVIAR

Figura 14 – Editar receita para aprovar (usuário administrativo)

Procurar por nome						
ID	Nome	Email	Cadastrado	Atualizado	Administrador	
115	Nome	email@email.com	16/11/2018 01:12:38	16/11/2018 01:12:38	Não	  
114	Nome	email@email.com	16/11/2018 00:55:05	16/11/2018 00:55:05	Não	  
113	Nome	email@email.com	20/10/2018 15:48:07	15/11/2018 18:43:31	Sim	  
112	Nome	email@email.com	20/10/2018 15:48:07	20/10/2018 15:48:07	Não	  
111	Nome	email@email.com	20/10/2018 15:48:07	20/10/2018 15:48:07	Não	  
110	Nome	email@email.com	20/10/2018 15:48:06	20/10/2018 15:48:06	Não	  
109	Nome	email@email.com	20/10/2018 15:48:06	20/10/2018 15:48:06	Não	  
108	Nome	email@email.com	20/10/2018 15:48:06	20/10/2018 15:48:06	Não	  
107	Nome	email@email.com	20/10/2018 15:48:06	20/10/2018 15:48:06	Não	  
106	Nome	email@email.com	20/10/2018 15:48:06	20/10/2018 15:48:06	Não	  
105	Nome	email@email.com	20/10/2018 15:48:05	20/10/2018 15:48:05	Não	  
104	Nome	email@email.com	20/10/2018 15:48:05	20/10/2018 15:48:05	Não	  
103	Nome	email@email.com	20/10/2018 15:48:05	20/10/2018 15:48:05	Não	  
102	Nome	email@email.com	20/10/2018 15:48:05	20/10/2018 15:48:05	Não	  
101	Nome	email@email.com	20/10/2018 15:48:05	20/10/2018 15:48:05	Não	  
100	Nome	email@email.com	20/10/2018 15:48:05	20/10/2018 15:48:05	Não	  

Figura 15 – Listagem de usuários (usuário administrativo)



ENTRAR CADASTRAR

Nome: _____

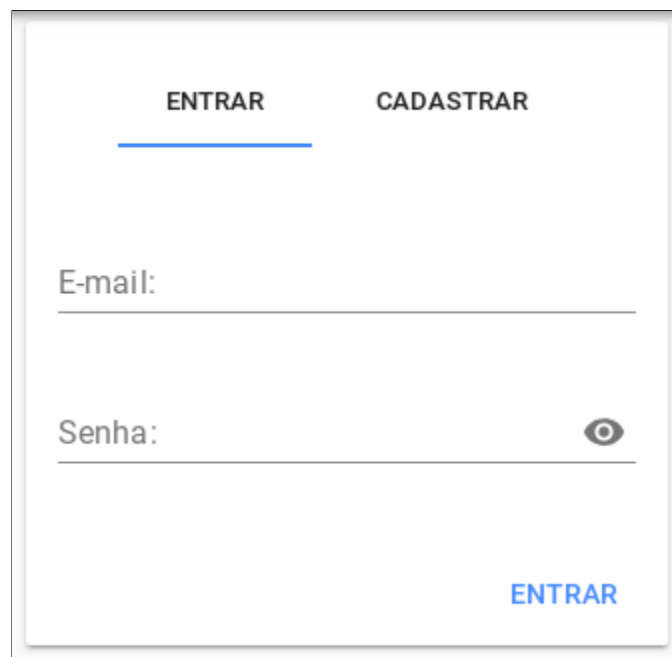
E-mail: _____

Senha: _____ 

Confirme senha: _____ 

CADASTRAR

Figura 16 – Formulário de cadastro



ENTRAR CADASTRAR

E-mail: _____

Senha: _____ 

ENTRAR

Figura 17 – Formulário de login

#114 - Usuário Teste

 Início

 Favoritos

 Receitas

 Usuários

Editar

Nome:
Usuário Teste

E-mail:
usuario@teste.com

Senha: 

Ativo?
☒ Não ☐ Sim

Administrador?
☒ Não ☐ Sim

[SALVAR](#)

Figura 18 – Formulário de edição de usuário (usuário administrativo)

4.5 PLANEJAMENTO DO TRABALHO

O [Quadro 1](#) contém as atividades relacionadas ao TCC e as datas aproximadas de início e conclusão das mesmas.

Atividades	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun
1. Revisão dos apontamentos da banca		X									
2. Revisão bibliográfica		X									
3. Levantamento de requisitos		X	X								
4. Redação do projeto de TCC		X	X	X							
5. Desenvolvimento do sistema			X	X	X	X	X				
6. Defesa do projeto de TCC					X						
7. Escrita da Monografia de TCC						X	X	X			
8. Elaboração da apresentação final								X	X		
9. Defesa final do TCC										X	

Quadro 1 – Cronograma de Atividades.

4.6 RECURSOS NECESSÁRIOS

Para o desenvolvimento do sistema é necessário um computador com acesso à Internet, editor de código fonte, sistema gerenciador de banco de dados, e uma ou mais linguagens de desenvolvimento para o servidor.

5 DESENVOLVIMENTO DO SISTEMA

O desenvolvimento teve início em 25 de setembro de 2018. Foi estimado pelo aluno que seriam necessárias 135 horas para transformar os requisitos em um produto de software. Com uma carga de trabalho de oito horas por semana/*sprint* e sem alterações dos requisitos, a conclusão do projeto ocorrerá no final de fevereiro de 2019. Esta seção contém um resumo das *Sprints* concluídas até o término deste documento e um gráfico do progresso (gráfico *burndown*).

5.1 SPRINTS

Cada *Sprint* tem duração de uma semana, iniciada em uma reunião no dia 25 de setembro de 2018 e tendo sua primeira entrega de funcionalidades sete dias depois (2 de outubro). O aluno deve destinar oito horas semanais para o projeto, sendo seis de desenvolvimento e duas para estudo e imprevistos.

5.1.1 Sprint 1: Registro e Autenticação de Usuário

Iniciada em 25 de setembro e encerrada em 2 de outubro. Foi definido o tempo de oito horas que o aluno destinará para desenvolvimento em cada *Sprint* e selecionada a *User Story* "Como usuário eu posso cadastrar receitas para que outras pessoas possam fazê-las" para desenvolver. Foi criado o projeto do servidor em *Ruby on Rails* e desenvolvidos cadastro de usuário, sistema de geração de *token* para identificar o usuário, cadastro de receitas (nome, ingredientes e modo de preparo) e testes seguindo o modelo TDD no servidor. Também foi criado o projeto cliente *web* com Vue.js com formulário de cadastro e formulário de login. Não foi possível criar o formulário de cadastro de receitas e os testes no lado do cliente por falta de conhecimento das tecnologias usadas pelo aluno.

5.1.2 Sprint 2: Cadastro de Receitas

Iniciada em 2 de outubro de 2018. Foi ajustado o tempo de oito para seis horas de desenvolvimento, deixando duas horas para estudos e imprevistos. Essa *Sprint* também foi reservada para a conclusão da *Sprint* anterior e calcular o tempo necessário para desenvolver cada *User Story*. Foram feitas pequenas correções no servidor e desenvolvido o formulário de cadastro de receitas no cliente.

5.1.3 Sprint 3: Edição de Receitas

Iniciada em 9 de outubro de 2018. Selecionada a *User Story* "Como administrador eu posso liberar receitas cadastradas pelos usuários para que outros usuários possam prepará-

las”. No servidor foi adicionado a opção de atualizar as receitas pelo usuário que tem status administrativo. No cliente *web* foi desenvolvido uma *dashboard*, com listagem e edição de receitas acessíveis apenas por usuários administrativos. Nesta etapa foi percebido um crescente débito técnico pela falta de testes no cliente *web* que terá que ser resolvido posteriormente.

5.1.4 Sprint 4: Listagem de Receitas e Busca Simples

Iniciada em 16 de outubro de 2018. Seleccionadas as *User Stories* “Como visitante eu posso ver receitas cadastradas para prepará-las” e “Como visitante eu posso procurar receitas cadastradas para prepará-las”. Nesta etapa foi feito a listagem de receitas, função de busca por palavra chave e exibição de receita individualmente, tanto no servidor quanto no cliente *web*.

5.1.5 Sprint 5: Comentários em Receitas

Iniciada em 23 de outubro de 2018. Seleccionadas as *User Stories* “Como usuário eu posso comentar receitas para compartilhar minha experiência com outras pessoas” e “Como visitante eu posso ver os comentários nas receitas para me ajudar a decidir se faço ou não a receita”. Foram desenvolvidas no servidor e cliente *web* as funcionalidades de comentar e ver comentários nas receitas.

5.1.6 Sprint 6: Sistema de Avaliação de Receitas

Iniciada em 30 de outubro de 2018. Seleccionadas as *User Stories* “Como usuário eu posso avaliar receitas para expressar o quanto gostei da mesma” e “Como visitante eu posso ver as avaliações das receitas para me ajudar a decidir qual receita fazer”. Foi desenvolvido um sistema de avaliação de receitas com escala de um a cinco. O servidor recalcula a média de avaliações a cada avaliação e o cliente *web* as exibe com um ícone de estrelas.

5.1.7 Sprint 7: Categorias

Iniciada em 6 de novembro de 2018. Seleccionada a *User Story* “Como visitante eu posso ver receitas por categoria para facilitar minha busca”. Foi desenvolvido listagem de categorias no servidor (adicionadas manualmente) e associado as receitas. No cliente *web* foi adicionado categoria no cadastro de receitas e menu de categorias nas áreas públicas. Também foi dedicado um tempo extra para produzir telas para este documento ([Seção 4.4](#)).

5.1.8 Sprint 8: Usuários Administrativos

Iniciada em 6 de novembro de 2018. Seleccionada a *User Story* “Como administrador eu posso cadastrar outro administrador para gerenciar receitas e administradores”. Foram desenvolvidas no cliente e servidor a listagem, cadastro e edição de usuários por usuários administradores. Também foi dedicado um tempo extra para desenvolver este documento.

5.2 GRÁFICO BURNDOWN

No desenvolvimento do projeto são estimadas 18 *Sprints* para sua conclusão, cada uma com duração de uma semana, com 8 horas de trabalho semanal. As atividades foram iniciadas no início de outubro de 2018 e tem previsão de encerrar em janeiro de 2019. Até o término deste documento 8 *Sprints* foram executadas (ver [Seção 5.1](#)), das quais foram desenvolvidas as primeiras 10 *User Stories* listadas na [Seção 4.1](#). São estimadas 135 horas totais para o desenvolvimento, contabilizados como 135 "pontos de desenvolvimento". A [Figura 19](#) contem o gráfico *burndown*, com a linha vermelha representando a distribuição ideal dos "pontos de desenvolvimento" estimados no período previsto, e a linha azul representando os pontos desenvolvidos pelo aluno. Em um cenário ideal as duas linhas seguem juntas ou a azul fica abaixo da vermelha.



Figura 19 – Gráfico Burndown com a Sprint 8 concluída

6 CONSIDERAÇÕES FINAIS

Este trabalho discorre brevemente sobre o problema de desperdício de alimentos e suas consequências e propõe o desenvolvimento do SABRE, uma ferramenta para auxiliar o consumidor à usar os ingredientes que já possui em casa, reduzindo desperdícios. Também apresenta alguns sistemas similares, mas que não possuem esta capacidade de exibir receitas possíveis de fazer de acordo com um conjunto de ingredientes informados.

O maior desafio para este projeto é definir o algoritmo que irá procurar as receitas. Este algoritmo é crucial para o desenvolvimento e tem o potencial de aumentar consideravelmente o processamento nos servidores e, portanto, deve ser analisado minuciosamente.

O resultado desejado - o desenvolvimento do sistema - é esperado que seja atingido seguindo os métodos do [Capítulo 3](#), usando as tecnologias da [Seção 4.2](#) e respeitando os prazos da [Seção 4.5](#).

Referências

- AGILEALLIANCE. **What is Test Driven Development (TDD)?** 2015. Disponível em: <<https://www.agilealliance.org/glossary/tdd/>>. Acesso em: 26 de agosto de 2018. Citado na página 6.
- AGILEMODELING. User Stories: An Agile Introduction. 2013. Disponível em: <<http://www.agilemodeling.com/artifacts/userStory.htm>>. Citado na página 7.
- AXIOS. **Axios**. 2018. Disponível em: <<https://github.com/axios/axios/>>. Citado na página 10.
- CYBERCOOK. **CyberCook**. 2018. Disponível em: <<https://cybercook.uol.com.br/>>. Acesso em: 19 de agosto de 2018. Citado na página 4.
- ESLINT. **About - ESLint**. 2018. Disponível em: <<https://eslint.org/docs/about/>>. Citado na página 10.
- FAO. **Global food losses and food waste**. Rome: [s.n.], 2011. 37 p. Disponível em: <<http://www.fao.org/docrep/014/mb060e/mb060e.pdf>>. Acesso em: 19 de agosto de 2018. Citado na página 1.
- FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. Tese (PhD in Computer Science) — UNIVERSITY OF CALIFORNIA, 2000. Citado na página 8.
- FOWLER, M. The New Methodology. **martinfowler.com**, 2005. Disponível em: <<https://martinfowler.com/articles/newMethodology.html>>. Citado na página 7.
- G1. Pão mais antigo do mundo tem receita de 14 mil anos revelada por arqueólogos. 2018. Disponível em: <<https://g1.globo.com/olha-que-legal/noticia/pao-mais-antigo-do-mundo-tem-receita-de-14-mil-anos-revelada-por-arqueologos.ghtml>>. Acesso em: 19 de agosto de 2018. Citado na página 1.
- GIT. **Git**. 2018. Disponível em: <<https://git-scm.com/>>. Citado na página 11.
- GITHUB. **The world's leading software development platform · GitHub**. 2018. Disponível em: <<https://github.com/>>. Citado na página 11.
- GSHOW. **Receitas Gshow**. 2018. Disponível em: <<https://gshow.globo.com/receitas-gshow/>>. Acesso em: 19 de agosto de 2018. Citado na página 5.
- JSON.ORG. **JSON**. 2018. Disponível em: <www.json.org/>. Citado na página 8.
- KARMA. **Karma - How It Works**. 2018. Disponível em: <<https://karma-runner.github.io/3.0/intro/how-it-works.html>>. Citado na página 10.
- LIPINSKI brian et al. Reducing food loss and waste. 2013. Disponível em: <<http://www.wri.org/publication/reducing-food-loss-and-waste>>. Acesso em: 19 de agosto de 2018. Citado na página 1.
- MOZILLA. **About JavaScript**. 2018. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript/>. Citado na página 10.

- PALMIRINHA, V. **Vovó Palmirinha**. 2018. Disponível em: <<https://www.vovopalmirinha.com.br/>>. Acesso em: 19 de agosto de 2018. Citado na página 4.
- PGSEARCH. **pg_search**. 2018. Disponível em: <https://github.com/Casecommons/pg_search/>. Citado na página 9.
- POSTGRESQL. **PostgreSQL: About**. 2018. Disponível em: <<https://www.postgresql.org/about/>>. Citado na página 9.
- RAILSGUIDES. **Getting Started with Rails**. 2018. Disponível em: <https://guides.rubyonrails.org/getting_started.html#creating-articles/>. Citado na página 9.
- RSPEC. **RSpec**. 2018. Disponível em: <<http://rspec.info/>>. Citado na página 9.
- RUBOCOP. **RuboCop**. 2018. Disponível em: <<https://github.com/rubocop-hq/rubocop/>>. Citado na página 9.
- RUBY. **About Ruby**. 2018. Disponível em: <<https://www.ruby-lang.org/en/about/>>. Citado na página 9.
- SCHWABER, K.; SUTHERLAND, J. The scrum guide. 2017. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>>. Acesso em: 26 de agosto de 2018. Citado na página 6.
- TRELLO. **Trello**. 2018. Disponível em: <<https://trello.com/>>. Citado na página 11.
- TUDOGOSTOSO. **TudoGostoso**. 2018. Disponível em: <<https://www.tudogostoso.com.br>>. Acesso em: 19 de agosto de 2018. Citado na página 3.
- VUE. **Introduction: What is Vue.js?** 2018. Disponível em: <<https://vuejs.org/v2/guide/>>. Citado na página 10.
- W3C. **HTML & CSS**. 2018. Disponível em: <<https://www.w3.org/standards/webdesign/htmlcss/>>. Citado na página 10.
- WRI. What's Food Loss and Waste Got to Do with Climate Change? A Lot, Actually. 2015. Disponível em: <<https://www.wri.org/blog/2015/12/whats-food-loss-and-waste-got-do-climate-change-lot-actually>>. Acesso em: 19 de agosto de 2018. Citado na página 1.