**APLICATIVO DE BUSCA DE RECEITAS**

***RangoJá-App***

|  |  |
| --- | --- |
|  | **Documento de especificação de requisitos do sistema**  Este documento contém a especificação dos requisitos para o sistema  **Rangojá**  **Andressa Zolet, Marcelo Menezes, Luiz Martins, Patricia Conceição e Lucas Camargo.**  Apresentando a situação atual e a pretendida, necessária ao projeto do sistema. |

**Versionamento do documento**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Data** | **Responsável** | **Observações** |
| 1.3 | 16/10/2019 | Lucas Camargo | Adicionado Diagramas de Sequência para users stories. |
| 1.2 | 16/10/2019 | Marcelo Menezes | Atualização do diagrama de classes. |
| 1.1 | 25/09/2019 | Marcelo Menezes | Protótipos, diagrama de atividades e diagrama de classes inicial. |
| 1.0 | 04/09/2019 | Andressa Zolet | Primeira versão publicada do documento com visão geral do sistema |

**SUMÁRIO**

[1. VISÃO DE NEGÓCIO 4](#_1t3h5sf)

[1.1 Escopo do sistema 4](#_4d34og8)

[1.2 Objetivo do sistema 4](#_2s8eyo1)

[1.3 Diferencial do Sistema 4](#_17dp8vu)

[1.4 Público alvo](#_3rdcrjn) 5

[1.5 Viabilidade](#_26in1rg) 5

[2. ESPECIFICAÇÃO DE REQUISITOS](#_lnxbz9) 6

[2.1 Requisitos funcionais](#_35nkun2) 6

[2.2 Requisitos não-funcionais](#_1ksv4uv) 6

[3. Use Stories](#_44sinio) 7

4. Protótipo 7

5. Diagrama de Atividades 9

6. Diagrama de Classes 11

6.1 Acesso ao Banco de Dados 11

6.2 Modelos 12

6.3 Views e Views-Models 13

7. Diagrama de Sequência 16

7.1 Estudante Atarefado 16

1. **VISÃO DE NEGÓCIO**

1.1 Escopo do sistema

Este projeto abrange todas as receitas culinárias que possam vir a ser identificadas através de pesquisas baseadas dentro do aplicativo. Para o uso do aplicativos foi previsto dois tipos de usuários.

O primeiro perfil de usuário será para pesquisas de receitas, onde o mesmo encontrará as receitas desejadas através de preenchimento de campos e ou marcações para filtros informados pelo aplicativo.

O segundo perfil de usuário será para a inserção de receitas, onde efetuando o cadastro das mesmas acabará auxiliando no enriquecimento de conteúdo no aplicativo.

Filtros para tolerâncias ou preferências será disponibilizado no aplicativo para auxiliar a busca do usuário. Estes mesmos filtros poderão ser selecionados/ cadastrados pelo segundo perfil de usuário que efetuará a marcação juntamente com o cadastro da receita desejada.

1.2 Objetivo do sistema

Através de pesquisas realizadas dentro do aplicativo RangoJÁ, os usuários poderão localizar receitas culinárias inserindo apenas ingredientes que possuem disponíveis em casa. A proposta do aplicativo é auxiliar ao usuário final a encontrar receitas práticas e rápidas sempre que necessário.  
Estes mesmos usuários poderão auxiliar os demais com informações de novas receitas sempre que desejado.

1.3 Diferencial do Sistema

O diferencial do aplicativo está ligada diretamente com o interesse da proposta. Hoje existem diversos canais de exibição, compartilhamento, leituras e afins onde atendem apenas à necessidades mais específicas de receitas e informações mais complexas, onde o usuário muitas vezes necessariamente precisa ter uma lista de ingredientes completos para finalizar a criação de qualquer pesquisa que lhe provem.   
Com o aplicativo RangoJÁ, este mesmo usuário poderá selecionar apenas os ingredientes que estão disponíveis em sua despensa e receber informações de centenas de usuários que compartilharam receitas práticas e rápidas através do aplicativo, facilitando e minimizando as dificuldades de uma receita.

1.4 Público Alvo

O objetivo do público alvo do aplicativo serão usuários, Jovens, trabalham ou estudam e que querem agilidade na hora de hora preparar a refeição. Procuram soluções para preparar a refeição

Como um(a) estudante, atarefado, gostaria de encontrar receitas para preparar uma refeição rapidamente, somente com os ingredientes disponíveis; (inserir ingredientes para sugestão de receitas pelo sistema)

Um segundo exemplo seria como um(a) estudante de gastronomia, gostaria de cadastrar receitas que conheço para compartilhar com outros usuários. (permitir cadastrar novas receitas)

1.5 Viabilidade

Do ponto de vista técnico, a equipe entende que é viável o desenvolvimento do projeto. Por escolha da equipe, será utilizado a plataforma Xamarin, de aplicativos híbridos, neste trabalho somente visando Android. Com Xamarin é possível desenvolver em C# utilizando todos os recursos existentes no .NET Framework, .NET Standard e .NET Core. Primeiramente foi cogitado utilizar o Firebase para agilidade, porém após pesquisa foi descoberto que o Firebase não possui SDK oficial para .NET. Existem apenas bibliotecas de terceiros que foram construídas em cima da API REST oficial do Firebase e em teste, não foram satisfatórias. A equipe então optou por utilizar serviços MYSQL remotos gratuitos.

**2. ESPECIFICAÇÃO DE REQUISITOS**

2.1 Requisitos funcionais

|  |
| --- |
| **RF1 Visualizar Receitas - Essencial** |
| Descrição: o usuário final poderá através do aplicativo visualizar as receitas inseridas |
| **RF2 Buscar receitas - Essencial** |
| Descrição: os usuários poderão efetuar buscas de receitas através de informações inseridas na busca. |
| **RF3 Inserir receitas - Essencial** |
| Descrição: os usuários poderão efetuar inserções de receitas através desta funcionalidade |
| **RF4 Campos obrigatórios - Essencial** |
| Descrição: os usuários deverão efetuar inserções de informações de receitas através de campos obrigatórios |
| **RF5 Filtro para opções veganas – Desejável** |
| Descrição: usuários poderão filtrar para receber somente resultados de opções veganas. |
| **RF6 Filtro para opções em micro-ondas/fogão/forno - Desejável** |
| Descrição: usuários poderão filtrar para receber somente resultados de opções que podem ser feitas em forno, fogão ou microondas. |

2.2 Requisitos não-funcionais

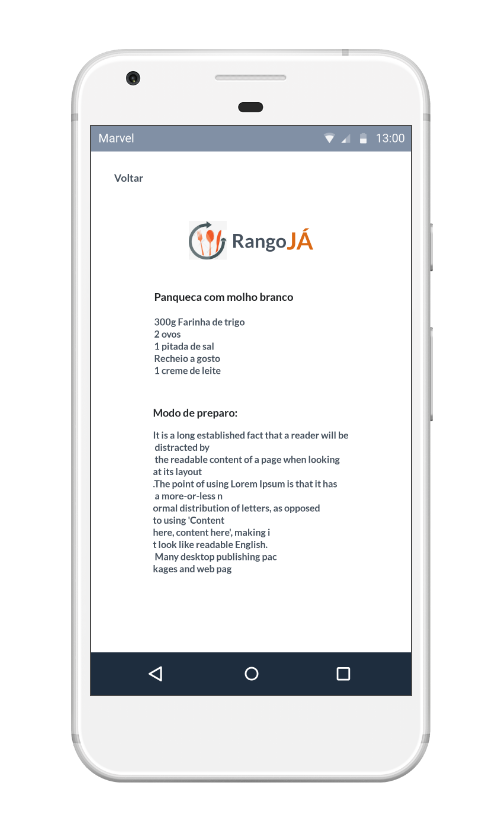
|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| **NFx.1 Sistema deve comunicar com banco MYSQL** | O sistema deve comunicar com banco MYSQL. |
| **NFx.2 Sistema deve ser implementado para sistemas Android** | Sistema deve ser implementado para sistemas Andoird. |

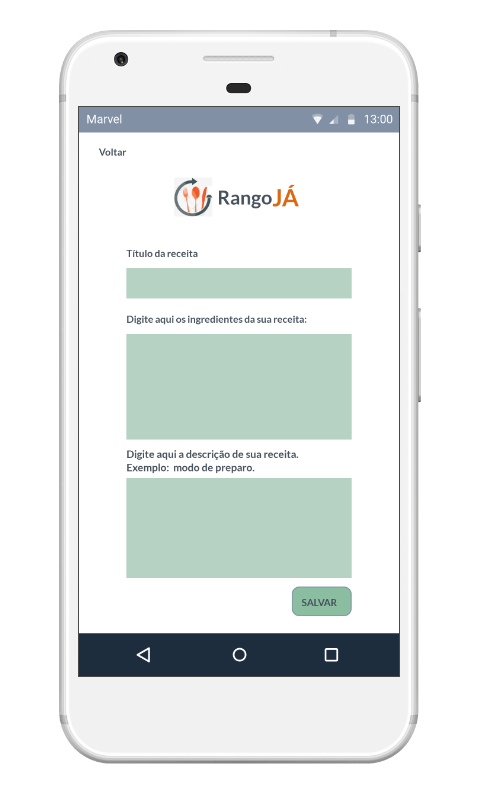
**3. Use Stories**

* Como um (a) estudante, atarefado, gostaria de encontrar receitas para preparar uma refeição rapidamente, somente com os ingredientes disponíveis; (inserir ingredientes para sugestão de receitas pelo sistema)
* Como um (a) estudante de gastronomia, gostaria de cadastrar receitas que conheço para compartilhar com outros usuários; (permitir cadastrar novas receitas);
* Como um (a) vegano (a), gostaria de encontrar receitas sem uso de produtos de origem animal para preparar uma refeição com ingredientes naturais disponíveis no momento; (filtro de opções veganas)
* Como um (a) publicitário (a), sem tempo, gostaria de apenas baixar e usar o app, para preparar a refeição o mais rápido possível. (Sem login)
* Como um (a) estudante que vive em uma república, gostaria de encontrar receitas específicas para serem preparados usando apenas micro-ondas. (Permitir opção de receitas para micro-ondas, forno ou fogão);

**4.Protótipo**





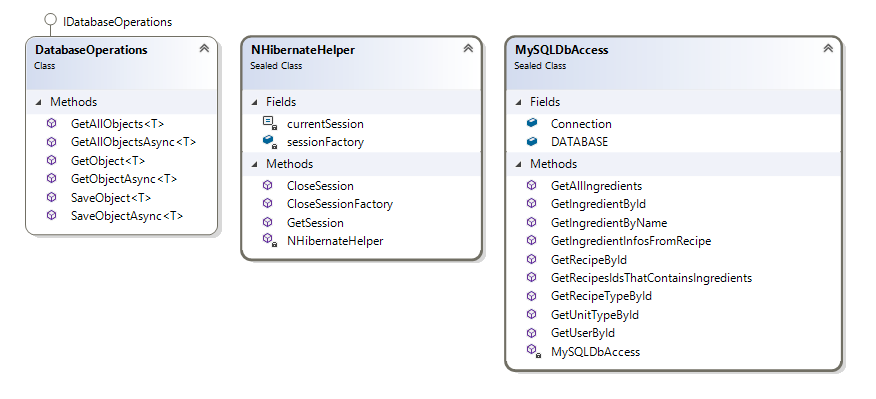
**5.Diagrama de Atividades**



**6. Diagrama de classes**

6.1 Acesso ao banco de dados

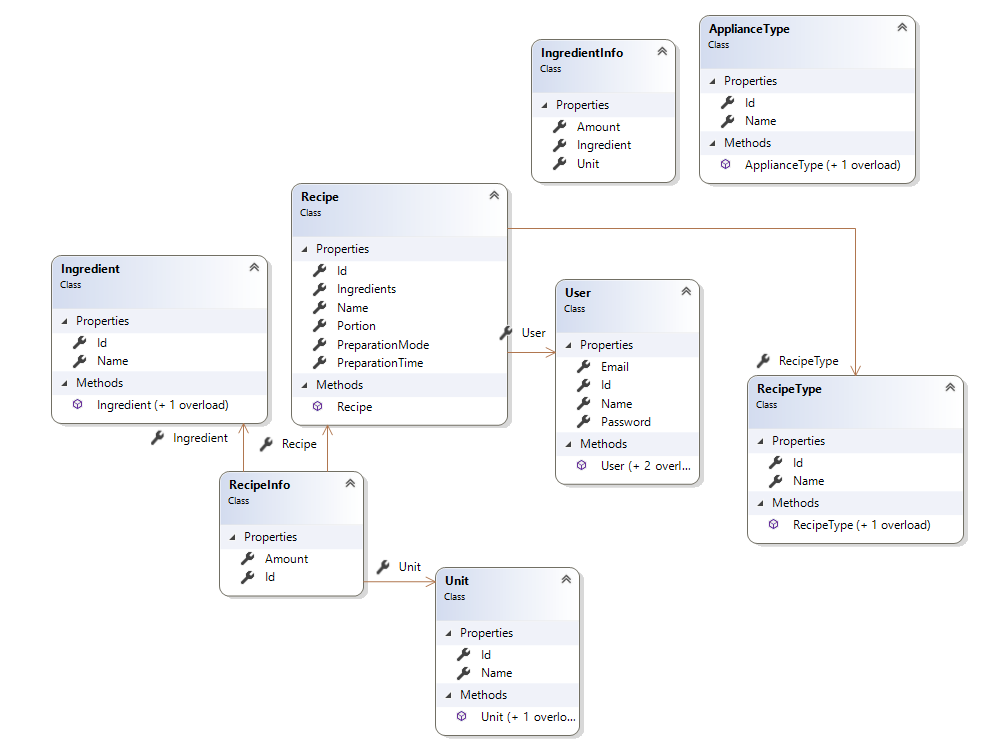
Para desenvolver a camada de banco de dados, foi criado um projeto do tipo “Biblioteca de classes” na solução do projeto. Este subprojeto fornece as classes e métodos necessários para o acesso ao banco de dados e os modelos do projeto. Quando este subprojeto é compilado, é gerado um arquivo com extensão “*.dll*” (*Dynamic Linked Library*) que é referenciado pelo projeto principal, que contém a regra de negócio e implementação das *views* e *view-models*. A seguir pode ser visto o diagrama de classes inicial desenvolvido:



* **NHibernateHelper**: Classe principal de controle do *NHibernate*. Esta classe fornece métodos estáticos que criam e controlam sessões de acesso;
* **IDatabaseOperations**: Interface desenvolvida que fornece o contrato de métodos básicos necessários para implementar o acesso;
* **DatabaseOperations**: Classe desenvolvida que utiliza os métodos estáticos da classe NHibernate e implementa a interface IDatabaseOperations. Esta classe é beneficiada pelo uso de *Generics,* que garantem que os métodos *Save*, *Update* e *Get* funcionem para objetos de qualquer tipo.
* **MySQLDbAccess**: Foi desenvolvido também esta classe estática com métodos para acesso e manipulação de dados do banco. Esta classe utiliza o pacote MySqlConnector do Bradley Grainger.

6.2 Modelos

Os modelos são necessários para que a regra de negócio funcione. Abaixo estão os modelos inicialmente pensados e desenvolvido para atender a necessidade do projeto.

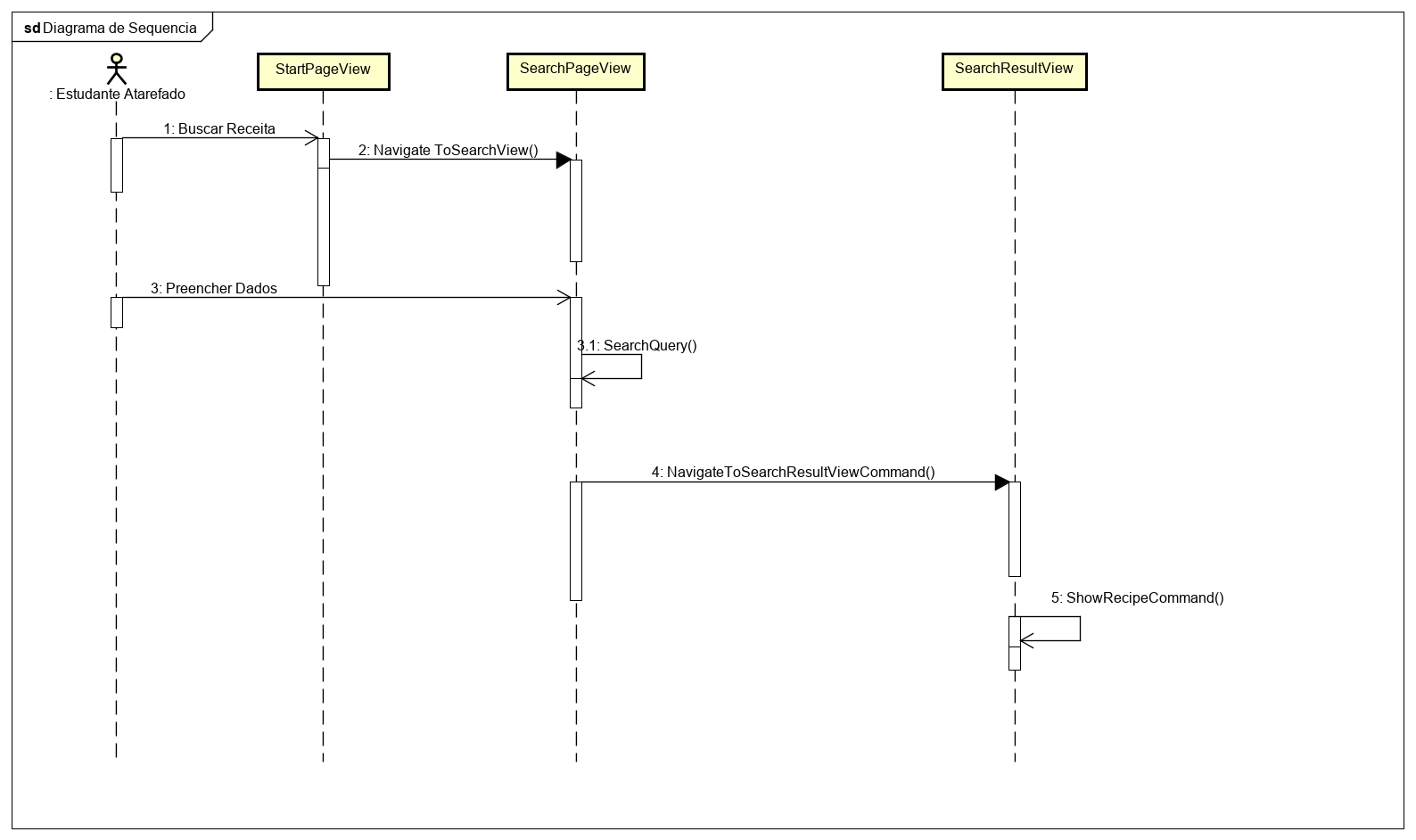


6.2 Views e View-Models

Cada tela proposta é uma classe. A parte visual da tela é desenvolvido através de XAML. O chamado *code-behind* da classe de tela, quando utilizando MVVM, não deve ser utilizado. Os controles devem ter seus valores ligados às propriedades da classe *view-model* respectiva da tela através de *Binding*. Portanto, para cada tela *view* existirá uma respectiva classe *view-model* que irá tratar as informações de entrada através dos controles visuais da *view*. Abaixo, estão as *views* e *view-models* inicialmente desenvolvidos:

**7. Diagrama de Sequencia**

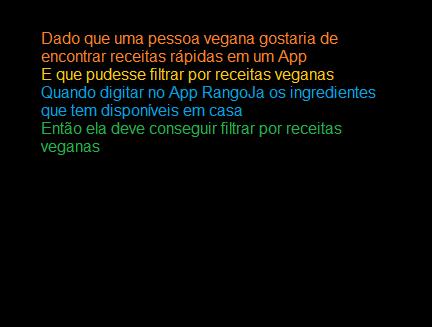
7.1 Estudante Atarefado

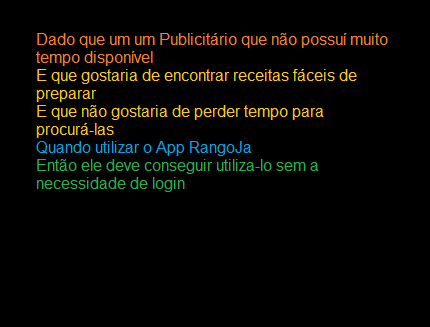


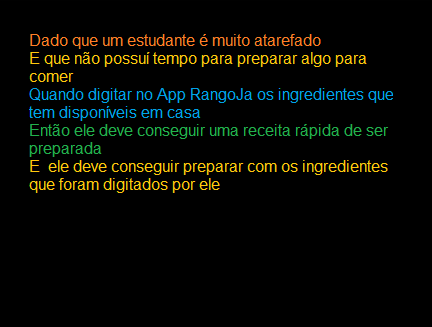
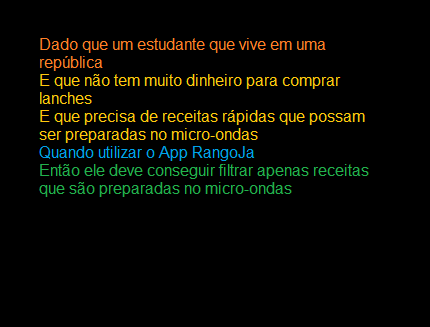
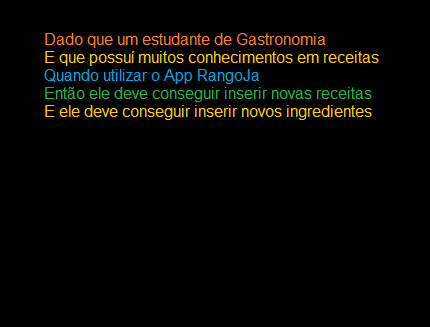
**8. Casos de Testes com BDD’S**

8.1 Cenários Positivos

BDD’s inspirados nos cartões do jogo criado por Ana Herman.







8.2 Cenários Negativos

