

# Fundamentos de Desenvolvimento de Software

## Trabalho Final

### TF

Lucas Piatnicki<sup>1</sup>

<sup>1</sup>Escola Politécnica – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)  
Av. Ipiranga, 6681 – Prédio 30 – Bloco C – Sala 101  
CEP 90.619-900 – Partenon – Porto Alegre – RS – Brasil

lucas.piatnicki@edu.pucrs.br

## 1. Introdução

O presente trabalho visa ao desenvolvimento de um sistema web, denominado *Recipes*, aplicando os conceitos de sistemas de gerência de configuração, sistemas de automação da compilação, teste unitário, padrões de projeto e arquitetura de software (arquitetura CLEAN). No entanto, os requisitos, objetivos, funcionalidades e mocks apresentados abaixo possuem caráter preliminar, e podem variar no decorrer do desenvolvimento do sistema.[Martin 2017]

## 2. Especificação de requisitos do sistema

Requisitos Funcionais

RF001 – O usuário pode criar uma nova receita.

RF002 – O usuário pode modificar uma receita existente.

RF003 – O sistema deve permitir ao usuário a criação de restrições em receitas novas e pré-existentes.

RF004 – O sistema deve permitir ao usuário a criação de notas extras.

RF005 – O sistema deve permitir ao usuário a criação de tags sobre as receitas.

RF006 – O sistema deve permitir ao usuário compartilhar as receitas com outros usuários.

RF007 – O sistema deve registrar as informações das receitas.

RF008 – O sistema deve permitir ao usuário copiar e colar o conteúdo das receitas.

RF009 – O usuário pode visualizar uma receita.

## 3. Objetivos do sistema

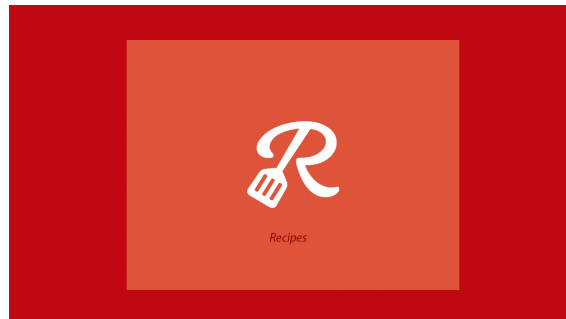
O produto será: um gerenciador/organizador de receitas de cozinha, gratuito, plataforma web, rodando em *localhost*, para fins de simplificação.

## 4. Relação de funcionalidades

O produto será utilizado por pessoas interessadas em expandir seus conhecimentos na cozinha, quer desejam encontrar novas receitas e aprimorar seus conhecimentos culinários. O produto armazenará e compartilhará receitas oriundas de diversas fontes.

Permitirá armazenar, salvar, compartilhar e comentar receitas, podendo separar, procurar por tags e aplicar restrições alimentícias.

## 5. Mocks das telas



(a) Tela de boas-vindas

(b) Tela de login

(c) Tela de criação de nova receita

(d) Tela de busca de receitas

Figura 1. Mocks do sistema

## **Referências**

Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Robert C. Martin Series. Prentice Hall, Boston, MA.