

## Open Rent

Andrey Calaca Resende <sup>1</sup> - 180062433

Felipe Luís Pinheiro <sup>2</sup> - 180052667

Wanderlan Alves de Jesus de Brito <sup>3</sup> - 160148782

William Coelho da Silva <sup>4</sup> - 180029274

<sup>1</sup><https://github.com/andreyresende>

<sup>2</sup><https://github.com/flpinheiro>

<sup>3</sup><https://github.com/Wander-lan>

<sup>4</sup><https://github.com/Williamcs1400>



# Contents

<b>1</b>	<b>Project Plan</b>	<b>5</b>
<b>2</b>	<b>Project Defined Process</b>	<b>7</b>
<b>3</b>	<b>Iteration Plan</b>	<b>11</b>
3.1	Sprint 1 . . . . .	12
3.2	Sprint 2 . . . . .	14
3.3	Sprint 3 . . . . .	16
3.4	Sprint 3 . . . . .	17
<b>4</b>	<b>Architecture Notebook</b>	<b>19</b>
<b>5</b>	<b>Use Case</b>	<b>21</b>
5.1	Autenticar Usuário . . . . .	23
5.2	listar imóveis disponíveis . . . . .	24
5.3	apresentar dados de imóvel . . . . .	25
5.4	descadastrar conta . . . . .	26
5.5	editar dados da conta . . . . .	27
5.6	Autenticar usuário . . . . .	28
5.7	Descadastrar Imóvel . . . . .	29
5.8	editar dados de imóvel . . . . .	30
5.9	cadastrar proposta de aluguel . . . . .	32
5.10	listar as propostas de aluguel . . . . .	33
5.11	apresentar dados de proposta de aluguel . . . . .	34
5.12	descadastrar proposta de aluguel . . . . .	35
<b>6</b>	<b>Smoke test</b>	<b>37</b>
	<b>Glossary</b>	<b>43</b>



# Chapter 1

## Project Plan

### Introduction

This plan consist of the entire project plan to construct all the Group work home of the discipline software engineer proposed by Fernando Antonio De Araujo Chacon from computer Science Department of University of Brasília (UnB).

This project is Licenced by Apache License.

### Project organization

See also <https://github.com/flpinheiro/ProjetoES>

This work is divided into the following content areas:

**Project Manager** Felipe Luís Pinheiro

**analyst** Wanderlan Alves de Jesus Brito

**architect** William Coelho da Silva

**Tester** Andrey Calaça Resende

### Project practices and measurements

The OpenUP component team will use OpenUP practices adapted to address the fact that we are doing content development rather than coding. Key artifacts include: Project defined process, project plan, iteration plan, tools, glossary, vision, system-wide requirement, usa-case model, use case,architecture notebook, user interface project, database physical project,infrastructure, test case. Progress is tracked using two primary measurements using a point system. It is estimated that 1 point represents 2h of work:

- Project backlog: The project backlog shows progress relative to overall work to be done within the project.
- Iteration backlog: The iteration backlog shows progress relative to work intended for the current iteration.

## Project milestone and objectives

Iteration	Primary objectives	milestone	Target velocity
I1	Objectives 1. Project Plan 2. Iteration Plan 1	25/02/2021 to 04/03/2021	7
I2	Objectives 1. Iteration Plan 2 2. Use-case Model 3. Architecture Notebook 4. Smoke Test 5. Glossary	05/03/2021 to 12/03/2021	7
I3	Objectives 1. project defined process 2. Documento de visão 3. system-wide requirements 4. Descrições detalhadas dos casos de uso	25/03/2021 to 02/04/2021	7
I2	Objectives 1. Lista de ferramentas (tools) 2. Projeto de interface com o usuário 3. Projeto físico de banco de dados 4. Descrição da infraestrutura de implantação 5. Protótipo de banco de dados	09/04/2021 to 23/04/2021	7

## Deployment

## Lessons learned

## Chapter 2

# Project Defined Process

### Introduction

#### Purpose

Este documento descreve o processo seguido pelo projeto do OpenRent.

#### Definitions, Acronyms, and Abbreviations

Veja o Glossario para uma lista completa dos termos usados nesse projeto.

#### Overview

O projeto do OpenRent será disciplinado mediante organização de metodologia ágil Scrum.

A equipe de Desenvolvimento está dividida em desenvolvedores, arquitetos, testers e analistas de requisitos.

#### Lifecycle Model

O Desenvolvimento será feito mediante modelo iterativo, que pode ser visto na imagem 2.1.

Onde início marca o início de uma nova Sprint, prazo médio de 15 dias, sendo a sprint desenvolvida em 4 fases:

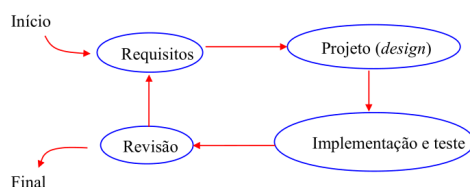


Figure 2.1: Modelo do ciclo de vida iterativo

- Requisitos
- Design e arquitetura
- Implementação
- Revisão e teste

onde essas 4 fases ocorrem de modo paralelo com cada equipe específica desenvolvendo a sua tarefa, de modo que um ciclo de Desenvolvimento completo e composto de 4 sprint sucessivas, sendo que ao final do ciclo completo se espera que seja colocado em produção a parte desenvolvida durante a sprint.

## Workflow

### Fase de Requisito

Desenvolvido pela equipe de analistas de requisitos em conjunto com o cliente, verificando-se todos os requisitos e problemas a serem resolvidos durante a sprint.

Nesta fase é desenvolvida protótipos de telas, verificado a interação do usuário.

Também é descrito com cuidado todos os casos de uso e histórias de usuário.

### Fase de Design e arquitetura

Desenvolvido pela equipe de arquitetura, a partir das especificações da equipe de requisitos.

Nesta fase é desenvolvida os diagramas de classe, diagramas de fluxo e pseudo código de regras de negócios específicas a serem desenvolvidas, preferencialmente demonstrando a completude e corretude do código quando possível e necessário.

### Fase de Implementação

Fase realizada pela equipe de Desenvolvimento frontend e backend em conjunto.

Nesta fase é implementado os diagramas de classe e de fluxo Desenvolvido pela equipe de arquitetura e também é implementado em código os pseudo código de negócios.



## **Fase de Revisão e teste**

Fase realizada pela equipe de teste, documentação e revisão.

Nesta fase é realizado teste intensos de cada aspecto do código desenvolvido pela equipe de desenvolvimento.

É implementado testes automatizados para serem executados durante as fases seguintes de desenvolvimento e implementação.

É realizado a documentação do sistema, tal como manual de usuário.

Por fim é realizado a revisão do sistema e caso seja aprovado deve ser colocado em produção, e o que não foi aprovado deve ser enviado reparado na próxima sprint de desenvolvimento.



## Chapter 3

# Iteration Plan

### 3.1 Sprint 1

#### Key Milestone

Milestone	Date
Iteration start	25/02/2021
Iteration stop	04/03/2021

#### High-level objectives

- Complete Project Plan
- Complete First Iteration Plan
- Construct Jira Project Board
- Implement Git Repository Basic Structure

#### Work Item assignments

The following Work Items will be addressed in this iteration:

Name	Priority	Size estimate (points)	State	Reference material	Target iteration	Assigned to	Hours worked	Estimate of hours remaining
Project Plan	1	2	Complete		1	Felipe	4	0
Iteration Plan week 1	1	2	Complete		1	Felipe	4	0
Jira Board	1	2	Complete		1	Felipe	4	0
Git Repository	1	2	Complete		1	Felipe	4	0

#### Issues

Issue	Status	Notes

#### Evaluation criteria

- Project Plan is complete
- First Iteration Plan is complete
- Jira board is complete
- Git Repository Basic Structure is complete

#### Assessment

Assessment target	Project Plan
Assessment Date	25/02/2021
Participants	Felipe, Wanderlan, William, Andrey
Project Status	Green
Assessment target	Git Repository Basic Structure
Assessment Date	25/02/2021
Participants	Felipe
Project Status	Green

Assessment target	Jira Board
Assessment Date	25/02/2021
Participants	Felipe
Project Status	Green

**Assessment against objectives**

Todas as tarefas especificadas foram realizadas conforme esperado.

**Work Items: Planned compared to actually completed**

Todas as tarefas foram realizadas durante a sprint dentro do prazo estimado.

**Assessment against Evaluation Criteria Test results**

O plano de projeto e o plano de iteração foram completamente realizados.

**Other concerns and deviations**

Não houve outras avaliações a serem realizadas.

## 3.2 Sprint 2

### Key Milestone

Milestone	Date
Iteration start	05/03/2021
Iteration stop	17/03/2021

### High-level objectives

- Delivery use-case model
- Delivery Smoke test
- Delivery architecture notebook
- Delivery Glossary

### Work Item assignments

The following Work Items will be addressed in this iteration:

Name	Priority	Size estimate (points)	State	Reference material	Target iteration	Assigned to	Hours worked	Estimate of hours remaining
Use Case model	1	4	On Work		2	Wanderlan	4	4
Smoke Test	1	4	Complete		2	Andrey	8	0
Architecture Notebook	1	4	Complete		2	Willian	8	0
Glossary	1	4	Complete		2	Felipe	8	0

### Issues

Issue	Status	Notes

### Evaluation criteria

#### Assessment

Assessment target	Use case Model
Assessment Date	05/03/2021
Participants	Wanderlan
Project Status	Green

Assessment target	Smoke Test
Assessment Date	05/03/2021
Participants	Andrey
Project Status	Green

Assessment target	Architecture Notebook
Assessment Date	05/03/2021
Participants	Willian
Project Status	Green

Assessment target	Glossary
Assessment Date	05/03/2021
Participants	Felipe
Project Status	Green

**Assessment against objectives**

Todos os itens da sprint foram entregues dentro do prazo, conforme o especificado.

**Work Items: Planned compared to actually completed**

Todos os itens da sprint foram entregues dentro do prazo, conforme o especificado.

**Assessment against Evaluation Criteria Test results****Other concerns and deviations**

Nada a reportar.

### 3.3 Sprint 3

#### Key Milestone

Milestone	Date
Iteration start	25/03/2021
Iteration stop	02/04/2021

#### High-level objectives

- project defined process
- Documento de visão (vision)
- system-wide requirements
- Descrições detalhadas dos casos de uso (use case)

#### Work Item assignments

The following Work Items will be addressed in this iteration:

Name	Priority	Size estimate (points)	State	Reference material	Target iteration	Assigned to	Hours worked	Estimate of hours remaining
project defined process	1	4	done		3	Felipe	8	0
Documento de visão (vision)	1	4	undone		3	Andrey	0	8
system-wide requirements	1	4	undone		3	William	0	8
Descrições detalhadas dos casos de uso (use case)	1	4	undone		3	Wanderlan	0	8

#### Issues

Issue	Status	Notes

#### Evaluation criteria

- project defined process

#### Assessment

Assessment target	project defined process
Assessment Date	02/04/2021
Participants	Felipe
Project Status	Green

#### Assessment against objectives

#### Work Items: Planned compared to actually completed

#### Assessment against Evaluation Criteria Test results

#### Other concerns and deviations



## 3.4 Sprint 3

### Key Milestone

Milestone	Date
Iteration start	09/04/2021
Iteration stop	23/04/2021

### High-level objectives

- Lista de ferramentas (tools)
- Projeto de interface com o usuário
- Projeto físico de banco de dados
- Protótipo de banco de dados

### Work Item assignments

The following Work Items will be addressed in this iteration:

Name	Priority	Size estimate (points)	State	Reference material	Target iteration	Assigned to	Hours worked	Estimate of hours remaining
Lista de ferramentas (tools)	1	4	undone		4	Felipe	8	0
Projeto de interface com o usuário	1	undone		4		Felipe	8	0
Projeto físico de banco de dados	1	undone		4		Wanderlan	8	0
Descrição da infraestrutura de implantação	1	undone		4		William	8	0
Protótipo de banco de dados	1	undone		4		Andrey	8	0

### Issues

Issue	Status	Notes

### Evaluation criteria

- project defined process

### Assessment

#### Assessment against objectives

#### Work Items: Planned compared to actually completed

#### Assessment against Evaluation Criteria Test results

#### Other concerns and deviations



## Chapter 4

# Architecture Notebook

### Purpose

Este processo de elaboração de software tem como filosofia a assertividade, velocidade, a boa qualidade, cumprimento de prazos estipulados, cumprimento integral das atividades delegadas para os colaboradores, além da eficiência e eficácia.

Foi decidido separar todo o desenvolvimento para os quatro colaboradores deste projeto, cada um com funções definidas, há um gerente de projeto, um arquiteto de software, um analista e um testador, com algumas restrições, todos devem se ajudar de acordo com o possível, porém sem que isso interfira no seu trabalho e sem tomar para si as tarefas de terceiros, buscando assim a maior velocidade e eficiência possível ao projeto a ser desenvolvido.

Com isso, a ferramenta a ser desenvolvida tem o papel de facilitar o uso de anúncio de aluguéis, negociações e cobranças, com a apresentação de propostas e descrição dos imóveis.

### Architectural goals and philosophy

O objetivo é construir um software que preste suporte ao aluguel de imóveis, sendo que proprietários podem anunciar, com várias informações sobre o objeto em questão, o interessado pode fazer propostas e conversar diretamente com o proprietário. Este dois precisando ser autenticados para tais funções, contendo dados pessoais para a sua precisa identificação em caso de imprevistos.

O software precisa ser capaz de entregar uma boa performance em qualquer dispositivo a ser utilizado, seja um computador potente ou fraco, uma vez que não é possível cobrar que o proprietário ou interessado tenha um bom hardware para executar o software, apenas requisitos mínimos, também deve ter um visual compreensivo, fácil de ser usado, moderno e leve, as-

sim evitando engargalos à aplicação que poderiam ser evitados. É necessário também que o software seja capaz o suficiente para comportar futuras atualizações pedidas pelos usuários do serviços e que façam sentido a proposta da aplicação.

Um problema que pode se tornar crítico é o de um proprietário ter várias propostas a se analisar e também muitos imóveis já alugados, é preciso fazer com que seja facilitada a visualização dos imóveis de maneira sucinta, para que possa se identificar de pronto de qual imóvel se trata e qual a condição dele.

## Architectural Mechanisms

É necessário também a identificação de mecanismos arquitetônicos, que nada mais são que soluções comuns para problemas comuns.

1 - Uma boa listagem de imóveis.

É necessário que os imóveis listados sejam adequados para os interessados, com base na sua pretensão de pagamento, localização do interessado e do imóvel, se tem estrutura para crianças ou para animais domésticos, no geral, suporte a filtros pré-aplicados.

2 - Visualização do preço do aluguel

Omitir do interessado o preço mínimo do aluguel informado pelo proprietário, e exibir apenas o desejado, se não há motivos para o interessado oferecer mais do que o mínimo, e então, assim, o proprietário só teria propostas com o valor mínimo, tornando a negociação bastante prejudicada.

## Chapter 5

# Use Case

### Overview

O sistema de software a ser desenvolvido tem o objetivo de prover suporte ao aluguel de imóveis.

Por meio desse sistema, proprietários podem anunciar imóveis para aluguel e os interessados podem apresentar propostas de aluguel.

Pode ser anunciado apartamento, casa ou quarto.

### Use-Case Diagram

The figure below shows the use-case diagram.

### Actors

#### Proprietários

Representa os usuários que desejam anunciar imóveis para aluguel.

#### Interessados

Representa os usuários que desejam alugar os imóveis oferecidos pelos proprietários.

### Use Cases

#### Autenticar usuário

Descreve como os usuários devem se autenticar para utilizar todas as funcionalidades do software.

**Listar imóveis disponíveis**

Descreve como os interessados veem a lista de imóveis disponíveis para suas propostas.

**Apresentar dados de imóvel**

Descreve como os interessados veem os dados a respeito do imóvel desejado.

**Descadastrar conta**

Descreve como os usuários podem descadastrar uma conta autenticada.

**Editar dados da conta**

Descreve como os usuários podem alterar os dados da conta autenticada.

**Cadastrar imóvel**

Descreve como os proprietários podem cadastrar imóveis para serem anunciados.

**Descadastrar imóvel**

Descreve como os proprietários podem descadastrar imóveis que foram cadastrados.

**Editar dados de imóvel**

Descreve como os proprietários podem alterar os dados dos imóveis cadastrados.

**Cadastrar proposta de aluguel**

Descreve como os interessados podem cadastrar propostas de aluguel.

**Listar as propostas de aluguel**

Descreve como os interessados podem visualizar sua lista de propostas de aluguel cadastradas.

**Apresentar dados de proposta de aluguel**

Descreve como os interessados podem visualizar os dados de suas propostas de aluguel cadastradas.

**Descadastrar proposta de aluguel**

Descreve como os interessados podem descadastrar suas propostas de aluguel cadastradas.

**5.1 Autenticar Usuário****Brief Description**

Este caso de uso descreve como os usuários devem se autenticar para utilizar todas as funcionalidades do software.

**Actors**

- Interessados
- Proprietários

**Preconditions**

O usuário deve ter uma conta cadastrada..

**Basic Flow of Events**

1. O usuário seleciona a opção referente à autenticação de usuário.
2. O usuário fornece seu endereço de e-mail.
3. O usuário digita sua senha.

**Alternative Flows****E-mail inválido**

O sistema verifica que o endereço de e-mail fornecido pelo usuário não é válido e informa o usuário sobre o problema, permitindo que ele digite o endereço de e-mail novamente.

**Senha inválida**

O sistema informa ao usuário que a senha fornecida não corresponde com a que foi criada no cadastro da conta, permitindo que ele informe a senha novamente.

**Post-conditions****Usuário autenticado com sucesso**

Se o procedimento de autenticação ocorrer corretamente, o sistema informa ao usuário que sua conta está autenticada e lhe-sugere as possíveis opções referentes ao aluguel de imóveis.

**Falha ao autenticar usuário**

Caso algum erro inesperado aconteça o sistema reinicia a operação de autenticação de usuários..

**Special Requirements**

O sistema deve manter o histórico de atividades dos usuários, incluindo o dia e o horário em que a autenticação foi realizada..

**5.2 listar imóveis disponíveis****Brief Description**

Este caso de uso descreve como os interessados veem a lista de imóveis disponíveis para suas propostas.

**Actors**

- Interessados
- Proprietários

**Preconditions**

O usuário deve ter acesso ao sistema.

**Basic Flow of Events**

1. O Interessado seleciona a opção referente à listar os imóveis disponíveis oferecidos pelos proprietários.
2. O sistema pede para que o Interessado escolha a classe do imóvel e a data inicial e final do período de disponibilidade.
3. O Interessado fornece as informações requeridas.
4. O sistema lista todas as possibilidades de imóveis com base nos requisitos do Interessado.

**Alternative Flows****Imóveis não encontrados**

O sistema avisa que não foram encontrados imóveis disponíveis com os requisitos informados pelo Interessado e retorna para a tela anterior.



**Post-conditions****Imóveis encontrados com sucesso**

O sistema permite que o interessado clique visualize os dados dos imóveis encontrados na lista.

**Special Requirements**

O sistema deve manter o histórico de busca de imóveis dos usuários.

**5.3 apresentar dados de imóvel****Brief Description**

Este caso de uso descreve como os interessados veem os dados a respeito do imóvel desejado.

**Actors**

- Interessados
- Proprietários

**Preconditions**

Devem ter sido encontrados imóveis disponíveis de acordo com os requisitos desejados pelo Interessado.

**Basic Flow of Events**

1. O Interessado seleciona o imóvel desejado entre os disponíveis na lista de imóveis encontrados.
2. O sistema fornece os dados que foram fornecidos pelo Proprietário do imóvel.
3. O Interessado confirma que tem interesse em alugar o determinado imóvel.
4. O sistema solicita que o Interessado esteja autenticado.
5. O Interessado se autentica e cadastra uma proposta de aluguel.

**Alternative Flows****O Interessado não tem interesse no imóvel**

O sistema permite que o Interessado retorne à página inicial. **O Interessado não é autenticado.** O sistema impede que o Interessado cadastre uma proposta de aluguel até que ele realize a autenticação.

### Post-conditions

**Proposta de aluguel cadastrada** A nova proposta de aluguel do Interessado é cadastrada e pode ser verificada pelo proprietário do imóvel.

### Special Requirements

O sistema deve manter o histórico de imóveis visualizados pelos Interessados.

## 5.4 descadastrar conta

### Brief Description

Este caso descreve como os usuários podem descadastrar uma conta autenticada.

### Actors

- Interessados
- Proprietários

### Preconditions

O usuário deve ter a conta autenticada no sistema.

### Basic Flow of Events

1. O usuário seleciona a opção referente a descadastrar sua conta.
2. O sistema pede para que o usuário informe seu endereço de e-mail e sua senha para confirmar a identidade do usuário.
3. O usuário informa os dados pedidos.
4. O sistema remove a autenticação da conta do usuário.
5. O sistema apaga os imóveis e as propostas cadastradas pelo usuário.
6. O usuário é levado de volta para a página inicial.

### Alternative Flows

#### E-mail ou senha incorretos

O sistema informa ao usuário que seus dados estão incorretos e não permite que a operação de descadastro aconteça até que os dados sejam inseridos corretamente.

#### Usuário não possui imóveis ou propostas de aluguel ativas

O sistema informa que não existem imóveis ou propostas de aluguel para serem removidos, e apenas remove a autenticação do usuário.

### Post-conditions

#### Conta descadastrada com sucesso

Se o usuário consegue descadastrar sua conta com sucesso o sistema o-redireciona para a página inicial.

#### Falha ao descadastrar conta

Caso algum erro inesperado aconteça, o sistema reinicia a operação de descadastro de conta.

### Special Requirements

O sistema deve manter o histórico de atividades dos usuários, incluindo a data e o horário em que eles realizaram o descadastramento de conta

## 5.5 editar dados da conta

### Brief Description

Este caso descreve como os usuários podem alterar os dados da conta autenticada.

### Actors

- Interessados
- Proprietários

### Preconditions

O usuário deve ter a conta autenticada no sistema.

### Basic Flow of Events

1. O usuário seleciona a opção referente a editar os dados da sua conta.
2. O sistema oferece as opções de edição para o nome, telefone, senha e endereço de e-mail.

3. O usuário escolhe a opção que deseja editar.
4. O sistema pede para que o usuário digite a alteração que deseja fazer.
5. O sistema envia um e-mail para o endereço de e-mail cadastrado até então pelo usuário para que ele confirme a operação.
6. O usuário realiza a confirmação por meio de seu endereço de e-mail.
7. O sistema atualiza e registra os novos dados do usuário.

### Alternative Flows

#### E-mail inválido

Caso o novo endereço de e-mail adicionado pelo usuário não seja válido, o sistema informa o erro ao usuário e pede para que ele digite um outro endereço de e-mail.

### Post-conditions

**Alterações realizadas com sucesso** Se o usuário consegue editar os dados da sua conta com sucesso o sistema o informa que os dados foram alterados e o redireciona para a página inicial.

**Falha ao editar os dados** Caso algum erro inesperado aconteça, o sistema reinicia a operação de edição de dados.

### Special Requirements

O sistema deve manter o histórico de atividades dos usuários, incluindo a data e o horário em que eles realizaram as alterações dos dados da conta

## 5.6 Autenticar usuário

### Brief Description

Este caso descreve como o proprietário deve prover os dados de seus imóveis que serão anunciados.

### Actors

- Proprietários

### Preconditions

O usuário deve ter a conta autenticada no sistema.

### Basic Flow of Events

1. O usuário seleciona a opção referente a cadastrar um imóvel.
2. O sistema pede para que sejam fornecidos os seguintes dados do imóvel: classe do imóvel (apartamento, casa ou quarto), descrição, endereço, número máximo de hóspedes, data inicial do período de disponibilidade, data final do período de disponibilidade e valor de diária mínimo.

### Alternative Flows

#### **usuário não autenticado**

O sistema oferece as opções de redirecionar o usuário para a página de autenticação ou para a página inicial.

#### **Endereço inválido**

O sistema informa que o endereço fornecido não é válido e pede para que o mesmo seja corrigido.

#### **Limite de imóveis cadastrados**

Cada usuário pode cadastrar apenas 5 imóveis, caso o usuário tente cadastrar um sexto imóvel o sistema informa que não é possível completar a ação e retorna o usuário para a página inicial.

### Post-conditions

#### **Imóvel cadastrado com sucesso**

Se o usuário consegue cadastrar o imóvel com sucesso o sistema o-redireciona para a página de consulta deste imóvel.

#### **Falha ao cadastrar imóvel**

Caso algum erro inesperado aconteça o sistema reinicia a operação de cadastro de imóveis.

### Special Requirements

O sistema deve manter o histórico de atividades dos usuários, incluindo o dia e o horário em que acessaram o sistema.

## 5.7 Descadastrar Imóvel

### Brief Description

Este caso de uso descreve como os proprietários podem descadastrar imóveis que foram cadastrados.

### Actors

- Proprietários

### Preconditions

O proprietário deve ter imóveis cadastrados.

### Basic Flow of Events

1. proprietário acessa a lista de imóveis cadastrados da sua conta.
2. Proprietário escolhe o imóvel que deseja descadastrar e seleciona a opção referente ao descadastramento.
3. O sistema pede para que o proprietário confirme sua senha para confirmar sua identidade.
4. O proprietário informa sua senha.
5. O sistema descadastra o respectivo imóvel.

### Alternative Flows

#### Senha inválida

O sistema informa ao usuário que a senha fornecida não corresponde com a que foi criada no cadastro da conta, permitindo que ele informe a senha novamente.

### Post-conditions

#### Imóvel descadastrado com sucesso

Se a ação ocorrer corretamente, o sistema informa ao usuário que o imóvel foi descadastrado e permite que ele retorne à lista de imóveis restantes.

#### Falha ao descadastrar imóvel

Caso algum erro inesperado aconteça o sistema reinicia a operação de descadastro de imóveis.

### Special Requirements

O sistema deve manter o histórico de atividades dos usuários, incluindo o dia e o horário em que o descadastro foi realizado

## 5.8 editar dados de imóvel

### Brief Description

Este caso de uso descreve como os proprietários podem alterar os dados dos imóveis cadastrados.

**Actors**

- **Proprietários**

**Preconditions**

O proprietário deve ter imóveis cadastrados.

**Basic Flow of Events**

1. O proprietário acessa a lista de imóveis cadastrados da sua conta.
2. O proprietário escolhe o imóvel do qual deseja editar os dados e seleciona a opção referente à edição de dados
3. O proprietário escolhe quais dados deseja alterar (classe do imóvel, descrição, endereço, número máximo de hóspedes, data inicial do período de disponibilidade, data final do período de disponibilidade e valor mínimo de diária).
4. O sistema pede para que o proprietário confirme sua senha para confirmar sua identidade.
5. O proprietário informa sua senha.
6. O sistema atualiza os novos dados do respectivo imóvel.

**Alternative Flows****Dados inválidos**

Caso algum dos novos dados informados não seja válido, o sistema informa o dado que está incorreto e pede para que o proprietário o-corrija.

**Senha inválida**

O sistema informa ao usuário que a senha fornecida não corresponde com a que foi criada no cadastro da conta, permitindo que ele informe a senha novamente.

**Post-conditions****Dados editados com sucesso**

Se a ação ocorrer corretamente, o sistema informa ao proprietário que os dados foram corretamente alterados e permite que ele retorne à lista de imóveis.

**Falha ao editar dados**

Caso algum erro inesperado aconteça o sistema informa ao proprietário que houve um erro e reinicia a operação de edição de dados.

### Special Requirements

O sistema deve manter o histórico de atividades dos usuários, incluindo o dia e o horário em que os dados foram alterados.

## 5.9 cadastrar proposta de aluguel

### Brief Description

Este caso de uso descreve como os interessados podem cadastrar propostas de aluguel.

### Actors

- Interessados

### Preconditions

O interessado deve ter uma conta cadastrada no sistema.

### Basic Flow of Events

1. O interessado busca um imóvel que se encaixe nas suas preferências.
2. O interessado seleciona a opção relacionada à cadastrar uma proposta de aluguel para o respectivo imóvel.
3. O sistema pede para que o interessado forneça os dados necessários para a realização da proposta (período do aluguel, número de hóspedes, valor mínimo do aluguel).
4. O interessado fornece os dados pedidos.
5. O sistema realiza o cadastro da proposta e envia uma notificação sobre o cadastro para o proprietário do imóvel.

### Alternative Flows

#### Período inválido

Caso o período informado pelo interessado não corresponda ao período permitido fornecido pelo proprietário, o sistema avisa ao interessado que ele deve inserir um novo período que seja válido.

#### Número de hóspedes inválido

Caso o número de hóspedes informado seja maior que o número máximo fornecido pelo proprietário, o sistema avisa ao interessado que ele deve inserir um número inferior.

#### Valor inválido



Caso o valor oferecido pelo interessado seja inferior ao requisitado pelo proprietário, o sistema informa ao interessado que ele deve escolher um valor que corresponda ao valor mínimo.

### **Post-conditions**

#### **Proposta realizada com sucesso**

Se a ação ocorrer corretamente, o sistema informa ao usuário que sua proposta de aluguel foi cadastrada e permite que ele retorne à lista de propostas alugadas.

#### **Falha ao cadastrar proposta de aluguel**

Caso algum erro inesperado aconteça o sistema reinicia a operação de cadastramento de proposta de aluguel.

### **Special Requirements**

O sistema deve manter o histórico de atividades dos usuários, incluindo o dia e o horário em que realizaram o cadastro de uma proposta de aluguel.

## **5.10 listar as propostas de aluguel**

### **Brief Description**

Este caso de uso descreve como os interessados podem visualizar sua lista de propostas de aluguel cadastradas

### **Actors**

- Interessados

### **Preconditions**

O interessado deve ter uma conta cadastrada no sistema.

### **Basic Flow of Events**

1. O interessado seleciona a opção relacionada à visualizar sua lista de propostas de aluguel.
2. O sistema mostra uma lista com todas as propostas de aluguel cadastradas pelo interessado.
3. O usuário pode selecionar as opções de visualizar os dados de alguma proposta ou a opção de descadastrar a determinada proposta.

### Alternative Flows

#### Nenhuma proposta cadastrada

Caso não haja nenhuma proposta cadastrada, o sistema informa ao interessado que ele não possui nenhuma proposta para ser visualizada.

### Post-conditions

#### Falha ao visualizar lista de propostas de aluguel

Caso algum erro inesperado aconteça, o sistema reinicia a operação de visualizar a lista de propostas de aluguel

### Special Requirements

O sistema deve manter o histórico de atividades dos usuários, incluindo as últimas propostas de aluguel visualizadas.

## 5.11 apresentar dados de proposta de aluguel

### Brief Description

Este caso de uso descreve como os interessados podem visualizar os dados de suas propostas de aluguel cadastradas.

### Actors

- Interessados

### Preconditions

O interessado deve ter alguma proposta de aluguel cadastrada no sistema.

### Basic Flow of Events

1. O interessado seleciona a opção relacionada à visualizar sua lista de propostas de aluguel.
2. O sistema mostra uma lista com todas as propostas de aluguel cadastradas pelo interessado.
3. O usuário seleciona a proposta ao qual deseja ver os dados.
4. O sistema mostra os seguintes dados sobre a respectiva proposta selecionada: valor à ser pago, período do aluguel e número de hóspedes.

### Alternative Flows

Sem alternative flows.

**Post-conditions****Falha ao visualizar os dados da proposta de aluguel**

Caso algum erro inesperado aconteça, o sistema reinicia a operação de visualizar os dados da proposta de aluguel.

**Special Requirements**

O sistema deve manter o histórico de atividades dos usuários, incluindo as últimas propostas de aluguel visualizadas.

## 5.12 descadastrar proposta de aluguel

**Brief Description**

Este caso de uso descreve como os interessados podem descadastrar suas propostas de aluguel cadastradas.

**Actors**

- Interessados

**Preconditions**

O interessado deve ter alguma proposta de aluguel cadastrada no sistema.

**Basic Flow of Events**

1. O interessado seleciona a opção relacionada à visualizar sua lista de propostas de aluguel.
2. O sistema mostra uma lista com todas as propostas de aluguel cadastradas pelo interessado.
3. O interessado seleciona a proposta ao qual deseja descadastrar.
4. O sistema requisita que o interessado forneça a senha de sua conta cadastrada para confirmar sua identidade.
5. O sistema confirma a senha do interessado e realiza o descadastro da respectiva proposta.

**Alternative Flows****Senha inválida**

Caso a senha fornecida seja inválida, o sistema pede para que o interessado forneça a senha novamente.

**Post-conditions****Proposta de aluguel descadastrada com sucesso**

Caso o procedimento ocorra corretamente, o sistema informa ao interessado que a proposta de aluguel foi descadastrada e permite que ele retorne à sua lista de propostas ainda cadastradas.

**Erro ao descadastrar proposta de aluguel**

Caso algum erro inesperado aconteça, o sistema reinicia a operação de descadastrar propostas de aluguel.

**Special Requirements**

O sistema deve manter o histórico de atividades dos usuários, incluindo a data e o horário em que os interessados descadastraram as suas propostas.

## Chapter 6

# Smoke test

The following tests are divided into 4 categories:

1. Log in system tests
2. Property tests
3. Bid tests
4. General Services

### 1.1 – Can create user

**Description** : A properly filled sign up form is submitted. It's expected that a new user's account gets created on the database.

**Pre-conditions** : There must be no other account with the same email on the database.

**Post-conditions** : All of the new account's information should be inserted on the database and the user should be able to log in with them.

**Data required** : Valid email, phone number, name and password.

### 1.2 – Can log in

**Description** : A properly filled sign in form is submitted. It's expected that the authenticated user can now access his account's information.

**Pre-conditions** : There must be an existing account on the database to log in.

**Post-conditions** : The user should be now authenticated and should be able to use the system's services.

**Data required** : Registered account with valid email, phone number, name and password.

### 1.3 – Can delete account

**Description** : An authenticated user tries to delete his account. It's expected that the user's information gets deleted from the database along with all associated bids and registered properties.

**Pre-conditions** : There must be an authenticated user and whose account information is properly registered with the bank

**Post-conditions** : There should be nothing left from that user on the database.

**Data required** : Registered account with valid email, phone number, name and password, there may be bids or properties associated with the account.

### 2.1 - Can register property

**Description** : An authenticated user submits a properly completed property registration form. A new property is created in the database associated with that user.

**Pre-conditions** : The user must be authenticated and must provide every information on the property registration form.

**Post-conditions** : A new property and its details should be created in the database.

**Data required** : Property class, description, address, maximum number of guests, start date of the availability period, end date of the availability period, and minimum daily rate.

### 2.2 – Can delete property

**Description** : An authenticated user tries to delete a property registered by him. That property and its associated bids should be deleted from the database.

**Pre-conditions** : The authenticated user must have a registered property.

**Post-conditions** : The property and its associated proposals should have been excluded from the database.

**Data required** : Previously registered property.

## 2.3 – Can edit property details

**Description** : An user tries to edit the details of one of its properties. The database should be updated with the new data.

**Pre-conditions** : The authenticated user must have a registered property.

**Post-conditions** : The property and its details should have been updated on the database.

**Data required** : Previously registered property.

## 2.4 – Can list associated bids

**Description** : An authenticated user tries to list every bid associated with his property.

**Pre-conditions** : The authenticated user must have a registered property and at least one associated bid.

**Post-conditions** : A list of bids should be displayed.

**Data required** : Previously registered property with at least one associated bid.

### 2.4.1 – Can see bids details

**Description** : An authenticated user tries to see the details of one bid associated with his property. A list with those details should be shown.

**Pre-conditions** : Test 2.4 must have been successful.

**Post-conditions** : The bid's details should be displayed.

**Data required** : Previously registered property with at least one associated bid.

## 3.1 – Can register proposal

**Description** : An authenticated user submits a properly completed proposal registration form associated to a property. A new proposal associated to that property should be created in the database.

**Pre-conditions** : The user must be authenticated and must provide every information on the proposal registration form.

**Post-conditions** : A new associated proposal should be created in the database.

**Data required** : Rental period start date, rental period end date, guests number, proposed daily rate.

## 3.2 – Can list proposals

**Description** : An authenticated user tries to list his registered proposals. A list of proposals should be displayed.

**Pre-conditions** : The authenticated user must have a registered proposal.

**Post-conditions** : A list of proposals should be displayed.

**Data required** : Previously registered proposal.

### 3.2.1 – Can see proposal details

**Description** : A authenticated user tries to see the details of one his registered proposals. A list with those details should be shown.

**Pre-conditions** : Test 3.2 must have been successful.

**Post-conditions** : A list with the proposal details should be displayed.

**Data required** : Previously registered proposal.

### 3.2.2 – Can delete proposal

**Description** : An authenticated user tries to delete one of his registered proposals. That proposal and its details should be excluded from the database.

**Pre-conditions** : The authenticated user must have a registered proposal.

**Post-conditions** : There should be no more data about that proposal in the database.

**Data required** : Previously registered proposal.

## 4.1 – Can list properties

**Description** : An user tries to list all properties. It's expected that a list of properties gets displayed.



**Pre-conditions** : There must be at least one property registered in the system.

**Post-conditions** : A list of properties is shown.

**Data required** : Previously registered properties.

#### **4.1.1 – Can see property details**

**Description** : An user tries to see a property details. A list of those details should be displayed.

**Pre-conditions** : Test 4.1 must have been successful.

**Post-conditions** : A list of details should be displayed.

**Data required** : Previously registered property.



# Glossary

**analyst** (1) member of the technical community who is skilled and trained to define problems and to analyze, develop, and express algorithms (ISO/IEC/IEEE 24765e:2015) Example: systems engineer, business analyst. 5

**architect** (1) person, team, or organization responsible for systems architecture (ISO/IEC/IEEE 24765e:2015). 5

**architecture** (1) [system] fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution (ISO/IEC/IEEE 12207:2017 Systems and software engineering–Software life cycle processes, 3.1.6) (ISO/IEC/IEEE 42010:2011 Systems and software engineering–Architecture description, 3.2) (ISO/IEC/IEEE 15288:2015 Systems and software engineering–System life cycle processes, 4.1.5) (ISO/IEC/IEEE 24748-1:2018 Systems and software engineering–Life cycle management–Part 1: Guidelines for life cycle management, 2.6) (2) set of rules to define the structure of a system and the interrelationships between its parts (ISO/IEC 10746-2:2009 Information technology – Open Distributed Processing – Reference Model: Foundations, 6.6) (3) fundamental concepts or properties of an entity in its environment and governing principles for the realization and evolution of this entity and its related life cycle processes (ISO/IEC/IEEE 42020:2019 Software, systems and enterprise – Architecture processes, 3.3) Note: Architectures can address a wide range of concerns, expressed, for example, through architecture views and models, as illustrated in the following examples associated with particular kinds of architectures such as security architecture, functional architecture, physical architecture, resilience architecture. See Also: component, module, subprogram, routine. 5

**backlog** (1) list of product requirements and deliverables not part of current work, to be prioritized and completed (ISO/IEC/IEEE 24765h:2019) (2) a set of software features awaiting development in a subsequent iteration (Software Extension to the PMBOK(R) Guide Fifth Edition)

(3) collection of agile features or stories of both functional and non-functional requirements that are typically sorted in an order based on value priority (ISO/IEC/IEEE 26515: 2018 Systems and software engineering: Developing information for users in an agile environment, 3.4) . 6

**infrastructure** (1) hardware and software environment to support computer system and software design, development, and modification (ISO/IEC/IEEE 12207:2017 Systems and software engineering–Software life cycle processes, 3.1.25) Syn: ecosystem. 5

**iteration** (1) process of performing a sequence of steps repeatedly (ISO/IEC/IEEE 24765:2017 Systems and software engineering–Vocabulary)  
 (2) a single execution of the sequence of steps in (1) (ISO/IEC/IEEE 24765:2017 Systems and software engineering–Vocabulary)  
 (3) short time frame in which a set of software features is developed, leading to a working product that can be demonstrated to stakeholders (ISO/IEC/IEEE 26515: 2018 Systems and software engineering: Developing information for users in an agile environment, 3.10) Note: One or more iterations comprise an instance. See Also: instance, invocation, mapping . 5

**milestone** (1) a list identifying all project milestones and normally indicates whether the milestone is mandatory or optional (A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide) – Sixth Edition). 6

**Project Manager** (1) the person assigned by the performing organization to lead the team that is responsible for achieving the project objectives (A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide) – Sixth Edition) (2) person with overall responsibility for the management and running of a project (ISO/IEC 26514:2008 Systems and software engineering–requirements for designers and developers of user documentation, 4.39) (3) stakeholder with overall responsibility for the planning, execution, and closure of a project (ISO/IEC/IEEE 24748-5:2017 Systems and software engineering–Life cycle management–Part 5: Software development planning, 3.10) Syn: PJM. 5

**project plan** document that describes the technical and management approach to be followed for a project (ISO/IEC/IEEE 24765:2017 Systems and software engineering–Vocabulary) . 5

**requirement** (1) statement that translates or expresses a need and its associated constraints and conditions (ISO/IEC/IEEE 12207:2017 Sys-

tems and software engineering–Software life cycle processes, 4.1.31) (ISO/IEC/IEEE 15288:2015 Systems and software engineering–System life cycle processes, 3.1.44) (2) condition or capability that must be met or possessed by a system, system component, product, or service to satisfy an agreement, standard, specification, or other formally imposed documents (IEEE 730-2014 IEEE Standard for Software Quality Assurance Processes, 3.2) (3) provision that conveys criteria to be fulfilled (ISO/IEC 14143-2:2011 Information technology – Software measurement – Functional size measurement – Part 2: Conformity evaluation of software size measurement methods to ISO/IEC 14143-1, 3.10) (4) a condition or capability that is necessary to be present in a product, service, or result to satisfy a business need (A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide) – Sixth Edition) (5) need or expectation that is stated, generally implied, or obligatory (ISO/IEC 19770-1:2017 Information technology – IT asset management – Part 1: IT asset management systems–Requirements, 3.47) Example: software component requirement Note: Requirements exist at different tiers and express the need in high-level form. A requirement is denoted by the word 'shall' and when used includes both the exclusive and applicable optional requirements. Requirements provide value when delivered, satisfied, or met. Requirements include the quantified and documented needs, wants, and expectations of the sponsor, customer, and other stakeholders. See Also: design requirement, functional requirement, implementation requirement, interface requirement, performance requirement, physical requirement. 5

**software** (1) computer programs, procedures and possibly associated documentation and data pertaining to the operation of a computer system (IEEE 828-2012 IEEE Standard for Configuration Management in Systems and Software Engineering, 2.1) (2) all or a part of the programs, procedures, rules, and associated documentation of an information processing system (IEEE 828-2012 IEEE Standard for Configuration Management in Systems and Software Engineering, 2.1) (ISO/IEC 19770-5:2015 Information technology–IT asset management–Overview and vocabulary, 3.34) (3) program or set of programs used to run a computer (ISO/IEC 26514:2008 Systems and software engineering–requirements for designers and developers of user documentation, 4.46) (4) all or part of the programs which process or support the processing of digital information (ISO/IEC 19770-1:2017 Information technology – IT asset management – Part 1: IT asset management systems–Requirements, 3.49) (5) part of a product that is the computer program or the set of computer programs (ISO/IEC/IEEE 26513:2017 Systems and software engineering–Requirements for testers and reviewers of information for users, 3.34) Note: Software excludes infor-

mation per se, such as the content of documents, audio and video recordings, graphics, and databases. Digital information which is managed by executable software (e.g. the content of documents and databases) is not considered software, even though program execution may depend on data values. Syn: SW See Also: application software.

5

**test case** (1) set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement (IEEE 1012-2016 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.3.31) (2) documentation specifying inputs, predicted results, and a set of execution conditions for a test item (IEEE 1012-2016 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1.31) (3) documentation specifying inputs, predicted results, and a set of execution conditions for a test item (ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) (4) set of test case preconditions, inputs (including actions, where applicable), and expected results, developed to drive the execution of a test item to meet test objectives, including correct implementation, error identification, checking quality, and other valued information (ISO/IEC/IEEE 29119-1:2013 Software and systems engineering-Software testing-Part 1: Concepts and definitions, 4.48) Note: A test case is the lowest level of test input (i.e. test cases are not made up of test cases) for the test subprocess for which it is intended.

5