A black background with green and brown text

Description automatically generated

Escola Superior de Tecnologia e Gestão  
Licenciatura em Engenharia Informática

Engenharia de Software

Plano de Desenvolvimento de Software Projeto Airbnb

**António Pedro Duarte Rosa, 24105**

**Rita Alexandra Lampreia Dias, 23240**

****

Beja, abril de 2025

Instituto Politécnico de Beja

Escola Superior de Tecnologia e Gestão  
Licenciatura em Engenharia Informática

Engenharia de Software

Plano de Desenvolvimento de Software Projeto Airbnb

**António Pedro Duarte Rosa, 24105**

**Rita Alexandra Lampreia Dias, 23240**

ORIENTAÇÃO

Professora Isabel Sofia Sousa Brito

Beja, abril de 2025

Índice

[Abstract 5](#_Toc198826977)

[1 Introdução 6](#_Toc198826978)

[2 Estrutura do Projeto 7](#_Toc198826979)

[2.1 Fase Preliminar – Planeamento e Gestão do Projeto 7](#_Toc198826980)

[2.1.1 Metodologia de desenvolvimento de Software 7](#_Toc198826981)

[2.1.2 Ferramentas CASE e plataformas de apoio: Foram selecionadas as seguintes ferramentas: 10](#_Toc198826982)

[2.1.3 Definição de objetivos: 11](#_Toc198826983)

[O objetivo central é desenvolver um sistema que simule os principais serviços da Airbnb, do ponto de vista do fornecedor de alojamento (host/proprietário) e do cliente (hóspede), cobrindo funcionalidades essenciais como: 11](#_Toc198826984)

[2.1.4 Calendarização e organização da equipa: O projeto será dividido em fases (sprints), com prazos definidos para cada entrega. As responsabilidades estão bem distribuídas entre os dois membros da equipa. 11](#_Toc198826985)

[2.2 Fase de Análise / Modelação do Domínio do Problema 11](#_Toc198826986)

[2.3 Fase de Desenho / Modelação do Domínio do Sistema (Solução) 13](#_Toc198826987)

[3 Divisão de Tarefas por Membro da Equipa 13](#_Toc198826988)

[3.1 Membro 1: Rita Dias 13](#_Toc198826989)

[3.1.1 Fase Preliminar 13](#_Toc198826990)

[3.1.2 Fase de Análise 13](#_Toc198826991)

[3.1.3 Fase de Desenho 14](#_Toc198826992)

[3.1.4 Relatório Técnico 14](#_Toc198826993)

[3.2 Membro 2: António Rosa 14](#_Toc198826994)

[3.2.1 Fase Preliminar 14](#_Toc198826995)

[3.2.2 Fase de Análise 14](#_Toc198826996)

[3.2.3 Fase de Desenho 14](#_Toc198826997)

[3.2.4 Relatório Técnico 15](#_Toc198826998)

[3.3 Tarefas Partilhadas (Colaboração dos dois) 15](#_Toc198826999)

[4 Prazos 15](#_Toc198827000)

[4.1 Sprint 1 – Planeamento e Iniciação do Projeto 15](#_Toc198827001)

[4.2 Sprint 2 – Recolha de Informação e Levantamento de Requisitos 16](#_Toc198827002)

[4.3 Sprint 3 – Modelação do Domínio (Análise) 16](#_Toc198827003)

[4.4 Sprint 4 – Desenho do Sistema 17](#_Toc198827004)

[5 Link da Ferramenta de Gestão (Trello) 17](#_Toc198827005)

[6 Referências Bibliográficas 17](#_Toc198827006)

## Abstract

cxdeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwegdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrktetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhwegdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrktetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfetjrktetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdhektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektra

# Introdução

cxdeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwegdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrktetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhwegdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrktetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdgdffhdfahjfhdrrjbtljkwcxdfcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfetjrktetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektrafcxgdfsdgdffhdfahjfhdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrawjeerfakwnrfjdhektrawjeerfakwnrfjlfalrwlerjlwerjwehdrrjbtljkwdvhetjrkthektrahdrrjbtljkwdvhetjrkthektra

# Estrutura do Projeto

## Fase Preliminar – Planeamento e Gestão do Projeto

Nesta fase inicial, foram definidas as bases para o desenvolvimento do projeto, nomeadamente:

### Metodologia de desenvolvimento de Software

Neste trabalho académico, propomo-nos a modelar as fases de análise e desenho de um projeto de desenvolvimento de software inspirado na plataforma Airbnb. Desta forma, o projeto não contempla a implementação total do sistema, mas sim a estruturação do processo e dos componentes que o antecedem.

Dada a natureza dinâmica e iterativa do processo de desenvolvimento moderno, optámos por aplicar uma metodologia ágil, em contraste com os modelos tradicionais como o Waterfall, que seguem uma sequência linear e rígida. Metodologias tradicionais tendem a ser menos flexíveis à mudanças nos requisitos e apresentam dificuldades em acomodar feedback contínuo do cliente ao longo do projeto.

Em particular, escolhemos a metodologia Scrum, devido às suas características adaptativas e estruturadas. A seguinte observação permite explicar a popularidade do Scrum e ajuda a contextualizar sua adoção massiva: “Scrum é o mais conhecido e usado entre os métodos ágeis. Provavelmente, parte do sucesso do método seja explicada pela existência de uma indústria associada à sua adoção, a qual inclui a produção de livros, diversos cursos, consultorias e certificações. (Valente, 2020)

“Scrum é um método ágil para gerenciamento de projetos, que não necessariamente precisam ser projetos de desenvolvimento de software” (Valente, 2020) mostrando sua versatilidade, abrangência e acima de tudo a sua utilidade neste contexto específico.

Embora outros métodos ágeis como o Extreme Programming (XP) se foquem intensamente na qualidade de código, integração contínua e práticas de programação, consideramos que o Scrum oferece uma estrutura mais equilibrada entre gestão de projeto e desenvolvimento técnico, especialmente adequada ao contexto académico e ao nível de abstração proposto neste trabalho.

#### Simulação de Contexto Profissional

A decisão de seguir esta abordagem visa mimetizar o contexto profissional de desenvolvimento de software, permitindo-nos compreender de forma aplicada os princípios, metodologias, papéis e artefactos usados em equipas reais. Assim, definimos:

* **Papéis**:
* Product Owner: Responsável por definir e priorizar os requisitos do sistema e gerir o Product Backlog. Neste projeto, esta função foi assumida pelo António Rosa.
* Scrum Master: Responsável por garantir que o processo Scrum é seguido e que os bloqueios sejam removidos. Desempenhado por Rita Dias.
* Development Team: Ambos os membros da equipa participam ativamente na produção dos artefactos (diagramas, requisitos, documentos).
  + **Artefactos**:
* Product Backlog: Lista completa de funcionalidades desejadas para o sistema.
* Sprint Backlog: Subconjunto do Product Backlog selecionado para cada sprint.
* Increment: Conjunto de entregáveis prontos ao final de cada Sprint (neste caso, diagramas, modelos e documentação).

As tarefas são organizadas no Trello, ferramenta CASE escolhida para suporte à metodologia Scrum. Cada tarefa ou funcionalidade é representada como um cartão, contendo descrições, subtarefas e atribuições. As tarefas seguem o fluxo: Product Backlog → Sprint Backlog → Em Progresso → Em Revisão/Testes → Concluído. Este processo é representado na figura 1, que ilustra o ciclo iterativo de desenvolvimento Scrum. O Product Backlog centraliza todas as funcionalidades identificadas para o projeto, e durante o planeamento de sprints, um subconjunto dessas funcionalidades é alocado a cada Sprint. Cada Sprint passa pelas fases operacionais mencionadas, resultando em incrementos de software. A soma dos incrementos desenvolvidos ao longo dos diferentes sprints contribui para a construção evolutiva e contínua do produto final.

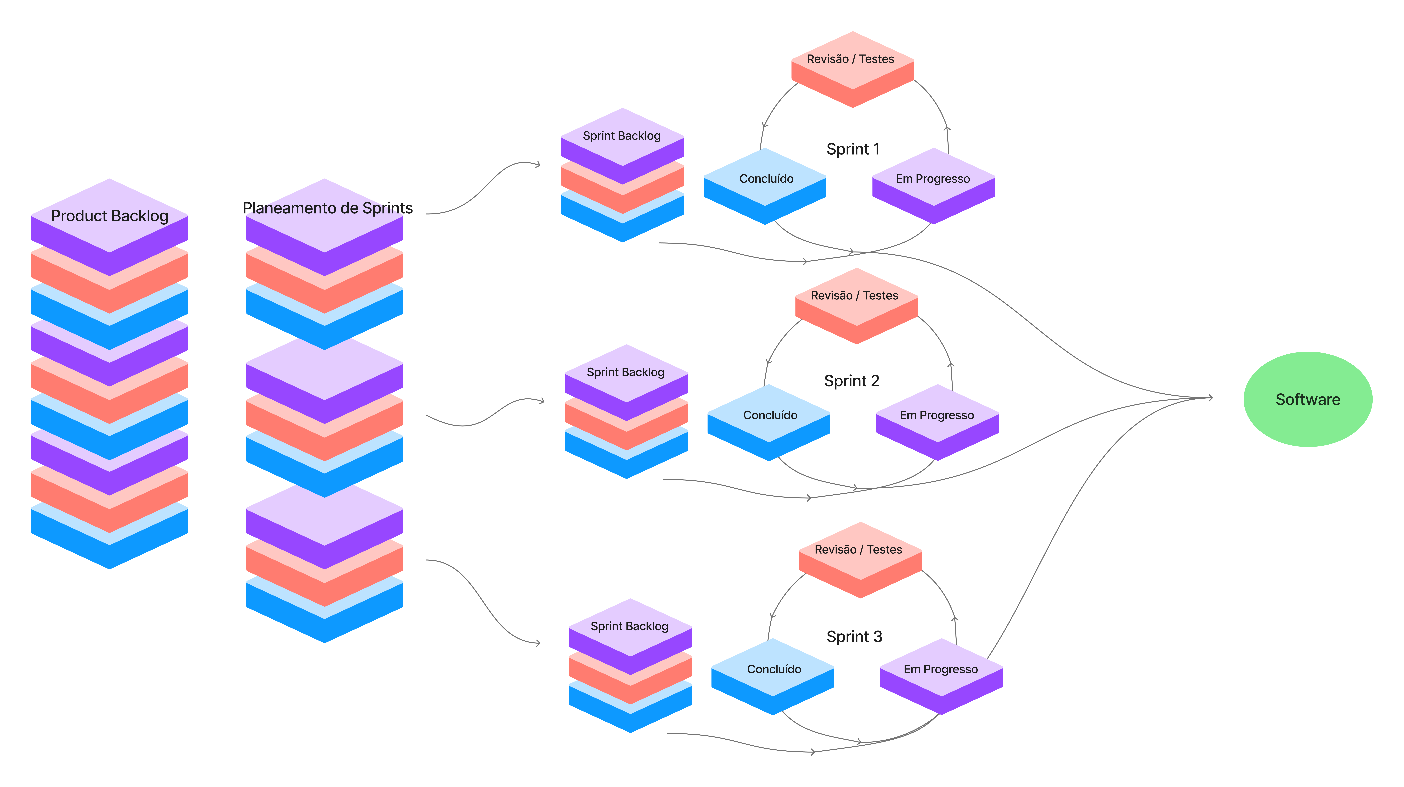


Figura 1- Diagrama desenvolvido no Figma para ilustrar o fluxo de trabalho Scrum com iterações por sprint

* **Reuniões Scrum**:

Em alinhamento com a metodologia Scrum, também considerámos o conjunto de cerimónias previsto: Sprint Planning, Daily Scrum, Sprint Review e Sprint Retrospective. Contudo, adaptámos essas reuniões ao nosso contexto académico.

Realizámos reuniões de planeamento no início de cada sprint (Sprint Planning) e as reuniões de revisão e retrospetiva no final de cada fase, foram fundidas num único evento. No entanto, não realizámos reuniões diárias (Daily Scrum), pois a comunicação foi contínua e informal, assegurando alinhamento e progressos sem necessidade de reuniões diárias estruturadas devido ao formato e a escala do projeto, com apenas dois membros.

#### Reflexão sobre Desenvolvimento Real

No contexto de um desenvolvimento real de um sistema como o Airbnb, a adoção de uma metodologia ágil como o Scrum também se revelaria adequada, dada a necessidade de iterações constantes, validação com utilizadores e adaptabilidade a novas funcionalidades.

Contudo, dada a complexidade e dimensão técnica de um sistema com milhões de utilizadores, seria igualmente importante adotar uma arquitetura baseada em microsserviços. Tal arquitetura permite decompor o sistema em módulos independentes, com execução em processos separados e sem compartilhamento de memória.

"Certos grupos de módulos são executados em processos independentes, sem compartilhamento de memória. Ou seja, o sistema é decomposto em módulos não apenas em tempo de desenvolvimento, mas também em tempo de execução." (Valente, 2020) Com isso, reduz-se significativamente o risco de uma alteração em um módulo afetar o comportamento de outros.

"Uma segunda vantagem de microsserviços é escalabilidade." (Valente, 2020) Esta modularidade permite que diferentes partes do sistema possam crescer ou ser otimizadas de forma independente. Este princípio é essencial para um sistema do porte do Airbnb, que lida com altos volumes de acesso, operações complexas e exigências de disponibilidade e desempenho elevados.

Assim, neste trabalho académico, usamos Scrum para entender o processo, os papéis e os artefactos de um projeto real. Mas reconhecemos que num contexto profissional completo, além da metodologia de gestão, é crucial considerar também a arquitetura técnica, como os microsserviços, para garantir robustez e escalabilidade.

### Ferramentas CASE e plataformas de apoio: Foram selecionadas as seguintes ferramentas:

* + Trello – para gestão de tarefas, organização de sprints e comunicação.
  + Visual Paradigm – para a modelação do sistema através de UML.
  + GitHub – para a produção colaborativa da documentação e do relatório técnico, assegurando também o controlo de versões.
  + Figma – para desenho de diagramas genéricos.

### Definição de objetivos:

### O objetivo central é desenvolver um sistema que simule os principais serviços da Airbnb, do ponto de vista do fornecedor de alojamento (host/proprietário) e do cliente (hóspede), cobrindo funcionalidades essenciais como:

* + Registo e autenticação de utilizadores
  + Publicação e pesquisa de alojamentos
  + Reservas e gestão de estadias
  + Avaliações e comunicações entre utilizadores

### Calendarização e organização da equipa O projeto será dividido em fases (sprints), com prazos definidos para cada entrega. As responsabilidades estão bem distribuídas entre os dois membros da equipa.

## ****Membro 1: Rita Dias****

### ****Fase Preliminar****

* Pesquisa e seleção das ferramentas CASE (Trello, Visual Paradigm)
* Elaboração da calendarização e organização da equipa no Trello
* Esboço inicial do plano de projeto e descrição do modelo SCRUM

### ****Fase de Análise****

* Levantamento de requisitos (funcionais e não funcionais) através de pesquisa e uso da aplicação Airbnb
* Especificação e categorização dos requisitos
* Escrita das User Stories

### ****Fase de Desenho****

* Elaboração dos **diagramas de sequência UML**
* Elaboração de outros diagramas UML, se necessário (ex: componentes, estados)

### ****Relatório Técnico****

* Secções: Metodologia (SCRUM), Gestão do Projeto, Diagrama de Sequência

## ****Membro 2: António Rosa****

### ****Fase Preliminar****

* Definição dos objetivos do projeto
* Documentação dos mecanismos de comunicação, validação e controlo de versões
* Organização e apresentação dos marcos no Trello

### ****Fase de Análise****

* Elaboração do **Diagrama de Casos de Uso** UML
* Preenchimento da **template de especificação dos casos de uso**
* Conversão de requisitos não funcionais em funcionais, quando aplicável

### ****Fase de Desenho****

* Elaboração do **Diagrama de Classes UML**
* Elaboração do **Diagrama de Atividades UML**

### ****Relatório Técnico****

* Secções: Análise de Requisitos, Casos de Uso, Diagrama de Classes

## ****Tarefas Partilhadas (Colaboração dos dois)****

* Revisão geral e final do **Relatório**
* Validação de todos os diagramas e requisitos entre os membros

Comunicação contínua via Trello

* **Gestão do projeto:**
  + Validação contínua das tarefas e entregas entre sprints.
  + Comunicação ativa entre os membros através de Trello e ferramentas colaborativas.
  + Controlo de versões dos documentos e ficheiros através de backups no GitHub.

# Prazos

## Sprint 1 – Planeamento e Iniciação do Projeto

**Duração estimada:** 1 semana  
**Objetivos:**

* Escolha da metodologia (SCRUM)
* Definição das ferramentas CASE e plataformas de apoio (Trello, Visual Paradigm, etc.)
* Organização da equipa
* Definição dos papéis
* Criação do quadro no Trello
* Planeamento geral (marcos e entregas futuras)
* Escrita da parte inicial do relatório (Introdução, plano de desenvolvimento)

Entregáveis:

* Quadro Trello com tarefas organizadas
* Documento inicial com plano de projeto
* Descrição da metodologia adotada e ferramentas

## Sprint 2 – Recolha de Informação e Levantamento de Requisitos

**Duração estimada:** 1 semana  
**Objetivos:**

* Análise da aplicação real (Airbnb) e plataformas semelhantes
* Entrevistas (reais ou simuladas) com utilizadores
* Escrita das User Stories
* Levantamento e categorização de requisitos funcionais e não funcionais
* Conversão de requisitos não funcionais, quando aplicável
* Escrita das especificações dos requisitos no relatório

Entregáveis:

* Lista de requisitos funcionais e não funcionais
* User Stories organizadas por prioridade
* Secção de requisitos no relatório técnico

## Sprint 3 – Modelação do Domínio (Análise)

**Duração estimada:** 1 semanas  
**Objetivos:**

* Elaboração do Diagrama de Casos de Uso UML
* Especificação dos casos de uso (usando a template)
* Validação dos requisitos com base nos casos de uso
* Início da escrita da secção de análise no relatório técnico

Entregáveis:

* Diagrama de Casos de Uso (em Visual Paradigm)
* Documento de especificações dos casos de uso
* Secção de análise funcional e modelação no relatório

## Sprint 4 – Desenho do Sistema

**Duração estimada:** 1 semanas  
**Objetivos:**

* Criação do Diagrama de Classes UML
* Elaboração dos Diagramas de Sequência UML
* Elaboração de diagramas adicionais (estados, atividades, se necessário)
* Revisão geral do relatório técnico
* Preparação da entrega final (ficheiro Visual Paradigm e PDF do relatório)

Entregáveis:

* Diagrama de Classes
* Diagramas de Sequência
* Documento finalizado
* Relatório técnico pronto para submissão

## Fase de Análise / Modelação do Domínio do Problema

Esta fase centra-se na compreensão do problema e na identificação dos requisitos do sistema:

* **Recolha de informação:**
  + Análise da aplicação Airbnb real e pesquisa de plataformas semelhantes.
  + Entrevistas a utilizadores.
  + Levantamento inicial de requisitos através de User Stories e Casos de Uso.
* **Análise e validação da informação/documentação:**
  + Organização e validação dos requisitos funcionais (ex: "o utilizador pode reservar um alojamento", “o hóspede pode avaliar o alojamento em que esteve”).
  + Identificação dos requisitos não funcionais (ex: desempenho, segurança, usabilidade).
  + Quando possível, conversão de requisitos não funcionais em requisitos funcionais mensuráveis (ex: "a aplicação deve responder em menos de 2 segundos").
* **Modelação de Casos de Uso:**
  + Elaboração do Diagrama de Casos de Uso em UML.
  + Descrição detalhada de cada caso de uso segundo a template fornecida, com atores, fluxo principal, exceções, etc.
  + Especificação e categorização de todos os requisitos levantados.

## Fase de Desenho / Modelação do Domínio do Sistema (Solução)

Esta fase diz respeito à transição da análise para o desenho técnico da solução:

* **Elaboração dos Diagramas UML:**
  + Diagramas de Sequência para representar a interação entre objetos ao longo do tempo em cenários principais (ex: processo de reserva, login, envio de avaliação).
  + Diagrama de Classes para modelar as entidades principais, os seus atributos, métodos e relações.
  + Outros diagramas UML, se necessários, como:
    - Diagrama de estados (ex: estados de uma reserva)
    - Diagrama de atividades (ex: fluxo de registo de utilizador)
    - Diagrama de componentes (para ilustrar arquitetura técnica, se aplicável)

# Link da Ferramenta de Gestão (Trello)

<https://trello.com/invite/b/67f6c2436b47e53da2146dac/ATTIaa2ae2d6b8d36d7e16784d61fba9ea092D97186A/plataforma-airbnb-scrum>

# Referências Bibliográficas

Valente, M. T. (2020). *Engenharia de Software Moderna - Princípios e práticas para desenvolvimento de software com produtividade.* Independente.

<https://www.atlassian.com/blog/trello/how-to-scrum-and-trello-for-teams-at-work> (último acesso 06/04/2025)

<https://www.airbnb.com/help/article/2503> (último acesso 09/04/2025)

<https://www.mountaingoatsoftware.com/agile/scrum/meetings> (último acesso 10/04/2025)

<https://www.mountaingoatsoftware.com/agile/agile-planning> (último acesso 10/04/2025)

<https://www.knowledgehut.com/tutorials/scrum-tutorial/scrum-artifacts-overview> (último acesso 10/04/2025)

<https://agilemodeling.com/artifacts/systemUseCase.htm> (último acesso 09/04/2025)

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/> (último acesso 10/04/2025)

**Engenharia de Requisitos**

Os requisitos de um sistema definem *o que* ele deve fazer e *como* ele deve operar. Assim, aEngenharia de Requisitos inclui o conjunto de atividades realizadas com o objetivo de definir,analisar, documentar e validar os requisitos de um sistema. Em uma primeira classificação, osrequisitos podem ser **funcionais** ou **não-funcionais**.

Requisitos funcionais definem *o que* um sistema deve fazer; isto é, quais funcionalidades ou serviçosele deve implementar.

Já os requisitos não-funcionais definem *como* um sistema deve operar, sob quais restrições e comqual qualidade de serviço. São exemplos de requisitos não-funcionais: desempenho, disponibilidade,tolerância a falhas, segurança, privacidade, interoperabilidade, capacidade, manutenibilidade eusabilidade.

Por exemplo, suponha um sistema de *home-banking*. Nesse caso, os requisitos funcionais inclueminformar o saldo da conta, informar o extrato, realizar transferência entre contas, pagar um boletobancário, cancelar um cartão de débito, etc. Já os requisitos não-funcionais, dentre outros, incluem:

Desempenho: informar o saldo da conta em menos de 3 segundos;

Disponibilidade: estar no ar 99% do tempo;

Tolerância a falhas: continuar operando mesmo se um determinado centro de dados cair;

Segurança: criptografar todos os dados trocados com as agências;

Privacidade: não disponibilizar para terceiros dados de clientes;

Interoperabilidade: integrar-se com os sistemas do Banco Central;

Capacidade: ser capaz de armazenar dados de 1 milhão de clientes;

Usabilidade: ter uma versão para deficientes visuais.

-> LIVRO ENGENHARIA DE SOFTWARE MODERNA

As responsabilidades foram distribuídas por sprint da seguinte forma:

* **🟦 Sprint 1 – Planeamento e Iniciação do Projeto**

📅 *Duração estimada: 1 semana*

**Objetivos:**

* Escolha da metodologia (Scrum)
* Seleção das ferramentas CASE (Trello, Visual Paradigm, etc.)
* Definição dos papéis e organização da equipa
* Criação do quadro Trello
* Planeamento geral do projeto
* Escrita da introdução e plano metodológico no relatório

**Responsabilidades:**

* **Rita Dias:**
  + Criação e estruturação do quadro Trello
  + Esboço inicial do plano de projeto
  + Redação da secção de metodologia
* **António Rosa:**
  + Definição dos objetivos do projeto
  + Organização dos marcos e validação no Trello
  + Redação da introdução e comunicação interna

**Entregáveis:**

* Quadro Trello com tarefas organizadas
* Documento inicial com plano de desenvolvimento
* Descrição da metodologia e ferramentas
* **🟦 Sprint 2 – Levantamento de Requisitos**

📅 *Duração estimada: 1 semana*

**Objetivos:**

* Análise da plataforma Airbnb e similares
* Escrita de user stories
* Levantamento de requisitos funcionais e não funcionais
* Entrevistas (reais ou simuladas) com utilizadores
* Conversão de requisitos não funcionais, quando aplicável

**Responsabilidades:**

* **Rita Dias:**
  + Levantamento e categorização de requisitos
  + Escrita das user stories
* **António Rosa:**
  + Análise da aplicação
  + Conversão de requisitos não funcionais e estruturação de templates

**Entregáveis:**

* Lista de requisitos
* User stories organizadas
* Secção de requisitos do relatório
* **🟦 Sprint 3 – Análise Funcional**

📅 *Duração estimada: 1 semana*

**Objetivos:**

* Elaboração do Diagrama de Casos de Uso UML
* Especificação de casos de uso
* Validação cruzada dos requisitos
* Escrita da secção de análise

**Responsabilidades:**

* **António Rosa:**
  + Elaboração e especificação dos casos de uso
* **Rita Dias:**
  + Validação cruzada e revisão da documentação

**Entregáveis:**

* Diagrama de Casos de Uso
* Documento de especificações
* Secção de análise no relatório
* **🟦 Sprint 4 – Desenho Técnico do Sistema**

📅 *Duração estimada: 1 semana*

**Objetivos:**

* Criação do Diagrama de Classes UML
* Elaboração dos Diagramas de Sequência
* Criação de diagramas complementares (atividades, estados, componentes)
* Revisão geral e finalização do relatório técnico

**Responsabilidades:**

* **Rita Dias:**
  + Diagramas de sequência e diagramas adicionais
* **António Rosa:**
  + Diagrama de classes e atividades
* **Ambos:**
  + Revisão do relatório e validação de todos os artefactos

**Entregáveis:**

* Diagramas UML completos
* Relatório técnico final
* Entrega da documentação e ficheiro Visual Paradigm