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, junho 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, junho de 2025

Índice

[Abstract 5](#_Toc199929553)

[1 Introdução 6](#_Toc199929554)

[2 Fase de Planeamento Inicial e Gestão do Projeto 7](#_Toc199929555)

[2.1 Definição de objetivos 7](#_Toc199929556)

[2.2 Metodologia de desenvolvimento de Software 7](#_Toc199929557)

[2.2.1 Simulação de Contexto Profissional 8](#_Toc199929558)

[2.2.2 Reflexão sobre Desenvolvimento Real 10](#_Toc199929559)

[2.3 Ferramentas CASE e plataformas de apoio 11](#_Toc199929560)

[2.4 Calendarização e organização da equipa 12](#_Toc199929561)

[2.4.1 Sprint 1 – Planeamento e Iniciação do Projeto 12](#_Toc199929562)

[2.4.2 Sprint 2 – Fase de análise 13](#_Toc199929563)

[2.4.3 Sprint 3 – Fase de Desenho 14](#_Toc199929564)

[2.4.4 Gestão do projeto 14](#_Toc199929565)

[3 Fase de Análise 15](#_Toc199929566)

[3.1 Engenharia de Requisitos 15](#_Toc199929567)

[3.2 Técnicas de obtenção de Requisitos 15](#_Toc199929568)

[3.2.1 Análise do sistema real Airbnb e estudo da utilização da aplicação 15](#_Toc199929569)

[3.2.2 Simulação de entrevista a utilizadores 15](#_Toc199929570)

[3.2.3 User stories para descrever funcionalidades centradas no utilizador 18](#_Toc199929571)

[3.2.4 Casos de uso (UML) para representação gráfica das interações 22](#_Toc199929572)

[3.3 Especificação de requisitos 22](#_Toc199929573)

[3.3.1 Documento de requisitos 22](#_Toc199929574)

[4 Fase de Desenho / Modelação do Domínio do Sistema (Solução) 25](#_Toc199929575)

[5 Conclusão 26](#_Toc199929576)

[6 Links para plataformas de apoio 27](#_Toc199929577)

[7 Referências Bibliográficas 27](#_Toc199929578)

Índice de Figuras

[Figura 1- Diagrama desenvolvido no Figma para ilustrar o fluxo de trabalho Scrum com iterações por sprint 9](#_Toc199933528)

[Figura 2 - Quadro Trello durante o Sprint 2 10](#_Toc199933529)

[Figura 3 – GitHub para controlo de versões e manutenção de histórico da documentação e diagramas 11](#_Toc199933530)

[Figura 4 - Criação de conta de hópede na plataforma Airbnb, recolha de dados pessoais 16](#_Toc199933531)

[Figura 5 - Criação de conta de hópede na plataforma Airbnb, criação do primeiro anúncio 17](#_Toc199933532)

[Figura 6 – Interface Principal do utilizador anfitrião 18](#_Toc199933533)

[Figura 7 - Seleção do modo viagem no perfil anfitrião 19](#_Toc199933534)

[Figura 8 - Mudança para o modo hospedagem 19](#_Toc199933535)

# Abstract

# Introdução

# Fase de Planeamento Inicial e Gestão do Projeto

## Definição de objetivos

O objetivo central deste projeto académico é simular o desenvolvimento de um sistema de reserva de alojamentos, inspirado na plataforma Airbnb, com o intuito de aplicar conceitos e metodologias de Engenharia de Software. Em particular, vamos focar-nos na modelação da fase de análise (domínio do problema) e da fase de desenho (domínio da solução), estruturando o processo de desenvolvimento sem implementar o sistema completo. De forma mais detalhada, pretendemos:

* Escolher a metodologia ágil mais adequada e selecionar as ferramentas CASE que darão suporte à gestão do projeto;
* Definir os papéis e responsabilidades da equipa, planear e calendarizar o projeto;
* Analisar a plataforma Airbnb e similares para levantar e categorizar requisitos funcionais e não funcionais, desenvolver user stories e elaborar diagramas de casos de uso UML, assegurando que os requisitos identificados respondem às necessidades do utilizador e do product owner (domínio do problema);
* Desenhar os modelos UML essenciais, incluindo diagramas de classes e sequência, representando as soluções do sistema (domínio da solução);
* Gerir o projeto com validação contínua das tarefas e entregas, comunicação eficaz e controlo de versões dos documentos e ficheiros.

## 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 foco aqui não é a implementação total do sistema, mas sim a estruturação do processo e dos componentes que o antecedem.

Como o desenvolvimento de software hoje em dia é dinâmico e iterativo, optámos por usar uma metodologia ágil, em vez de modelos mais tradicionais, como o Waterfall, que seguem uma sequência rígida e linear. Estes modelos mais antigos tendem a ser menos flexíveis a mudanças nos requisitos e têm dificuldade em integrar o feedback contínuo do cliente ao longo do projeto.

Em particular, escolhemos a metodologia Scrum porque, na nossa perspectiva oferece uma combinação equilibrada entre estrutura e capacidade de adaptação. Como refere (Valente, 2020), “Scrum é um método ágil para gerenciamento de projetos, que não necessariamente precisam ser projetos de desenvolvimento de software” mostrando sua versatilidade, abrangência e acima de tudo a sua relevância para o contexto académico e para a estrutura que estamos a desenvolver.

### 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 no final de cada Sprint (neste caso, diagramas, modelos e documentação).

As tarefas são organizadas no Trello, ferramenta CASE escolhida para dar suporte à metodologia Scrum. Cada tarefa ou funcionalidade é representada como um cartão, contendo descrições 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 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. Na figura 2, apresentamos um print do quadro trello durante o sprint 2 para ilustrar o processo de organização, evolução e monitorização do progresso das tarefas.

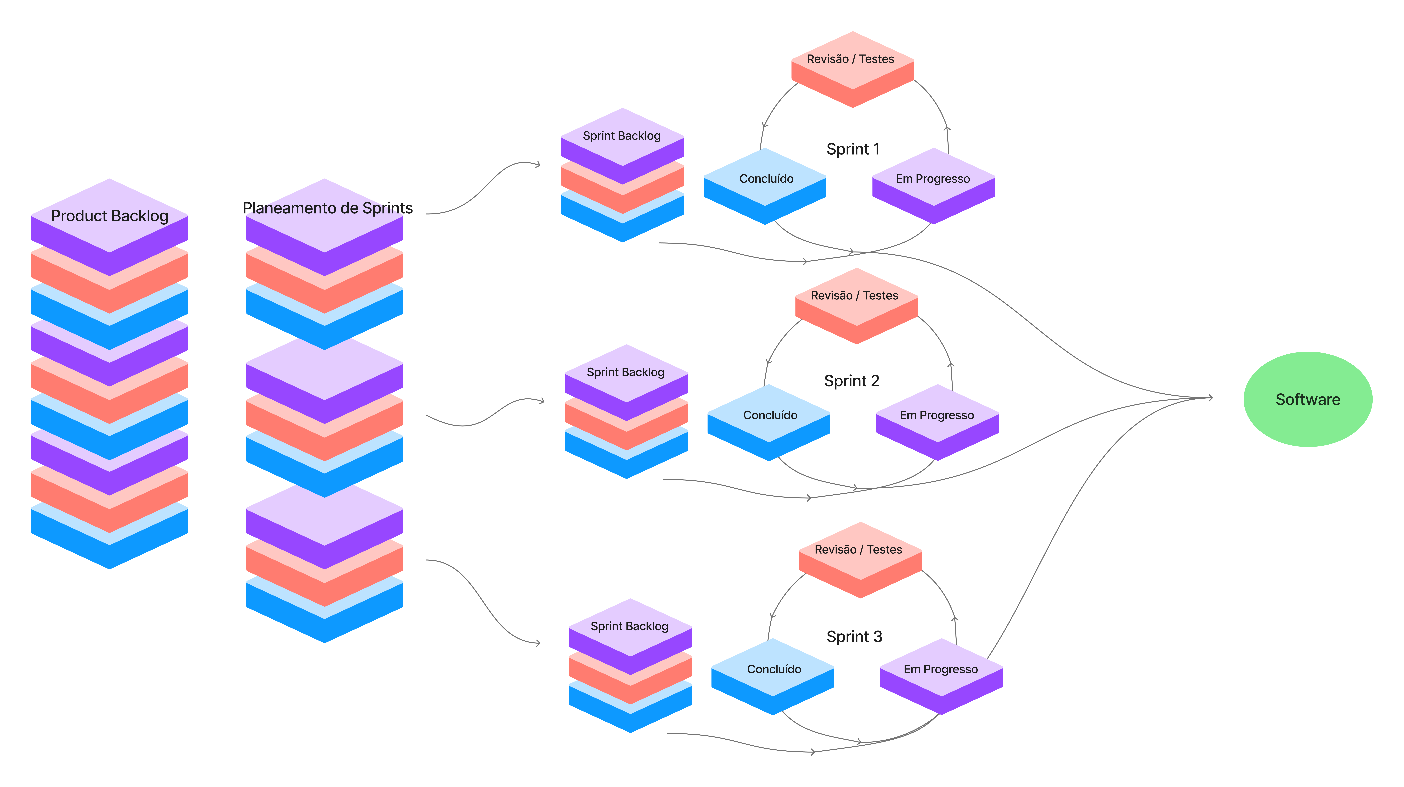


Figura 1- Diagrama desenvolvido no Figma para ilustrar o fluxo de trabalho Scrum com iterações por sprint

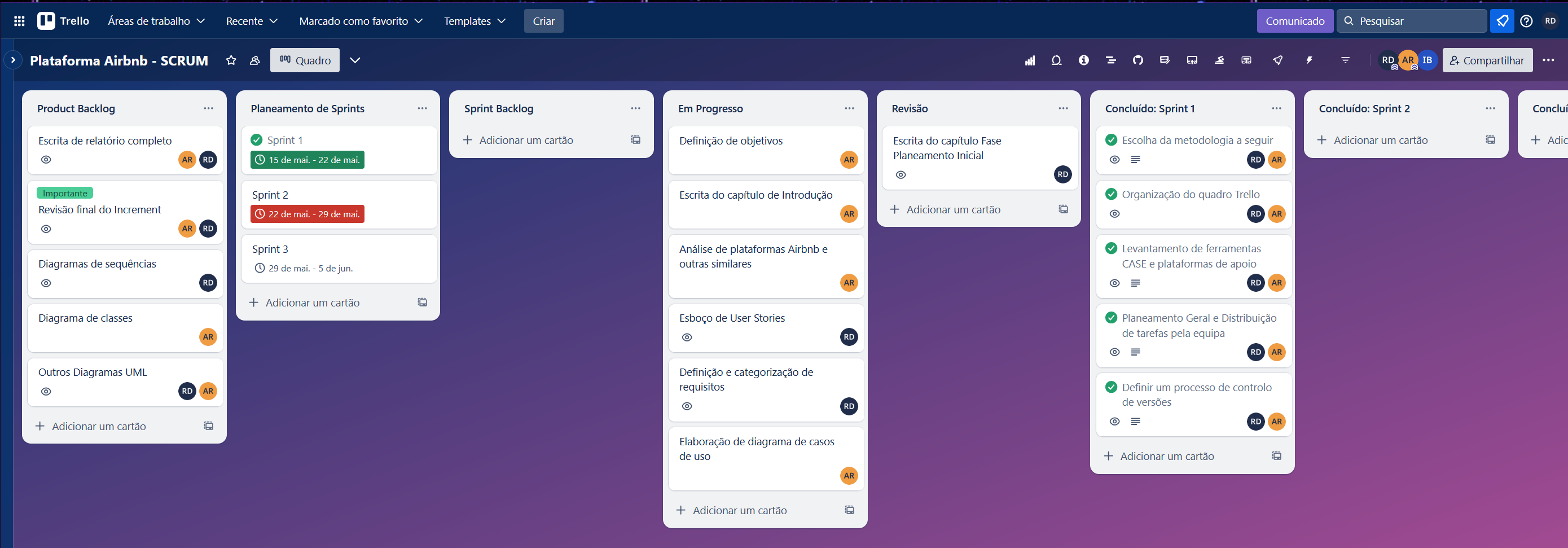


Figura 2 - Quadro Trello durante o Sprint 2

* **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 tantos utilizadores, seria igualmente importante adotar uma arquitetura baseada em microsserviços. Tal arquitetura permite dividir o sistema em "Certos grupos de módulos que 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 num 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 plataformas como o Airbnb, que processam grandes volumes de dados, operações complexas e requerem alta disponibilidade e desempenho.

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

Para apoiar o desenvolvimento do projeto, foram selecionadas ferramentas CASE e plataformas de gestão que facilitam a modelação, o controlo de versões e a colaboração da equipa, otimizando o fluxo de trabalho e a qualidade dos artefactos. Assim, optámos pelas seguintes soluções:

* + 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.

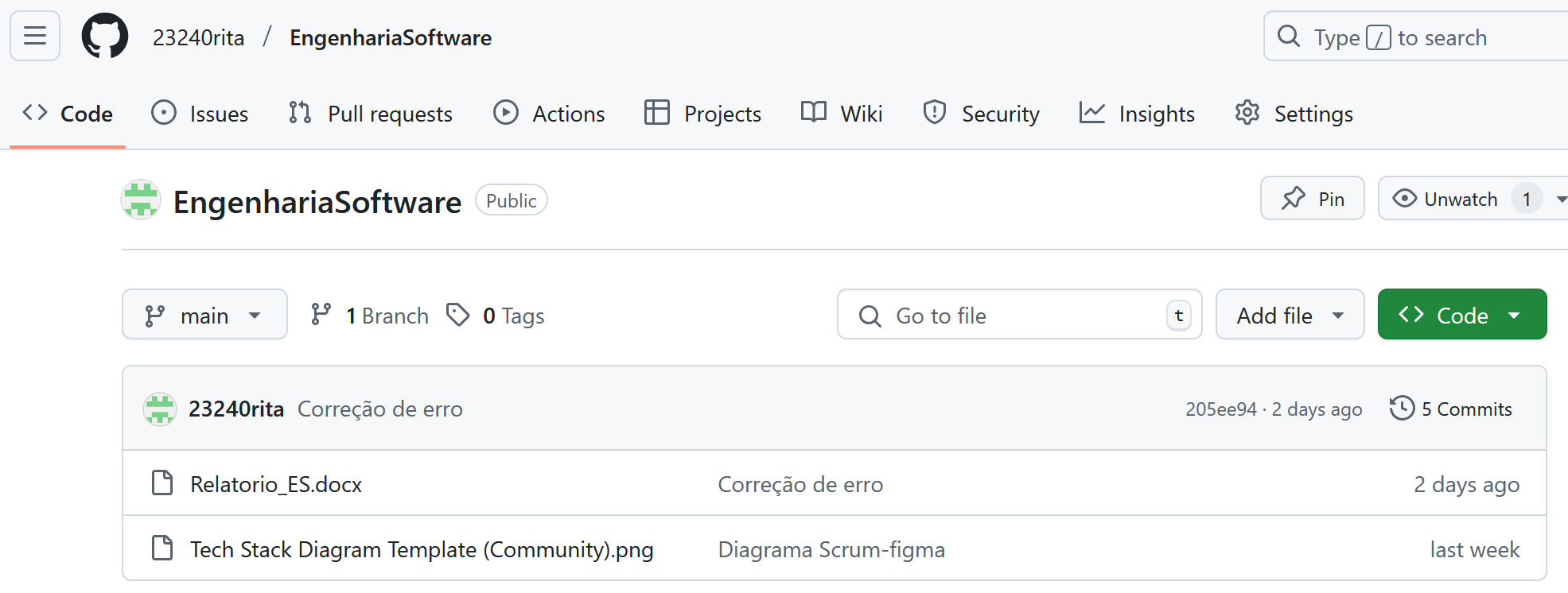


Figura 3 – GitHub para controlo de versões e manutenção de histórico da documentação e diagramas

## Calendarização e organização da equipa

O projeto foi dividido em fases (sprints), com prazos definidos para cada entrega. As responsabilidades encontram-se distribuídas entre os dois membros da equipa, da seguinte forma:

### Sprint 1 – Planeamento e Iniciação do Projeto

(Duração estimada: 1 semana)

**Objetivos:**

* Escolha da metodologia
* Pesquisa e seleção das ferramentas CASE
* Definição dos papéis e organização da equipa
* Planeamento geral e calendarização do projeto
* Criação do quadro Trello
* Escrita dos capítulos de introdução e Fase de Planeamento Inicial e Gestão do Projeto

**Responsabilidades:**

* **Rita Dias:**
  + Criação e estruturação do quadro Trello
  + Esboço inicial do plano de projeto
  + Redação da secção da fase de planeamento Inicial do relatório final
* **António Rosa:**
  + Definição dos objetivos do projeto
  + Definir um processo de controlo de versões e manter o histórico (usar uma ferramenta para garantir a rastreabilidade)
  + Redação da introdução do relatório final

**Entregáveis:**

* Quadro Trello com tarefas organizadas
* Documento inicial com plano de desenvolvimento
* Descrição da metodologia e ferramentas
* Parte inicial do relatório

### Sprint 2 – Fase de análise

(Duração estimada: 1 semana)

**Objetivos:**

* Análise da plataforma Airbnb e similares
* Escrita de user stories
* Elaboração do Diagrama de Casos de Uso UML
* Levantamento e categorização de requisitos funcionais e não funcionais
* Escrita de entrevista para levantamento de requisitos
* Validação cruzada dos requisitos (Garantir que os requisitos identificados correspondem realmente às necessidades dos utilizadores e do product owner)
* Escrita dos capítulos da fase de análise no relatório

**Responsabilidades:**

* **Rita Dias:**
  + Escrita das user stories
  + Escrita da entrevista
  + Levantamento e categorização de requisitos
* **António Rosa:**
  + Análise da plataforma Airbnb e similares
  + Elaboração do diagrama de casos de uso
* **Ambos:**
  + Escrita dos capítulos da fase de análise no relatório

**Entregáveis:**

* Lista de requisitos funcionais e não funcionais
* User stories
* Entrevista para utilizadores do sistema Airbnb
* Secção de requisitos do relatório final

### Sprint 3 – Fase de Desenho

(Duração estimada: 1 semana) (Mike Beedle, 2001)

**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 UML e outros diagramas UML, se necessário (ex: componentes, estados)
* **António Rosa:**
  + Diagrama de classes e outros diagramas UML, se necessário (ex: componentes, estados)
* **Ambos:**
  + Escrita final e revisão do relatório
  + Validação de todos os artefactos entre os membros
  + Comunicação contínua via Trello

**Entregáveis:**

* Diagramas UML completos – ficheiro Visual Paradigm
* Relatório técnico final

### Gestão do projeto

* + Validação contínua das tarefas e entregas entre sprints.
  + Comunicação ativa entre os membros através de Trello e outros meios.
  + Controlo de versões dos documentos e ficheiros através de backups no GitHub.

# Fase de Análise

## Engenharia de Requisitos

A fase de análise é essencial no desenvolvimento de software, pois permite compreender de forma aprofundada o domínio do problema e estabelecer as bases para a fase de desenho. O seu foco reside na identificação, análise e validação dos requisitos do sistema.

No âmbito da Engenharia de Requisitos, são realizadas atividades específicas para definir o que o sistema deve fazer (requisitos funcionais) e como ele deve operar (requisitos não funcionais) (Valente, 2020). Requisitos funcionais descrevem as funcionalidades e os serviços a serem implementados, como, por exemplo, permitir a pesquisa e a reserva de alojamentos. Requisitos não funcionais especificam atributos de qualidade e restrições, tais como desempenho, segurança, usabilidade e disponibilidade, que condicionam o comportamento do sistema.

Para a recolha de informação, recorremos a técnicas como a análise do sistema real do Airbnb e o estudo da utilização da aplicação, identificando funcionalidades e características essenciais. Além disso, consideramos a simulação de uma entrevista com utilizadores reais da plataforma, que, apesar de não ser aplicada nesta fase académica, seria fundamental num projeto real para obter feedback direto e levantar requisitos relevantes.

A modelação inclui também a utilização de user stories, que permitem descrever de forma centrada no utilizador as funcionalidades desejadas, e de diagramas de casos de uso (UML), que representam graficamente as interações entre os atores e o sistema.

No que diz respeito à especificação dos requisitos, será elaborado um documento formal baseado na norma IEEE/ANSI 830-1993, que, embora tenha sido proposto no contexto de processos waterfall na década de 1990 (Valente, 2020), ainda possui relevância para este trabalho académico. Esta norma proporciona uma estrutura clara e completa para a documentação dos requisitos, assegurando a consistência e a rastreabilidade, mesmo num contexto ágil e académico.

## Técnicas de obtenção de Requisitos

### Análise do sistema real Airbnb e estudo da utilização da aplicação

Com o objetivo de fundamentar a fase de levantamento de requisitos, foi realizada uma análise exploratória da plataforma real Airbnb. No contexto real de desenvolvimento de software, este tipo de análise pode ocorrer em dois cenários distintos: quando o sistema ainda não existe, e é necessário estudar sistemas semelhantes para extrair requisitos comparáveis; ou quando o sistema já existe, sendo alvo de evolução ou melhoria, o que justifica uma análise detalhada da solução atual.

No âmbito do nosso projeto académico analisámos a plataforma existente do Airbnb com o intuito de compreender o seu funcionamento prático e, a partir daí, identificar elementos essenciais para a nossa fase de análise e desenho. Esta observação permitiu explorar processos como o registo de utilizadores, criação de anúncios, pesquisa de alojamentos, realização de reservas e comunicação entre anfitriões e hóspedes. A análise direta da interface e das funcionalidades disponíveis revelou-se essencial para identificar casos de uso relevantes e inspirar a modelação de requisitos funcionais adaptados à realidade do sistema que estamos a desenvolver.

Começámos por analisar o processo de criação de conta na perspetiva do utilizador anfitrião. Para além da recolha de dados pessoais básicos, como nome, data de nascimento e endereço de email (figura 4), observámos que o sistema permite avançar de forma bastante rápida para a criação de um anúncio de alojamento.

Este processo é segmentado em três fases principais (figura 5):

1. Informações básicas — onde são introduzidos dados como a localização do alojamento, tipo de espaço e capacidade máxima de hóspedes;
2. Descrição do anúncio — que inclui o título, a descrição detalhada e o carregamento de fotografias do espaço;
3. Definição de preços — com a possibilidade de configurar valores por noite, descontos por estadia prolongada e opções de publicação imediata.

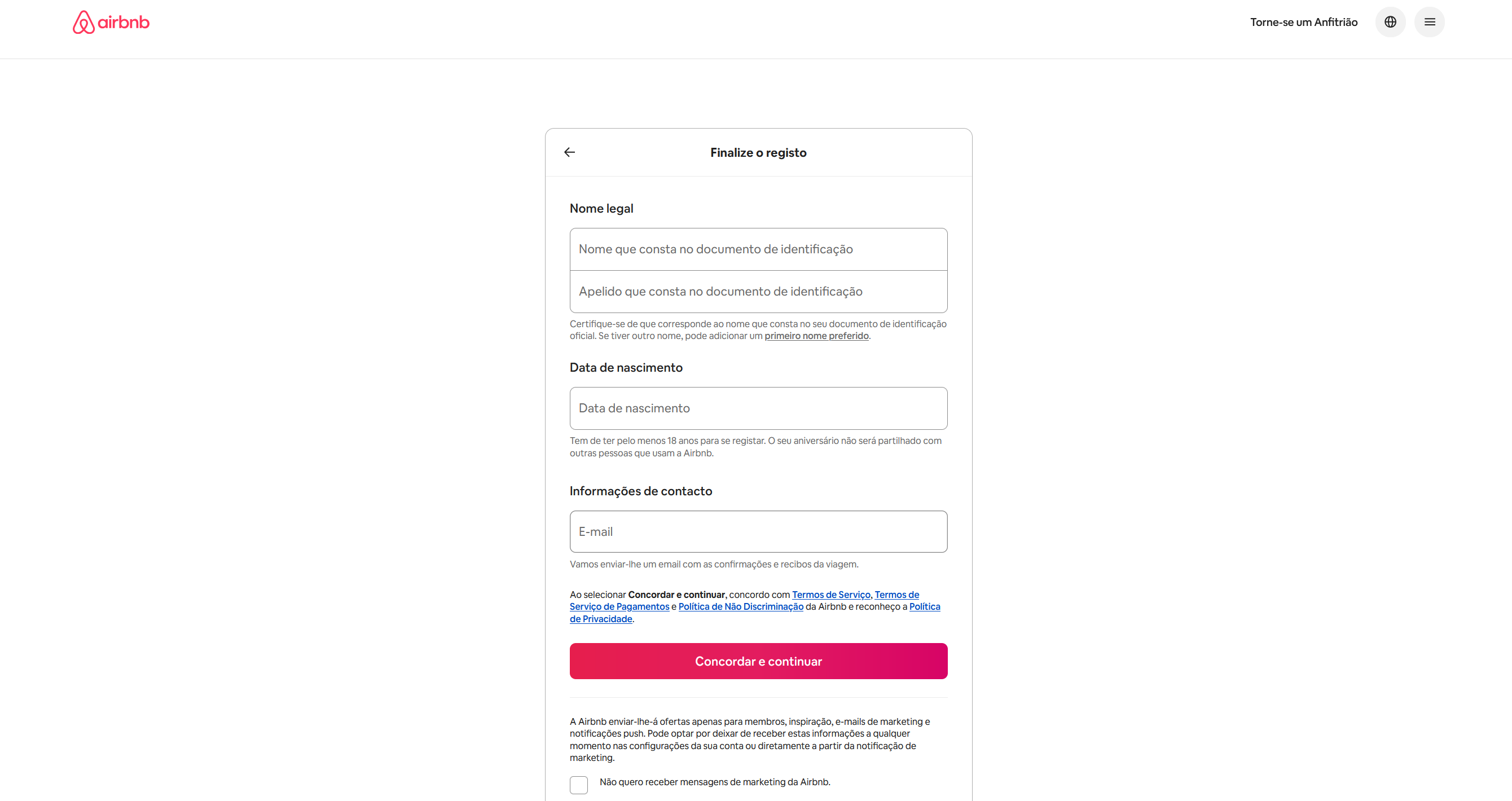


Figura 4 - Criação de conta de hópede na plataforma Airbnb, recolha de dados pessoais



Figura 5 - Criação de conta de hópede na plataforma Airbnb, criação do primeiro anúncio

Adicionalmente, foi analisada a interface principal (figura 6) disponibilizada ao utilizador após o início de sessão, que apresenta um menu superior com várias opções essenciais para a gestão da atividade na plataforma. Entre os itens observados destacam-se:

* **Hoje**, que resume as próximas reservas e tarefas relevantes;
* **Calendário**, que permite visualizar a disponibilidade e ocupação dos alojamentos;
* **Anúncios**, onde o utilizador pode consultar, editar ou remover os seus espaços publicados;
* **Mensagens**, que centraliza a comunicação com hóspedes ou anfitriões;
* **Reservas**, onde é possível acompanhar o histórico e estado atual das reservas;
* **Rendimentos**, com informações sobre ganhos, pagamentos e estatísticas de desempenho;
* **Criar novo anúncio**, para adicionar um novo espaço à plataforma;
* E o **Perfil do utilizador**, que permite aceder e configurar dados pessoais, definições de conta, preferências e verificações de segurança.

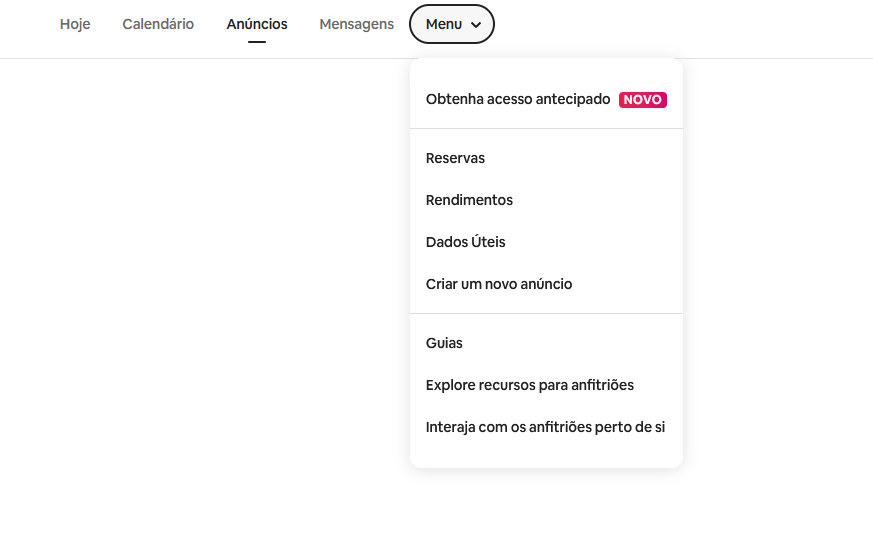


Figura 6 – Interface Principal do utilizador anfitrião

De salientar a funcionalidade "Modo Viagem" (figura 7) disponível no perfil do utilizador anfitrião, que permite alternar para a visualização do site na perspetiva de hóspede (figura 8). Esta opção oferece uma experiência direta e simplificada do ponto de vista de quem procura alojamento, ocultando funcionalidades específicas de gestão de anúncios. Ao ativar este modo, o utilizador pode explorar a plataforma tal como um hóspede faria, incluindo a pesquisa de alojamentos, filtragem por preferências, visualização de avaliações e simulação de reservas. Esta alternância entre modos revelou-se particularmente útil para compreender a separação de responsabilidades e funcionalidades entre perfis distintos, o que influenciou diretamente a definição de requisitos e o desenho da interface no nosso projeto.

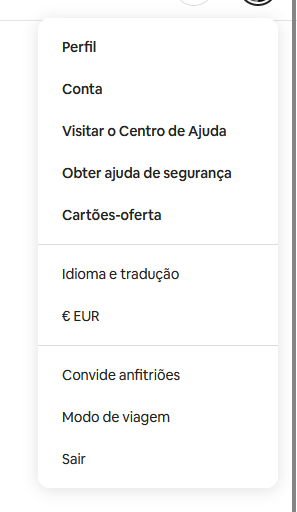


Figura 7 - Seleção do modo viagem no perfil anfitrião

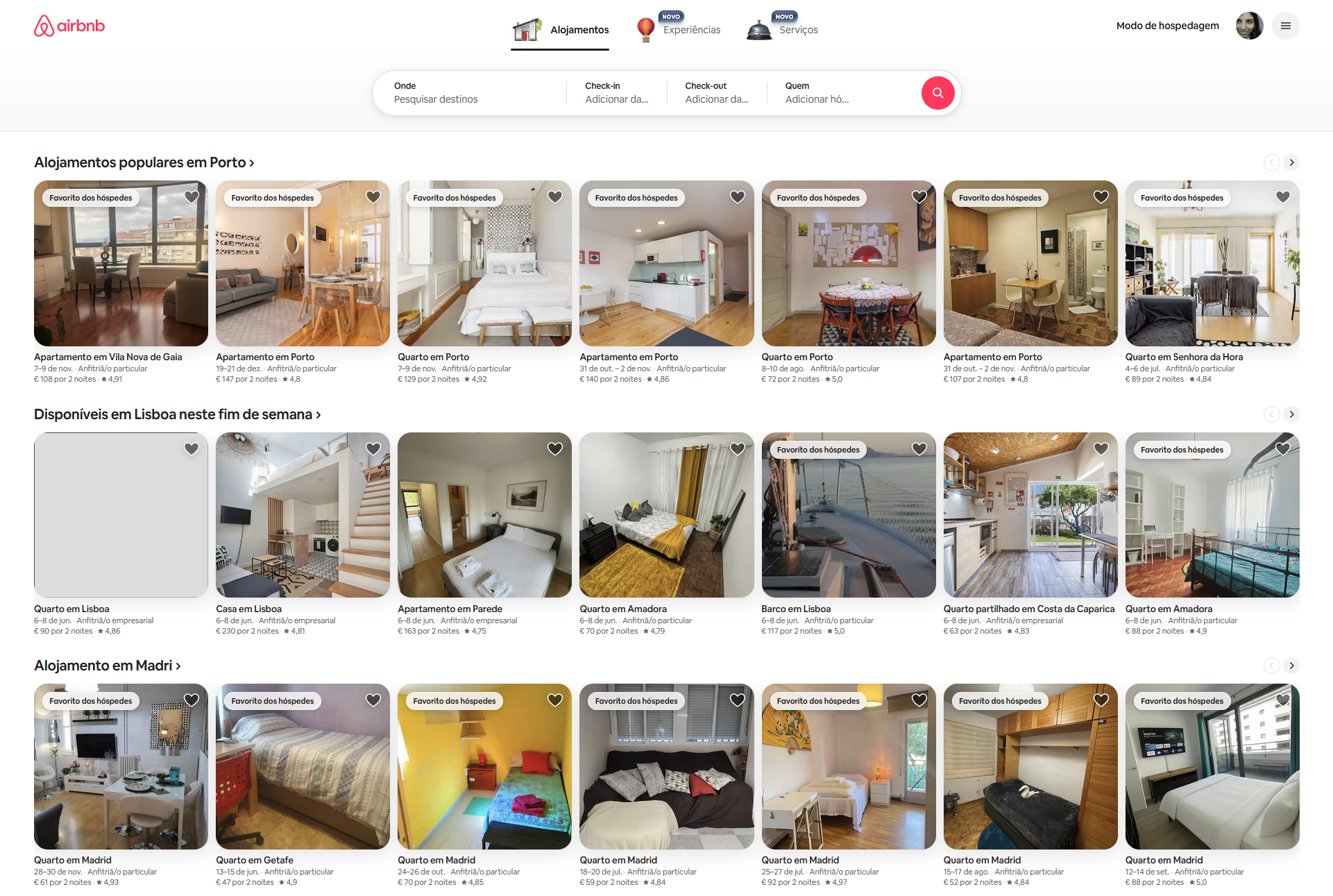


Figura 8 - Mudança para o modo hospedagem

De seguida foi explorada a experiência do utilizador em modo hóspede, com especial atenção à funcionalidade de pesquisa. Antes mesmo de iniciar a procura, o sistema permite ao utilizador selecionar o tipo de oferta pretendido, distinguindo entre alojamentos, experiências ou serviços relacionados com viagens (figura 9). Esta separação inicial ajuda a direcionar a navegação de forma clara e eficiente.

A funcionalidade de pesquisa (figura 9) em si começa com uma barra principal onde o hóspede introduz os dados essenciais: destino, datas da estadia e número de hóspedes. Após esta pesquisa inicial, os resultados apresentados podem ser refinados através de múltiplos filtros, incluindo: faixa de preços, tipo de alojamento (apartamento, vivenda, cabana, etc.), comodidades disponíveis (Wi-Fi, ar condicionado, cozinha equipada, entre outras), regras da casa (permitir fumar, animais, festas), política de cancelamento e ainda idioma falado pelo anfitrião.

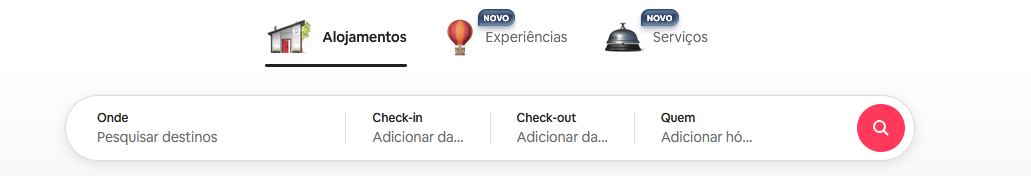


Figura 9 - Seleção do tipo de oferta e funcionalidade de pesquisa

A visualização detalhada de um anúncio disponibiliza ao hóspede informações completas sobre o alojamento, tais como descrição textual, galeria de imagens, localização no mapa, regras da casa, política de cancelamento e comentários de hóspedes anteriores. Além disso, é possível visualizar dados do anfitrião, com uma avaliação geral e histórico de resposta.

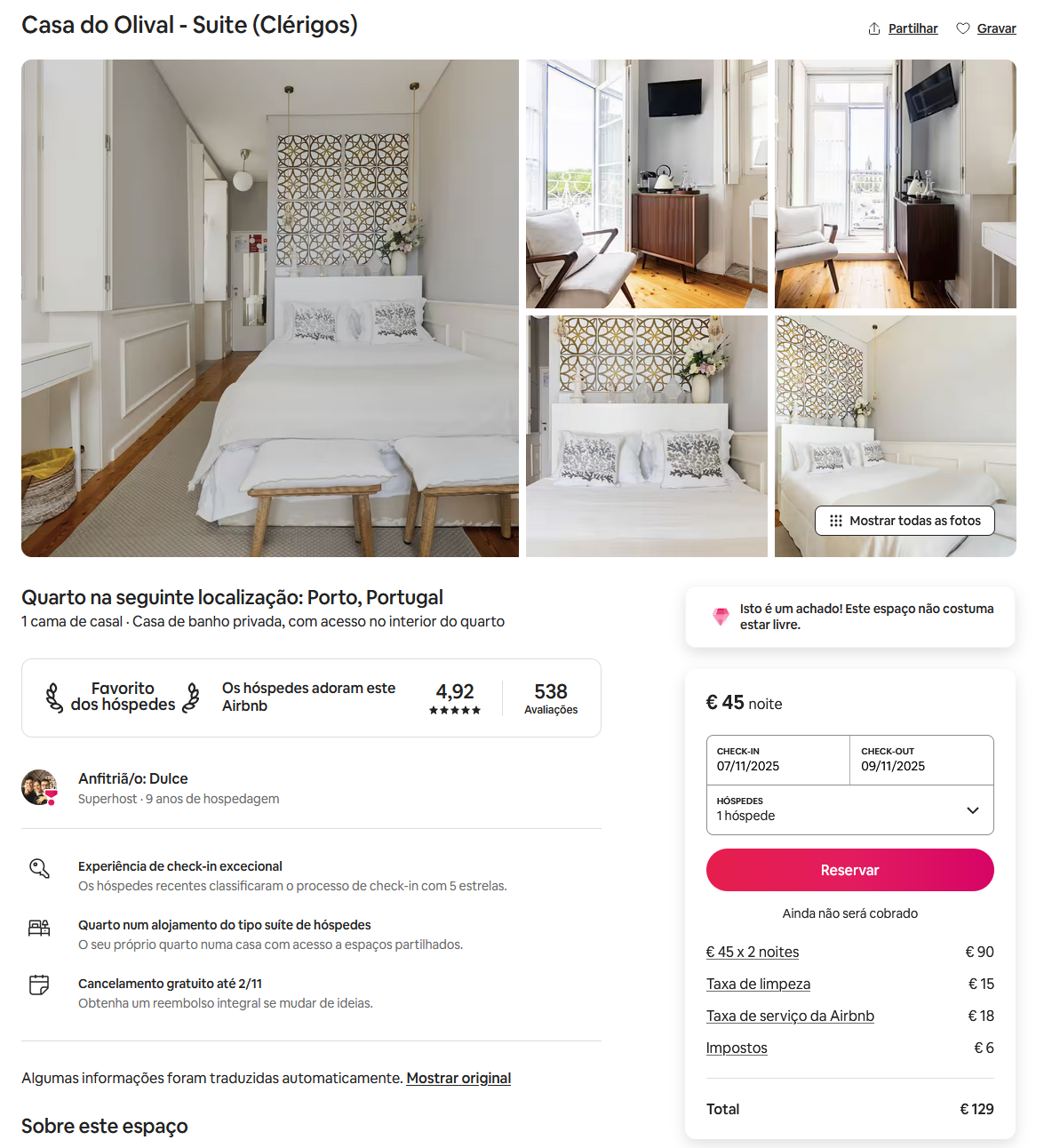


Figura 10 - Visualização detalhada de um exemplo de anúncio

Um dos elementos observados com particular interesse foi a funcionalidade de mensagens privadas entre hóspedes e anfitriões, acessível a partir de um botão direto nos anúncios. Esta comunicação permite esclarecer dúvidas antes da reserva e é fundamental para reforçar a confiança entre utilizadores.

Por fim, destacamos o sistema de avaliação mútua entre hóspedes e anfitriões, com atribuição de classificações e comentários públicos após cada estadia. Este mecanismo de reputação tem impacto direto na confiança e visibilidade dos utilizadores, sendo um fator diferenciador da plataforma.

### Simulação de entrevista a utilizadores

De forma a obter uma melhor compreensão das necessidades e expectativas dos potenciais utilizadores da plataforma, foi desenvolvida uma entrevista para utilizadores reais do serviço Airbnb. Esta técnica, embora não aplicada na prática neste projeto académico, seria fundamental num cenário real para identificar requisitos funcionais e não funcionais antes do desenvolvimento da plataforma. A entrevista seria orientada para recolher feedback sobre as necessidades, expectativas e preocupações dos utilizadores, de modo a produzir um sistema alinhado com as suas preferências. As perguntas formuladas teriam o intuito de explorar o comportamento atual e as necessidades relacionadas com a reserva de alojamentos, mesmo sem uma aplicação concreta já desenvolvida.

Para além das perguntas da entrevista, seria igualmente importante recolher um formulário de perfil do utilizador, incluindo características pessoais e profissionais e contexto de utilização, como por exemplo: idade, profissão, frequência de viagens e experiência prévia com plataformas similares. Estas informações permitem segmentar os utilizadores e enriquecer a análise dos requisitos.

Adicionalmente, seria essencial obter o consentimento informado dos participantes, garantindo que os dados recolhidos são tratados com confidencialidade e apenas para o propósito do desenvolvimento da aplicação. Esta prática assegura a conformidade ética e legal na recolha de informações.

**Formulário de Perfil do Utilizador**

Idade: \_\_\_\_\_\_\_\_\_\_\_

Profissão: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Utiliza plataformas similares (ex.: Booking, Airbnb)? Sim ( ) Não ( )

Se sim, quais? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Frequência de viagens:

( ) Nunca viajo

( ) Raramente (1 a 2 vezes por ano)

( ) Ocasionalmente (3 a 5 vezes por ano)

( ) Frequentemente (mais de 5 vezes por ano)

Experiência com reservas online:

( ) Nenhuma experiência – nunca utilizei plataformas online para reservar alojamentos

( ) Pouca experiência – usei uma ou duas vezes

( ) Experiência moderada – uso ocasionalmente, em viagens

( ) Experiência elevada – faço reservas online regularmente

( ) Outra: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Consentimento Informado**

O participante declara ter sido devidamente informado sobre os objetivos desta entrevista, realizada no âmbito do desenvolvimento de uma plataforma de reserva de alojamentos. Compreende que os dados recolhidos, incluindo o perfil e as respostas fornecidas, serão utilizados exclusivamente para fins de desenvolvimento desta plataforma e respeitarão integralmente os princípios de confidencialidade e proteção de dados. O participante poderá, a qualquer momento, recusar ou interromper a sua participação.

□ Concordo em participar na entrevista.

Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Data:\_\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_\_

**Perguntas da Entrevista**

* + 1. Quando procura alojamento para as suas viagens, o que considera mais importante?
    2. Quais são os principais fatores que o levam a escolher um alojamento?
    3. Que tipo de informações gosta de ter antes de fazer uma reserva?
    4. Como gostaria que fosse o processo de pesquisa e seleção de alojamentos?
    5. Que tipo de funcionalidades consideraria úteis numa nova plataforma de reserva de alojamentos?
    6. Como gostaria de poder comunicar com o responsável pelo alojamento antes da reserva?
    7. Que preocupações teria em relação à segurança dos dados e pagamentos?
    8. O que tornaria a experiência mais simples e agradável para si?
    9. Já utilizou serviços semelhantes (ex.: Booking, Airbnb)? Que pontos positivos e negativos destaca?
    10. Se pudesse imaginar a “plataforma ideal” para reservas de alojamentos, como seria?
    11. O que pensa sobre a possibilidade de integrar novas funcionalidades, como experiências (atividades locais) e serviços adicionais? Considera que ter tudo junto na mesma plataforma seria uma vantagem?
    12. Se respondeu sim na pergunta anterior, que tipo de funcionalidades ou serviços gostaria de ver integrados?

### User stories para descrever funcionalidades centradas no utilizador

Para desenvolver user stories de forma completa e realista, começámos por criar personas que representam os diferentes tipos de utilizadores da plataforma. Estas personas ajudaram a identificar as necessidades e objetivos dos utilizadores, permitindo escrever user stories centradas no utilizador.

Para cada user story, definimos critérios de aceitação usando a estrutura Dado que, Quando, Então, cobrindo tanto os cenários de sucesso (em que tudo corre bem) como os de exceção ou erro (em que o sistema deve responder adequadamente). Esta abordagem assegura que cada funcionalidade está bem definida e pode ser validada com facilidade.

Personas:

P1: **Maria Saraiva (Hóspede)**

Maria Saraiva é uma engenheira informática de 33 anos, que adora viajar e procura alojamentos confortáveis e modernos para as suas férias. Como utilizadora desta plataforma, ela valoriza a facilidade de reservar estadias, a possibilidade de comunicar diretamente com os anfitriões para esclarecer dúvidas e a capacidade de poder ler avaliações de outros hóspedes. O seu principal objetivo é planear as suas viagens de forma simples e segura para garantir uma experiência agradável.

P2: **João Almeida (Anfitrião)**

João Almeida tem 59 anos e é professor. Ele possui uma casa de férias que pretende alugar para alcançar uma fonte de rendimento adicional. Na plataforma, João valoriza poder gerir reservas de forma eficaz, comunicar com hóspedes e manter o seu anúncio atualizado. O seu objetivo principal é possuir uma ocupação do seu alojamento perto de 100%, proporcionando uma experiência positiva aos seus hóspedes, de forma a obter boas avaliações.

User Stories Hóspede:

* US1: Como Maria, quero registar-me para reservar alojamentos.

Critérios de Aceitação:

Cenário 1 (sucesso):

**Dado** que Maria preenche o formulário de registo com dados válidos,

**Quando** submete o formulário,

**Então** a conta é criada e Maria pode aceder.

Cenário 2 (excepção):

**Dado** que Maria preenche o formulário com dados inválidos ou incompletos,

**Quando** tenta submeter o formulário,

**Então** o sistema exibe uma mensagem de erro indicando os campos a corrigir.

* US2: **Como** Maria, **quero** pesquisar alojamentos com filtros **para** encontrar o que melhor se adapta à minha viagem.

Critérios de Aceitação:

Cenário 1 (sucesso):

**Dado** que Maria seleciona filtros de pesquisa,

**Quando** clica em "Pesquisar",

**Então** o sistema apresenta alojamentos correspondentes aos filtros.

Cenário 2 (excepção):

**Dado** que nenhum alojamento corresponde aos filtros,

**Quando** clica em "Pesquisar",

**Então** o sistema informa que não existem resultados.

* US3: **Como** Maria, **quero** reservar um alojamento **para** garantir a minha estadia.

Critérios de Aceitação:

Cenário 1 (sucesso):

**Dado** que Maria seleciona um alojamento disponível e preenche os dados,

**Quando** clica em "Reservar",

**Então** a reserva é confirmada e uma notificação é enviada.

Cenário 2 (excepção):

**Dado** que as datas estão ocupadas,

**Quando** Maria tenta finalizar a reserva,

**Então** o sistema exibe uma mensagem de que não há disponibilidade.

* US4: **Como** Maria, **quero** comunicar com João **para** esclarecer dúvidas sobre o alojamento.

Critérios de Aceitação:

Cenário 1 (sucesso):

**Dado** que Maria está autenticada,

**Quando** envia uma mensagem ao anfitrião,

**Então** a mensagem é enviada e João recebe uma notificação.

* US5: **Como** Maria, **quero** avaliar a minha estadia **para** ajudar outros utilizadores.

Critérios de Aceitação:

Cenário 1 (sucesso):

**Dado** que Maria finalizou a estadia,

**Quando** acede à área de avaliações e insere um comentário e nota,

**Então** a avaliação é publicada e visível para outros utilizadores.

User Stories Anfitrião:

* US6: **Como** João, **quero** criar um anúncio **para** a receber hóspedes na minha casa.

Critérios de Aceitação:

Cenário 1 (sucesso):

**Dado** que João preenche todos os dados do anúncio (nome, categoria, preço, endereço, datas disponíveis, imagens),

**Quando** carrega em “Submeter”,

**Então** a plataforma publica o novo anúncio após um período de verificação.

Cenário 2 (excepção – dados incompletos):

**Dado** que o João não preenche todos os campos obrigatórios do anúncio,

**Quando**  tenta submeter o anúncio,

**Então** a plataforma impede a publicação e exibe uma mensagem a solicitar os dados em falta.

Cenário 3 (excepção – dados inválidos após verificação):

**Dado** que o anúncio contém dados inválidos detetados no processo de verificação,

**Quando**  a plataforma processa o anúncio,

**Então** a publicação é recusada e João é informado com uma mensagem de erro.

* US7: **Como** João, **quero** gerir as reservas dos hóspedes **para** controlar a disponibilidade e confirmar estadias.

Critérios de Aceitação:

Cenário 1 (sucesso):

**Dado** que João recebe um pedido de reserva,

**Quando** clica em aceitar ou recusar,

**Então** a decisão é registada e o hóspede é notificado.

* US8: **Como** João, **quero** editar o anúncio **para** manter a informação atualizada.

Critérios de Aceitação:

Cenário 1 (sucesso):

**Dado** que João acede ao seu anúncio,

**Quando** edita o anúncio e confirma,

**Então** as alterações são guardadas e aplicadas.

* US9: **Como** João, **quero** comunicar com Maria **para** responder a dúvidas e dar informações.

Critérios de Aceitação:

Cenário 1 (sucesso):

**Dado** que João está autenticado,

**Quando** envia uma mensagem ao hóspede,

**Então** a mensagem é entregue e Maria é notificada.

### Casos de uso (UML) para representação gráfica das interações

## Especificação de requisitos

Para a produção do documento de requisitos deste projeto, adaptámos o standard IEEE/ANSI 830-1993 (Sheldon Linker, 2021), alinhando-o com os princípios do Manifesto Ágil, que valoriza a documentação apenas na medida em que esta seja útil e eficaz. “Valorizamos mais software funcional do que documentação abrangente” (Mike Beedle, 2001).

### Documento de requisitos

**Introdução**

**Objetivos do documento**

Este documento especifica os requisitos do sistema a ser desenvolvido, inspirado na plataforma Airbnb, com o objetivo de simular as funcionalidades essenciais e processos do ponto de vista dos utilizadores (anfitrião e hóspede).

**Âmbito do produto**

Desenvolver uma plataforma de reserva de alojamentos, experiências e serviços que permita o registo e gestão de utilizadores, publicação e pesquisa de alojamentos, experiências, serviços, reservas, avaliações e comunicações entre utilizadores.

**Definições, acrónimos e abreviaturas**

RF: Requisito Funcional

RNF: Requisito Não Funcional

**Referências**

IEEE/ANSI 830-1993

Slides Engenharia de Requisitos (Brito, 2025)

**Descrição geral**

**Perspetiva do produto**

A plataforma é um sistema web que permite a interação entre hóspedes e anfitriões para reserva de alojamentos, inspirado na plataforma Airbnb.

**Funções do produto**

Registo e login de utilizadores (hóspedes e anfitriões)

Publicação de anúncios de alojamentos, experiências e serviços

Pesquisa e reserva de alojamentos, experiências e serviços

Avaliação de estadias, experiências e serviços

Comunicação interna entre hóspedes e anfitriões

**Características dos utilizadores**

Hóspedes: utilizadores registados que reservam alojamentos, experiências e serviços

Anfitriões: utilizadores que oferecem alojamentos, experiências e serviços

Administrador: gestão global do sistema

**Considerações, pressupostos e dependências**

O projeto é académico e contempla apenas as fases de análise e desenho, não incluindo a fase de produção.

**Especificação de requisitos**

**Requisitos Funcionais (RF)**

RF1 – O sistema deve permitir o registo de utilizadores (hóspedes e anfitriões).

RF2 – O sistema deve permitir o login de utilizadores (hóspedes e anfitriões).

RF3 – O sistema deve permitir a publicação de anúncios de alojamentos, experiências ou serviços com fotos, descrição, preço e localização.

RF4 – O sistema deve permitir a pesquisa de alojamentos, experiências ou serviços com filtros.

RF5 – O sistema deve permitir reservas.

RF6 – O sistema deve permitir o cancelamento de reservas.

RF7 – O sistema deve permitir avaliações de estadias.

RF8 – O sistema deve permitir comunicação interna entre hóspedes e anfitriões.

RF9 – O sistema deve permitir a edição e remoção de anúncios por anfitriões.

RF10 – O sistema deve permitir a gestão de perfil.

RF11 – O sistema deve permitir o pagamento de reservas.

**Requisitos Não Funcionais (RF)**

RNF1 – O sistema deve assegurar segurança com encriptação.

RNF2 – O sistema deve ter uma interface intuitiva.

RNF3 – O sistema deve possuir um tempo de resposta máximo de 2 segundos.

RNF4 – O sistema deve cumprir com a legislação de proteção de dados.

RNF5 – O sistema deve ser de fácil manutenção para permitir futuras alterações.

# 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)

# Conclusão

A adoção da metodologia Scrum ao longo deste projeto permitiu-nos compreender de forma prática e aprofundada o que está envolvido no planeamento e desenvolvimento de software. A organização do trabalho em sprints, com tarefas bem definidas e revisões constantes, proporcionou uma visão clara da evolução do projeto e ajudou-nos a desenvolver competências fundamentais de gestão de equipa.

O trabalho em grupo revelou-se particularmente enriquecedor, pois tornou evidente a complexidade da coordenação entre membros, bem como a importância de uma comunicação eficaz e da partilha de responsabilidades. A utilização de ferramentas CASE, com destaque para o quadro Trello, foi fundamental na fase inicial do projeto: ajudou a desbloquear obstáculos organizacionais e funcionou como um verdadeiro catalisador na estruturação do trabalho e no acompanhamento do progresso.

Inicialmente, optámos por redigir o relatório final colaborativamente através do SharePoint Word, permitindo edição simultânea por todos os membros. No entanto, esta abordagem revelou-se limitada no que respeita à partilha de versões e histórico de alterações com a docente, o que nos levou a migrar para o GitHub, que se revelou uma solução mais robusta para controlo de versões e colaboração técnica.

Relativamente à gestão de tempo, esta foi maioritariamente bem-sucedida. Contudo, o último sprint acabou por ter uma duração inferior à prevista (3 dias em vez de uma semana), exigindo um esforço adicional para cumprir os objetivos estabelecidos.

Este projeto foi importante para consolidar os conhecimentos adquiridos durante as aulas, nomeadamente ao nível da engenharia de requisitos e da modelação de sistemas com diagramas UML. A partir da identificação das funcionalidades e das necessidades dos utilizadores, elaborámos diagramas de casos de uso, classes e sequência, garantindo a coerência entre a análise do problema e o desenho da solução. Esta prática reforçou a nossa compreensão sobre a importância de alinhar os modelos com os requisitos funcionais e não funcionais, tal como é exigido em ambientes reais de desenvolvimento de software.

# Links para plataformas de apoio

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

GitHub:

<https://github.com/23240rita/EngenhariaSoftware>

# Referências Bibliográficas

Brito, I. (2025). *Engenharia de Requisitos.* Obtido de https://cms.ipbeja.pt/pluginfile.php/8479/mod\_resource/content/8/aula2EngeRequisitos24.pdf

Mike Beedle, A. v. (2001). *Manifesto for Agile Software Development.* Obtido de https://agilemanifesto.org/

Sheldon Linker, M. B. (2021). *Requirements Engineering - Appendix C: IEEE 830 Template.* Obtido de https://press.rebus.community/requirementsengineering/back-matter/appendix-c-ieee-830-template/

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