



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



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

Resumo	6
Abstract.....	7
1 Introdução.....	8
2 Fase de Planeamento Inicial e Gestão do Projeto.....	9
2.1 Definição de objetivos	9
2.2 Metodologia de desenvolvimento de Software.....	9
2.2.1 Simulação de Contexto Profissional.....	10
2.2.2 Reflexão sobre Desenvolvimento Real	13
2.3 Ferramentas CASE e plataformas de apoio	14
2.4 Calendarização e organização da equipa.....	15
2.4.1 Sprint 1 – Planeamento e Iniciação do Projeto	15
2.4.2 Sprint 2 – Fase de análise.....	16
2.4.3 Sprint 3 – Fase de Desenho	17
2.4.4 Gestão do projeto	17
3 Fase de Análise.....	18
3.1 Engenharia de Requisitos.....	18
3.2 Técnicas de obtenção de Requisitos.....	19
3.2.1 Análise do sistema real Airbnb e estudo da utilização da aplicação	19
3.2.2 Simulação de entrevista a utilizadores	25
3.2.3 User stories.....	27
3.2.4 Casos de Uso e Diagrama de Casos de Uso	31
3.3 Especificação de requisitos	33
3.3.1 Documento de requisitos.....	33

4	Fase de Desenho.....	36
4.1	Diagrama de classes.....	36
4.2	Diagramas de Sequência.....	38
4.3	Diagrama de estados	45
5	Conclusão.....	46
6	Links para plataformas de apoio	47
7	Referências Bibliográficas.....	47

Índice de Figuras

Figura 1- Diagrama desenvolvido no Figma para ilustrar o fluxo de trabalho Scrum com iterações por sprint	11
Figura 2 - Quadro Trello durante o Sprint 2.....	12
Figura 3 – GitHub para controlo de versões e manutenção de histórico da documentação e diagramas	14
Figura 4 - Criação de conta de hópede na plataforma Airbnb, recolha de dados pessoais	20
Figura 5 - Criação de conta de hópede na plataforma Airbnb, criação do primeiro anúncio.....	20
Figura 6 – Interface Principal do utilizador anfitrião.....	21
Figura 7 - Seleção do modo viagem no perfil anfitrião	22
Figura 8 - Mudança para o modo hospedagem	22
Figura 9 - Seleção do tipo de oferta e funcionalidade de pesquisa	23
Figura 10 - Visualização detalhada de um exemplo de anúncio.....	24
Figura 11 - Diagrama Casos de Uso para o sistema em estudo	32
Figura 12 - Diagrama de Classes para o sistema em estudo	37
Figura 13 - Diagrama de Sequência Registrar Utilizador	38
Figura 14 - Diagrama de Sequência Efetuar Login	39
Figura 15 - Diagrama de Sequência Editar Perfil	39
Figura 16 - Diagrama de Sequência Publicar Anúncio	40
Figura 17 - Diagrama de Sequência Editar Anúncio.....	40
Figura 18 - Diagrama de Sequência Remover Anúncio	41
Figura 19 - Diagrama de Sequência Pesquisar Anúncios	41
Figura 20 - Diagrama de Sequência Enviar Mensagem ao Anfitrião	41
Figura 21 - Diagrama de Sequência Criar Reserva	42
Figura 22 - Diagrama de Sequência Efetuar Pagamento.....	42
Figura 23 - Diagrama de Sequência Cancelar Reserva.....	43
Figura 24 - Diagrama de Sequência Efetuar Reembolso.....	43
Figura 25 - Diagrama de Sequência Trocar Mensagem.....	43
Figura 26 - Diagrama de Sequência Avaliar Reserva.....	44
Figura 27 - Diagrama de Estados para a Entidade Reserva.....	45

Resumo

Este relatório documenta o trabalho desenvolvido no âmbito da unidade curricular de Engenharia de Software, cujo objetivo foi simular o processo de desenvolvimento de um sistema de reservas de alojamentos, baseado na plataforma Airbnb. A equipa recorreu à metodologia ágil Scrum, aplicando princípios de planeamento incremental, definição e análise de requisitos, modelação de casos de uso e, posteriormente, o desenho da solução, com recurso a diagramas UML — nomeadamente de classes e de sequência — para representar a estrutura e o comportamento do sistema. O projeto incidiu principalmente nas fases iniciais do ciclo de vida do software, com especial atenção à análise do domínio do problema e à representação estruturada da solução proposta. A observação direta da plataforma Airbnb serviu como base para fundamentar e justificar decisões técnicas. Foram utilizadas ferramentas CASE para apoiar o processo de desenvolvimento, nomeadamente o Visual Paradigm para a modelação UML, o Trello para a organização de tarefas e o GitHub para controlo de versões. Este relatório apresenta o processo seguido, as decisões tomadas e os artefactos produzidos ao longo do projeto.

Abstract

This report documents the work carried out within the scope of the Software Engineering course, aimed at simulating the development process of an accommodation booking system, based on the Airbnb platform. The team adopted the agile Scrum methodology, applying principles of incremental planning, requirements elicitation and analysis, use case modelling, and, subsequently, system design using UML diagrams — specifically class and sequence diagrams — to represent the structure and behavior of the system. The project focused primarily on the initial stages of the software development life cycle, with particular emphasis on the analysis of the problem domain and the structured representation of the proposed solution. Direct observation of the Airbnb platform served as a basis to support and justify technical decisions. CASE tools were used to support the development process, namely Visual Paradigm for UML modelling, Trello for task management, and GitHub for version control. This report presents the process followed, the decisions made, and the artefacts produced throughout the project.

1 Introdução

O trabalho desenvolvido no âmbito da unidade curricular de Engenharia de Software teve como principal objetivo simular, de forma prática e colaborativa, as fases iniciais do processo de desenvolvimento software de um sistema de reservas de alojamentos, tendo como referência o funcionamento da plataforma Airbnb.

O projeto iniciou-se com a escolha do modelo de desenvolvimento, tendo sido adotada a metodologia ágil Scrum, pela sua natureza iterativa, flexível e centrada na entrega incremental. Com base nesta metodologia, procedeu-se à organização da equipa e ao planeamento inicial, definindo-se os papéis dos elementos, a distribuição de tarefas e os prazos de execução. A gestão do tempo, das tarefas e da comunicação foi assegurada com recurso a ferramentas colaborativas, como o Trello, e a coordenação técnica do trabalho foi mantida através do controlo de versões com o GitHub. Este processo é encontrado no capítulo 2 do relatório.

O corpo do trabalho divide-se em duas fases principais: análise do problema e desenho da solução.

A fase de análise, correspondente ao processo de Engenharia de Requisitos, envolveu a aplicação de diversas técnicas de obtenção de requisitos, como a análise do sistema real (a plataforma Airbnb), o esboço de entrevista a utilizadores e a formulação de user stories para descrever as funcionalidades esperadas. Os requisitos funcionais foram organizados em casos de uso, complementados pelo respetivo Diagrama de Casos de Uso UML, permitindo uma representação clara das interações entre os diferentes tipos de utilizadores e o sistema. Esta análise culminou na elaboração de um documento de requisitos, que serviu de base à fase seguinte. Esta fase está documentada no capítulo 3.

Na fase de desenho, os requisitos foram traduzidos em modelos UML que representam a estrutura e o comportamento do sistema. O Diagrama de Classes permitiu identificar os principais elementos do sistema, os seus atributos, métodos e relações, enquanto os Diagramas de Sequência ilustraram as interações entre os componentes em diferentes cenários. A modelação destes diagramas foi orientada pelo padrão arquitetural Controller–Model–View (CMV), que serviu de base para a organização lógica do sistema, facilitando a transição para a implementação. Além disto, foi também efetuado um diagrama de estados para a entidade Reserva. Esta parte encontra-se descrita no capítulo 4.

A modelação e documentação técnica foram apoiadas por ferramentas CASE, destacando-se o Visual Paradigm para a criação dos modelos UML, o que garantiu uma representação formal e padronizada dos artefactos produzidos.

2 Fase de Planeamento Inicial e Gestão do Projeto

2.1 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, planejar 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.

2.2 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.

2.2.1 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

Figura 1, apresentamos uma captura de ecrã do quadro trello durante o sprint 2 para ilustrar o processo de organização, evolução e monitorização do progresso das tarefas.

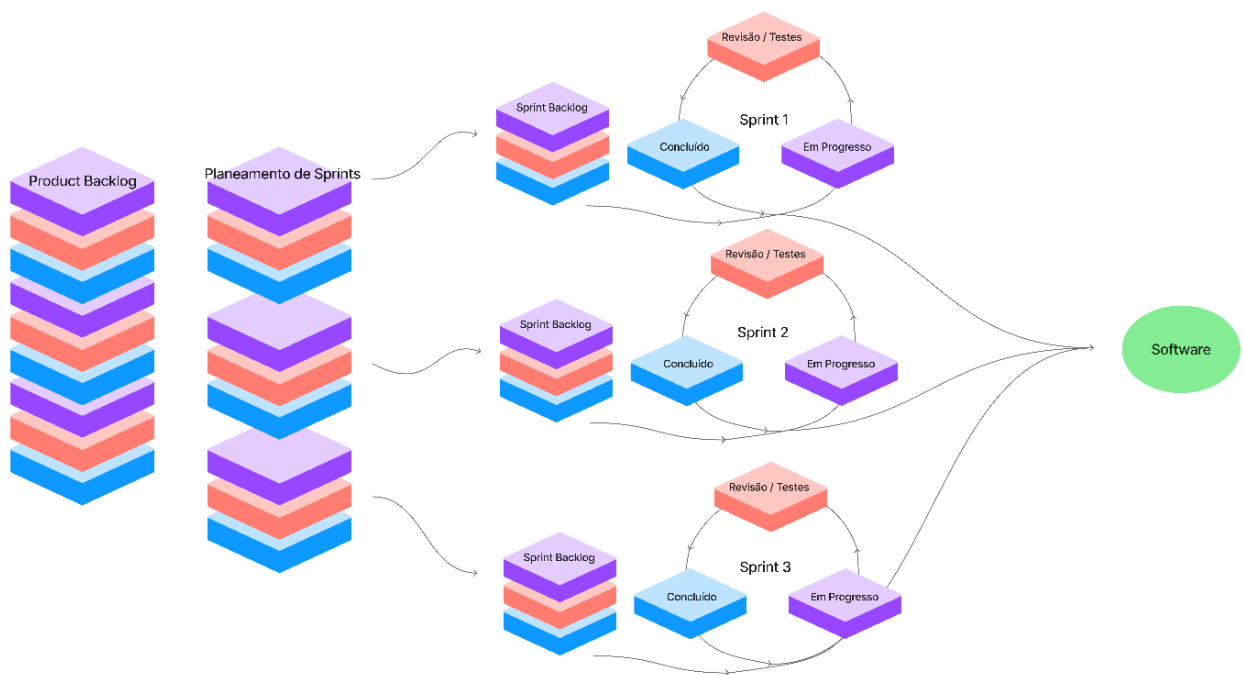


Figura 1- Diagrama desenvolvido no Figma para ilustrar o fluxo de trabalho Scrum com iterações por sprint

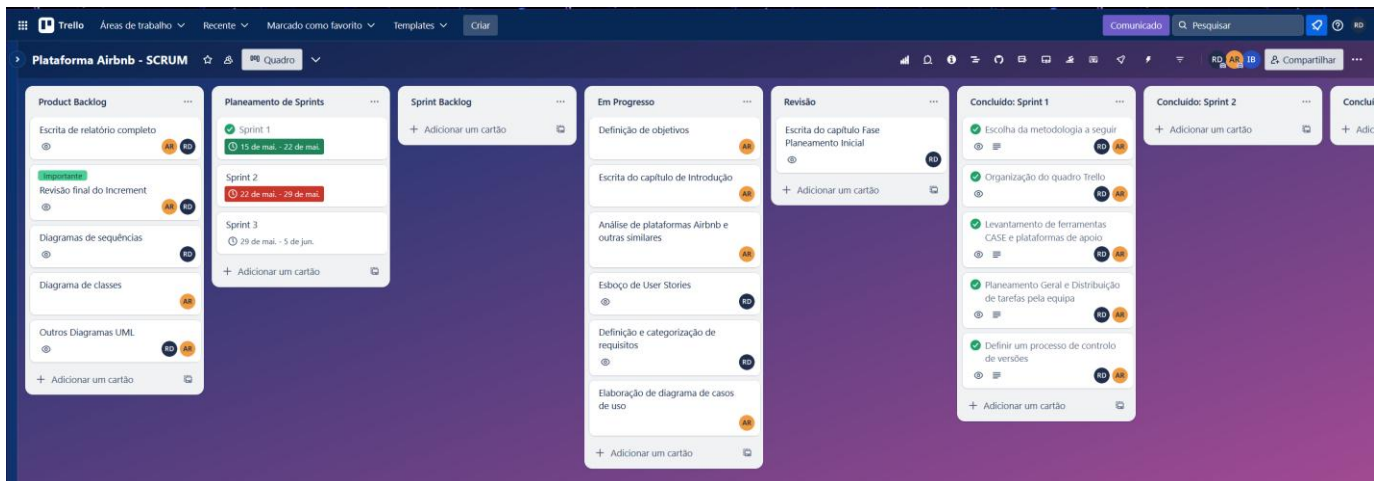


Figura 2 - Quadro Trello durante o Sprint 2

- **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 retrospectiva 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.

2.2.2 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.

2.3 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.

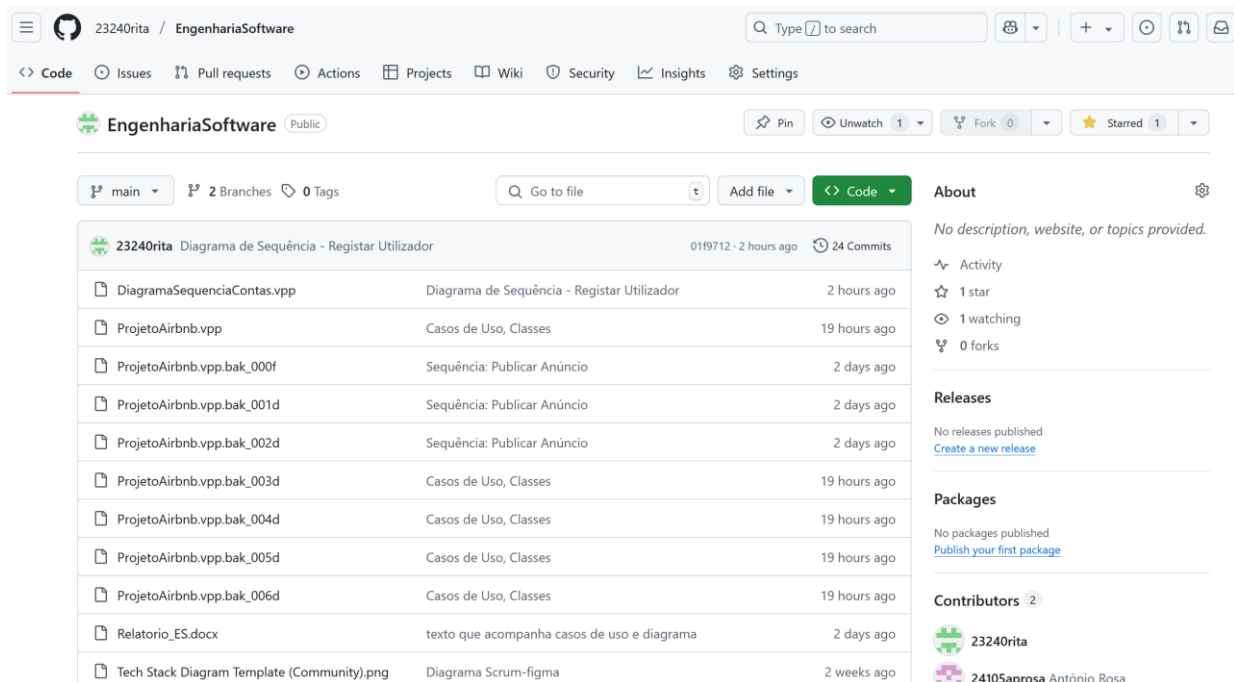


Figura 3 – GitHub para controlo de versões e manutenção de histórico da documentação e diagramas

2.4 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:

2.4.1 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

2.4.2 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

2.4.3 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

2.4.4 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.

3 Fase de Análise

3.1 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, recorreremos 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 diagrama de casos de uso (UML), que representa 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.

3.2 Técnicas de obtenção de Requisitos

3.2.1 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 acadêmico 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.

← Finalize o registo

Nome legal

Nome que consta no documento de identificação

Apelido que consta no documento de identificação

Certifique-se de que corresponde ao nome que consta no seu documento de identificação oficial. Se tiver outro nome, pode adicionar um [primeiro nome preferido](#).

Data de nascimento

Data de nascimento

Tem de ter pelo menos 18 anos para se registar. O seu aniversário não será partilhado com outras pessoas que usam a Airbnb.

Informações de contacto

E-mail

Vamos enviar-lhe um email com as confirmações e recibos da viagem.

Ao seleccionar **Concordar e continuar**, concordo com [Termos de Serviço](#), [Termos de Serviço de Pagamentos](#) e [Política de Não Discriminação](#) da Airbnb e reconheço a [Política de Privacidade](#).

Concordar e continuar

A Airbnb enviar-lhe-á ofertas apenas para membros, inspiração, e-mails de marketing e notificações push. Pode optar por deixar de receber estas informações a qualquer momento nas configurações da sua conta ou diretamente a partir da notificação de marketing.

☐ Não quero receber mensagens de marketing da Airbnb.

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

1 Fale-nos sobre o seu espaço

Partilhe algumas informações básicas, como a localização e quantos hóspedes podem ficar.



2 Faça com que se destaque

Adicione 5 ou mais fotos, um título e uma descrição. Nós ajudamos com o resto.



3 Conclua o seu anúncio e publique-o

Escolha um preço inicial, verifique alguns detalhes e publique o anúncio.



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.

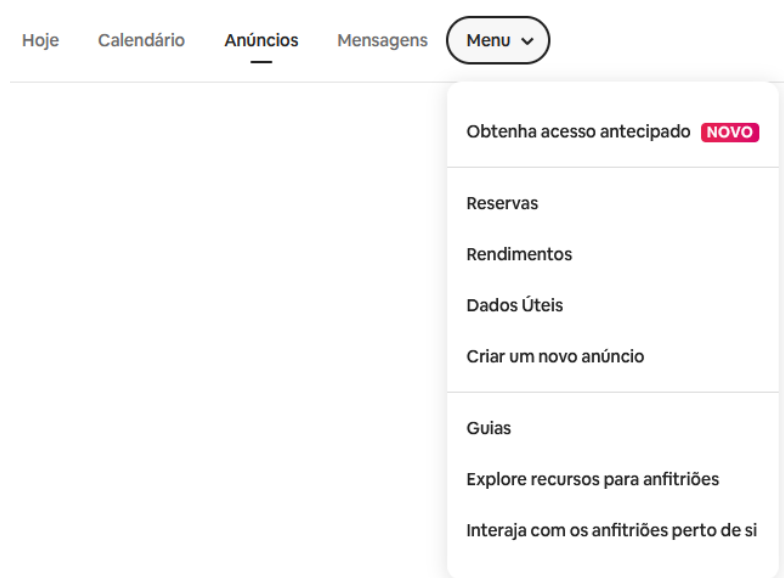


Figura 6 – Interface Principal do utilizador anfitrião

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.

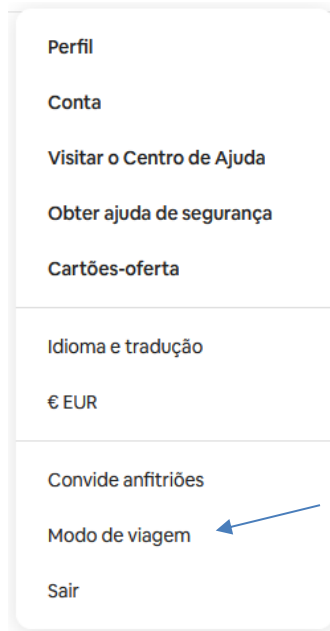


Figura 7 - Seleção do modo viagem no perfil anfitrião

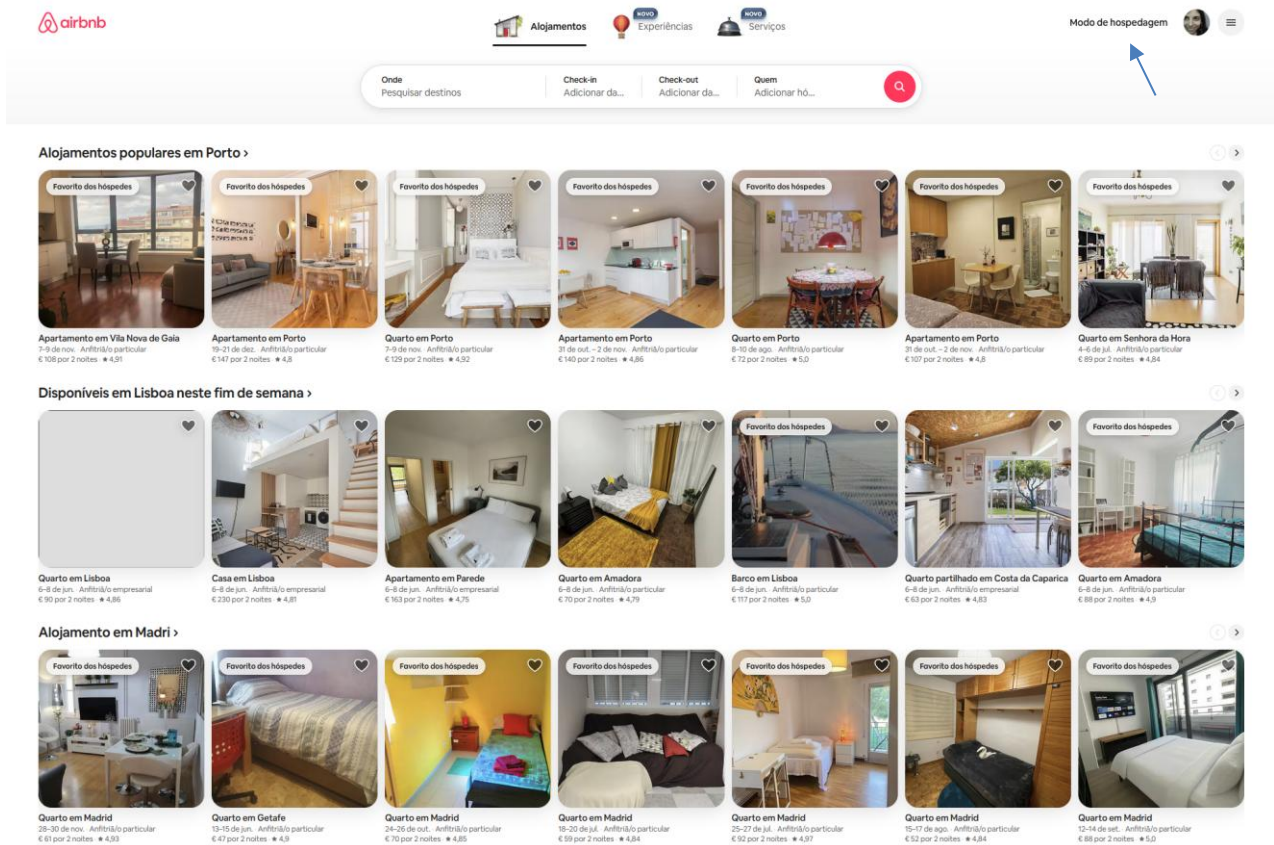


Figura 8 - Mudança para o modo hospedagem

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 (topo da Figura 9). Esta separação inicial ajuda a direcionar a navegação de forma clara e eficiente.

A funcionalidade de pesquisa (base da 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.

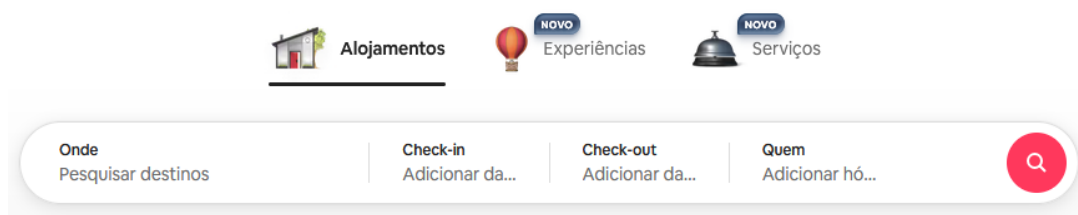


Figura 9 - Seleção do tipo de oferta e funcionalidade de pesquisa

A visualização detalhada de um anúncio (

Figura 10) 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.

Casa do Olival - Suite (Clérigos)


[Partilhar](#) [Gravar](#)



Quarto na seguinte localização: Porto, Portugal

1 cama de casal · Casa de banho privada, com acesso no interior do quarto

 **Favorito dos hóspedes**

 **Os hóspedes adoram este Airbnb**

4,92
★★★★★

538
Avaliações



Anfitriã/o: Dulce

Superhost · 9 anos de hospedagem



Experiência de check-in excepcional

Os hóspedes recentes classificaram o processo de check-in com 5 estrelas.



Quarto num alojamento do tipo suite de hóspedes

O seu próprio quarto numa casa com acesso a espaços partilhados.




Cancelamento gratuito até 2/11

Obtenha um reembolso integral se mudar de ideias.

Algumas informações foram traduzidas automaticamente. [Mostrar original](#)

Sobre este espaço

 **Isto é um achado! Este espaço não costuma estar livre.**

€ 45 noite

CHECK-IN 07/11/2025	CHECK-OUT 09/11/2025
HÓSPEDES 1 hóspede	

Reservar

Ainda não será cobrado

€ 45 x 2 noites	€ 90
Taxa de limpeza	€ 15
Taxa de serviço da Airbnb	€ 18
Impostos	€ 6
Total	€ 129

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.

3.2.2 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?

3.2.3 User stories

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 (exceção):

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 (exceção):

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 (exceção):

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 (exceção – 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 exhibe uma mensagem a solicitar os dados em falta.

Cenário 3 (exceção – 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.

3.2.4 Casos de Uso e Diagrama de Casos de Uso

Para complementar o levantamento de requisitos e as user stories, foram definidos casos de uso, que descrevem de forma estruturada as principais interações entre os utilizadores e o sistema. Cada caso de uso identifica um objetivo específico do utilizador, detalhando os passos necessários para atingir esse objetivo, bem como os cenários alternativos ou exceções que possam ocorrer. Esta abordagem permite clarificar o comportamento esperado do sistema e orientar a modelação subsequente. Os cenários base e alternativos, assim como as respetivas pré e pós-condições, foram preenchidos diretamente nos detalhes de cada caso de uso no Visual Paradigm. Estes elementos encontram-se integrados no ficheiro do projeto (.vpp) e podem ser consultados através da opção "*Open Use Case Details...*" da ferramenta.

Com base nos casos de uso identificados, foi elaborado o Diagrama de Casos de Uso UML, apresentado na Figura 11. Este diagrama oferece uma representação gráfica das interações entre os atores primários (hóspede e anfitrião) e secundário (banco) e os principais serviços disponibilizados pela aplicação. A visualização das relações entre atores e casos de uso facilita a compreensão das operações previstas no sistema e constitui uma base de referência para a fase de desenho.

Para melhorar a organização e legibilidade do Diagrama de Casos de Uso, os casos foram agrupados em *packages* correspondentes a áreas funcionais do sistema: Contas de Utilizadores, Reservas, e Anúncios. Esta separação reflete a modularidade lógica do sistema e facilita a identificação das responsabilidades associadas a cada tipo de utilizador.

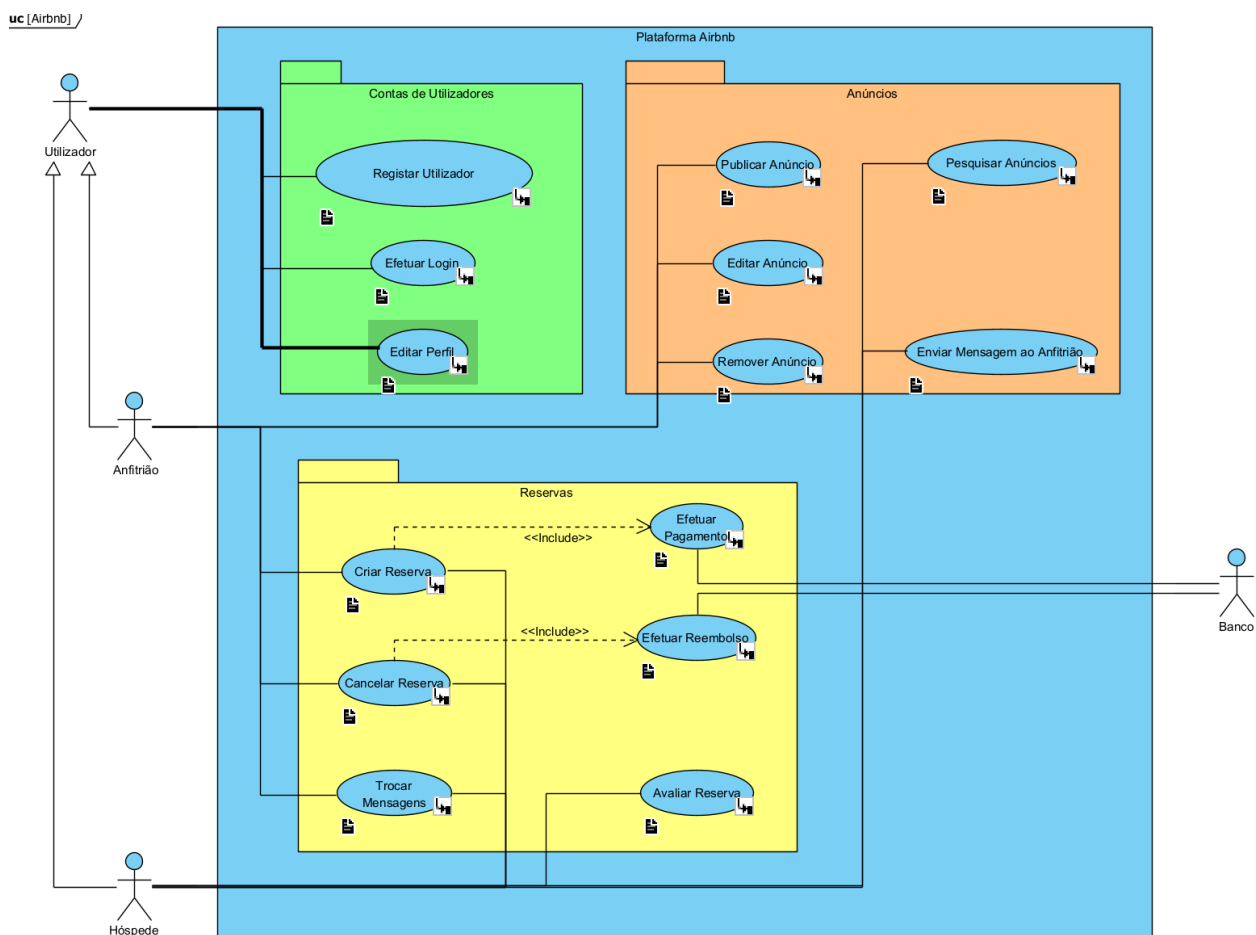


Figura 11 - Diagrama Casos de Uso para o sistema em estudo

3.3 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).

3.3.1 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.

4 Fase de Desenho

Esta fase corresponde à transição entre a análise de requisitos e o desenho técnico da solução, sendo essencial para estruturar a implementação futura de forma coerente com os objetivos definidos. O desenho do sistema foi orientado pelo padrão arquitetural Controller–Model–View (CMV), que nos permitiu organizar logicamente os componentes da aplicação, separando responsabilidades entre a lógica de negócio, a interface com o utilizador e o controlo das interações.

A modelação foi realizada com recurso à notação UML (Unified Modeling Language), permitindo representar graficamente a estrutura e o comportamento do sistema. Foram elaborados os seguintes diagramas:

Diagrama de Classes – utilizado para modelar as principais entidades do sistema, os seus atributos, métodos e relações, fornecendo uma visão estruturada da lógica de dados e regras de negócio;

Diagramas de Sequência – aplicados aos casos de uso identificados na fase de análise, com o objetivo de representar a dinâmica da interação entre objetos ao longo do tempo;

Diagrama de Estados – utilizado especificamente para descrever os diferentes estados possíveis de uma reserva, assim como as transições entre esses estados em função das ações do sistema ou do utilizador.

Estes diagramas garantem uma representação consistente dos requisitos identificados, servindo como ponte entre a análise funcional e a implementação técnica da solução.

4.1 Diagrama de classes

O Diagrama de Classes, apresentado na Figura 12, descreve a estrutura estática do sistema, funcionando como a base para a organização lógica da aplicação. Este modelo define as principais classes, os seus atributos, métodos e os relacionamentos entre elas, permitindo compreender como os diferentes elementos do sistema interagem entre si.

A construção do diagrama foi orientada pelo padrão arquitetural Model–View–Controller (MVC), garantindo uma separação clara de responsabilidades nas três camadas da aplicação:

- **Model (Modelo):** Inclui as classes que representam os dados e as regras de negócio. Neste grupo encontram-se as classes Utilizador, ContaBancaria, Anuncio, Reserva, TabelaPrecos, Avaliacao, Imagem e Mensagem.

- **Controller (Controlador):** Abrange os processadores responsáveis pela lógica de coordenação das operações do sistema. Estão representados os controladores AccountProcessor, ListingProcessor, MessageProcessor, PaymentProcessor, ReservationProcessor, ReviewProcessor e SearchProcessor, cada um com métodos específicos que implementam as funcionalidades principais da aplicação.

- **View (Vista):** Compreende as interfaces utilizadas para interação com o utilizador. Estão incluídas as interfaces MenuConta, MenuAnfitriao, MenuMensagem, MenuReservas, MenuPesquisa e Display.

A estrutura apresentada assegura uma organização coesa e modular do sistema, facilitando a implementação orientada a objetos e promovendo a reutilização e manutenção do código.

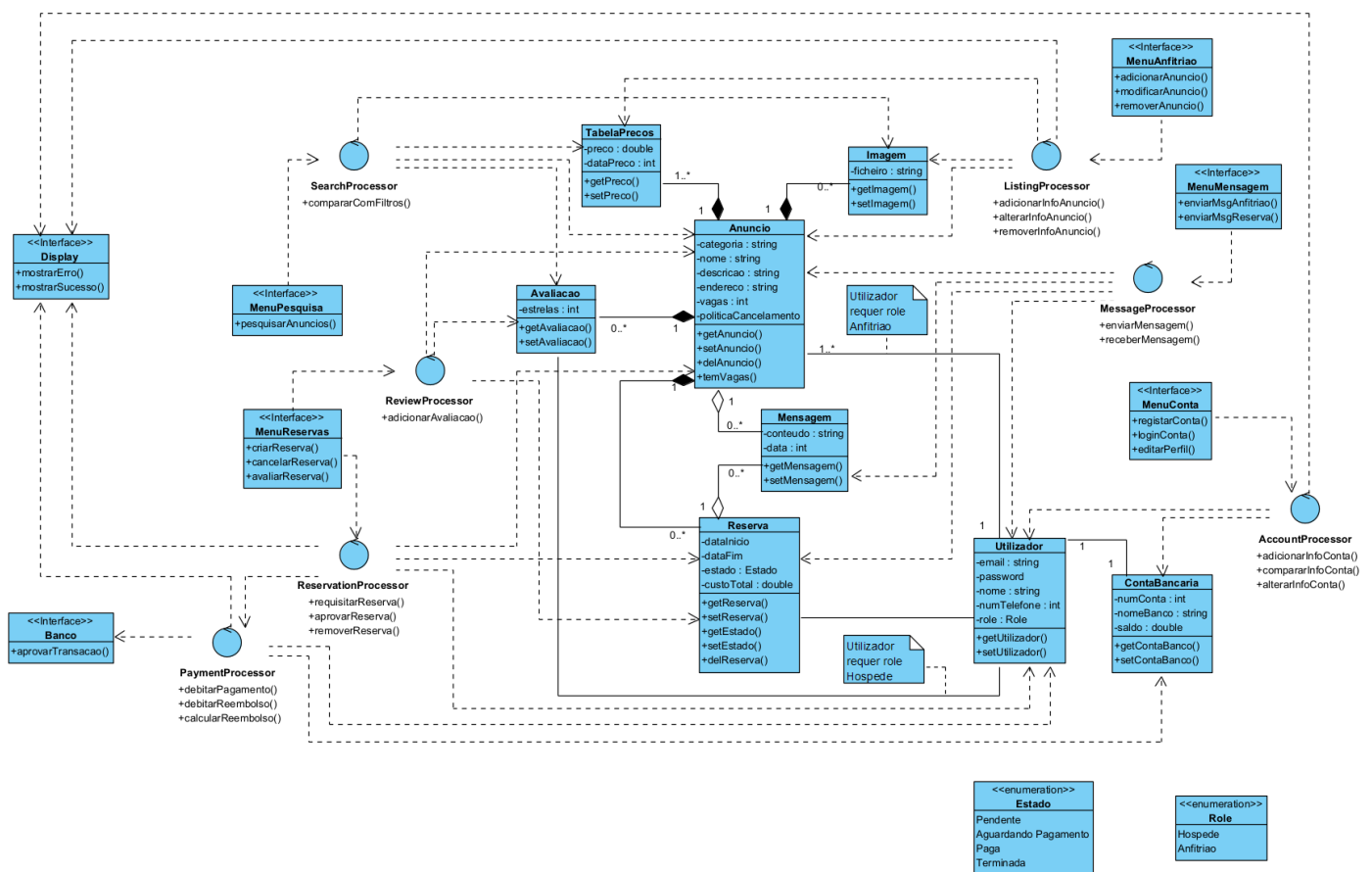


Figura 12 - Diagrama de Classes para o sistema em estudo

4.2 Diagramas de Sequência

Os Diagramas de Sequência foram elaborados com o objetivo de representar, de forma dinâmica, as interações entre os diferentes objetos do sistema ao longo do tempo, em resposta às ações dos utilizadores. Para cada caso de uso identificado na fase de análise, foi desenvolvido um diagrama correspondente, permitindo visualizar a sequência de mensagens trocadas entre a interface, os controladores e os elementos do modelo. Esta abordagem garante a consistência com os requisitos definidos e facilita a posterior implementação das funcionalidades. A seguir apresentam-se os diagramas de sequência obtidos:

- para a Package Contas de Utilizadores temos na Figura 13 - Diagrama de Sequência Registrar Utilizador, na Figura 14 - Diagrama de Sequência Efetuar Logine e na Figura 15 - Diagrama de Sequência Editar Perfil;

- para a Package Anúncios temos na Figura 16 - Diagrama de Sequência Publicar Anúncio, na Figura 17 - Diagrama de Sequência Editar Anúncio, na Figura 18 - Diagrama de Sequência Remover Anúncio, na Figura 19 - Diagrama de Sequência Pesquisar Anúncios e na Figura 20 - Diagrama de Sequência Enviar Mensagem ao Anfitrião;

- para a Package Reservas temos na Figura 21 - Diagrama de Sequência Criar Reserva, na Figura 22 - Diagrama de Sequência Efetuar Pagamento, na Figura 23 - Diagrama de Sequência Cancelar Reserva, na Figura 24 - Diagrama de Sequência Efetuar Reembolso, na Figura 25 - Diagrama de Sequência Trocar Mensagem e finalmente na Figura 26 - Diagrama de Sequência Avaliar Reserva.

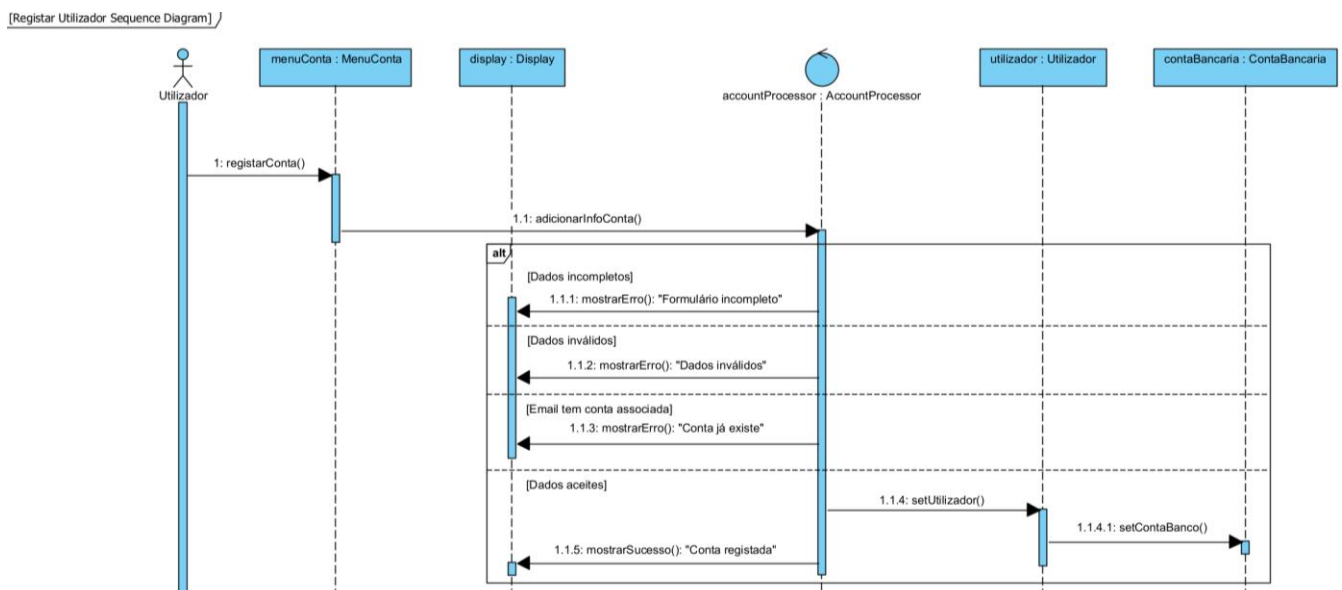


Figura 13 - Diagrama de Sequência Registrar Utilizador

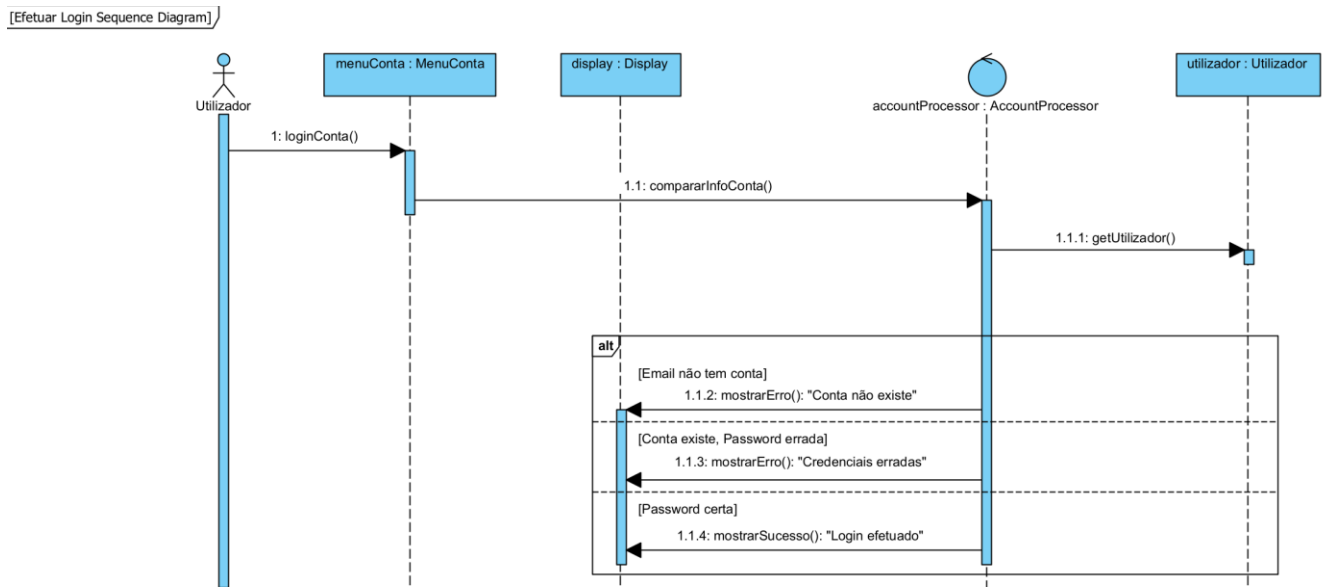


Figura 14 - Diagrama de Sequência Efetuar Login

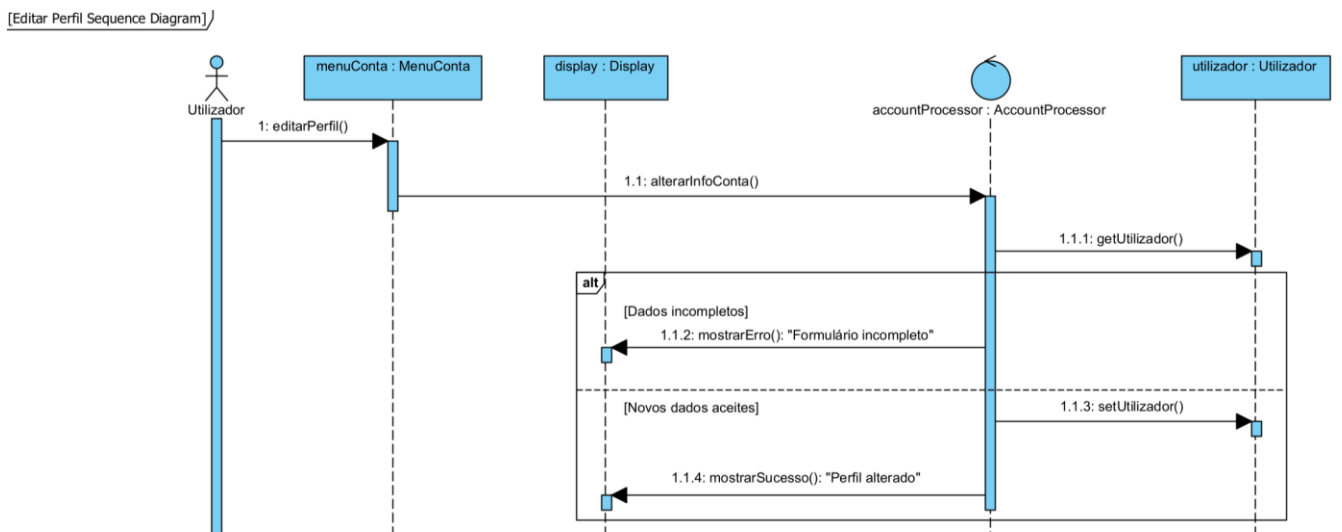


Figura 15 - Diagrama de Sequência Editar Perfil

[Publicar Anúncio Sequence Diagram]

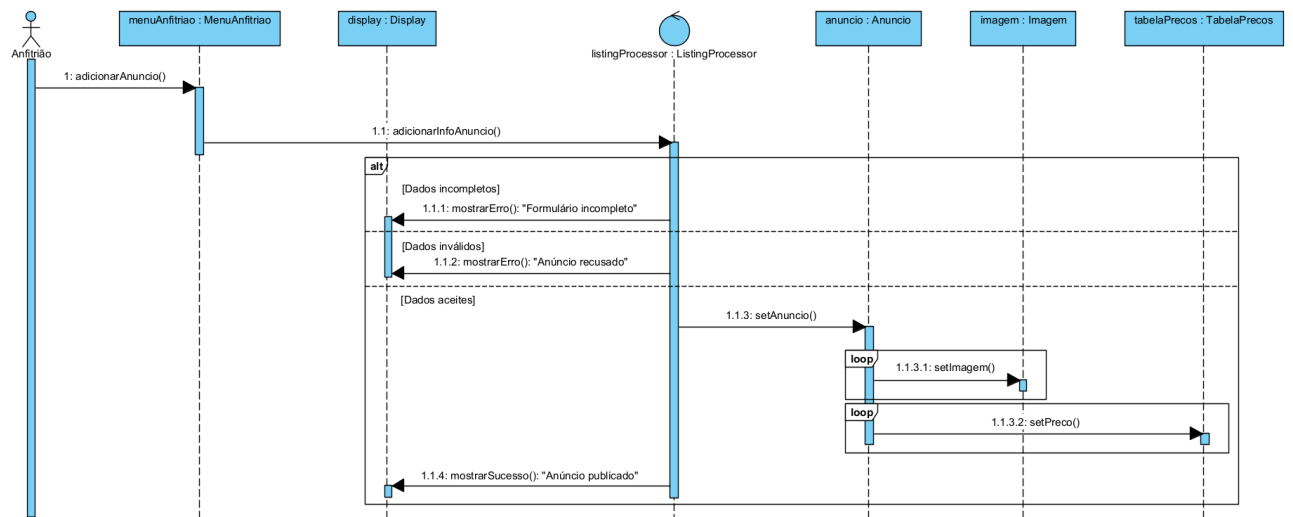


Figura 16 - Diagrama de Sequência Publicar Anúncio

[Editar Anúncio Sequence Diagram]

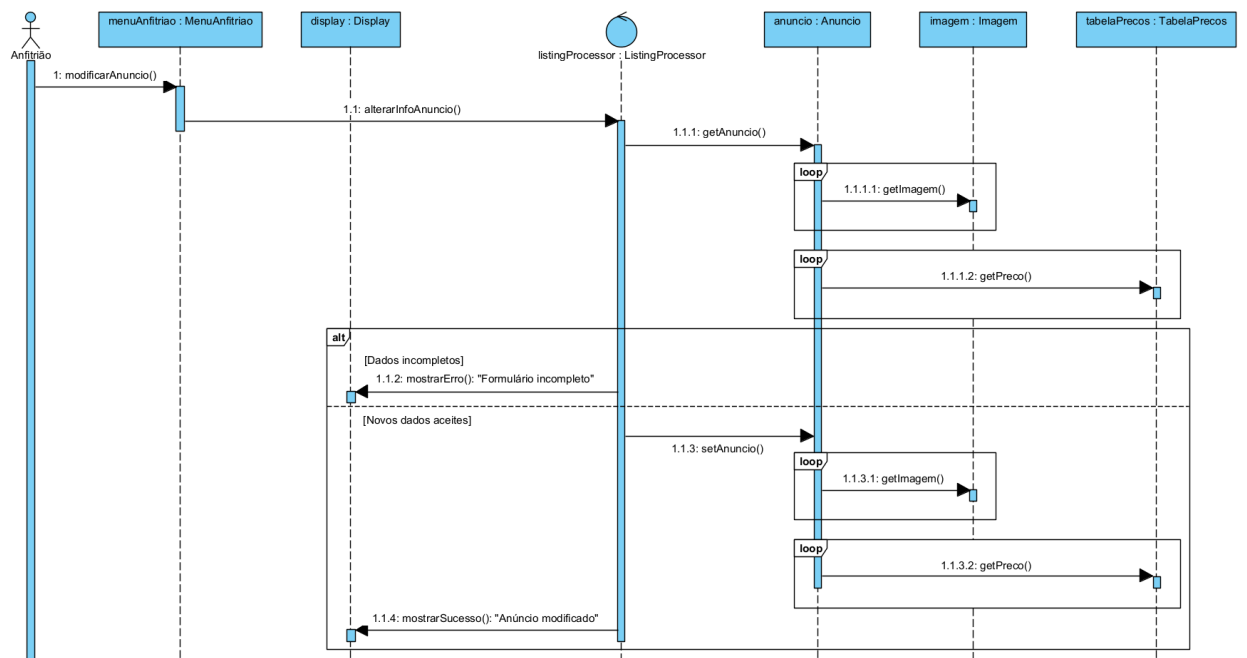


Figura 17 - Diagrama de Sequência Editar Anúncio

[Remover Anúncio Sequence Diagram]

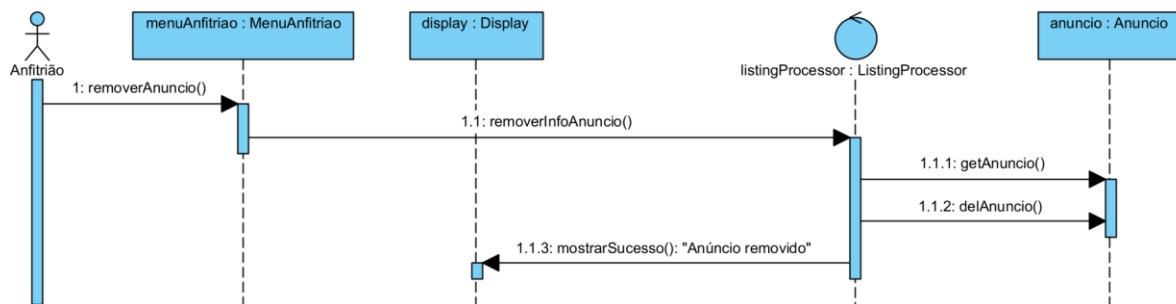


Figura 18 - Diagrama de Sequência Remover Anúncio

[Pesquisar Anúncios Sequence Diagram]

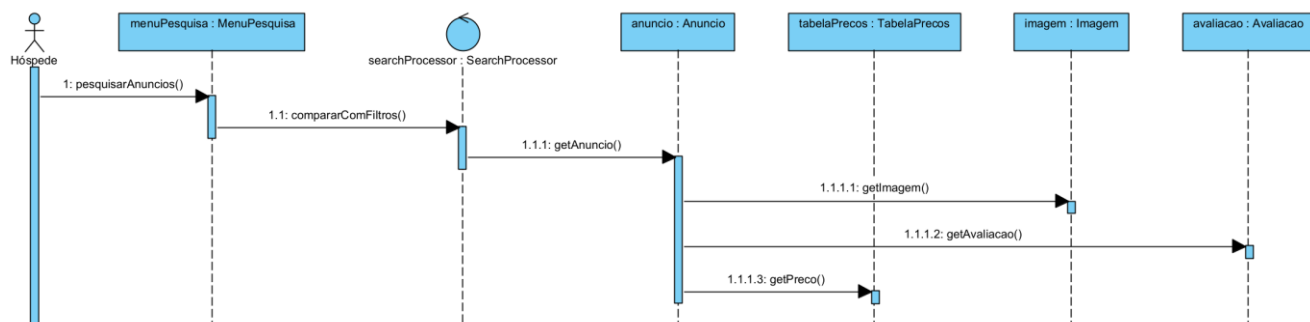


Figura 19 - Diagrama de Sequência Pesquisar Anúncios

[Enviar Mensagem ao Anfitrião Sequence Diagram]

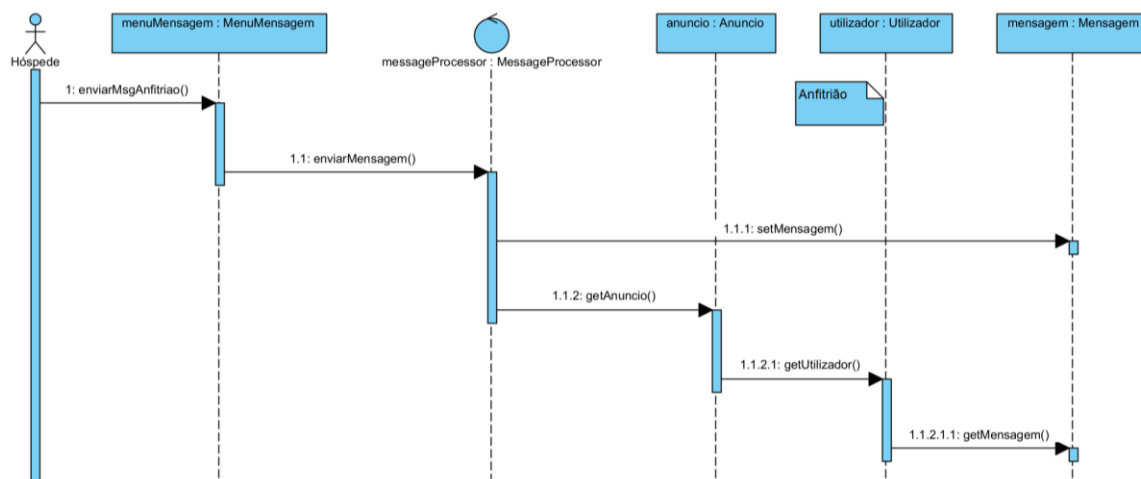


Figura 20 - Diagrama de Sequência Enviar Mensagem ao Anfitrião

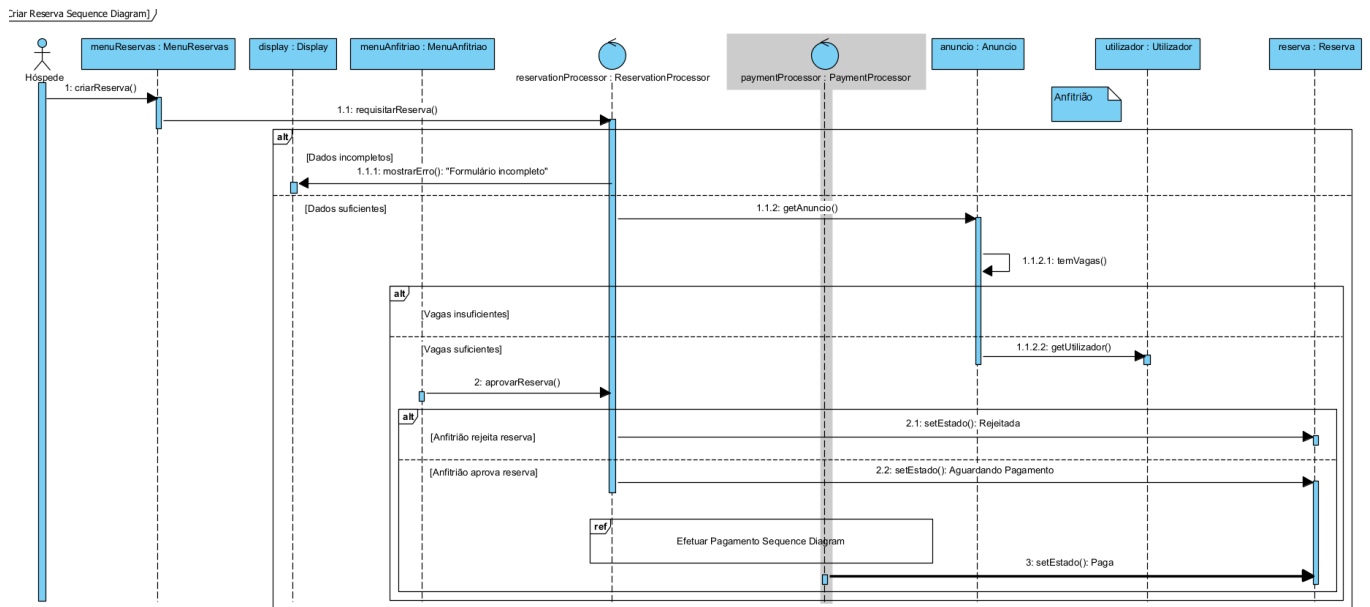


Figura 21 - Diagrama de Sequência Criar Reserva

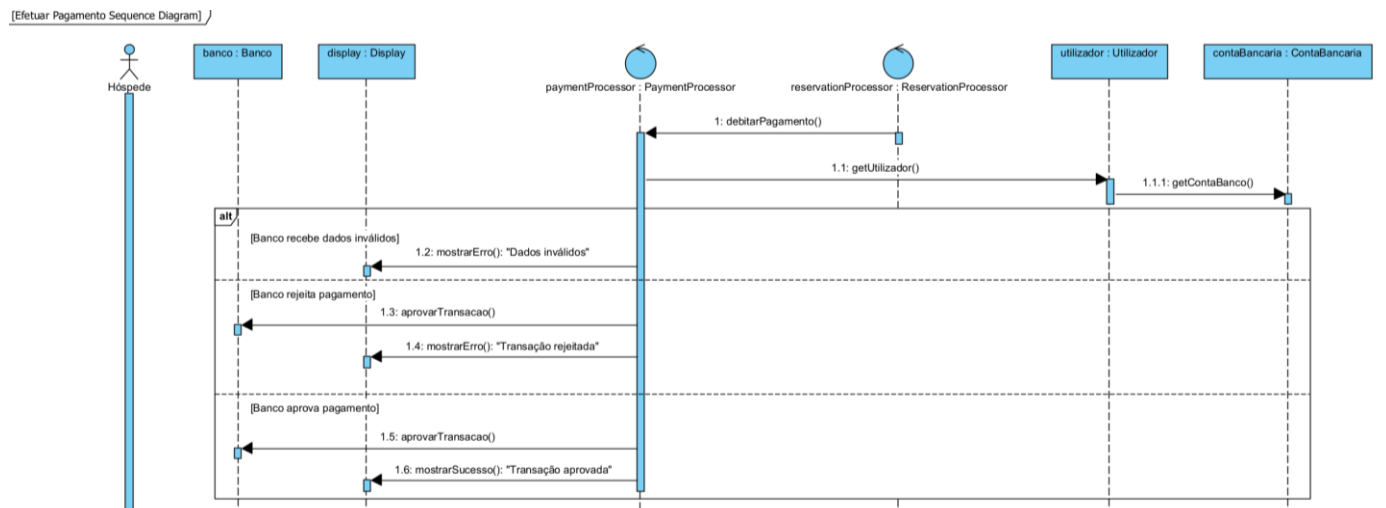


Figura 22 - Diagrama de Sequência Efetuar Pagamento

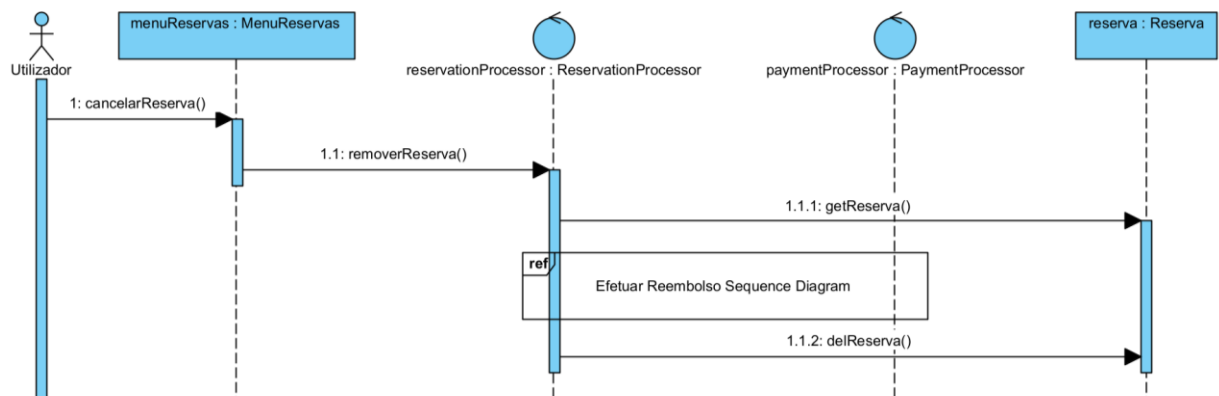


Figura 23 - Diagrama de Sequência Cancelar Reserva

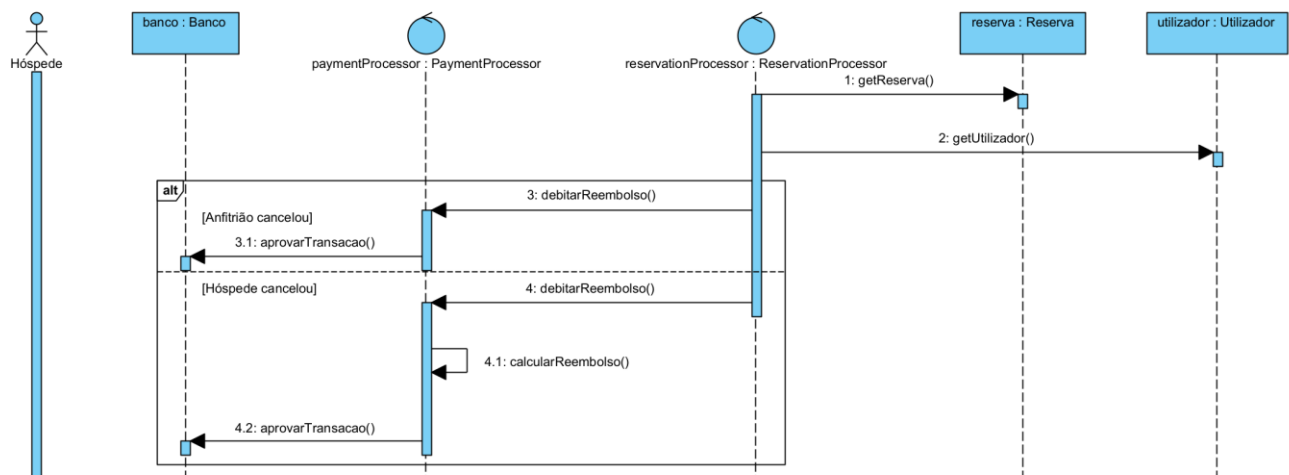


Figura 24 - Diagrama de Sequência Efetuar Reembolso

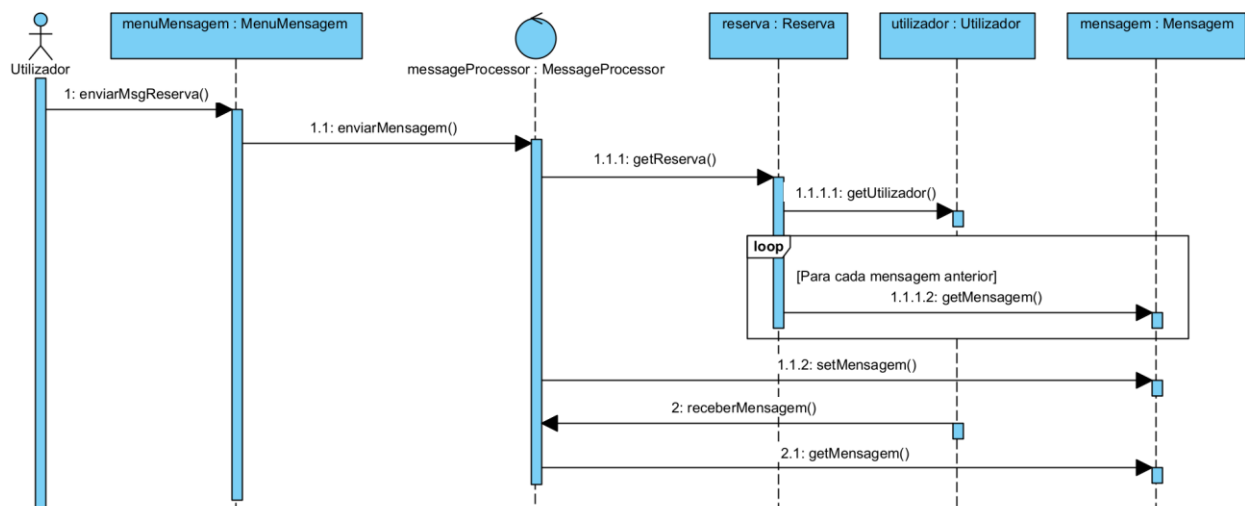


Figura 25 - Diagrama de Sequência Trocar Mensagem

[Avaliar Reserva Sequence Diagram]

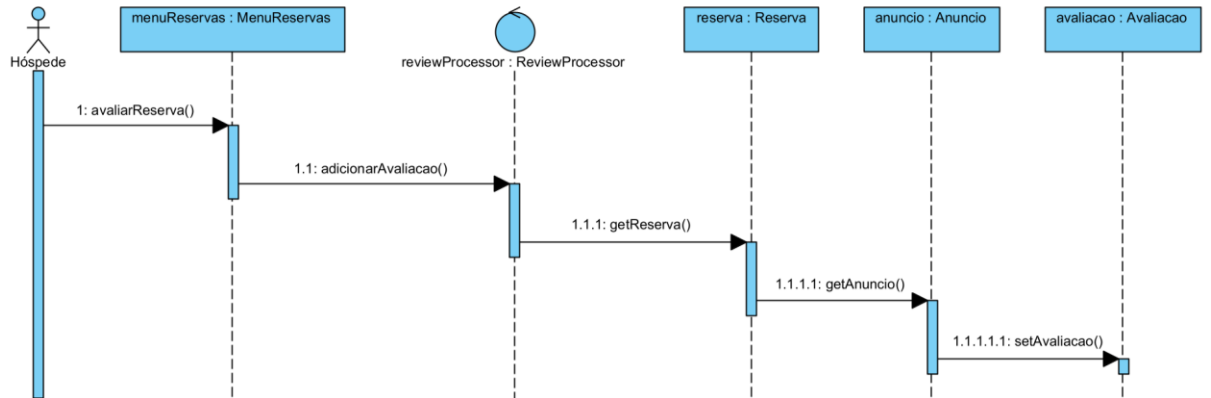


Figura 26 - Diagrama de Sequência Avaliar Reserva

4.3 Diagrama de estados

O Diagrama de Estados (Figura 27) foi desenvolvido com o objetivo de representar o ciclo de vida de uma reserva ao longo do seu percurso funcional no sistema. Este tipo de diagrama é particularmente útil para descrever o comportamento de uma entidade que pode assumir diferentes estados em resposta a eventos específicos.

Neste caso, o modelo incide sobre a entidade Reserva, ilustrando os vários estados possíveis — desde a criação até à sua conclusão ou cancelamento — bem como as transições entre esses estados, desencadeadas por ações dos utilizadores (hóspede e anfitrião) ou por condições do processo. Os eventos considerados incluem a criação da reserva, a aceitação ou rejeição por parte do anfitrião, o pagamento por parte do hóspede, o eventual cancelamento e a conclusão da estadia.

Este diagrama permite clarificar as regras de negócio associadas à gestão de reservas, assegurando que cada cenário relevante está devidamente representado e facilitando, assim, a futura implementação da lógica de estados na aplicação.

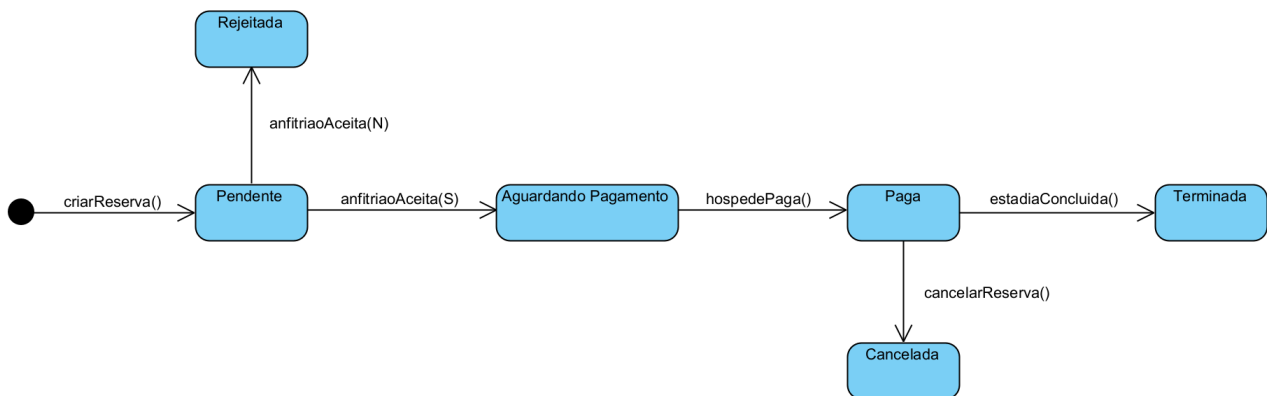


Figura 27 - Diagrama de Estados para a Entidade Reserva

5 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, sequência e estados, 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.

6 Links para plataformas de apoio

Trello:

<https://trello.com/invite/b/67f6c2436b47e53da2146dac/ATTIaa2ae2d6b8d36d7e16784d61bfa9ea092D97186A/plataforma-airbnb-scrum>

GitHub:

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

7 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.