

IPBeja

INSTITUTO POLITÉCNICO
DE BEJA

Escola Superior de Tecnologia e Gestão
Licenciatura em Engenharia Informática

Projeto Final de Engenharia de Software

Airbnb

Ângelo Teresa - 25441 / Denis Cicau - 25442

Beja, 5 de junho de 2025

INSTITUTO POLITÉCNICO DE BEJA

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática

Projeto Final de Engenharia de Software

Airbnb

Ângelo Teresa - 25441 / Denis Cicau - 25442

Orientado por :

Docente : Isabel Sofia Brito, IPBeja

Relatório do Trabalho Prático

Conteúdo

Conteúdo	i
Lista de Figuras	iii
1 Introdução	1
1.1 Objetivos	2
2 Metodologia de Desenvolvimento e Ferramentas de Apoio	3
2.1 Metodologia de Desenvolvimento	3
2.2 Arquitetura utilizada	3
2.3 Ferramentas de Apoio	4
3 Análise	5
3.1 Recolha de Informação	5
3.1.1 Entrevistas a utilizadores de plataformas de alojamento	5
3.1.2 Pesquisa na internet	5
3.1.3 Utilização direta da aplicação Airbnb	6
3.2 Análise da Informação e Identificação de Requisitos	6
3.2.1 Requisitos Funcionais	6
3.2.2 Requisitos Não Funcionais	7
3.3 Personas	8
3.3.1 Persona 1 – Hóspede	8
3.3.2 Persona 2 – Administrador	8
3.3.3 Persona 3 – Anfitrião	9
3.4 User Stories	9
4 Desenho	11
4.1 Elaboração do Diagrama de Casos de Uso [Bri25c]	11
4.2 Elaboração dos Diagramas de Classes UML [Bri25a]	12
4.3 Elaboração dos Diagramas de Sequência UML [Bri25b]	13
5 Gestão Prática do Projeto e Organização da Equipa	15

5.1	Sistema de Controlo de Versões	15
5.1.1	Sistema de Controlo Distribuído com Git	15
5.1.2	Repositório Remoto e Colaboração via GitHub	15
5.2	Gestão e Acompanhamento de Tarefas	16
5.2.1	Github Issue Board	16
5.3	Comunicação e Coordenação da Equipa	17
5.3.1	Coordenação através de Reuniões de Sincronização	17
5.3.2	Comunicação Assíncrona e Síncrona com Discord e Microsoft Teams	17
6	Conclusão	19
	Bibliografia	21

Lista de Figuras

4.1	Diagrama de casos de uso	11
4.2	Diagrama de Classes	12
4.3	Exemplo de diagrama de sequência UML	13
5.1	Quadro de Tarefas (Issue Board) no GitHub	16
5.2	Itens do Quadro de Tarefas	16
5.3	Exemplo de Cartão de Tarefa	17

Capítulo 1

Introdução

O presente trabalho prático insere-se no âmbito da unidade curricular de Engenharia de Software tendo como base o serviço disponibilizado pela plataforma Airbnb. O principal objetivo consiste na conceção e especificação de um sistema de alojamentos temporários que replique, de forma simplificada, o modelo de negócio da referida plataforma.

Neste contexto, procura-se simular o ciclo de vida de desenvolvimento de software, aplicando metodologias e práticas adequadas às fases de análise, desenho e gestão do projeto. A proposta visa não apenas a criação de um sistema funcional, mas também a produção de artefactos técnicos que evidenciem o rigor e a profundidade do trabalho realizado. Pretende-se, assim, elaborar um projeto completo e realista, capaz de satisfazer as necessidades dos principais intervenientes do sistema: anfitriões e hóspedes.

Este relatório técnico documenta as metodologias adotadas, os requisitos identificados, os modelos UML elaborados e os procedimentos de gestão de projeto aplicados ao longo do desenvolvimento da solução.

1.1 Objetivos

Os principais objetivos do trabalho são os seguintes:

- Simular o processo de desenvolvimento de um sistema de gestão de alojamentos de curta duração, com base no modelo da Airbnb;
- Identificar e especificar os requisitos funcionais e não funcionais do sistema;
- Produzir diagramas UML (Casos de Uso, Sequência, Classes) que representem a arquitetura e o comportamento dinâmico do sistema;
- Aplicar boas práticas de engenharia de software, nomeadamente no que diz respeito à gestão de versões e à comunicação eficiente entre os membros da equipa;
- Redigir um relatório técnico que documente detalhadamente todas as fases do projeto, evidenciando a aprendizagem e o conhecimento adquirido;
- Propor e integrar funcionalidades inovadoras que acrescentem valor à solução e diferenciem o projeto de abordagens convencionais.

Capítulo 2

Metodologia de Desenvolvimento e Ferramentas de Apoio

2.1 Metodologia de Desenvolvimento

A metodologia adotada para o desenvolvimento deste projeto foi o **Scrum**, um framework ágil amplamente utilizado em Engenharia de Software. A escolha desta metodologia permitiu uma abordagem iterativa e incremental ao desenvolvimento, promovendo a colaboração contínua entre os membros da equipa, a adaptabilidade a mudanças de requisitos e a entrega progressiva de valor.

Ao longo do projeto, organizámos o trabalho em sprints, com planeamentos regulares para definição de tarefas e objetivos, reuniões diárias para acompanhamento da evolução e sprints de retrospectiva que possibilitaram a reflexão sobre o desempenho da equipa e a identificação de oportunidades de melhoria.

2.2 Arquitetura utilizada

Adicionalmente, foi adotada a arquitetura **MVC (Model-View-Controller)** como padrão de organização do código. Esta arquitetura permitiu uma separação clara entre os dados (modelo), a interface com o utilizador (vista) e a lógica de controlo (controlador), facilitando a manutenção, escalabilidade e reutilização do código. A aplicação desta arquitetura contribuiu significativamente para a clareza estrutural do projeto e para uma melhor organização das responsabilidades de cada componente.

2.3 Ferramentas de Apoio

Para suportar a aplicação da metodologia Scrum e assegurar uma gestão eficiente do projeto, foram utilizadas as seguintes ferramentas:

- **Visual Paradigm** – ferramenta utilizada para a modelação dos diagramas UML (Casos de Uso, Classes, Sequência, entre outros), facilitando o desenho e a visualização da arquitetura e do comportamento do sistema;
- **GitHub** – plataforma usada como sistema de controlo de versões e repositório colaborativo para o código-fonte e artefactos do projeto, promovendo a integração contínua e o trabalho em equipa;

Capítulo 3

Análise

3.1 Recolha de Informação

A recolha de informação revelou-se fundamental para a compreensão das necessidades do sistema e das funcionalidades que este deve disponibilizar. Para tal, foram utilizados os seguintes métodos:

3.1.1 Entrevistas a utilizadores de plataformas de alojamento

Foram realizadas entrevistas a utilizadores de plataformas de alojamento, nomeadamente anfitriões e hóspedes, com o objetivo de identificar as suas expectativas e problemas recorrentes. Estes testemunhos permitiram-nos perceber as principais dificuldades e as funcionalidades mais valorizadas pelos utilizadores.

3.1.2 Pesquisa na internet

A pesquisa na internet envolveu a análise de plataformas de reserva de alojamento, como o Booking.com e o Hoteis.com, que funcionam como intermediários na reserva de acomodações, oferecendo funcionalidades semelhantes ao modelo de negócio da Airbnb. Estas plataformas permitem a pesquisa por tipo de acomodação, localização, preço e avaliações de hóspedes anteriores, além de garantirem métodos de pagamento seguros e sistemas de avaliação mútua para promover a confiança. Através desta análise, identificámos boas práticas e funcionalidades essenciais que serão adaptadas e integradas no nosso próprio sistema, sem intenção de replicar ou copiar qualquer solução existente.

3.1.3 Utilização direta da aplicação Airbnb

Foi também realizada a utilização direta da aplicação Airbnb, para adquirir uma experiência em primeira mão das principais funcionalidades e da interface. Esta análise crítica serviu de inspiração para o desenho do nosso projeto, que visa oferecer um sistema de reserva de alojamentos que partilha alguns princípios e práticas recomendadas, mas que é desenvolvido de raiz, respeitando a originalidade e adaptando as funcionalidades às necessidades identificadas.

3.2 Análise da Informação e Identificação de Requisitos

A partir da análise da informação recolhida durante a fase de levantamento de requisitos, foram identificadas e categorizadas as necessidades do sistema em dois grandes grupos: requisitos funcionais e requisitos não funcionais. Esta classificação visa assegurar uma compreensão clara das funcionalidades esperadas e das características de qualidade a serem garantidas pelo sistema.

3.2.1 Requisitos Funcionais

Os requisitos funcionais referem-se às operações e serviços que o sistema deverá fornecer aos seus utilizadores. Foram identificadas as seguintes funcionalidades essenciais:

- **Gestão de Anúncios:** O sistema deve permitir aos anfitriões criar, editar e eliminar anúncios de alojamento. Cada anúncio deverá conter uma descrição detalhada, localização, imagens e o preço por noite.
- **Pesquisa e Filtragem:** Os hóspedes devem poder procurar alojamentos com base em múltiplos critérios, nomeadamente localização, faixa de preços, datas disponíveis, tipo de alojamento e comodidades.
- **Realização de Reservas:** O sistema deverá possibilitar aos hóspedes a reserva de alojamentos disponíveis para as datas pretendidas.
- **Avaliação e Comentários:** Após a estadia, tanto anfitriões como hóspedes poderão avaliar a experiência e deixar comentários, fomentando a confiança entre os utilizadores da plataforma.
- **Gestão de Reservas:** Os anfitriões deverão poder aceitar, recusar ou cancelar pedidos de reserva. Os hóspedes, por sua vez, deverão ter acesso à gestão das suas reservas ativas ou futuras.
- **Gestão de Contas de Utilizador:** O sistema deverá permitir a criação, autenticação e atualização de contas, bem como a recuperação de credenciais e a gestão de perfis tanto de anfitriões como de hóspedes.

3.2.2 Requisitos Não Funcionais

Os requisitos não funcionais referem-se às propriedades de qualidade que o sistema deve possuir, de forma a garantir a sua eficácia, fiabilidade e acessibilidade:

- **Segurança:** - Utilização do protocolo HTTPS para comunicação segura, implementação de autenticação multifator para acessos críticos, e auditorias periódicas para monitorizar vulnerabilidades.
- **Desempenho:** O sistema deverá apresentar tempos de resposta reduzidos nas operações principais, mesmo em situações de elevada concorrência de utilizadores.
- **Usabilidade:** A interface da aplicação deverá ser intuitiva, acessível e de fácil utilização, proporcionando uma experiência de navegação fluida e agradável, realizando testes de usabilidade a utilizadores reais.
- **Compatibilidade:** Testes de compatibilidade em Chrome, Firefox, Edge e Safari, garantindo suporte para telas de 768px a 1920px, além de otimização para sistemas operacionais Windows, macOS, Android e iOS.
- **Confiabilidade:** Implementação de monitorização contínua com alertas em caso de falha, backups automáticos diários, e um plano de recuperação de desastre que permite retomar o sistema em menos de 30 minutos após um incidente.
- **Escalabilidade:** A arquitetura da aplicação deverá suportar o crescimento da base de utilizadores e a adição de novas funcionalidades, sem comprometer a performance do sistema, para isso utilizamos uma arquitetura de microserviços.

3.3 Personas

Para garantir que o desenvolvimento do sistema estivesse alinhado com as necessidades dos utilizadores finais, foi realizada a definição de **personas**. Uma persona é uma representação fictícia de um tipo de utilizador típico do sistema, baseada em dados reais e suposições fundamentadas sobre os comportamentos, objetivos e necessidades desses utilizadores.

A criação das personas permitiu à equipa compreender melhor o público-alvo e tomar decisões de design centradas no utilizador. Cada persona foi descrita com base em características como idade, profissão, objetivos, frustrações e nível de literacia tecnológica, proporcionando uma visão clara dos diferentes perfis que irão interagir com a aplicação.

3.3.1 Persona 1 – Hóspede

Nome	João Silva
Idade	29 anos
Profissão	Engenheiro Informático
Descrição	Jovem profissional que viaja frequentemente em trabalho e lazer. Valoriza plataformas intuitivas que lhe permitam reservar alojamento de forma rápida e segura. Costuma usar o telemóvel para fazer reservas.
Objetivos	Procura alojamentos cómodos, bem localizados e com processo de reserva simples. Frustra-se com falta de clareza nos preços, informações desatualizadas e interfaces confusas.

3.3.2 Persona 2 – Administrador

Nome	Ana Moreira
Idade	43 anos
Profissão	Gestora de Sistemas
Descrição	Responsável pela gestão da plataforma. Tem perfil técnico e valoriza interfaces administrativas organizadas e eficientes.
Objetivos	Precisa de ferramentas para gerir contas, monitorizar reservas e assegurar o funcionamento da plataforma. Irrita-se com interfaces confusas, falta de controlo sobre conteúdo e relatórios limitados.

3.3.3 Persona 3 – Anfitrião

Nome	Pedro Carvalho
Idade	35 anos
Profissão	Proprietário de Alojamentos Locais
Descrição	Gere vários apartamentos turísticos. Utiliza plataformas digitais para promover e gerir alojamentos. Dá importância à facilidade de comunicação com hóspedes.
Objetivos	Quer publicar e atualizar anúncios facilmente, gerir reservas e comunicar com hóspedes. Frustra-se com burocracias, dificuldades na edição dos anúncios e falta de histórico visível.

3.4 User Stories

Com base nas personas definidas, foram elaboradas diversas user stories, seguindo a estrutura habitual da metodologia ágil: “Como [tipo de utilizador], quero [objetivo] para [benefício].”

- **Gestão de Anúncios:**

Como *anfitrião (Pedro Carvalho)*, quero *criar, editar e eliminar anúncios de alojamento* para *gerir facilmente os meus apartamentos turísticos e manter as informações atualizadas*.

- **Pesquisa e Filtragem:**

Como *hóspede (João Silva)*, quero *procurar alojamentos com base em critérios como localização, preço, datas e comodidades* para *encontrar rapidamente opções que se ajustem às minhas necessidades de viagem*.

- **Realização de Reservas:**

Como *hóspede (João Silva)*, quero *reservar alojamentos disponíveis para as datas que pretendo* para *garantir a minha estadia com antecedência e segurança*.

- **Avaliação e Comentários:**

Como *utilizador da plataforma (anfitrião ou hóspede)*, quero *deixar uma avaliação e comentário após a estadia* para *partilhar a minha experiência e ajudar outros utilizadores a tomarem decisões informadas*.

- **Gestão de Reservas:**

- Como *anfitrião (Pedro Carvalho)*, quero *aceitar, recusar ou cancelar pedidos de reserva* para *ter controlo sobre a disponibilidade dos meus alojamentos*.
- Como *hóspede (João Silva)*, quero *consultar e gerir as minhas reservas ativas ou futuras* para *acompanhar as minhas estadias e fazer alterações quando necessário*.

- **Gestão de Contas de Utilizador:**

Como utilizador da plataforma (Ana Moreira, Pedro Carvalho ou João Silva), quero criar, autenticar e atualizar a minha conta, bem como recuperar credenciais para aceder de forma segura à plataforma e manter o meu perfil atualizado.

Estas user stories funcionaram como pontos de partida para a definição dos requisitos funcionais do sistema e permitiram organizar o desenvolvimento com base nas reais necessidades dos utilizadores.

A adoção de user stories contribuiu para uma abordagem centrada no utilizador, promovendo a entrega de funcionalidades com valor e garantindo que o produto final respondesse às expectativas dos seus utilizadores-alvo.

Capítulo 4

Desenho

4.1 Elaboração do Diagrama de Casos de Uso [Bri25c]

Após a identificação dos requisitos, procedeu-se à elaboração do **diagrama de casos de uso**, o qual ilustra as principais interações entre os utilizadores (anfitriões e hóspedes) e o sistema. Este diagrama foi desenvolvido com recurso ao software Visual Paradigm.

O diagrama de casos de uso está diretamente associado aos requisitos funcionais levantados, representando visualmente as funcionalidades que o sistema deve oferecer. Cada caso de uso corresponde a um ou mais requisitos funcionais, garantindo que todas as ações essenciais esperadas dos utilizadores estejam contempladas e possam ser validadas durante o desenvolvimento.

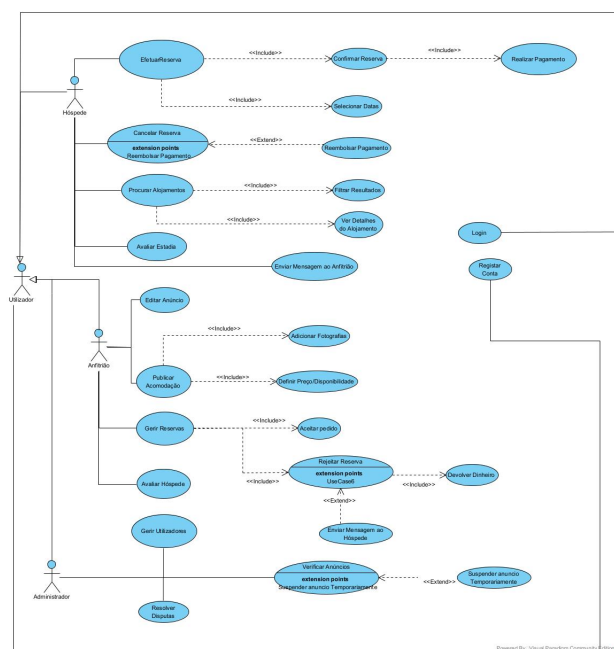
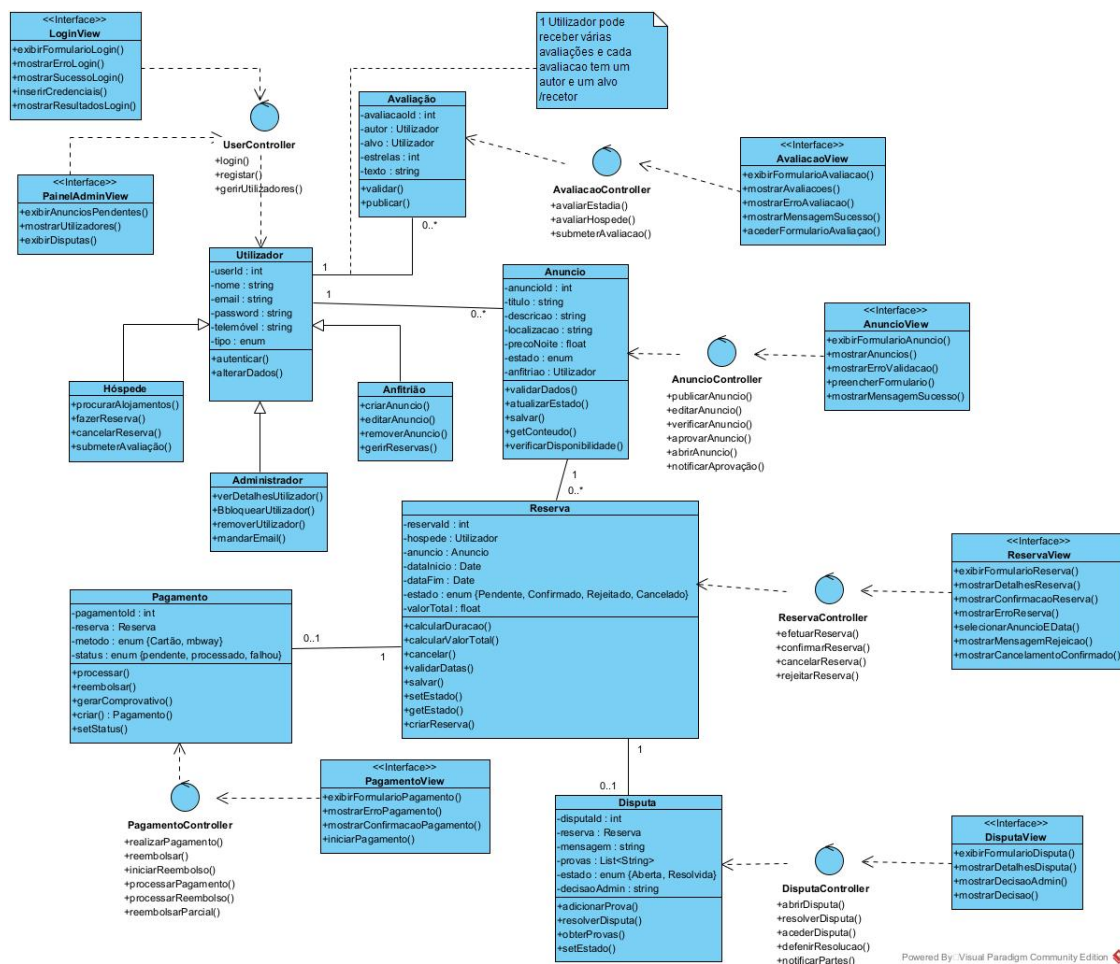


Figura 4.1: Diagrama de casos de uso



4.3 Elaboração dos Diagramas de Sequência UML [Bri25b]

Complementando a modelação estrutural do sistema, foram elaborados um total de **12 diagramas de sequência UML**, com o objetivo de representar o comportamento dinâmico do sistema em diferentes funcionalidades.

Estes diagramas descrevem, de forma temporal, a interação entre os objetos envolvidos em cada cenário, através da troca de mensagens. Cada diagrama está associado a um caso de uso identificado anteriormente, como por exemplo o *registo de utilizador*, a *realização de uma reserva*, ou a *edição de um alojamento por parte do anfitrião*.

Devido à quantidade de diagramas produzidos, considerou-se excessivo incluir todos no relatório. Assim, optou-se por apresentar apenas alguns dos mais representativos, que ilustram com clareza os fluxos de interação e as decisões de design adotadas durante a modelação do sistema.

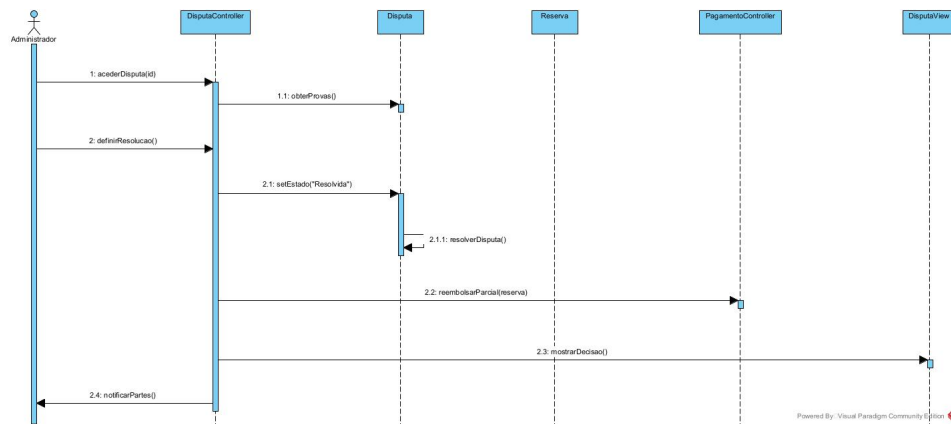


Figura 4.3: Exemplo de diagrama de sequência UML

Capítulo 5

Gestão Prática do Projeto e Organização da Equipa

Neste capítulo, descreve-se a abordagem adotada para a gestão prática do projeto, detalhando os mecanismos e ferramentas utilizados para o controlo de versões, a distribuição e acompanhamento de tarefas, bem como os métodos de comunicação entre os membros da equipa.

5.1 Sistema de Controlo de Versões

O controlo de versões foi uma componente essencial para garantir a rastreabilidade e a integridade do código-fonte ao longo do desenvolvimento. Para esse efeito, foram utilizadas as ferramentas Git e GitHub, que permitiram à equipa trabalhar de forma colaborativa, minimizando conflitos e promovendo a integração contínua.

5.1.1 Sistema de Controlo Distribuído com Git

O *Git* foi utilizado como sistema de controlo de versões distribuído, permitindo que cada membro da equipa mantivesse uma cópia local do repositório. Esta abordagem possibilitou o desenvolvimento assíncrono, com *commits* frequentes que registaram alterações incrementais ao código.

5.1.2 Repositório Remoto e Colaboração via GitHub

O *GitHub* foi utilizado como plataforma de repositório remoto, centralizando todas as contribuições da equipa. Através do GitHub, foi possível gerir *branches*, abrir *pull requests* e efetuar revisões de código, promovendo a colaboração e a qualidade do software.

Link do Repositório

5.2 Gestão e Acompanhamento de Tarefas

A organização das tarefas foi feita através do *GitHub Issue Board*, complementada pela metodologia *Scrum* com sprints e reuniões diárias para acompanhamento do progresso e resolução rápida de impedimentos.

5.2.1 Github Issue Board

O quadro de tarefas foi estruturado segundo a metodologia Scrum, com colunas como “Por Fazer”, “Em Progresso” e “Concluído”, permitindo o acompanhamento do estado de cada tarefa e facilitando a priorização das atividades.

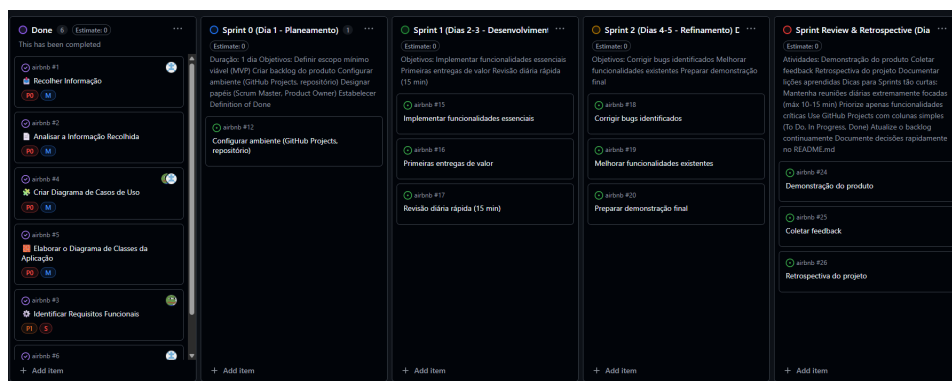


Figura 5.1: Quadro de Tarefas (Issue Board) no GitHub

Cada membro da equipa teve tarefas específicas associadas a cartões no Issue Board do GitHub. Cada cartão incluía descrição, prioridade, estimativa de esforço e datas de entrega, facilitando a organização, o acompanhamento do progresso e a responsabilização individual.

Status	Title	Priority	Size	Start date	End date
Todo	1. Recolher Informação #1	P0	M	Jun 1, 2025	Jun 2, 2025
In Progress	2. Analisar a Informação Recolhida #2	P0	M	Jun 2, 2025	Jun 3, 2025
In Progress	3. Identificar Requisitos Funcionais #3	P1	S	Jun 2, 2025	Jun 3, 2025
Done	4. Criar Diagrama de Casos de Uso #4	P0	M	Jun 3, 2025	Jun 4, 2025
Done	5. Elaborar o Diagrama de Classes da Aplicação #5	P0	M	Jun 3, 2025	Jun 4, 2025
Done	6. Elaborar Diagrama(s) de Sequência #6	P1	M	Jun 3, 2025	Jun 4, 2025

Figura 5.2: Itens do Quadro de Tarefas

Cada cartão representa uma tarefa ou funcionalidade, incluindo descrições, *labels*, listas de verificação, datas de entrega e prioridades. Esta abordagem contribuiu para uma visão global e clara do progresso do projeto.



Figura 5.3: Exemplo de Cartão de Tarefa

5.3 Comunicação e Coordenação da Equipa

A comunicação eficaz entre os membros da equipa revelou-se essencial para a coordenação das atividades e para a rápida resolução de impedimentos. Para esse fim, foram combinadas práticas síncronas e assíncronas, nomeadamente reuniões diárias e comunicação contínua via plataformas digitais.

5.3.1 Coordenação através de Reuniões de Sincronização

Realizaram-se reuniões diárias curtas, com o objetivo de sincronizar o trabalho da equipa. Cada membro partilhava as tarefas realizadas, os objetivos do dia e eventuais bloqueios. Esta prática, inspirada no modelo de *daily stand-up* do Scrum, permitiu manter a equipa alinhada, promover a colaboração e antecipar problemas de integração.

5.3.2 Comunicação Assíncrona e Síncrona com Discord e Microsoft Teams

A comunicação contínua foi assegurada através das plataformas *Discord* e *Microsoft Teams*. Estas ferramentas permitiram a troca rápida de mensagens, partilha de ficheiros e realização de chamadas de voz sempre que necessário. O Discord foi mais utilizado para interações informais e técnicas durante o desenvolvimento, enquanto o Teams foi reservado para reuniões mais formais e contacto com docentes, quando aplicável.

Capítulo 6

Conclusão

O desenvolvimento deste projeto permitiu aplicar de forma prática os princípios da metodologia ágil, em particular o framework *Scrum*, proporcionando uma abordagem iterativa e incremental ao longo das diferentes fases do trabalho. A definição de *personas*, a identificação de requisitos, a elaboração de *user stories* e a modelação através de diagramas UML contribuíram para uma compreensão clara das necessidades dos utilizadores e das funcionalidades a implementar.

A utilização de *sprints* curtos, com planeamento e revisão diária, promoveu uma melhor organização da equipa, facilitou a deteção precoce de problemas e garantiu um progresso contínuo e alinhado com os objetivos definidos. Paralelamente, o uso de ferramentas colaborativas como o *GitHub Issues Board* ajudou a gerir eficazmente as tarefas, mantendo uma visão global e atualizada do estado do projeto.

Conclui-se, assim, que a combinação entre técnicas de engenharia de software e práticas ágeis resultou num processo de desenvolvimento mais estruturado, centrado no utilizador e adaptável a mudanças. Este projeto não só reforçou conhecimentos técnicos, como também destacou a importância da comunicação, da organização e da colaboração em ambientes de desenvolvimento de software.

Bibliografia

- [Bri25a] Isabel Brito. *Diagrama De Classes - Arquitetura MVC Ficheiro*. Ipbeja.pt, 2025. URL: https://cms.ipbeja.pt/pluginfile.php/365448/mod_resource/content/1/aula5classesMVC.pdf (acedido em 06/06/2025) (ver p. 12).
- [Bri25b] Isabel Brito. *Diagramas De Sequência*. Ipbeja.pt, 2025. URL: https://cms.ipbeja.pt/pluginfile.php/184424/mod_resource/content/4/aula6SequenciaUML.pdf (acedido em 06/06/2025) (ver p. 13).
- [Bri25c] Isabel Brito. *Regras Para Os Casos De Uso Ficheiro*. Ipbeja.pt, 2025. URL: https://cms.ipbeja.pt/pluginfile.php/139224/mod_resource/content/9/RegrasCenariosCasosUso.pdf (acedido em 06/06/2025) (ver p. 11).