UNIVERSIDADE
LUSÓFONA

BudgetBreak

Trabalho Final de curso

Relatório Intercalar 2º Semestre

Bernardo António Agostinho Jerónimo, a22402766, LEI

Henrique Valente Marçalo, a22302483, LEI

Orientador: Pedro De Almeida Perdigão

Entidade Externa: CGI

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

12/04/2026

Direitos de cópia

BudgetBreak, Copyright de *Bernardo Jerónimo* e *Henrique Marçalo*, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Este trabalho está abrangido por Acordo de Não Divulgação (NDA); qualquer disponibilização pública fica condicionada à eliminação/anonimização de informação confidencial e/ou à autorização escrita prévia da CGI; a versão pública depositada será necessariamente expurgada.

Agradecimentos

Resumo

O presente Trabalho Final de Curso descreve uma plataforma de planeamento de viagens, *BudgetBreak*, que permite aos utilizadores criar automaticamente pacotes de voo e alojamento ajustados ao seu orçamento pré-definido. No contexto atual do turismo digital, marcado por um elevado volume de informação, diversidade de opções e crescente exigência de personalização, os utilizadores enfrentam desafios significativos na organização eficiente das suas viagens, nomeadamente em manter-se dentro do orçamento e potencializar o equilíbrio entre custo e qualidade.

A *BudgetBreak* propõe uma abordagem diferenciadora, invertendo o paradigma tradicional de decisão, permitindo que o orçamento seja o ponto de partida para a seleção de destinos e serviços. A plataforma oferece recomendações personalizadas, recorrendo a um motor de Inteligência Artificial (*LLM*s) capaz de interpretar as preferências semânticas do utilizador num contexto controlado, gerando sugestões que respeitam rigorosamente os critérios financeiros. Adicionalmente, inclui funcionalidades inovadoras, como o modo "*Randomizer*", que sugere destinos inesperados e promove experiências de descoberta enriquecedoras.

O projeto evidencia relevância prática e académica, ao preencher uma lacuna existente no mercado de planeamento turístico, combinando eficiência operacional, acessibilidade e experiência de utilização intuitiva. Esta, por um lado, contribui para a democratização do acesso a viagens personalizadas, permitindo escolhas mais conscientes, otimizadas e alinhadas com os interesses do utilizador, com potencial impacto positivo no setor do turismo digital.

Em termos de conceção e implementação, a plataforma foi projetada para validar um *MVP* funcional, centrando-se na integração de dados externos, na lógica de recomendação e na experiência do utilizador, garantindo aplicabilidade em contextos reais. A solução distingue-se das existentes pelo foco no orçamento como ponto de partida, personalização, descoberta e simplicidade de utilização, oferecendo uma proposta inovadora e única no mercado.

Palavras-chave: Planeamento de viagens, otimização de orçamento, personalização, experiência do utilizador, inovação digital, turismo, recomendações automatizadas, pacotes de viagem inteligentes, tecnologia de viagens, descoberta de destinos, eficiência, acessibilidade, automatização, transformação digital.

Abstract

This Final Year Project describes a travel planning platform, *BudgetBreak*, which allows users to automatically create flight and accommodation packages tailored to their predefined budget. In the current context of digital tourism, characterized by a high volume of information, diversity of options, and increasing demand for personalization, users face significant challenges in efficiently organizing their trips, particularly in staying within budget and optimizing the balance between cost and quality.

BudgetBreak proposes a differentiating approach, inverting the traditional decision-making paradigm by allowing the budget to be the starting point for selecting destinations and services. The platform offers personalized recommendations, utilizing an Artificial Intelligence engine (LLMs) capable of interpreting the user's semantic preferences within a controlled context, generating suggestions that strictly adhere to financial criteria. Additionally, it includes innovative features, such as the "Randomizer" mode, which suggests unexpected destinations and promotes enriching discovery experiences.

The project demonstrates practical and academic relevance by filling an existing gap in the travel planning market, combining operational efficiency, accessibility, and an intuitive user experience. This contributes to the democratization of access to personalized travel, enabling more conscious, optimized choices aligned with user interests, with a potential positive impact on the digital tourism sector.

In terms of design and implementation, the platform was designed to validate a functional *MVP* focusing on external data integration, recommendation logic, and user experience, ensuring applicability in real-world contexts. The solution distinguishes itself from existing ones by focusing on the budget as the starting point, along with personalization, discovery, and ease of use, offering an innovative and unique proposal in the market.

Key words: Travel planning, budget optimization, personalization, user experience, digital innovation, tourism, automated, recommendations, intelligent travel packages, travel technology, destination discovery, efficiency, accessibility, automation, digital transformation

Este relatório diz respeito a um projeto sujeito a restrições de confidencialidade (NDA). Os detalhes sensíveis foram abstraídos

Índice

1 Introdução 1
1.1 Enquadramento 1
1.2 Motivação e Identificação do Problema 2
1.3 Objetivos 2
1.3.1 Objetivo geral 2
1.3.2 Objetivos específicos 2
1.4 Objetivos de negócio e tecnológicos 3
1.5 Estrutura do Documento 3
2 Pertinência e Viabilidade 5
2.1 Pertinência 5
2.1.1 Motivações dos Utilizadores 5
2.1.2 Fatores de Valorização (Custo e Qualidade) 5
2.1.3 Problemas Identificados 5
2.1.4 Alinhamento das Funcionalidades com as Necessidades 6
2.1.5 Existência de Mercado Ativo 7
2.1.6 Adoção Tecnológica 7
2.1.7 Validação da Proposta de Valor 7
2.1.8 Oportunidade de Mercado (Insatisfação Atual) 7
2.2 Viabilidade 8
2.2.1 Objetivos de Desenvolvimento Sustentável (ODS) 8
2.2.2 Viabilidade económica 8
2.3 Análise Comparativa com Soluções Existentes 9
2.3.1 Soluções existentes 9
2.3.2 Análise de benchmarking 11
2.3.3 Inovação da solução 13
2.3.4 Mais-valias e benefícios 13

2.4 Identificação de oportunidade de negócio 13
3 Especificação e Modelação 15
3.1 Análise de Requisitos 15
3.1.1 Enumeração de Requisitos 15
3.1.2 Descrição detalhada dos requisitos principais 23
3.1.3 Casos de Uso/*User Stories* 28
3.2 Modelação 28
3.2.1 Modelo de Dados (Diagrama de Entidade-Relação) 28
3.2.2 Modelo de Classes (Diagrama de Classes) 29
3.2.3 Modelo Comportamental (Diagramas de Sequência) 30
3.3 Protótipos de Interface 31
3.3.1 *Landing Page* (*Search Engine* & *Definition*) 31
3.3.2 *Authentication Screen* (Ecrã de Autenticação) 31
3.3.3 *User Dashboard* (Painel do Utilizador) 31
3.3.4 *Results List* (Lista de Resultados) 32
3.3.5 Package Detail (Detalhe do Pacote) 32
3.3.6 Checkout 32
3.4 Mockups e Experiência Visual 32
4 Solução Proposta 33
4.1 Apresentação 33
4.2 Arquitetura 33
4.3 Tecnologias e Ferramentas Utilizadas 33
4.4 Ambientes de Teste e de Produção 34
4.4.1 Infraestrutura Computacional (Hardware) 34
4.4.2 Capacidade de Processamento e Software 34
4.4.3 Armazenamento e Rede 34
4.5 Abrangência 34
4.6 Componentes 35
4.6.1 Componente 1 – *Backend Core* (*Spring Boot* & *APIs* externas): 35

4.6.2 Componente 2 - Base de dados (*PostgreSQL*): ... 35

4.6.3 Componente 3 - *Frontend* (*Next.js + Tailwind CSS*): ... 35

4.6.4 Componente 4 - *Kafka*: ... 35

4.6.5 Componente 5 - *gRPC*: ... 35

4.6.6 Componente 6 - Motor de recomendações (*LLM*s + *MCP*): ... 35

4.6.7 Componente 7 – Docker: ... 35

4.7 Interfaces ... 35

4.7.1 Landing Page & Search ... 36

4.7.2 Package List ... 36

4.7.3 Package detail ... 36

4.7.4 Checkout & Payment ... 36

5 Testes e Validação ... 37

5.1 Abordagem Metodológica ... 37

5.2 Testes de Qualidade e Código (Automatizado) ... 37

5.3 Validação Operacional e Integração (Manual) ... 37

6 Método e Planeamento ... 38

6.1 Planeamento inicial ... 38

6.2 Análise Crítica ao Planeamento ... 38

7 Resultados ... 39

7.1 Resultados dos Testes ... 39

7.2 Cumprimento de requisitos ... 39

8 Conclusão ... 40

8.1 Conclusão ... 40

8.2 Trabalhos Futuros ... 40

Bibliografia ... 41

Anexo 1 – Resultados do Inquérito de Pertinência ... 43

Anexo 2 – Artefactos de Modelação ... 48

Anexo 3 – Arquitetura de Sistema e Lógica ... 52

Anexo 4 – Protótipos de Interface (*Mockups*) ... 54

Anexo 5 – Testes ... 60

Anexo 6 – Formulário de declaração de uso de ferramentas de Inteligência Artificial 62

1. Utilização de IA 62

2. Ferramentas utilizadas 62

3. Fases do trabalho em que foi utilizada IA 63

4. Tipo de utilização 63

5. Partes do trabalho afetadas 63

6. Exemplos de *prompt* 64

9. Trabalhos em parceria 65

10. Declaração de responsabilidade 65

11. Identificação dos autores 65

Glossário 66

Lista de Figuras

Figura 1 - Gantt Chart 38
Figura 2 - Frequência com que os inquiridos realizam viagens 43
Figura 3 - Principais motivos que levam os utilizadores a viajar 43
Figura 4 - Ferramentas utilizadas atualmente para organizar viagens 44
Figura 5 - Quantidade de tempo investido no processo de planeamento 44
Figura 6 - Fatores mais valorizados na escolha de uma viagem 45
Figura 7 - Frequência com que os utilizadores têm dificuldade em manter-se dentro do orçamento 45
Figura 8 - Nível de satisfação com as plataformas atuais (Escala 1-5) 46
Figura 9 - Funcionalidades consideradas mais úteis pelos utilizadores 46
Figura 10 - Interesse na utilização de uma app que cria pacotes por orçamento 47
Figura 11 - Diagrama de Entidade-Relação (ERD) 48
Figura 12 - Diagrama de Classes (UML) 49
Figura 13 - Diagrama de Sequência: Stateless Authorization 49
Figura 14 - Diagrama de Sequência: Login 50
Figura 15 - Diagrama de Sequência: Logout 50
Figura 16 - Diagrama de Sequência: Refresh Token 51
Figura 17 - Diagrama da Arquitetura Global do Sistema 52
Figura 18 - Bounded context 52
Figura 19 - DAG de agregação de pacotes 53
Figura 20 - Mapa Aplicacional 54
Figura 21 - Mockup: Landing Page 54
Figura 22 - Mockup: Login 55
Figura 23 - Mockup: Register 55
Figura 24 - Mockup: Reset Password 56
Figura 25 - Mockup: User Dashboard 56
Figura 26 - Mockup: User Trips 57
Figura 27 - Mockup: Favorite Accommodation 57
Figura 28 - Mockup: Profile settings & Preferences 58
Figura 29 - Mockup: Package list 58
Figura 30 - Mockup: Package details 59
Figura 31 - Mockup: Checkout 59
Figura 32 - Coverage: Application 60
Figura 33 - Coverage: User Controller 60
Figura 34 - Coverage: User Service 60
Figura 35 - Coverage: Preference Controller 60
Figura 36 - Coverage: Authentication Controller 60
Figura 37 - Coverage: Authentication Service 60
Figura 38 – Coverage: Booking Controller 61
Figura 39 – Coverage: Booking Service 61
Figura 40 – Coverage: Package Controller 61
Figura 41 – Coverage: Package Service 61

Lista de Tabelas

Tabela 1 - Benchmarking 12
Tabela 2 - Requisitos Funcionais 15
Tabela 3 - Requisitos Não Funcionais 21

Lista de Siglas

AI
Inteligência Artificial, 11
API
Interface de Programação de Aplicações, iii, iv, 5, 1, 2, 8, 2, 3, 5, 10, 11, 12
CSS
Cascading Style Sheets, 11
HTML
Hypertext Markup Language, 11
HTTP
Hypertext Transfer Protocol, 5
IA
Inteligência Artificial, ii, iv, 1, 11, 14
JPA
Java Persistence API, 7
JSON
JavaScript Object Notation, 5
LLM
Large Language Model, iv, 10, 12
MCP
Model Context Protocol, iv, 10, 12
MSSQL
Microsoft SQL Server, 11
MVP
Minimum Viable Product, iv, v, 10
MySQL
My Structured Query Language, 11
NPU
Neural Processing Unit, 11
NVMe
Non-Volatile Memory Express, 11
PIB
Produto Interno Bruto, ii
RAM
Random Access Memory, 11
SSD
Solid State Drive, 11
TLS
Transport Layer Security, 10
XML
Extensible Markup Language, 5

1 Introdução

Este capítulo contextualiza o projeto BudgetBreak, evidenciando o problema real do planeamento de viagens eficiente e ajustado ao orçamento. Identificam-se as necessidade e lacunas do mercado, destacando as limitações das soluções existentes.

Apresentam-se, de forma resumida, os objetivos gerais e específicos, bem como os objetivos de negócio e tecnológicos, estabelecendo a base para a análise de pertinência, viabilidade e especificação da solução proposta nos capítulos subsequentes.

1.1 Enquadramento

O setor do turismo está atualmente assente como um pilar de extrema relevância na economia global, e representa cerca de 10% do PIB mundial. Este valor reflete não apenas a sua dimensão económica, mas também o seu impacto social, tecnológico e cultura[1].

Nos últimos anos, tem-se verificado um crescimento contínuo e sustentado no número de viajantes internacionais. De acordo com os dados mais recentes[2], foram registados mais de 300 milhões de turistas no primeiro trimestre de 2025, o que representa um aumento de aproximadamente 5% face ao mesmo período de 2024, o que representa um crescimento de 14 milhões de pessoas. Este aumento reforça a tendência de recuperação total do setor pós-pandemia, e demonstra a vitalidade e o dinamismo que o turismo continua a evidenciar à escala global.

Paralelamente, o avanço tecnológico tem sido determinante na transformação dos comportamentos de consumo turístico. Em 2024, mais de 70% da receita do mercado de viagens e turismo foi gerada através de vendas online, sendo que as aplicações móveis representam mais de metade deste volume[3].

No entanto, este crescimento acentuado também introduziu novos desafios relacionados com a sobrecarga de informação, a padronização das experiências e as dificuldades em personalizar escolhas de acordo com as preferências individuais dos viajantes. Assim, o modelo contemporâneo de consumo turístico tende a privilegiar experiências personalizadas, sustentáveis e economicamente otimizadas.

Dados recentes indicam que 22% dos turistas abandonaram processos de reserva online devido à desconfiança na segurança das páginas, e 19% devido a falhas no pagamento. Estas estatísticas sublinham a urgência de implementar soluções digitais mais transparentes e fiáveis. O cenário é particularmente crítico entre a Geração Z (18-27 anos), onde 70% dos viajantes já desistiram de concluir uma reserva, evidenciando que os nativos digitais possuem expectativas extremamente elevados quanto à usabilidade e à confiança nas plataformas[4].

Para reforçar estes dados, o relatório anterior revela ainda que há uma média global de 78% de viajantes que, tanto durante o planeamento, como na sua estadia, estão dispostos a usar IA em 2025. Isto revela a necessidade de desenvolvimento de plataformas digitais inteligentes, capazes de adaptar-se dinamicamente ao perfil e às preferências de cada utilizador, promovendo uma experiência de viagem mais eficiente e orientada ao valor.

Neste contexto, a criação de soluções inteligentes e centradas no utilizador, como a *BudgetBreak*, torna-se particularmente relevante. A aplicação propõe-se a combinar simplicidade, personalização e otimização orçamental, respondendo aos principais desafios identificados no mercado no mercado e às novas exigências do turista digital contemporâneo.

1.2 Motivação e Identificação do Problema

O planeamento de viagens, apesar de entusiasmante, permanece um processo exigente devido à elevada quantidade de informação disponível, à necessidade de comparar diferentes opções e ao impacto significativo que o orçamento tem nas decisões de consumo turístico. O viajante contemporâneo procura soluções rápidas, fiáveis e personalizadas, mas depara-se frequentemente com plataformas que, embora completas, não conseguem simplificar verdadeiramente o processo de decisão.

A motivação deste trabalho assenta na necessidade de simplificar e tornar o planeamento de viagens mais acessível, ajustando-o ao perfil do utilizador atual, que valoriza autonomia, rapidez e personalização. O aumento do interesse por ferramentas inteligentes e orientadas por dados reforça a importância de criar soluções que tornem o processo de decisão mais objetivo e eficiente.

O problema identificado é a falta de soluções que foquem o orçamento como ponto central do planeamento, apresentando resultados otimizados que combinem voo e alojamento. A *BudgetBreak* propõe-se resolver esta lacuna ao inverter o modelo tradicional: em vez de o utilizador adaptar o seu orçamento às opções existentes, é a aplicação que gera automaticamente pacotes de viagem ajustados ao valor definido.

1.3 Objetivos

1.3.1 Objetivo geral

O desenvolvimento da plataforma *BudgetBreak*, que tem como objetivo ajudar os utilizadores a planear pacotes de viagem (voo + alojamento), de forma simples e automática, ajustando os mesmos ao orçamento pré-definido, de forma a garantir uma experiência prática e personalizada.

1.3.2 Objetivos específicos

- Desenvolver um sistema de integração com *API*s externas (*Skyscanner*, *Booking*, *Airbnb*, …) para recolha de dados atualizados sobre voos e alojamentos.
- Desenvolver um motor de recomendação inteligente, capaz de sugerir destinos e pacotes de viagem ajustados ao orçamento, preferências e localização do utilizador.
- Criar uma *interface web* responsiva e clara (*frontend*), que simplifique o processo de pesquisa e reserva.
- Desenvolver uma arquitetura *backend* modular, segura e escalável, capaz de centralizar a lógica de negócio, orquestrar fluxos de dados por via de múltiplas *API*s externas e garantir a consistência das operações. A arquitetura deve suportar separação clara de responsabilidades (integrações, recomendação,

autenticação, gestão de dados e controlo de acessos), assegurar comunicação eficiente entre serviços.

- Analisar a viabilidade técnica e económica do projeto, avaliando o potencial de mercado e as oportunidades de integração futura com parceiros comerciais do setor turístico, bem como em eventos como a Web Summit.
- Explorar o modo "*Randomizer*", que pretende oferecer sugestões de destinos aleatórios, promovendo o elemento surpresa e estimulando a descoberta.

1.4 Objetivos de negócio e tecnológicos

- **Negócio:**
  - Posicionar a *BudgetBreak* como uma solução inovadora e diferenciadora no mercado de turismo digital;
  - Reduzir o tempo de planeamento de viagens e aumentar a taxa de conversão entre pesquisa e reserva;
  - Estimular parcerias comerciais com companhias aéreas, plataformas de alojamento e agências de viagens.
- **Tecnológicos:**
  - Assegurar integração eficaz e segura com múltiplas *API*s externas;
  - Desenvolver um sistema de recomendação inteligente suportado por *IA*, apto a interpretar preferências, restrições e padrões de comportamento do utilizador através de um contexto controlado via *MCP*, garantindo respostas fiáveis, rastreáveis e alinhadas com as fontes autorizadas.
  - Desenvolver uma infraestrutura escalável, robusta e compatível com diferentes dispositivos.

1.5 Estrutura do Documento

O presente relatório encontra-se organizado em oito capítulos principais, que refletem as diversas fases de conceção, análise e desenvolvimento do projeto BudgetBreak. A estrutura detalhada é a seguinte:

- **Na Secção 1**, é feita a introdução ao projeto, apresentando o enquadramento no setor do turismo, a motivação, a identificação do problema e os objetivos definidos.
- **Na Secção 2**, analisa-se a pertinência e viabilidade da solução, sustentada pelos resultados de um inquérito aos utilizadores, uma análise comparativa com a concorrência (*benchmarking*) e a identificação da oportunidade de negócio.
- **Na Secção 3**, detalha-se a especificação e modelação do sistema, incluindo o levantamento de requisitos, casos de uso, modelação de dados e apresentação dos protótipos de interface.
- **Na Secção 4**, descreve-se a solução proposta, focando na arquitetura do sistema, nas tecnologias e ferramentas utilizadas (como *Spring Boot* e *Next.js*), bem como na descrição dos componentes e interfaces.
- **Na Secção 5**, serão abordados os testes e a validação da solução, garantindo a qualidade e o cumprimento dos requisitos estabelecidos.

- **Na Secção 6**, apresenta-se o método de trabalho e o planeamento do projeto, incluindo cronogramas e a análise crítica ao cumprimento dos prazos.
- **Na Secção 7**, expõem-se os resultados alcançados, tanto ao nível dos testes como do cumprimento global dos requisitos.
- **Na Secção 8**, apresentam-se as conclusões finais, a síntese do trabalho desenvolvido e as perspetivas para trabalhos futuros.
- **Por fim**, encontram-se a Bibliografia, os Anexos e o Glossário, que complementam a informação apresentada ao longo do documento.

2 Pertinência e Viabilidade

2.1 Pertinência

A validação da pertinência deste projeto constitui um passo fundamental para assegurar que este responde efetivamente às necessidades reais de potenciais utilizadores. Neste sentido, foi desenvolvido e aplicado um inquérito direcionado a indivíduos com idade igual ou superior a 18 anos, tendo sido recolhido 115 respostas. Este inquérito procurou compreender os hábitos de viagem do público-alvo, as suas motivações, os métodos de planeamento atualmente utilizados, as principais dificuldades enfrentadas, e o interesse numa solução que integre funcionalidades de automatização e controlo orçamental. Os resultados obtidos fornecem dados quantitativos que sustentam tanto a necessidade quanto a viabilidade da solução proposta.

Os gráficos detalhados com os resultados brutos de todas as questões do inquérito encontram-se disponíveis para consulta no *Anexo 1 – Resultados do Inquérito de Pertinência*

2.1.1 Motivações dos Utilizadores

Relativamente às motivações, o lazer destaca-se claramente como o principal motivo, representando 85,2% das respostas [Figura 3]. Esta predominância do turismo de lazer indica que os utilizadores procuram experiências enriquecedoras e personalizadas, reforçando a pertinência de uma aplicação que facilite o planeamento dessas viagens de forma eficiente e adaptada às suas preferências individuais. Os restantes motivos, incluindo trabalho, visitas a familiares e amigos, e a procura por experiências culturais, embora minoritários, demonstram a diversidade de contextos em que uma ferramenta de planeamento pode ser útil.

2.1.2 Fatores de Valorização (Custo e Qualidade)

No que toca à valorização de fatores no planeamento de uma viagem, o custo emerge como o elemento-chave, com 45,2% das preferências, seguido pela qualidade com 37,4% e pelas avaliações e ratings com 16,5% [Figura 6]. Esta hierarquização demonstra que os utilizadores procuram otimizar o equilíbrio entre preço e qualidade, valorizando particularmente a confiança transmitida por avaliações de outros viajantes. Uma aplicação que integre comparação de preços, informações de qualidade e sistemas de avaliação transparentes tem, portanto, elevado potencial para responder às necessidades reais identificadas no mercado.

2.1.3 Problemas Identificados

2.1.3.1 Tempo de Planeamento

O tempo investido no planeamento apresenta uma distribuição considerável que revela ineficiências no processo atual: 35,7% dos utilizadores gastam entre 3 e 6 horas, 33,9% investem entre 1 e 3 horas, e 27% gastam mais de 6 horas neste processo [Figura 5]. Estes dados sugerem que o planeamento é moroso e fragmentado, exigindo que os

utilizadores consultem múltiplas plataformas e recursos dispersos. Mesmo os utilizadores que gastam apenas 1 a 3 horas representam um investimento significativo de tempo que poderia ser otimizado. Esta fragmentação e o tempo considerável despendido reforçam a necessidade de uma solução integrada que simplifique e centralize o processo, reduzindo o tempo e esforço necessários.

2.1.3.2 Controlo Orçamental

Um dos resultados mais reveladores do inquérito relaciona-se precisamente com a dificuldade em manter-se dentro do orçamento durante o planeamento de viagens. Os dados demonstram que 87,8% dos inquiridos enfrentam esta dificuldade com alguma frequência [Figura 7]: 59,1% indicam que "às vezes" têm este problema, enquanto 28,7% afirmam que isso acontece "frequentemente". Apenas 11,3% raramente enfrentam esta dificuldade, e uma percentagem residual nunca a experiência. Estes números evidenciam uma lacuna significativa nas ferramentas atualmente disponíveis, que não conseguem auxiliar eficazmente os utilizadores a gerir e controlar os seus gastos durante o processo de planeamento. A dificuldade em controlar o orçamento, combinada com o facto de o custo ser o fator-chave para 45,2% dos inquiridos, cria um desalinhamento crítico entre as necessidades dos utilizadores e as capacidades das ferramentas existentes. Esta dificuldade generalizada reforça fortemente a pertinência de uma solução que integre o controlo orçamental como funcionalidade central, permitindo aos utilizadores visualizar e comparar opções dentro dos seus limites financeiros predefinidos.

2.1.4 Alinhamento das Funcionalidades com as Necessidades

A análise das funcionalidades consideradas mais úteis pelos potenciais utilizadores revela preferências claras que devem orientar as prioridades de desenvolvimento. "Sugestões automáticas de viagens baseadas no orçamento" foi a funcionalidade mais valorizada, com 76,5% das preferências [Figura 9], reforçando diretamente a necessidade identificada de auxiliar os utilizadores no controlo orçamental e na descoberta de opções viáveis dentro das suas possibilidades financeiras. Esta funcionalidade responde diretamente ao problema mais crítico identificado: a dificuldade em manter-se dentro do orçamento. Os alertas de preços e ofertas de viagem foram valorizados por 53% dos inquiridos, sugerindo que os utilizadores procuram otimizar ativamente os seus gastos e aproveitar oportunidades, o que justifica a implementação de sistemas de monitorização e notificação proativa. A "integração com plataformas já estabelecidas, como *Booking* e *Airbnb*", foi considerada útil por 45,2% dos respondentes, indicando que os utilizadores não procuram necessariamente substituir completamente as ferramentas que já conhecem e confiam, mas sim uma solução que as centralize e complemente de forma eficiente, eliminando a necessidade de alternar constantemente entre múltiplas aplicações e *websites*. O "histórico de viagens e favoritos" foi valorizado por 28,7% dos utilizadores, demonstrando interesse em personalização e em manter um registo organizado das suas experiências que possa informar futuras decisões de viagem. Por fim, o "*Randomizer*", embora menos prioritário com 19,1%, ainda revela algum interesse em elementos de descoberta e espontaneidade que podem enriquecer a experiência do utilizador e diferenciar a aplicação.

2.1.5 Existência de Mercado Ativo

Os dados recolhidos demonstram que a grande maioria dos inquiridos viaja com regularidade: 46,1% afirma viajar várias vezes por ano e 36,5% viaja pelo menos uma vez por ano [Figura 2]. Apenas uma minoria reduzida (13,9%) viaja raramente, e, há um conjunto ainda menor que viaja frequentemente ao ponto de o fazer mensalmente. Estes resultados comprovam que viajar é uma atividade regular e significativa para o público-alvo, o que sustenta a existência de um mercado consistente e ativo para ferramentas dedicadas ao planeamento de viagens.

2.1.6 Adoção Tecnológica

A análise dos métodos atualmente utilizados para planear viagens revela que 82,6% dos inquiridos recorrem a aplicações e websites como *Booking*, *SkyScanner* e *Airbnb*, enquanto apenas 13,9% utilizam agências de viagens tradicionais [Figura 4]. Esta tendência digital evidencia que o público está familiarizado e confortável com soluções tecnológicas para organizar as suas deslocações, validando a escolha de desenvolver uma aplicação como resposta ao problema identificado. A adoção massiva de ferramentas digitais demonstra ainda que não existe resistência tecnológica no público-alvo, mas sim uma procura ativa por soluções que melhorem o processo de planeamento já digitalmente estabelecido.

2.1.7 Validação da Proposta de Valor

A validação direta da solução proposta foi obtida mediante de uma questão específica sobre o interesse numa aplicação capaz de criar pacotes de viagem, integrando voo e alojamento, baseados num orçamento máximo definido pelo utilizador. A resposta foi extremamente positiva: 60% dos inquiridos afirmaram que "definitivamente" gostariam de ter acesso a esta funcionalidade, enquanto 37,4% demonstraram interesse condicional, referindo que talvez a utilizassem se fosse acessível ou fácil de usar [Figura 10]. Apenas uma percentagem residual, aproximadamente 2,6%, manifestou desinteresse. Estes resultados validam claramente a proposta de valor da aplicação em desenvolvimento, confirmando que existe uma procura real e substancial por uma solução que simplifique o planeamento através da automatização e personalização baseada no orçamento disponível. A condição de facilidade de uso, mencionada por mais de um terço dos inquiridos, sublinha ainda a importância crítica de desenvolver uma interface intuitiva e um processo de planeamento verdadeiramente simplificado, onde a tecnologia trabalha em prol do utilizador sem adicionar complexidade.

2.1.8 Oportunidade de Mercado (Insatisfação Atual)

A avaliação da satisfação com as aplicações de planeamento de viagens atualmente disponíveis no mercado revela um nível moderado-baixo de contentamento, com uma classificação média de 3,32 numa escala de 1 a 5 [Figura 8]. A distribuição das respostas é particularmente reveladora: a maioria dos inquiridos concentra-se nos níveis 3, com 39,1%, e 4, com 38,3%, indicando uma satisfação mediana que sugere que as soluções existentes cumprem apenas parcialmente as expectativas dos utilizadores. Adicionalmente, 11,3% dos utilizadores atribuíram classificação 2 e 4,3% classificação 1, demonstrando insatisfação clara com as ferramentas disponíveis. Apenas 7%

classificaram a sua satisfação com o nível máximo de 5, um valor notavelmente baixo que indica que muito poucas pessoas consideram as soluções atuais verdadeiramente excelentes. Esta avaliação mediana, combinada com a elevada percentagem de utilizadores que demonstraram interesse na solução proposta, sugere que, embora existam ferramentas disponíveis e amplamente utilizadas no mercado, estas não respondem plenamente às necessidades e expectativas dos utilizadores. Existe, portanto, espaço considerável para inovação e melhoria, especialmente nas áreas identificadas ao longo do inquérito: controlo orçamental efetivo, automatização inteligente de sugestões personalizadas, redução do tempo de planeamento, e integração de múltiplas plataformas numa experiência verdadeiramente unificada e simplificada.

2.2 Viabilidade

2.2.1 Objetivos de Desenvolvimento Sustentável (ODS)

Embora a *BudgetBreak* seja uma aplicação centrada na área do turismo e planeamento de viagens, o projeto apresenta contributos relevantes para diversos Objetivos de Desenvolvimento Sustentável (ODS) definidos pelas Nações Unidas, reforçando a sua viabilidade e impacto social.

2.2.1.1 Trabalho Digno e Crescimento Económico (ODS 8)

O setor do turismo é um dos principais motores económicos a nível mundial. Ao facilitar o acesso dos utilizadores a opções de viagem mais económicas e ajustadas ao seu orçamento, a nossa aplicação contribui para o aumento da atividade turística e, indiretamente, para o crescimento económico sustentável. Uma maior acessibilidade a viagens promove também um fluxo turístico mais distribuído, incentivando comércio local e pequenas economias regionais.

2.2.1.2 Produção e Consumo Sustentáveis (ODS 12)

A aplicação incentiva decisões de viagem mais conscientes, ao permitir aos utilizadores definir um orçamento máximo e visualizar escolhas transparentes e ajustadas à sua capacidade financeira. Isto reduz comportamentos de consumo impulsivo, evita gastos excessivos e promove uma gestão mais sustentável dos recursos pessoais. Além disso, ao consolidar várias fontes num único sistema, reduz a necessidade de múltiplas pesquisas dispersas, o que melhora a eficiência no fluxo de informação.

2.2.1.3 Parcerias para a Implementação dos Objetivos (ODS 17)

A *BudgetBreak* depende da integração de múltiplos serviços externos (como *Skyscanner*, *Booking*, *Airbnb*), exemplificando o ODS 17 ao promover parcerias digitais para criar um serviço mais completo e eficiente. A colaboração com diferentes fornecedores de dados reforça a importância da interoperabilidade e das tecnologias colaborativas no desenvolvimento de soluções escaláveis.

2.2.2 Viabilidade económica

A análise económica do projeto demonstra um equilíbrio sólido entre os custos de desenvolvimento e o potencial de receitas no mercado das soluções digitais de

planeamento de viagens. O modelo de negócio assenta principalmente em comissões obtidas através de reservas de voos e alojamentos, uma fonte de rendimento estável e amplamente validada no setor.

Além disso, existem oportunidades adicionais como funcionalidades premium agregadas a uma subscrição, parcerias comerciais e publicidade dirigida que reforçam a sustentabilidade financeira do produto ao longo do tempo.

A estrutura de custos mantém-se controlada graças à natureza digital da plataforma, centrada em desenvolvimento, manutenção, infraestrutura *cloud* e integrações por *API*. A escalabilidade tecnológica assegura que o aumento de utilizadores não implica uma subida significativa das despesas operacionais, permitindo maior eficiência à medida que o serviço cresce.

Os resultados do estudo realizado mostram um interesse claro dos utilizadores em ferramentas que simplifiquem o planeamento de viagens, sobretudo quando incluem gestão por orçamento. Esta evidência confirma a existência de procura real e aponta para uma taxa de adesão favorável.

Em conjunto, estes fatores demonstram que o projeto apresenta fundamentos económicos sólidos para avançar, gerar retorno e manter um modelo financeiro sustentável num mercado competitivo.

2.3 Análise Comparativa com Soluções Existentes

2.3.1 Soluções existentes

2.3.1.1 Hopper

O *Hooper* é uma plataforma de viagens que se destaca por sua abordagem baseada em previsões de preços e comportamentos de mercado. Utiliza algoritmos para identificar os melhores momentos para reservar voos e hotéis, com base em padrões históricos de variação de preços. A plataforma é focada na otimização de custos e na eficiência do planeamento, mas não oferece funcionalidades de planeamento inverso ou descoberta inspiradora, limitando-se a um enfoque puramente económico[5].

2.3.1.2 Kiwi

O *kiwi* diferencia-se pela funcionalidade "*Nomad*", que permite planear viagens com múltiplos destinos, tendo em conta o orçamento disponível e o tempo de viagens. Essa abordagem mais flexível e personalizada atende a um público que busca soluções integradas e adaptáveis, combinando eficiência com criatividade no planeamento de viagens[6].

2.3.1.3 Google Travel

O *Google Travel* atua como um agregador massivo (*Metasearch Engine*), centralizando voos (*Google Flights*), hotéis e atividades. Tira partido do vasto histórico de dados do utilizador para sugerir destinos e apresenta a funcionalidade "Explorar", que visualiza preços num mapa global. No entanto, por operar como um intermediário que redireciona o tráfego para terceiros, não garante a integridade transacional da reserva

(o utilizador paga em sites diferentes), nem oferece uma verdadeira personalização orçamental que combine voo e hotel num produto único[7].

2.3.1.4 Skyscanner

O *Skyscanner* é uma referência na comparação de preços global, facilitando a pesquisa de voos, hotéis e aluguer de carros. A "Explore Everywhere" incentiva a descoberta baseada no preço mínimo, listando destinos do mais barato para o mais caro. Apesar da eficiência na pesquisa, esta lista é genérica e desprovida de contexto: não considera as preferências semânticas do utilizador (ex: "praia tranquila" vs. "cidade vibrante") nem cria pacotes, limitando-se a ser uma ferramenta utilitária de comparação[8].

2.3.1.5 Booking

A *Booking* é líder no segmento de alojamento, mas a sua oferta de pacotes (voo + alojamento) não é uma funcionalidade nativa do seu núcleo tecnológico. Esta oferta é operacionalizada através de uma parceria *white-label* com a *lastminute.com[9]* (na Europa). Isto significa que, ao reservar um pacote, o utilizador está tecnicamente a interagir com um agregador externo integrado na interface da *Booking*. Esta dependência de terceiros limita a flexibilidade do planeamento inverso e confirma que a combinação dinâmica de pacotes não é o foco central da plataforma.[10].

2.3.1.6 Airbnb

O *Airbnb* privilegia experiências autênticas e alojamento local, aproximando viajantes e anfitriões num ambiente de confiança e conectividade, embora também não ofereça planeamento por orçamento nem descoberta aleatória, o que indica que a sua estratégia se concentra mais no aspeto social e de experiência do que no aspeto operacional e financeiro[11].

2.3.1.7 Trivago

O Trivago posiciona-se exclusivamente como um motor de pesquisa focado em alojamento. A sua funcionalidade central é a comparação do preço do mesmo quarto em múltiplos sites de reserva. Ao contrário da proposta da *BudgetBreak*, o Trivago não realiza vendas (redireciona o utilizador), não integra voos e não permite qualquer tipo de planeamento de itinerário. Representa o exemplo clássico da fragmentação do mercado, onde o foco está na micro-otimização do preço de um item isolado, ignorando a experiência da viagem completa[12].

2.3.1.8 Expedia

A *Expedia* é uma das maiores OTAs (*Online Travel Agencies*) do mundo e o concorrente mais direto na componente de "Pacotes Dinâmicos" (*Bundles*). A plataforma permite agrupar Voo + Hotel numa única transação, oferecendo descontos cruzados. Contudo, o seu modelo mental permanece convencional: o utilizador deve saber para onde quer ir. A Expedia carece de funcionalidades de descoberta inspiracional baseadas no orçamento ("*Budget-First*") e a sua recente aposta em IA (assistente *Romie*) funciona

como um complemento conversacional, e não como o motor central de decisão do pacote[13].

2.3.2 Análise de benchmarking

A tabela infra apresenta uma análise comparativa entre a solução proposta *BudgetBreak* e outras plataformas existentes no mercado, destacando as funcionalidades principais e aferindo com o símbolo “x” a presença de cada característica

Tabela 1 - Benchmarking

Benchmarking									
Características	**BudgetBreak**	**Hooper**	**Kiwi**	**Google Travel**	**Booking**	**Airbnb**	**Skyscanner**	**Trivago**	**Expedia**
Pesquisa "*Budget-First*" (O orçamento define o destino)	x		Parcial ("*Nomad*")				Parcial ("*Explore*")		
Foco na inspiração emocional (Descoberta vs. Transação)	x					x	Parcial		
Modo "*Randomizer*" (Sugestão de destino surpresa)	x						Parcial (Genério)		
Pacotes Dinâmicos (voo + alojamento)	x				x (Parceiro)				x
Integração APIs externas	x	x	x	x	x	x	x	x	x
Cashback ou Recompensas		x			x (*Genius*)				x
Personalização e Descoberta via IA (LLM + Preferências do utilizador)	x			Parcial (Gemini)					Parcial (Romie)
Integridade Transacional (ACID)	x	x	x		x	x			x

2.3.3 Inovação da solução

A *BudgetBreak* transforma como o planeamento de viagens é realizado ao inverter o modelo tradicional de decisão. Em vez de o utilizador adaptar as suas preferências às ofertas existentes, a plataforma gera automaticamente pacotes de viagem personalizados com base no orçamento definido utilizando:

- Mecanismos de *IA*, capazes de sugerir destinos, otimizar pacotes de viagens e indicar oportunidades de poupança;
- Integração com *API*s externas (*Skyscanner*, *Booking*, *Airbnb*, …) para acesso a dados atualizados e reais.
- Modo "*Randomizer*", que propõe destinos inesperados, inspiradores e personalizados, criando um elemento de descoberta e estímulo cognitivo

Esta abordagem representa uma inovação significativa, ao integrar eficiência operacional (otimização de custo e tempo), personalização e experiência inspiradora, distinguindo-se das plataformas tradicionais que focam apenas na pesquisa por destino ou data.

2.3.4 Mais-valias e benefícios

- Eficiência: Reduzir a complexidade do planeamento de viagens e fornecer resultados objetivos conforme o orçamento pré-definido.
- Acessibilidade: Democratizar o acesso a pacotes de viagens personalizadas, permitindo que diferentes viajantes planeiem viagens mais económicas e de qualidade.
- Impacto social: Incentivar experiências culturais diversificadas e acessíveis, promovendo inclusão turística.
- Sustentabilidade: Permitir escolhas mais conscientes ao sugerir voos e alojamentos que otimizam custos e distâncias.

2.4 Identificação de oportunidade de negócio

O mercado atual de planeamento de viagens apresenta uma fragmentação significativa que cria oportunidades claras para inovação. Plataformas como *Booking* e *Airbnb* especializam-se em alojamento, enquanto *Skyscanner* e *Google Flights* focam-se exclusivamente em voos. Aplicações agregadoras como a Kiwi oferecem comparação de preços, mas sem verdadeira integração e automatização inteligente. Esta fragmentação obriga os utilizadores a navegar entre múltiplas plataformas, construindo manualmente o *puzzle* completo de uma viagem, um processo demorado, fragmentado e sem garantia de otimização real do orçamento disponível.

A principal lacuna identificada é a ausência de soluções que partam do orçamento como ponto de partida. As plataformas existentes funcionam num modelo tradicional onde o utilizador define primeiro o destino e depois procura opções que se adequem à sua capacidade financeira, frequentemente resultando em frustração quando as escolhas excedem o orçamento ou em compromissos desnecessários por falta de visão global das alternativas. Nenhuma plataforma *mainstream* oferece a criação automática de pacotes

completos (voo + alojamento) otimizados para um limite financeiro predefinido, apesar do custo ser frequentemente o fator decisivo na escolha de viagens.

Esta oportunidade foi validada através do inquérito de pertinência anteriormente mencionado, aplicado a 115 utilizadores, maioritariamente viajantes regulares que já utilizam ferramentas digitais para planeamento (82,6%). Os resultados demonstram que 87,8% dos inquiridos enfrentam dificuldades em manter-se dentro do orçamento durante o planeamento, e 45,2% consideram o custo o fator mais importante na decisão de viagem. Quando questionados diretamente sobre o interesse numa aplicação que crie automaticamente pacotes baseados num orçamento máximo, 97,4% manifestaram interesse, sendo que 60% afirmaram que "definitivamente" utilizariam a solução. Adicionalmente, a satisfação média com as ferramentas atuais situa-se em apenas 3,32 numa escala de 5, evidenciando espaço significativo para melhoria e inovação no mercado.

A *BudgetBreak* posiciona-se para preencher esta lacuna através de uma abordagem que inverte o paradigma tradicional de planeamento. Utilizando mecanismos de Inteligência Artificial e integração com *API*s externas (*Skyscanner*, *Booking*, *Airbnb*, ...), a plataforma gera automaticamente pacotes personalizados partindo do orçamento definido pelo utilizador, otimizando simultaneamente custo, qualidade e coerência logística. O modo "*Randomizer*" adiciona um elemento único de descoberta, sugerindo destinos alternativos e inesperados dentro do orçamento, promovendo exploração criativa e distribuição mais equilibrada do turismo. Esta combinação de eficiência operacional e controlo financeiro efetivo distingue a *BudgetBreak* das ferramentas transacionais existentes.

3 Especificação e Modelação

3.1 Análise de Requisitos

3.1.1 Enumeração de Requisitos

Tabela 2 - Requisitos Funcionais

Requisitos Funcionais					
Requisito	**Nome**	**Descrição**	**Prioridade**	**Importância**	**Critério de Aceitação**
RQF01	Pesquisa de Alojamento por Orçamento e Local	O sistema deve permitir pesquisar alojamentos por cidade/país ou via "Randomizer", filtrando resultados que caibam no orçamento máximo definido.	Crítica	Alta	O preço total da estadia nos resultados deve ser menor ou igual ao do Orçamento proposto. O Randomizer deve sugerir destinos válidos dentro do preço.
RQF02	Filtros Temporais e de Capacidade	O sistema deve filtrar alojamentos por datas (check-in/out) e capacidade (n.º hóspedes/quartos).	Crítica	Alta	Devem ser mostrados apenas alojamentos com disponibilidade total nas datas e capacidade suficiente para o grupo.
RQF03	Filtros Qualitativos de Alojamento	O sistema deve permitir filtrar por Tipo (Hotel, Hostel), Comodidades (Wi-Fi, Piscina) e Intervalo de Preço.	Importante	Média	A seleção múltipla de filtros deve atualizar a lista de resultados corretamente.
RQF04	Ordenação de Resultados de Alojamento	O sistema deve permitir a ordenação dos resultados por Preço (cresc/decresc) e Rating (melhores avaliações).	Importante	Média	A reordenação da lista deve ser imediata e refletir o critério escolhido.
RQF05	Visualização de Detalhes e Multimédia	O sistema deve apresentar a ficha do alojamento com galeria de fotos, descrição e mapa.	Crítica	Alta	As fotos devem carregar corretamente. A descrição deve ser legível.

Requisitos Funcionais					
Requisito	**Nome**	**Descrição**	**Prioridade**	**Importância**	**Critério de Aceitação**
RQF06	Sistema de Avaliações e Favoritos	O sistema deve permitir ler/escrever reviews e guardar alojamentos nos favoritos.	Importante	Média	O utilizador deve conseguir adicionar aos favoritos, e deve conseguir escrever reviews.
RQF07	Políticas e Sustentabilidade	O sistema deve destacar e permitir filtrar por políticas de cancelamento flexíveis e selos eco-friendly.	Desejável	Baixa	O ícone de sustentabilidade deve ser visível.
RQF08	Motor de Recomendação "Randomizer" contextual	O algoritmo deve sugerir alojamentos baseados no perfil e histórico do utilizador	Crítica	Alta	As sugestões devem variar consoante as preferências guardadas no perfil do utilizador.
RQF09	Pesquisa de Voos por Orçamento e Datas	O sistema deve pesquisar voos (ida/volta) que respeitem o orçamento do pacote e datas selecionadas.	Crítica	Alta	Resultados devem ser apresentados apenas se disponíveis para as datas e dentro do limite de preço.
RQF10	Filtros de Características de Voo	O sistema deve filtrar por Classe, Companhia Aérea e Tipo de Voo (Direto/Escalas).	Importante	Média	O filtro "Direto" deve eliminar voos com escalas. Seleção de classe deve atualizar o preço.
RQF11	Filtros de Serviços Incluídos	O sistema deve permitir filtrar voos que já incluam bagagem de porão e refeições no preço base.	Desejável	Baixa	A lista deve mostrar apenas tarifas "com bagagem" se o filtro for ativado.
RQF12	Ordenação de Voos	O sistema deve ordenar voos por Preço, Duração e Horário de partida.	Importante	Média	O utilizador deve conseguir identificar rapidamente o voo mais barato ou mais rápido.

Requisitos Funcionais					
Requisito	**Nome**	**Descrição**	**Prioridade**	**Importância**	**Critério de Aceitação**
RQF13	Seleção de Serviços	O sistema deve permitir adicionar bagagem extra, escolher lugar e solicitar assistência especial.	Importante	Média	O custo da bagagem extra e lugar deve ser somado ao total antes do pagamento.
RQF14	Disponibilidade em Tempo Real (Alojamentos)	O sistema deve validar a disponibilidade dos alojamentos no momento da seleção.	Crítica	Alta	Se o quarto deixar de estar disponível durante a navegação, o sistema deve alertar e bloquear a reserva.
RQF15	Disponibilidade em Tempo Real (Voos)	O sistema deve validar a disponibilidade dos voos no momento da seleção.	Crítica	Alta	Se o lugar deixar de estar disponível durante a navegação, o sistema deve alertar e bloquear a reserva.
RQF16	Partilha de Itinerário	O sistema deve permitir partilhar os detalhes do voo com terceiros.	Desejável	Baixa	A partilha deve gerar um link ou texto formatado com os horários e números de voo.
RQF17	Motor de Combinação de Pacotes	O sistema deve gerar combinações automáticas (Voo + Alojamento) que não excedam o orçamento total.	Crítica	Alta	A soma (Preço Voo + Preço Alojamento) deve ser menor ou igual ao Orçamento proposto.
RQF18	Estratégias do "Randomizer" de Pacotes	O sistema deve sugerir mixes variados: Voo Barato/Hotel Bom vs. Voo Premium/Hotel Simples.	Desejável	Baixa	O sistema deve apresentar pelo menos 3 opções de pacotes com racios de preço diferentes.
RQF19	Criação e Confirmação de Reserva Unificada	O sistema deve processar a reserva de múltiplos items (pacote) numa única transação.	Crítica	Alta	O checkout deve resultar na reserva efetiva em ambos os fornecedores (aéreo e hotelaria).

Requisitos Funcionais					
Requisito	**Nome**	**Descrição**	**Prioridade**	**Importância**	**Critério de Aceitação**
RQF20	Gestão de Cancelamento de Reservas	O sistema deve permitir cancelar reservas (total ou parcial) e calcular reembolsos automaticamente.	Importante	Alta	O sistema deve informar o valor exato a reembolsar segundo as políticas vigentes.
RQF21	Alteração de Reservas	O sistema deve permitir modificar datas ou detalhes da reserva sujeito a disponibilidade e taxas.	Importante	Baixa	A alteração deve recalcular o preço e exigir pagamento da diferença, se houver.
RQF22	Validação de Resumo Final	O sistema deve exigir a validação de um resumo completo (termos, condições, preços finais) antes do pagamento.	Crítica	Alta	O botão de "Pagar" só fica ativo após visualização do resumo.
RQF23	Autenticação e Registo	O sistema deve permitir registo/login via e-mail e Social Login, e recuperação de password.	Crítica	Alta	Login com Google/Facebook deve funcionar. E-mail de recuperação deve chegar em menos de 2 minutos.
RQF24	Gestão de Perfil Pessoal	O utilizador deve poder editar dados pessoais, password e contactos.	Importante	Média	As alterações devem ser persistidas na base de dados imediatamente.
RQF25	Preferências de Viagem e Acessibilidade	O sistema deve guardar preferências (destino, moeda, idioma) e necessidades especiais.	Importante	Média	O sistema deve iniciar na moeda/idioma preferido do utilizador.
RQF26	Histórico de Atividade	O sistema deve listar todas as reservas passadas e futuras com o respetivo estado.	Importante	Baixa	A lista deve permitir clicar numa reserva antiga para ver detalhes.

Requisitos Funcionais					
Requisito	**Nome**	**Descrição**	**Prioridade**	**Importância**	**Critério de Aceitação**
RQF27	Personalização via Histórico	O sistema deve usar o histórico de reservas para afinar as sugestões do "Randomizer".	Desejável	Baixa	Utilizadores com histórico de "Hotéis 5*" devem ver menos sugestões de "Hostels".
RQF28	Notas na Reserva	O utilizador deve poder adicionar pedidos especiais (ex: cama extra) no momento da reserva.	Importante	Baixa	As notas devem ser enviadas para o fornecedor do serviço.
RQF29	Processamento de Cartões	O sistema deve processar cartões Crédito/Débito via gateway seguro com encriptação	Crítica	Alta	Dados sensíveis não podem ser guardados em plain text. Suporte a 3DSecure.
RQF30	Pagamentos Alternativos	O sistema deve suportar pagamentos via MBWay e PayPal.	Crítica	Alta	A integração deve redirecionar corretamente e confirmar o pagamento via callback.
RQF31	Comprovativos e Faturação	O sistema deve gerar recibo digital e enviar confirmação de pagamento imediata.	Crítica	Alta	O e-mail de confirmação deve conter anexo ou link para a fatura/recibo.
RQF32	Tratamento de Falhas e Reembolsos	O sistema deve notificar falhas de pagamento e permitir nova tentativa ou gestão de reembolsos na app.	Importante	Alta	Em caso de erro, o carrinho não deve ser esvaziado, permitindo nova tentativa.
RQF33	Carteira de Pagamento	O sistema deve permitir guardar métodos de pagamento tokenizados para uso futuro.	Desejável	Baixa	O utilizador deve conseguir pagar uma nova reserva sem reintroduzir o cartão completo.

Requisitos Funcionais					
Requisito	**Nome**	**Descrição**	**Prioridade**	**Importância**	**Critério de Aceitação**
RQF34	Notificações Transacionais	O sistema deve enviar confirmações de reserva e vouchers digitais (Email/App).	Crítica	Alta	O voucher deve conter QR Code ou localizador válido.
RQF35	Alertas Operacionais de Viagem	O sistema deve notificar sobre Check-in (24h/2h) e alterações de porta/horário.	Importante	Média	As notificações push devem ser entregues no tempo estipulado
RQF36	Sistema de Ticketing	O sistema deve permitir abrir tickets de suporte associados a uma reserva específica.	Importante	Média	O ticket deve ficar visível na área de "Suporte" com o estado "Aberto".
RQF37	Canais de Ajuda Rápida	O sistema deve disponibilizar FAQ contextual e Chat Widget para dúvidas simples.	Desejável	Baixa	O widget deve estar acessível em todas as páginas do processo de reserva.

Tabela 3 - Requisitos Não Funcionais

Requisitos Não Funcionais					
Requisito	**Nome**	**Descrição**	**Tipo**	**Prioridade**	**Critério de Aceitação**
RQNF01	Encriptação de Dados	Todos os dados sensíveis (pagamento, passwords) devem ser encriptados em trânsito (TLS 1.3)	Segurança	Crítica	Nenhuma password ou n.º cartão visível na BD ou logs.
RQNF02	Autenticação Multifator (MFA)	O sistema deve suportar 2FA (Email/SMS) para operações sensíveis ou login suspeito.	Segurança	Importante	Solicitação de código extra ao alterar password ou pagar.
RQNF03	Conformidade RGPD	O sistema deve permitir ao utilizador descarregar os seus dados ou solicitar "esquecimento" (eliminação).	Legal	Crítica	Funcionalidade de exportação de dados disponível no perfil.
RQNF04	Sincronismo em Tempo Real	A verificação de disponibilidade com fornecedores externos deve ocorrer em menos de 5 segundos no checkout.	Performance	Crítica	Feedback imediato se o fornecedor rejeitar.
RQNF05	Internacionalização (i18n)	O sistema deve suportar múltiplos idiomas (PT, EN) na interface e notificações.	Usabilidade	Importante	Troca de idioma instantânea sem recarregar dados.
RQNF06	Gestão Multimoeda	O sistema deve apresentar preços convertidos na moeda do utilizador, atualizando taxas de câmbio diariamente.	Usabilidade	Importante	Preço final exibido na moeda selecionada.
RQNF07	Responsividade (Mobile-First)	A interface deve adaptar-se fluidamente a ecrãs móveis, tablets e desktop.	Usabilidade	Crítica	Sem barras de scroll horizontal ou elementos sobrepostos em mobile.

Requisitos Não Funcionais					
Requisito	**Nome**	**Descrição**	**Tipo**	**Prioridade**	**Critério de Aceitação**
RQNF08	Integração com APIs de Terceiros	O sistema deve integrar via REST com provedores de voos e hotéis	Integração	Crítica	Comunicação bem-sucedida e tratamento de erros de API
RQNF09	Integridade Transacional	As reservas de pacotes devem seguir o princípio ACID (Atomicidade): ou reserva voo E hotel, ou não reserva nada.	Fiabilidade	Crítica	Sem reservas "órfãs" (voo pago sem hotel reservado).

3.1.2 Descrição detalhada dos requisitos principais

Nesta secção, são detalhados os requisitos funcionais de maior impacto para o sucesso do sistema. Estes requisitos representam o "core business" da aplicação, focando-se na pesquisa orçamentada, na integridade das reservas e na segurança dos pagamentos.

3.1.2.1 RQF01 – Pesquisa de Alojamento por Orçamento e Local

Descrição: Este requisito permite ao utilizador iniciar a pesquisa definindo um destino ou usando o "*Randomizer*", aplicando como critério base o orçamento máximo da estadia. O sistema deve filtrar e devolver apenas alojamentos compatíveis com o limite financeiro, funcionando como o ponto de entrada do fluxo de reserva.

Objetivo: Permitir ao utilizador pesquisar alojamentos de forma rápida e intuitiva, garantindo que os resultados respeitam o orçamento proposto.

Dependências:

- RQF02 - Filtros temporais e de capacidade
- RQF03 - Filtros qualitativos
- RQF05 - Visualização de detalhes e multimédia
- RQF08 - Motor de Recomendação "*Randomizer*" contextual
- RQF09 - Pesquisa de Voos por Orçamento e datas.
- RQF22 - Validação de Resumo Final
- RQF23 - Autenticação e Registo
- RQF14 – Disponibilidade em Tempo Real (Alojamentos)
- RQF17 - Motor de Combinação de Pacotes

Critério de aceitação: O preço total da estadia nos resultados deve ser menor ou igual ao do orçamento proposto. O "*Randomizer*" deve sugerir destinos válidos dentro do preço.

3.1.2.2 RQF02 – Filtros Temporais e de Capacidade

Descrição: Estabelece os filtros essenciais de datas de *check-in*/*check-out* e de capacidade do alojamento. O sistema deve validar disponibilidade real para as datas e garantir que o alojamento suporta o número de hóspedes/quartos selecionado.

Objetivo: Permitir filtrar alojamentos por datas e capacidade, assegurando que apenas opções compatíveis são apresentadas.

Dependências:

- RQF01 - Pesquisa de Alojamento por Orçamento e Local
- RQF03 - Filtros Qualitativos de Alojamento
- RQF14 – Disponibilidade em Tempo Real (Alojamentos)
- RQF08 - Motor de Recomendação "*Randomizer*" contextual

Critério de Aceitação: Devem ser mostrados apenas alojamentos com disponibilidade total nas datas e capacidade suficiente para o grupo.

3.1.2.3 RQF05 - Visualização de Detalhes e Multimédia

Descrição: O sistema deve apresentar a página completa do alojamento, incluindo descrição estruturada, galeria de fotos, mapa e informações relevantes. Esta página é essencial para decisão de compra, devendo ser clara, rápida e completa.

Objetivo: Permitir ao utilizador visualizar detalhes completos do alojamento antes de avançar com a reserva.

Dependências:

- RQF01 – Pesquisa de Alojamento por Orçamento e Local
- RQF02 - Filtros Temporais e de Capacidade
- RQF03 - Filtros Qualitativos de Alojamento
- RQF22 - Validação de Resumo Final
- RQF14 Disponibilidade em Tempo Real (Alojamentos)

Critério de Aceitação: As fotos devem carregar corretamente. A descrição deve ser legível.

3.1.2.4 RQF08 - Motor de Recomendação "*Randomizer*" contextual

Descrição: O "*Randomizer*" deve analisar o perfil e histórico do utilizador para apresentar recomendações personalizadas e coerentes com orçamento, preferências guardadas e comportamento passado. O sistema ajusta dinamicamente sugestões a partir destes dados.

Objetivo: Gerar sugestões personalizadas baseadas no perfil e histórico do utilizador.

Dependências:

- RQF01 - Pesquisa de Alojamento por Orçamento e Local
- RQF03 - Filtros Qualitativos de Alojamento
- RQF17 - Motor de Combinação de Pacotes
- RQF27 - Personalização via histórico
- RQF14 – Disponibilidade em Tempo Real (Alojamento)

Critério de Aceitação: As sugestões devem variar consoante as preferências guardadas no perfil do utilizador.

3.1.2.5 RQF09 - Pesquisa de Voos por Orçamento e Datas

Descrição: O sistema deve pesquisar voos reais via integração com APIs externas, filtrando resultados por preço máximo e datas pretendidas. Apenas voos disponíveis e dentro do orçamento devem ser apresentados.

Objetivo: Permitir pesquisar voos ida/volta que respeitem o orçamento do pacote e as datas selecionadas pelo utilizador, suportando integração com APIs externas para resultados reais.

Dependências:

- RQF01 - Pesquisa de Alojamento por Orçamento e Local
- RQF10 - Filtros de Características de Voos
- RQF12 - Ordenação de Voos
- RQF15 - Disponibilidade em Tempo Real (Voos)
- RQF17 - Motor de Combinação de Pacotes

Critério de Aceitação: Resultados devem ser apresentados apenas quando disponíveis para as datas e dentro do limite de preço.

3.1.2.6 RQF14 – Disponibilidade em Tempo Real (Alojamentos)

Descrição: Valida, no momento da seleção, a existência real de quartos disponíveis através de *API*s externas. Previne erros de reserva e overbooking

Objetivo: Garantir disponibilidade real do alojamento no ato da seleção.

Dependências:

- RQF01 - Pesquisa de Alojamento por Orçamento e Local
- RQF17 - Motor de Combinação de Pacotes
- RQF19 - Criação e Confirmação de Reserva Unificada

Critério de Aceitação: Se o quarto deixar de estar disponível durante a navegação, o sistema deve alertar e bloquear a reserva.

3.1.2.7 RQF15 - Disponibilidade em Tempo Real (Voos)

Descrição: Valida, no momento da seleção, a existência real de lugares no voo através de *API*s externas. Previne erros de reserva e *overbooking*.

Objetivo: Garantir disponibilidade real do voo no ato da seleção.

Dependências:

- RQF09 - Pesquisa de Voos por Orçamento e Datas
- RQF17 - Motor de Combinação de Pacotes
- RQF19 - Criação e Confirmação de Reserva Unificada

Critério de Aceitação: Se o lugar deixar de estar disponível durante a navegação, o sistema deve alertar e bloquear a reserva.

3.1.2.8 RQF17 - Motor de Combinação de Pacotes

Descrição: O sistema deve gerar automaticamente combinações entre voos e alojamentos, calculando preços, disponibilidades e garantindo que o pacote não excede o orçamento total definido. Produz múltiplas opções viáveis.

Objetivo: Criar combinações automáticas de Voo + Alojamento dentro do orçamento.

Dependências:

- RQF01 - Pesquisa de Alojamento por Orçamento e Local
- RQF08 - Motor de Recomendação "*Randomizer*" contextual

- RQF09 - Pesquisa de Voos por Orçamento e Datas
- RQF19 - Criação e Confirmação de Reserva Unificada

Critério de Aceitação: A soma (Preço Voo + Preço Alojamento) deve ser menor ou igual ao Orçamento proposto.

3.1.2.9 RQF19 - Criação e Confirmação de Reserva Unificada

Descrição: Este requisito centraliza todo o processo de *checkout*. Permite reservar voo, alojamento e serviços adicionais numa única transação, garantindo atomicidade: ou tudo confirma, ou nada é reservado.

Objetivo: Processar a reserva completa numa única transação consistente.

Dependências:

- RQF15 - Disponibilidade em Tempo Real (Voos)
- RQF14 – Disponibilidade em Tempo Real (Alojamentos)
- RQF17 - Motor de Combinação de Pacotes
- RQF20 - Gestão de Cancelamento de Reservas
- RQF29 - Processamento de Cartões
- RQF30 - Pagamentos Alternativos
- RQF22 - Validação de Resumo Final

Critério de Aceitação: O *checkout* deve resultar na reserva efetiva em ambos os fornecedores (aéreo e hotelaria).

3.1.2.10 RQF22 - Validação de Resumo Final

Descrição: Antes do pagamento, o sistema deve apresentar todas as informações da compra como preços, termos, detalhes do voo e alojamento e bloquear o botão de pagamento até o utilizador validar.

Objetivo: Garantir que o utilizador confirma o resumo antes de pagar.

Dependências:

- RQF05 - Visualização de Detalhes e Multimédia
- RQF19 - Criação e Confirmação de Reserva Unificada
- RQF23 – Autenticação e Registo

Critério de Aceitação: O botão de "Pagar" só fica ativo após visualização/validação do resumo.

3.1.2.11 RQF23 - Autenticação e Registo

Descrição: O sistema deve permitir criar conta, iniciar sessão, recuperar palavra-passe e efetuar *login* social (*Google*/*Facebook*), permitindo guardar preferências e aceder ao histórico.

Objetivo: Permitir autenticação segura e acesso personalizado.

Dependências:

- RQF22 - Validação de Resumo Final
- RQF24 - Gestão de Perfil Pessoal

Critério de Aceitação: Login com *Google*/*Facebook* deve funcionar. E-mail de recuperação deve ser enviado em menos de 2 minutos.

3.1.2.12 RQNF08 – Integração com *APIs* de Terceiros

Descrição: O sistema deve possuir uma camada de integração robusta (*Backend-for-Frontend* ou *Middleware*) capaz de comunicar via protocolo *REST* com os sistemas externos de gestão de inventário de voos e alojamentos. Esta integração deve gerir a autenticação segura, a serialização de dados (*JSON*/*XML*) e o mapeamento de campos entre o modelo de dados interno e as estruturas externas.

Objetivo: Permitir que a aplicação consuma e apresente dados dinâmicos (preços, disponibilidade, metadados) em tempo real provenientes de múltiplos fornecedores globais, centralizando a operação numa interface única para o utilizador.

Dependências:

- RQF01 – Pesquisa de Alojamento por Orçamento e Local
- RQF09 - Pesquisa de Voos por Orçamento e Datas
- RQF14 - Disponibilidade em Tempo Real (Alojamentos)
- RQF15 - Disponibilidade em Tempo Real (Voos)
- RQF19 - Criação e Confirmação de Reserva Unificada
- RQNF01 - Encriptação de Dados (Segurança na transmissão)
- RQNF04 - Sincronismo em Tempo Real (Performance da API

Critério de Aceitação: A comunicação com as *API*s externas deve retornar códigos de sucesso (HTTP 200) com dados estruturados corretamente. O sistema deve ser capaz de capturar erros de comunicação (4xx/5xx) e apresentar mensagens de erro amigáveis ou tentar rotas alternativas sem "crashar" a aplicação.

3.1.2.13 RQNF09 – Integridade Transacional

Descrição: O sistema deve implementar mecanismos de controlo de transações distribuídas (como o padrão *Saga* ou *Two-Phase Commit*) para assegurar que a compra de um pacote (Voo + Alojamento) é tratada como uma operação atómica. O sistema deve orquestrar as chamadas de reserva de modo que, se um dos componentes (ex: o hotel) falhar a confirmação, o outro componente (ex: o voo) seja automaticamente revertido ou cancelado antes da captura final do pagamento.

Objetivo: Garantir a consistência dos dados e financeira, assegurando que o utilizador nunca fica com uma “reserva órfã” (ex: um voo pago sem ter onde dormir, ou vice-versa) e que o pacote vendido respeita integralmente as condições selecionadas.

Dependências:

- RQF17 - Motor de Combinação de Pacotes
- RQF19 - Criação e Confirmação de Reserva Unificada
- RQF29 - Processamento de Cartões (Autorização vs. Captura)

- RQF32 - Tratamento de Falhas e Reembolsos
- RQNF08 - Integração com *API*s de Terceiros (Necessário para enviar comandos de *rollback*)

Critério de Aceitação: Em testes de falha simulada (onde o serviço de hotel rejeita a reserva após o voo estar reservado), o sistema deve executar automaticamente o cancelamento do voo e não deve capturar o valor financeiro do cartão do utilizador, retornando o estado da encomenda para "Falha/Cancelado" sem intervenção manual.

3.1.3 Casos de Uso/*User Stories*

Para o planeamento inicial e definição dos requisitos do sistema, adotou-se a metodologia *Waterfall* (Cascata). Esta abordagem permitiu estabelecer a base estrutural e o âmbito do projeto apresentados nos capítulos anteriores.

Para a etapa de desenvolvimento e implementação, o projeto transita para um Modelo de Integração Contínua (CI/CD) Orientado a Módulos. Esta abordagem substitui a metodologia Ágil tradicional, permitindo gerir tarefas de forma centralizada por domínio arquitetural e garantindo extrema agilidade na validação do código produzido.

Neste contexto, as *User Stories* não são listadas exaustivamente neste documento, sendo geridas num ambiente centralizado. A organização adotada reflete a estrutura funcional do projeto, dividindo-se em categorias que correspondem a cada *Epic* (E1 – Reserva de alojamentos; E2 – Reserva de voo; E3 – Reserva de pacotes; E4 – Utilizador; E5 – Pagamentos; E6 – Notificações), agregando as respetivas *features*, *user stories* e *technical user stoires* associadas.

Todos os materiais, documentação e recursos complementares encontram-se disponíveis para consulta na plataforma *Trello*.

3.2 Modelação

O desenvolvimento da aplicação *BudgetBreak* seguiu rigorosamente as práticas fundamentais de Engenharia de Software, visando criar uma arquitetura robusta, escalável e de fácil manutenção. Nesta secção, apresentamos os artefactos de modelação que serviram de base para a implementação do sistema.

3.2.1 Modelo de Dados (Diagrama de Entidade-Relação)

Conforme representado no Diagrama de Entidade-Relação [Anexo 2 – Artefactos de Modelação - Figura 11], a estrutura relacional organiza-se da seguinte forma:

- Núcleo da Transação (*Booking*, *Bundle*, *Calendar*): A entidade *Booking* (Reserva) é o elemento central que liga o *User* ao serviço adquirido. Ao contrário de modelos tradicionais, a reserva não se liga diretamente a voos soltos, mas sim a um *Bundle*. A entidade *Bundle* agrega as componentes de *Flight* e *Accommodation*, garantindo que o utilizador adquire um conjunto coerente. A entidade *Calendar* está associada à reserva para gerir a sincronização temporal do itinerário do utilizador.

- Gestão de Utilizador & Preferências: O domínio de identidade (*User*) suporta um forte Controlo de Acessos Baseado em Funções (*RBAC*), com as entidades *Role* e *Permission*. Para alimentar o algoritmo de recomendações, o modelo inclui a entidade *Preference* (associada bidirecionalmente ao *User*), que armazena dados como o ritmo de viagem (*TravelPace*) e o tipo de destino (DestinationType).
- Componentes do Serviço: Refletindo da nossa arquitetura, o núcleo do sistema não armazena a totalidade dos dados turísticos, mas atua como orquestrador. Os detalhes físicos de *Flight* (com as suas respetivas *AvailabilityFligh*t) e *Accommodation* (com *RoomType*, *RoomUnit*, etc.) são consumidos via *API*s externas (*Fake-API*s). O modelo relacional interno armazena apenas os identificadores únicos (*UUID*s) para garantir a consistência sem duplicação de dados voláteis. A localização geográfica é normalizada através das entidades *City*, e *Address*.
- Transacional & Social: O modelo contempla ainda a entidade *Payment*, que regista o estado e método das transações financeiras, e as entidades *Review* e *Notification*, que enriquecem a interação social e a comunicação com o utilizador. Este esquema encontra-se normalizado, garantindo que não existe duplicação desnecessária de dados (ex: a morada está separada do hotel) e que todas as relações possuem chaves estrangeiras (*Foreign Keys*) explícitas.

3.2.2 Modelo de Classes (Diagrama de Classes)

O Diagrama de Classes [Anexo 2 – Artefactos de Modelação - Figura 12] representa a estrutura orientada a objetos do sistema (arquitetura *backend*), refletindo não apenas no armazenamento de dados, mas também na lógica de orquestração.

Destacam-se as seguintes decisões de modelação e negócio:

- Estratégia abordada: Para garantir legibilidade do modelo, os atributos de coleção (listas de entidades) foram intencionalmente omitidos das tabelas. Por outro lado, a cardinalidade (*ex: **1..***) e o sentido justificam a implementação estrutural. O modelo distingue claramente dependências de Agregação (permite a existência independente na base de dados) e Composição (onde entidades filhas dependem da existência do pai).
- Classes de Domínio Principal: A classe *Booking* atua como o controlador da lógica de reserva, possuindo métodos como "*confirmBooking()*", "*cancelBooking()*". Ela compõe-se de instâncias de *User*, *Payment* e, crucialmente, de *Package*.
- Gestão de Pacotes & Associações Complexas: A classe *Package* agrega *Accommodation* e *Flight*. Um detalhe técnico relevante neste diagrama é a introdução da classe associativa *PackageFlight*. Esta classe intermediária é necessária para gerir atributos específicos da relação entre um pacote e um voo, como o *seatNumber* e *baggageWeight*, que não pertencem exclusivamente nem ao voo nem ao pacote, mas sim à sua ligação.
- Serviços & Disponibilidade: A classe *Accommodation* relaciona-se com *RoomType* , que por sua vez gere múltiplas instâncias físicas (*RoomUnit*). O sistema valida a

capacidade real através do método "*checkAvailability()*", se existe capacidade para satisfazer um pedido (*RoomAvailability*).

- Integração de Funcionalidades: As classes *Payment* e *Notification* expõem métodos de negócio essenciais, como "*processPayment()*" e "*markAllAsRead()*", respetivamente, demonstrando como o sistema manipula o estado das operações e a interação com o utilizador.

Este diagrama serve como base para o desenvolvimento das entidades *JPA* e dos serviços no *backend* (*Spring Boot*), assegurando que a estrutura do código está alinhada com os requisitos de dados e funcionalidade.

3.2.3 Modelo Comportamental (Diagramas de Sequência)

Para complementar a visão estrutural do sistema, foram elaborados Diagramas de Sequência [Anexo 2 – Artefactos de Modelação], que ilustram o comportamento dinâmico e a interação entre os diversos componentes ao longo do tempo. Devido à sua criticidade para a integridade da plataforma, a modelação comportamental focou-se detalhadamente nos fluxos do módulo de Autenticação e Autorização (*Security & JWT*).

Estes artefactos demonstram a separação de responsabilidades (SoC - *Separation of Concerns*) entre os controladores, os serviços de negócio e os componentes de segurança do *Spring Security*:

- Interceção e Validação de Acessos (Filtro *JWT*): Este diagrama descreve o ciclo de vida de um pedido a um *endpoint* protegido. O *JwtAuthenticationFilter* atua como a primeira linha de defesa, intercetando o cabeçalho *Authorization*. Um detalhe arquitetural vital aqui demonstrado é: o JwtTokenService extrai o utilizador e os seus perfis (*Roles*) diretamente da assinatura do *token*, evitando consultas dispendiosas à base de dados. Posteriormente, o *SecurityContextHolder* é populado, permitindo que o *Spring Security* valide as anotações ao nível do método, decidindo se permite o acesso (*200 OK*) ou se devolve *403 Forbidden*.[Figura 13]
- Fluxo de Autenticação e Geração de Tokens (*Login*): O processo é iniciado quando o *Web Client* envia as credendicais para o *AuthController*. O diagrama ilustra como o *AuthService* delega a validação ao *AuthenticationManager*. Em caso de falha, entramos na exceção, retornando um erro *401 Unauthorized*. Sendo credenciais válidas, o *JwtTokenService* é encarregue de gerar um par de tokens (*Access Token* e *Refresh Token*), devolvendo-os ao cliente (*200 OK*). Este padrão garante que a *API* mantém-se *statless*, não guardando sessões em memória.[Figura 14]
- Renovação de Sessão (*Refresh Token*): Para mitigar o riscos de tokens de acesso com o tempo de vida longo, modelou-se o fluxo de *Refresh Token*. O diagrama ilustra os mecanismos de segurança adicionais: mesmo que o formato do *token* seja válido, o *AuthService* cruza a informação com o *UserDetailsService* para garantir que a conta do utilizador continua ativa antes de imitir um novo par de *tokens*. Caso a conta tenha sido desativada, o fluxo é interrompido com um *403 Forbidden*.[Figura 16]

- Encerramento de Sessão (*Logout*): Demonstra o fluxo simplificado onde o *AuthController* recebe o pedido de fim de sessão, extraindo o token atual (removendo o prefixo *"Bearer"*) e o processando a invalidação no *AuthService* garantindo o fecho seguro do clico de vida da autenticação do lado do cliente.[Figura 15]

Todos os materiais, documentação e recursos complementares encontram-se disponíveis para consulta na plataforma *Trello*.

3.3 Protótipos de Interface

O mapa aplicacional foi desenhado para refletir uma experiência de utilização moderna e direta, onde a hierarquia da informação privilegia a rapidez de conversão e a clareza na tomada de decisão. A estrutura organiza-se em três fluxos essências Autenticação, Gestão Pessoal e Reserva, que convergem a partir de um ponto de entrada unificado [Anexo 4 – Protótipos de Interface (*Mockups*) - Figura 20]

3.3.1 *Landing Page* (*Search Engine* & *Definition*)

Este é o ecrã principal e ponto de entrada da aplicação. Ao contrário das plataformas tradicionais, funde a componente institucional com o motor de pesquisa.

- Função: Permite ao utilizador definir imediatamente o orçamento máximo, datas de viagem e o número de viajantes.
- Caraterísticas: inclui o seletor para o modo "*Randomizer*", que ativa a pesquisa de destinos surpresa baseada no perfil do utilizador, e encaminha o fluxo para a lista de resultados ou para a autenticação.

3.3.2 *Authentication Screen* (Ecrã de Autenticação)

Módulo central de segurança e gestão de identidade que serve de barreira ou facilitador para o acesso a funcionalidades personalizadas.

- *Login*: Interface para a entrada de credenciais ou autenticação *google*.
- *Register*: Formulário para criação de uma nova conta, recolhendo dados essenciais para a personalização futura.

3.3.3 *User Dashboard* (Painel do Utilizador)

A área reservada onde o utilizador gere a sua relação com a plataforma após a autenticação

- *Profile Settings* & *Preferences*: Permite editar dados pessoais, métodos de pagamento guardados e definir preferências
- *My Trips* (*History & Upcoming*): Lista cronológica que separa as viagens futuras (com acesso a voucher e contagens decrescentes) do histórico de viagens passadas

3.3.4 *Results List* (Lista de Resultados)

O "coração" da experiência de descoberta, onde o motor de recomendação apresenta as opções geradas.

- Função: Exibe "*cards*" de pacotes combinados (Voo + Alojamento) que respeitam estritamente o orçamento definido na *Landing Page*.
- Interação: Permite ao utilizador filtrar (por comodidades, horários, rating) e ordenar as opções, facilitando a comparação rápida entre diferentes destinos ou hotéis.

3.3.5 Package Detail (Detalhe do Pacote)

O ecrã de decisão final, acedido ao selecionar um resultado específico.

- Função: Apresenta a informação granular dos dois componentes da viagem. Exibe a galeria de fotos e avaliações do alojamento, bem como os detalhes técnicos dos voos (escalas, companhias aéreas, bagagem).
- Objetivo: Fornecer toda a informação necessária para que o utilizador valide a escolha antes de avançar para o pagamento.

3.3.6 Checkout

A interface transacional e segura onde a reserva é efetivada.

- Função: Recolhe os dados finais dos passageiros e processa o pagamento unificado do pacote.
- Características: Inclui um resumo final de custos (para garantir transparência total) e integra-se com *gateways* de pagamento para confirmar a reserva em tempo real.

3.4 Mockups e Experiência Visual

Os *mockups* desenvolvidos materializam o mapa aplicacional através de uma estética minimalista, simples e apelativa, onde a limpeza visual privilegia a ausência de distrações e a rápida compreensão dos dados.[Anexo 4 – Protótipos de Interface (*Mockups*)]

- Minimalismo e legibilidade: Recurso predominante a fundos neutros e a um amplo espaço em branco para reduzir a sobrecarga cognitiva, direcionando o foco estritamente para a informação crítica da viagem.
- Organização Modular: Estruturação da interface em blocos visuais independentes que agrupam variáveis complexas (como detalhes do voo, alojamento e comodidades) num formato prático, acessível e semelhante.

Todos os materiais, documentação e recursos complementares encontram-se disponíveis para consulta na plataforma *Trello*.

4 Solução Proposta

4.1 Apresentação

A solução implementada corresponde a um *MVP* que valida o conceito proposto, integrando dados de *API*s externas de turismo, um motor de recomendações inteligente e funcionalidades de visualização de resultados. Dado que este projeto se concentra na integração do *backend,* mecanismos de recomendação e segurança, com um *frontend* minimalista, desenvolvido apenas com o propósito de demonstrar as funcionalidades *core* do sistema.

O seguinte capítulo está organizado em diversas secções como; Arquitetura, Tecnologias e Ferramentas Utilizadas, Ambientes de Teste e Produção, Abrangência Académica e Componentes, incluindo a descrição detalhada de cada módulo e das interfaces implementadas.

4.2 Arquitetura

A solução desenvolvida segue a arquitetura de uma aplicação monolítica organizada internamente em módulos [Figura 18], permitindo separação lógica de responsabilidades e preparação para futura escalabilidade horizontal. Esta arquitetura integra o *backend* onde serão implementadas a regras de negócio, *frontend* para interação do utilizador e base de dados para persistência de dados, facilitando a manutenção, evolução e futuras extensões do sistema de uma forma unificada [Figura 17].

O *frontend*, desenvolvido em *Next.js* com *Tailwind CSS*, fornece uma interface clara e funcional, permitindo visualizar resultados e recomendações. A integração com o *backend* é feita através de *REST*/*gRPC*, garantindo comunicação eficiente entre camadas.

Todo o sistema é containerizado com *Docker*, garantindo portabilidade, isolamento e facilidade de *deploy* em diferentes ambientes.

4.3 Tecnologias e Ferramentas Utilizadas

- *Spring Boot (Java)*: *framework* principal para desenvolvimento do *backend* modular, com suporte a segurança, *REST* e integração com *Kafka/gRPC*.
- *Next.js + Tailwind CSS*: *frontend* minimalista para visualização de funcionalidades do *MVP*.
- *PostgreSQL / MongoDB*: persistência de dados estruturados ou semi-estruturados, flexível conforme necessidade de escalabilidade.
- *Kafka*: Sistema de mensagens assíncronas, usado para fluxo de eventos interno.
- *gRPC*: comunicação eficiente para serviços externos de alta *performance*.
- *Docker*: containerização de toda a solução para portabilidade e facilidade de *deploy*.
- *LLMs + MCP*: inteligência artificial para motor de recomendações, garantindo contexto controlado e rastreável.

- Autenticação & segurança de dados: adoção de um modelo de segurança *stateless* (*Spring Security* + *JWT*) para garantir o escalonamento independe do servidor operando em conjunto com encriptação *TLS* e validação de *inputs*.

4.4 Ambientes de Teste e de Produção

4.4.1 Infraestrutura Computacional (Hardware)

O desenvolvimento e testes são sustentados por um *cluster* local de alto desempenho composto por quatro estações de trabalho (dois desktops e dois portáteis), equipadas com processadores Intel Core de última geração (i7 e Ultra 7) e 128 GB de memória RAM total. O ambiente possui aceleração gráfica dedicada (NVIDIA RTX 4070/5060 e Intel Arc) e processamento neural (NPU Intel AI Boost), essenciais para a execução de modelos de IA e renderização.

4.4.2 Capacidade de Processamento e Software

Os recursos disponíveis garantem a execução simultânea de toda a arquitetura baseada em contentores *Docker*, suportando o sistema em *Spring Boot*, bases de dados (*PostgreSQL*/*MongoDB*) e o sistema de mensagens *Kafka* sem latência. A capacidade gráfica e a *NPU* permitem testar localmente os algoritmos de recomendação e o "*Randomizer*" baseado em *IA*.

4.4.3 Armazenamento e Rede

Todas as unidades utilizam armazenamento *SSD NVMe* para garantir alta performance de I/O na gestão de dados. A conectividade é assegurada por rede *Gigabit* e *Wi-Fi 6E*, permitindo a comunicação eficiente entre os serviços distribuídos e a integração rápida com *APIs* externas (*Skyscanner*, *Booking*, *Airbnb*) via *REST* e *gRPC*.

4.5 Abrangência

O projeto integra conhecimentos de várias disciplinas do curso de engenharia informática, nomeadamente:

- Computação Distribuída: implementação de arquiteturas para aplicações escaláveis e resilientes, incluindo o uso de *APIs REST* e *gRPC*, integração de *Kafka* para comunicação assíncrona entre componentes e utilização de *PostgreSQL* para persistência de dados. Esta disciplina foi fundamental para compreender como estruturar aplicações complexas e coordenar fluxos de dados de forma eficiente.
- Engenharia de *Software* & Engenharia de Requisitos e Testes: planeamento, levantamento de requisitos e integração de tecnologias, assegurando que o protótipo funcional está alinhado com os objetivos definidos, aplicando práticas de análise, *design* e validação de *software*.
- Base de Dados: utilização de *MySQL* e *MSSQL* para persistência de dados estruturados, modelação de esquemas e implementação de consultas eficientes, aplicando os conceitos aprendidos na disciplina de Base de Dados.
- Programação *Web* & Desenvolvimento de *Interfaces Web*: construção do *frontend* utilizando *Next.js* e *Tailwind CSS*, integração com *backend* e base de dados, aplicação

de conceitos de *HTML*, *CSS*, *JavaScript* e *frameworks* modernas de desenvolvimento *web fullstack*. Para persistência no ambiente de programação *web*, foram utilizados *SQLite* e *PostgreSQL*, consolidando práticas de integração *frontend-backend*.

4.6 Componentes

4.6.1 Componente 1 – *Backend Core* (*Spring Boot* & *APIs* externas):

- Centraliza toda a lógica de negócio.
- Processa requisições do *frontend*.
- Coordena chamadas a *APIs* externas via *REST* e/ou *gRPC*.
- Orquestra eventos assíncronos com *Kafka*.

4.6.2 Componente 2 - Base de dados (*PostgreSQL*):

- Armazena dados de viagens, utilizadores e preferências.
- Permite consultas otimizadas e persistência eficiente.

4.6.3 Componente 3 - *Frontend* (*Next.js + Tailwind CSS*):

- Visualiza resultados de pesquisa e recomendações.
- Permite interação direta com o *backend* via *REST*/*gRPC*.
- Interface simples, responsiva e funcional.

4.6.4 Componente 4 - *Kafka*:

- Coordena eventos assíncronos entre módulos internos.
- Suporta comunicação futura com outros serviços distribuídos ou microserviços.

4.6.5 Componente 5 - *gRPC*:

- Comunicação de alta *performance* entre módulos internos e serviços externos.

4.6.6 Componente 6 - Motor de recomendações (*LLMs* + *MCP*):

- Interpreta preferências do utilizador, critérios de orçamento e localização.
- Gera recomendações personalizadas, mantendo o contexto limitado e auditável.
- Facilita a futura integração com funcionalidades adicionais, como o modo "*Randomizer*" de sugestões aleatórias.

4.6.7 Componente 7 – Docker:

- Containeriza todos os componentes da aplicação.
- Garante replicabilidade, isolamento e facilidade de *deployment*.

4.7 Interfaces

Esta secção detalha o processo de tradução dos requisitos funcionais em artefactos visuais de alta fidelidade. O desenvolvimento da interface de utilizador (UI) e da experiência do utilizador (UX) teve como principal objetivo a criação de uma plataforma intuitiva, cujo foco central é a pesquisa de viagens guiada por um orçamento predefinido. [Anexo 4 – Protótipos de Interface (*Mockups*)]

Para a conceção dos ecrãs, priorizaram-se princípios como a clareza da informação, a redução da carga cognitiva (evitando a fadiga de decisão) e a transparência. Todos os protótipos apresentados foram desenvolvidos com recurso ao software Figma.

4.7.1 Landing Page & Search

- Função: Atua como o ponto de entrada da aplicação, combinando a proposta de valor da marca com a ferramenta principal de pesquisa orientada ao orçamento. Apresenta ainda sugestões imediatas de destinos.
- Implementação: Estruturada com base num sistema de grelha (*grid*), dividindo a página em duas áreas de foco: o bloco superior de ação (pesquisa) e o bloco inferior de exploração (pacotes em destaque).[Figura 21]

4.7.2 Package List

- Função: Apresenta as opções de pacotes (alojamento + voo(s)) que cumprem estritamente os critérios de pesquisa do utilizador, permitindo a filtragem e ordenação.
- Implementação: Baseada num design modular (*Card-Based Design*). O layout foi dividido numa barra lateral (filtros) e uma área central de listagem, otimizando o espaço do ecrã para facilitar a leitura.[Figura 29]

4.7.3 Package detail

- Função: Fornece a informação aprofundada sobre o alojamento e os voos selecionados.
- Implementação: Desenvolvido com uma hierarquia de módulos de apresentção de dados. Esta abordagem separa claramente a informação descritiva do alojamento, os horários dos transportes e a área de transação (preços e reserva), permitindo agregar diferentes fontes de dados numa única vista fluida para o utilizador.[Figura 30]

4.7.4 Checkout & Payment

- Função: O ecrã transacional onde o utilizador introduza os dados dos viajantes e efetua o pagamento.
- Implementação: Interface em formato de formulário linear simples, com um resumo da reserva.[Figura 31]

Todos os materiais, documentação e recursos complementares encontram-se disponíveis para consulta na plataforma *Trello*.

5 Testes e Validação

O plano de testes deste projeto foi concebido para assegurar a estabilidade técnica e a qualidade do código desenvolvido, demonstrando ao mesmo tempo que a solução proposta cumpre de facto o seu propósito central – orquestrar pacotes de viagem rigorosamente dentro de um orçamento fechado.

5.1 Abordagem Metodológica

A estratégia de testes funcionais foi diretamente derivada de requisitos, e especialmente de Critérios de Aceitação. Adotámos uma abordagem em *Behavior-Driven-Development* (BDD), onde os cenários de teste foram descritos na sintaxe: **Given-When-Then-And.**

Todos os cenários de teste foram documentados previamente nas *User Stories* geridas no *Trello*. Isto permite-nos rastrear e garantir que cada funcionalidade implementada (ex: filtragem de voos, orquestração de pacotes) tem um cenário ou teste correspondente que atesta a sua conclusão (*Definition of Done*).

5.2 Testes de Qualidade e Código (Automatizado)

No nível estrutural, a qualidade da solução foi assegurada através de testes unitários desenvolvidos em *JUnit* dentro do ambiente de desenvolvimento *Intellij IDEA*.[Anexo 5 – Testes]

- Cobertura: Os testes incidiram essencialmente nas classes de Serviço (ex: *BundleServiceImpl*) e Controladores, garantindo que a lógica de negócio central (cruzamento de voos e hotéis face ao orçamento) funciona isoladamente e sem falhas.
- Métricas de Qualidade: Utilizámos o plugin *JaCoCo* (*Java Code Coverage*) para medir e gerar relatórios de percentagem de código coberto pelos testes, estabelecendo métricas de fiabilidade das funções de cálculo e filtragem.

5.3 Validação Operacional e Integração (Manual)

Para simular um ambiente de produção e validar o funcionamento operacional, os testes de integração focaram-se na comunicação entre o orquestrador principal e os serviços externos.

- Validação via *Swagger UI*:Todos os *endpoints* foram testados manualmente através das interfaces *OpenAPI* (*Swagger*). Isto permitiu-nos simular o comportamento de chamadas reais de *frontend,* validar os *payloads* de entrada/saída (JSON) e atestar a correta devolução de códigos de estado HTTP (*200 OK*, *400 Bad Request*, *403 Forbidden*).

Todos os materiais, documentação e recursos complementares encontram-se disponíveis para consulta na plataforma *Trello*.

6 Método e Planeamento

6.1 Planeamento inicial

O desenvolvimento do projeto segue uma metodologia de gestão híbrida, ajustada às fases do ciclo de vida do *software* e aos momentos de avaliação definidos no Regulamento de TFC.

- Fase de Conceção (*Waterfall*): Para a etapa inicial, correspondente à 1.ª Entrega Intercalar (outubro e novembro), adotou-se uma abordagem em cascata. Esta escolha foi determinante para estabilizar os requisitos funcionais, o modelo de dados e a arquitetura distribuída antes do início da implementação técnica, reduzindo a probabilidade de riscos de reengenharia precoce.
- Fase de Implementação (*CI/CD*): Para a fase de desenvolvimento (dezembro a junho), o trabalho é estruturado através de Ciclos de Validação de 3 dias, geridos visualmente num *Kanban* de módulos estáticos. Esta cadência estática permite a integração progressiva dos componentes técnicos através de marcadores visuais de estado, garantindo a estabilidade imediata e a validação contínua das funcionalidades críticas (IA e Comunicação).

O planeamento temporal do projeto encontra-se alinhado com as datas-chave do regulamento: a conclusão da conceção em novembro, a entrega do protótipo funcional na 2.ª Entrega Intercalar (12 de abril de 2026) e a defesa final em junho (26 de junho de 2026)

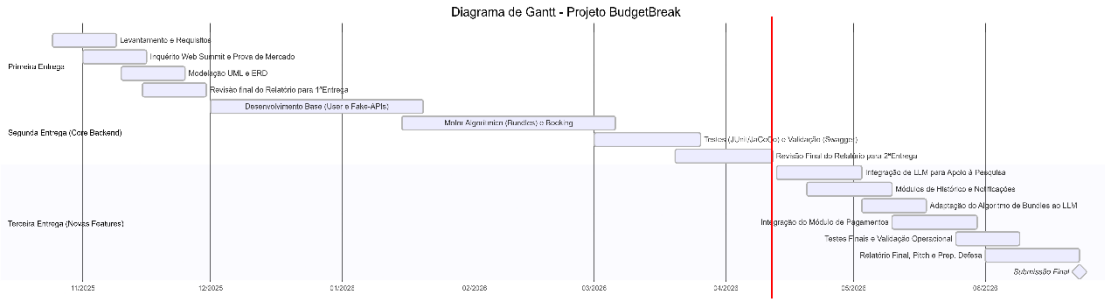

Figura 1 - Gantt Chart

6.2 Análise Crítica ao Planeamento

7 Resultados

7.1 Resultados dos Testes

7.2 Cumprimento de requisitos

8 Conclusão

8.1 Conclusão

8.2 Trabalhos Futuros

Bibliografia

[DEISI24] DEISI, Regulamento de Trabalho Final de Curso, Out. 2024.

[DEISI24b] DEISI, www.deisi.ulusofona.pt, Out. 2024.

[TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6ª Edição, Prentice Hall, 2020.

[ULHT21] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt, acedido em Out. 2024.

[1] «Travel & Tourism Economic Impact Research (EIR)». [Em linha]. Disponível em: https://wttc.org/research/economic-impact

[2] «Turismo tem aumento de 5% no primeiro trimestre do ano, diz agência da ONU | ONU News». [Em linha]. Disponível em: https://news.un.org/pt/story/2025/05/1848856

[3] «Online travel market statistics & facts | Statista». [Em linha]. Disponível em: https://www.statista.com/topics/2704/online-travel-market/#topicOverview

[4] «SiteMinder's Changing Traveller Report 2026». [Em linha]. Disponível em: https://www.siteminder.com/changing-traveller-report/

[5] «Hopper | Book Hotels, Flights, Car Rentals & More». [Em linha]. Disponível em: https://hopper.com/

[6] «Viagens de a». [Em linha]. Disponível em: https://www.kiwi.com/pt/?inboundDate=anytime&outboundDate=anytime&destination=-&origin=-

[7] «Google». [Em linha]. Disponível em: https://www.google.com/travel/

[8] «Compare voos baratos e reserve bilhetes de voos low cost | Skyscanner». [Em linha]. Disponível em: https://www.skyscanner.pt/

[9] «Booking.com powered by lastminute.com». [Em linha]. Disponível em: https://booking-pt.lastminute.com/?utm_source=booking&utm_medium=whitelabel&utm_campaign=homepage_tab&token=tJK3nStbF0FveAfBAUvvYjgtm7Sirg5rwuUmQ_yDPEMxNzY0Mzc0ODQ4&aid=2311236

[10] «Booking.com | Website oficial | Os melhores hotéis, voos, alugueres de carros e alojamentos». [Em linha]. Disponível em: https://www.booking.com/index.pt-pt.html?label=gen173nr-10CAEoggI46AdIM1gEaLsBiAEBmAEzuAEXyAEM2AED6AEB-AEBiAIBqAIBuALWoaPJBsACAdICJGJkMjZiYWU2LWY5NzYtNDYxZi05OTNjLTJkZjgzMDE4MzYxNtgCAeACAQ&sid=c7659e49e0ee178b032413842186e68c&keep_landing=1&sb_price_type=total&chal_t=1764282583127&force_referer=

[11] «Airbnb | Alojamentos para férias, cabanas, casas de praia e muito mais». [Em linha]. Disponível em: https://www.airbnb.pt/?_set_bev_on_new_domain=1761305283_EAMzM4NzZiYTU3OT&set_everest_cookie_on_new_domain=1761305283.EAMWZkMjNjZTZiOWE1Ym.L8Y3UKeUjj7Gn12HGPgPdjL97XrMbAcHnRJbMmqoXx0

[12] «trivago.pt - Comparar preços de hotéis em todo o mundo». [Em linha]. Disponível em: https://www.trivago.pt/pt?themeId=280&sem_keyword=trivago&sem_creativeid=713963771034&sem_matchtype=e&sem_network=g&sem_device=c&sem_placement=&sem_target=&sem_adposition=&sem_param1=&sem_param2=&sem_campaignid=14643174652&sem_adgroupid=126812097083&sem_targetid=kwd-5593367084&sem_location=20873&cipc=br&cip=3511900005&gad_source=1&gad_campaignid=14643174652&gbraid=0AAAAAD6-gN_V2RkJpkat8IMuh8nf3bvOP

[13] «Expedia Travel: Vacation Homes, Hotels, Car Rentals, Flights & More». [Em linha]. Disponível em: https://www.expedia.com/

Anexo 1 – Resultados do Inquérito de Pertinência

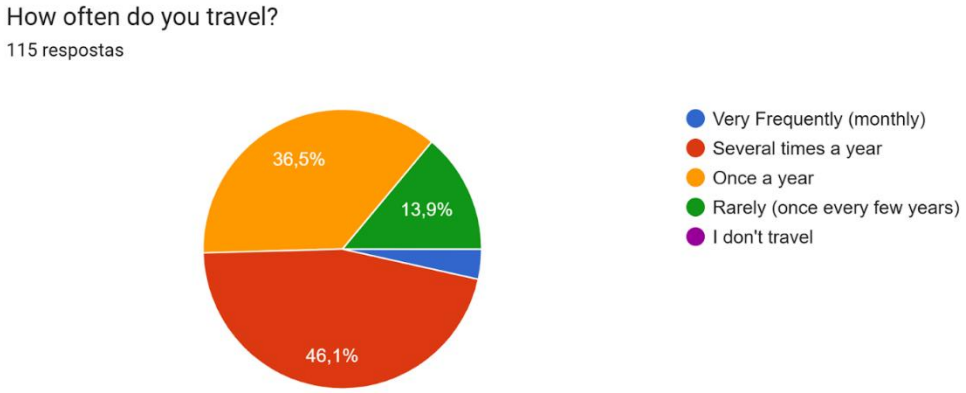

Figura 2 - Frequência com que os inquiridos realizam viagens

- Análise: Confirma-se a existência de um mercado ativo, onde a maioria dos inquiridos viaja com regularidade: 46,1% viajam várias vezes por ano e 36,5% pelo menos uma vez por ano.

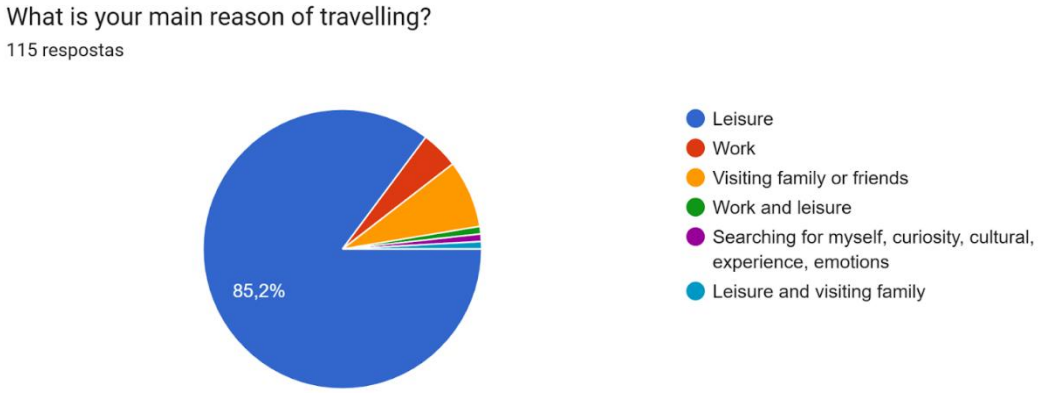

Figura 3 - Principais motivos que levam os utilizadores a viajar

- Análise: O turismo de lazer é predominante, representando 85,2% das respostas, o que valida o foco da aplicação em experiências pessoais e enriquecedoras.

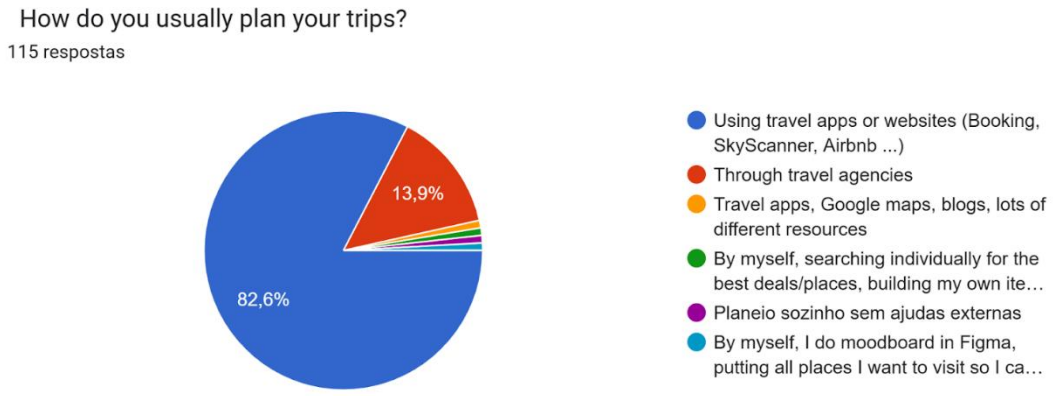

Figura 4 - Ferramentas utilizadas atualmente para organizar viagens

- Análise: Existe uma elevada adoção tecnológica, com 82,6% dos utilizadores a recorrerem já a websites e aplicações (*Booking*, *Skyscanner*, ...), contra apenas 13,9% que usam agências tradicionais.

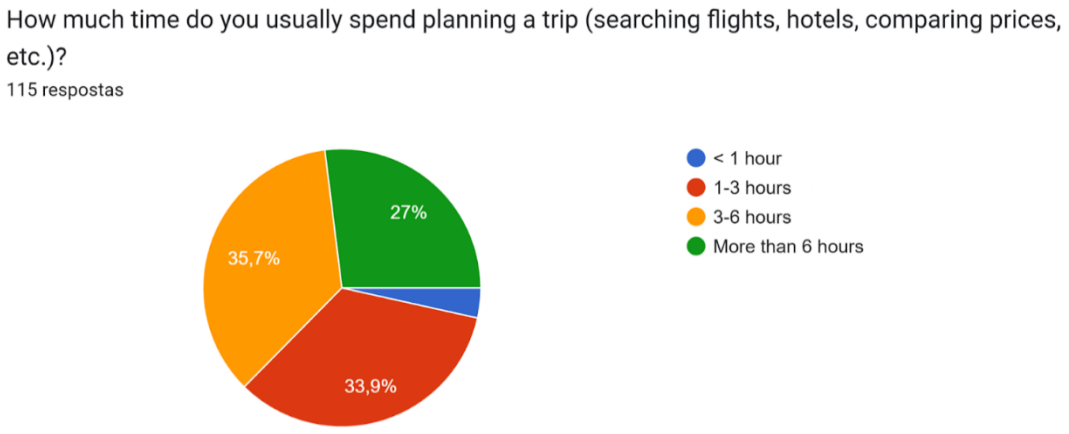

Figura 5 - Quantidade de tempo investido no processo de planeamento

- Análise: O processo é considerado moroso: 35,7% gastam entre 3 a 6 horas e 27% gastam mais de 6 horas, evidenciando a necessidade de uma solução que agilize a pesquisa.

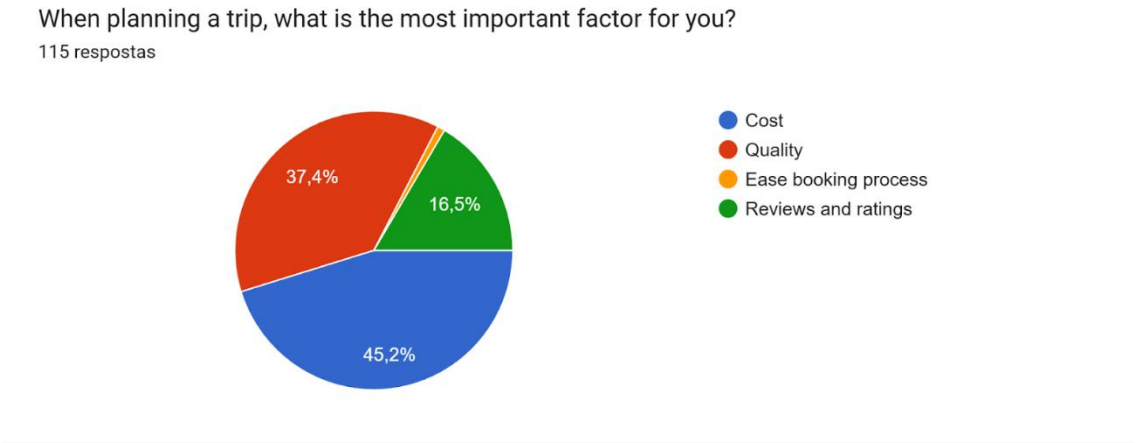

Figura 6 - Fatores mais valorizados na escolha de uma viagem

- Análise: O custo é o fator decisivo para 45,2% dos inquiridos, seguido pelo equilíbrio entre preço e qualidade (37,4%), justificando o foco da *BudgetBreak* na otimização orçamental.

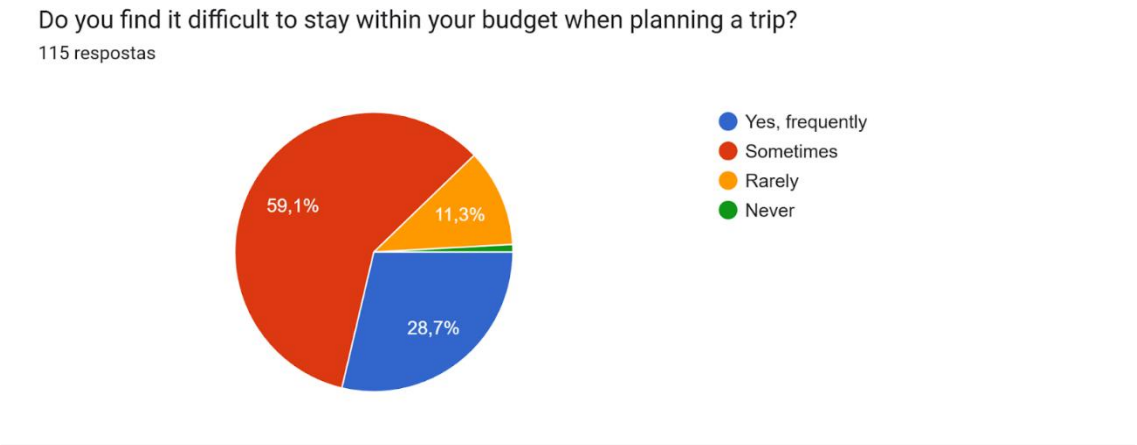

Figura 7 - Frequência com que os utilizadores têm dificuldade em manter-se dentro do orçamento

- Análise: Este é o principal problema identificado: 87,8% dos inquiridos admitem ter dificuldades (frequentes ou ocasionais) em respeitar o orçamento inicial durante o planeamento.

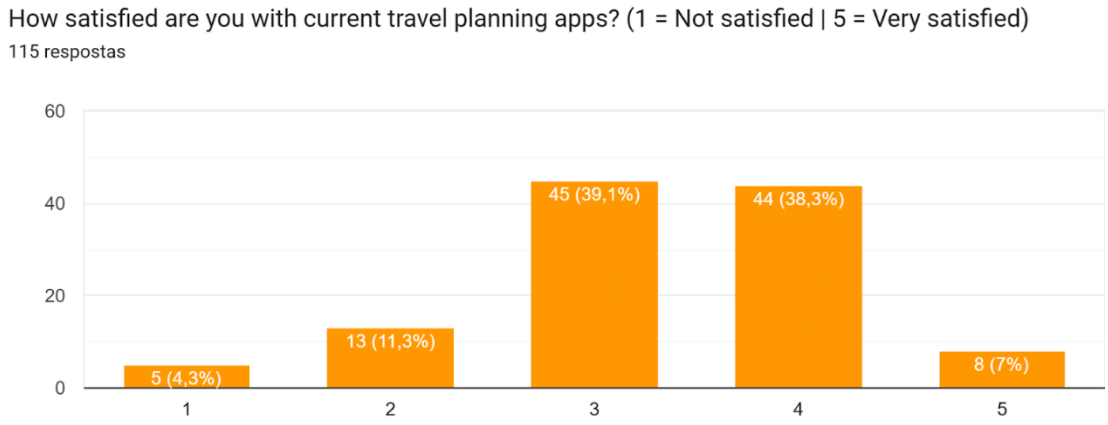

Figura 8 - Nível de satisfação com as plataformas atuais (Escala 1-5)

- Análise: A satisfação média é apenas moderada (3,32). Apenas 7% dos utilizadores estão plenamente satisfeitos (nível 5), o que demonstra uma clara oportunidade para melhoria no mercado.

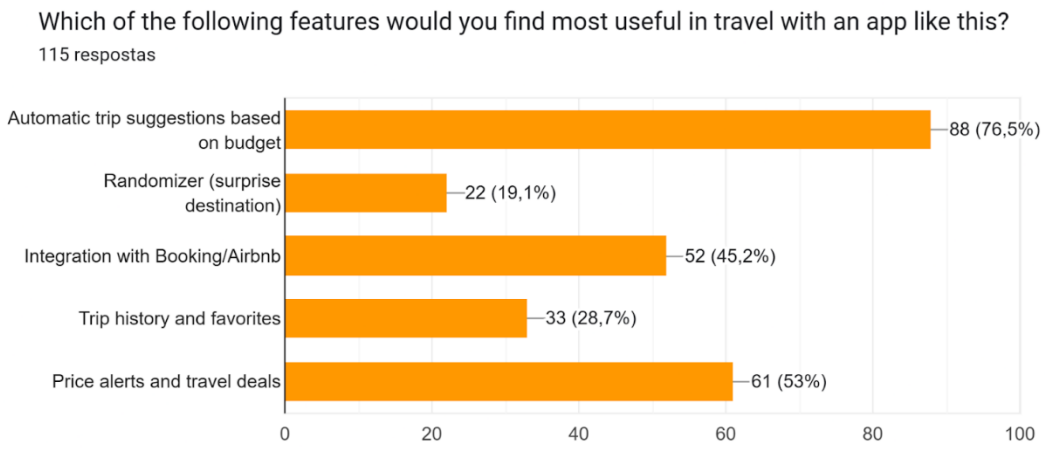

Figura 9 - Funcionalidades consideradas mais úteis pelos utilizadores

- Análise: A funcionalidade *core* do projeto "Sugestões automáticas baseadas no orçamento" foi a mais votada (76,5%), validando diretamente a proposta de valor da aplicação.

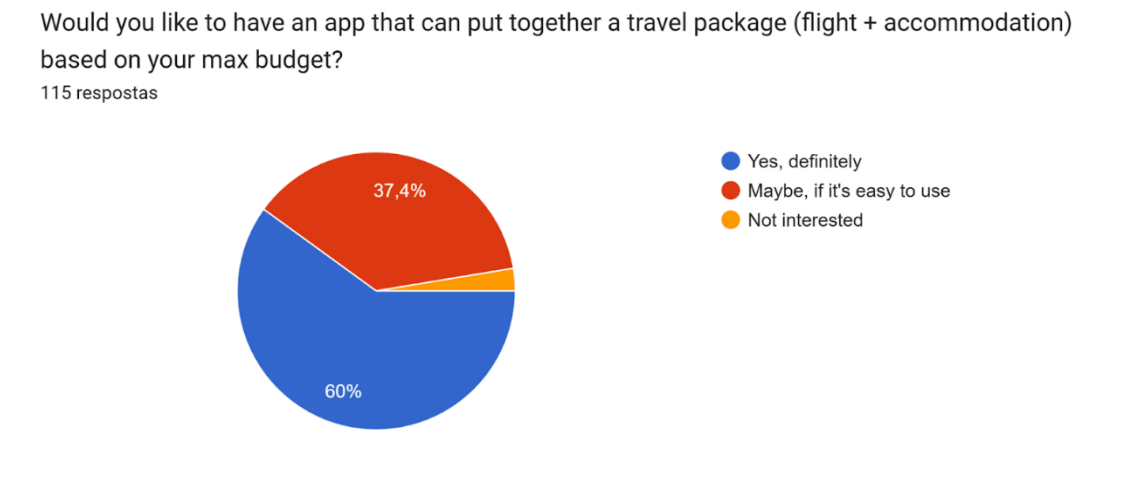

Figura 10 - Interesse na utilização de uma app que cria pacotes por orçamento

- Análise: A aceitação é extremamente positiva: 60% afirmam que "definitivamente" usariam a solução e 37,4% mostram interesse condicional, totalizando 97,4% de potenciais utilizadores.

Anexo 2 – Artefactos de Modelação

Este anexo reúne os diagramas técnicos fundamentais que suportam a arquitetura e o desenho da solução *BudgetBreak*, nomeadamente o Modelo de Dados, o Modelo de Classes e os fluxos de atividade principais.

Devido à dimensão e complexidade técnica dos artefactos apresentados infra, as versões originais em alta resolução podem ser consultadas no quadro *Trello*.

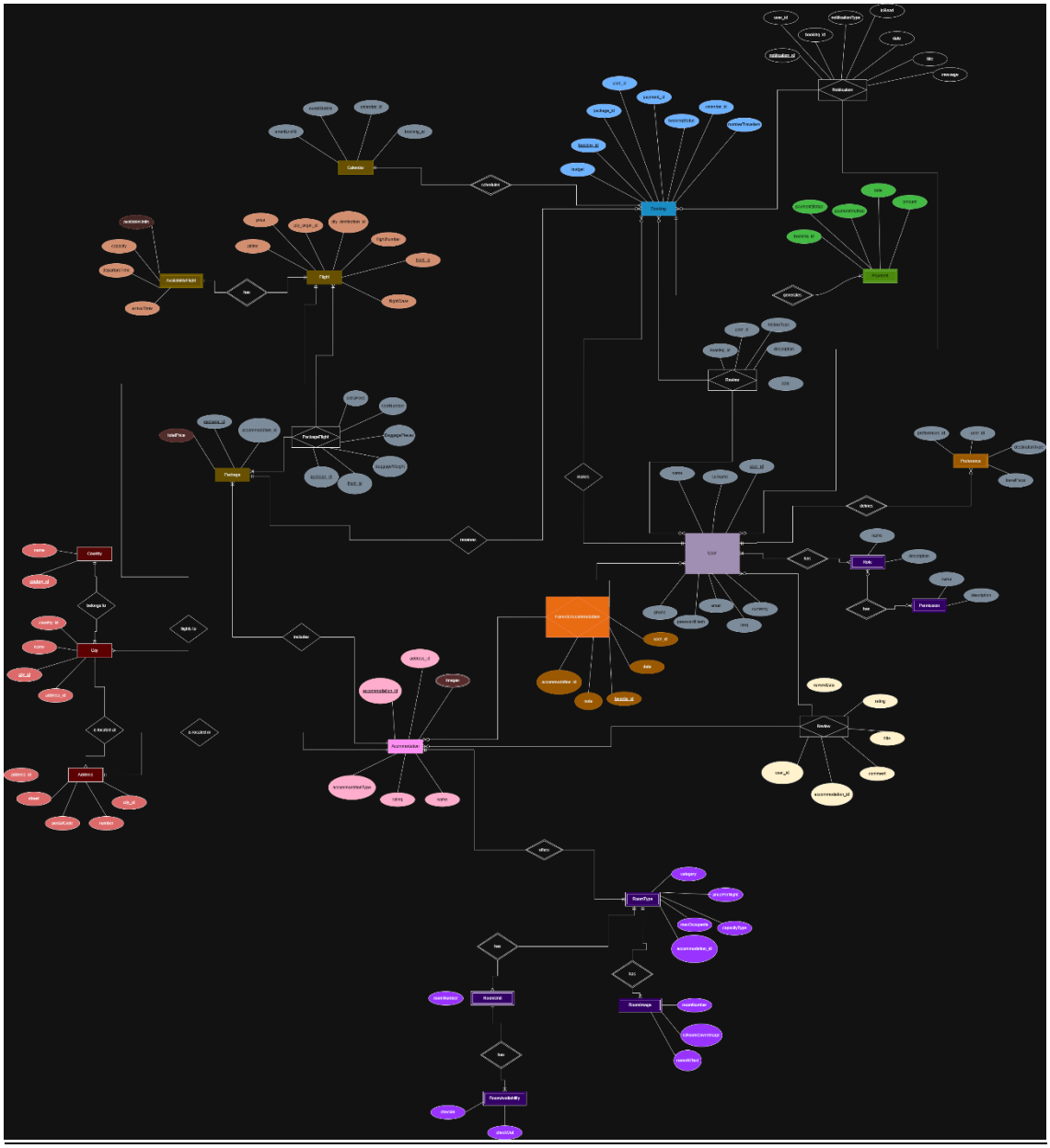

Figura 11 - Diagrama de Entidade-Relação (ERD)

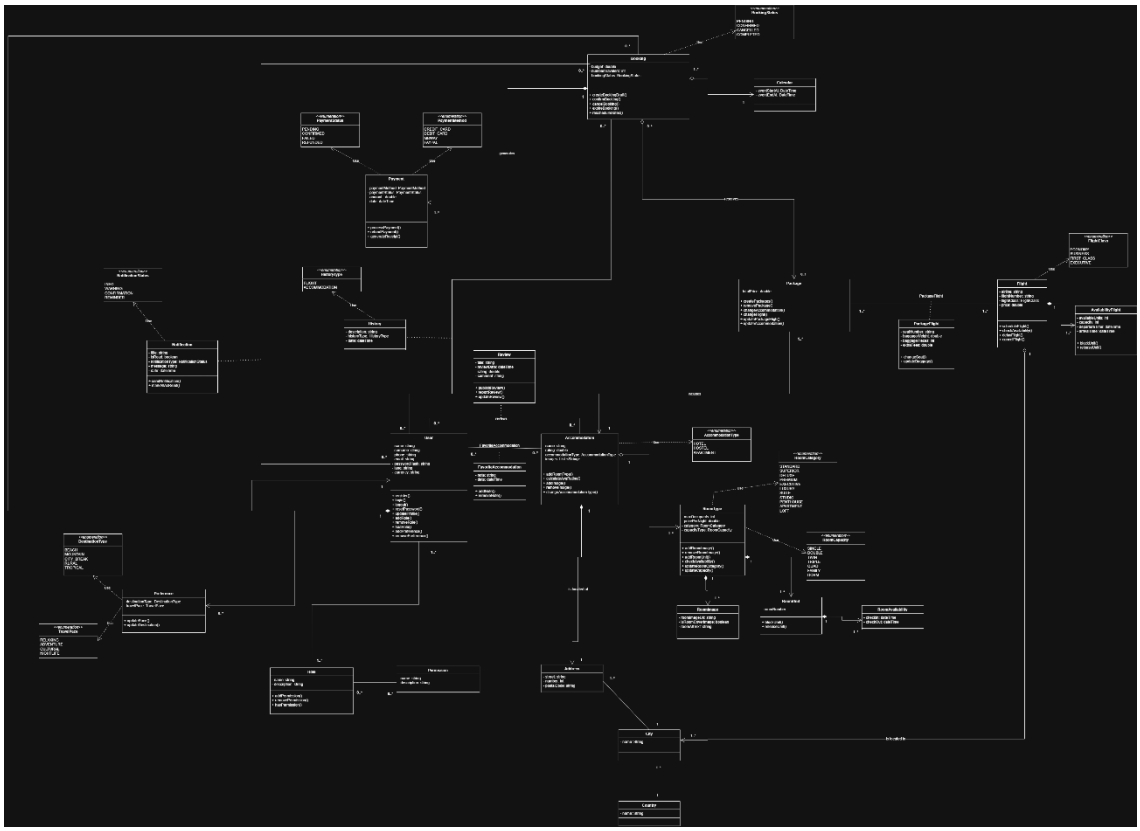

Figura 12 - Diagrama de Classes (UML)

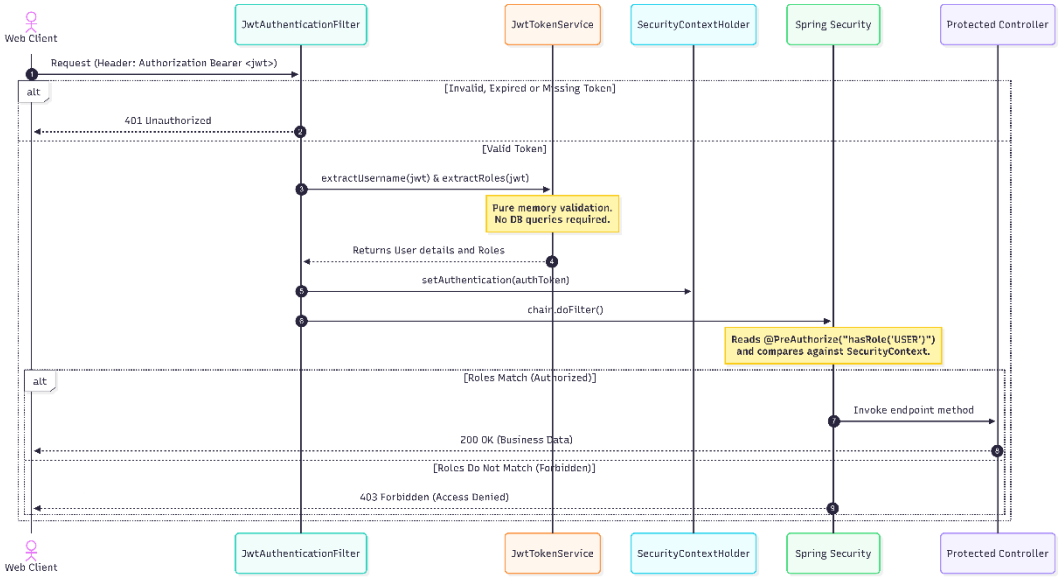

Figura 13 - Diagrama de Sequência: Stateless Authorization

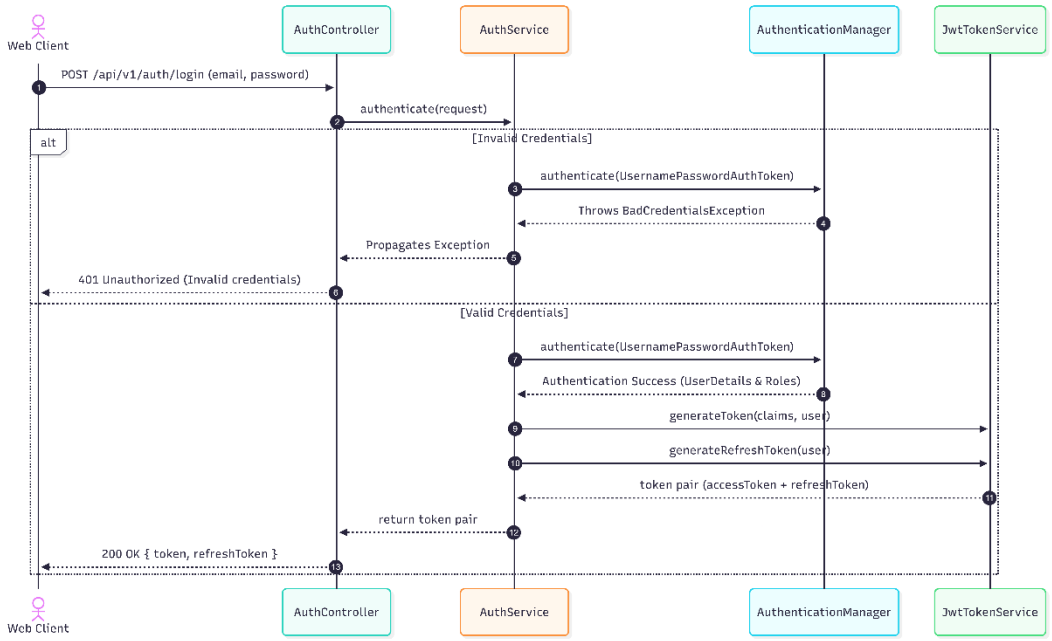

Figura 14 - Diagrama de Sequência: Login

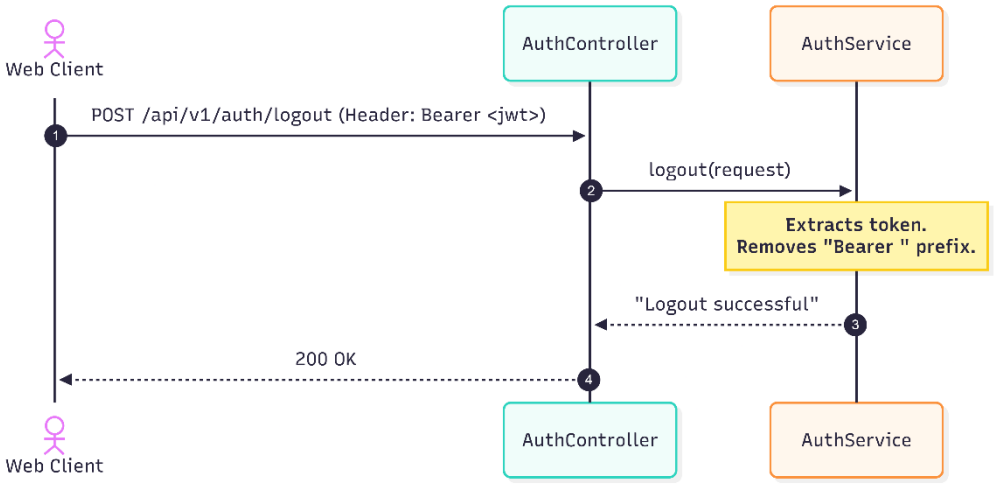

Figura 15 - Diagrama de Sequência: Logout

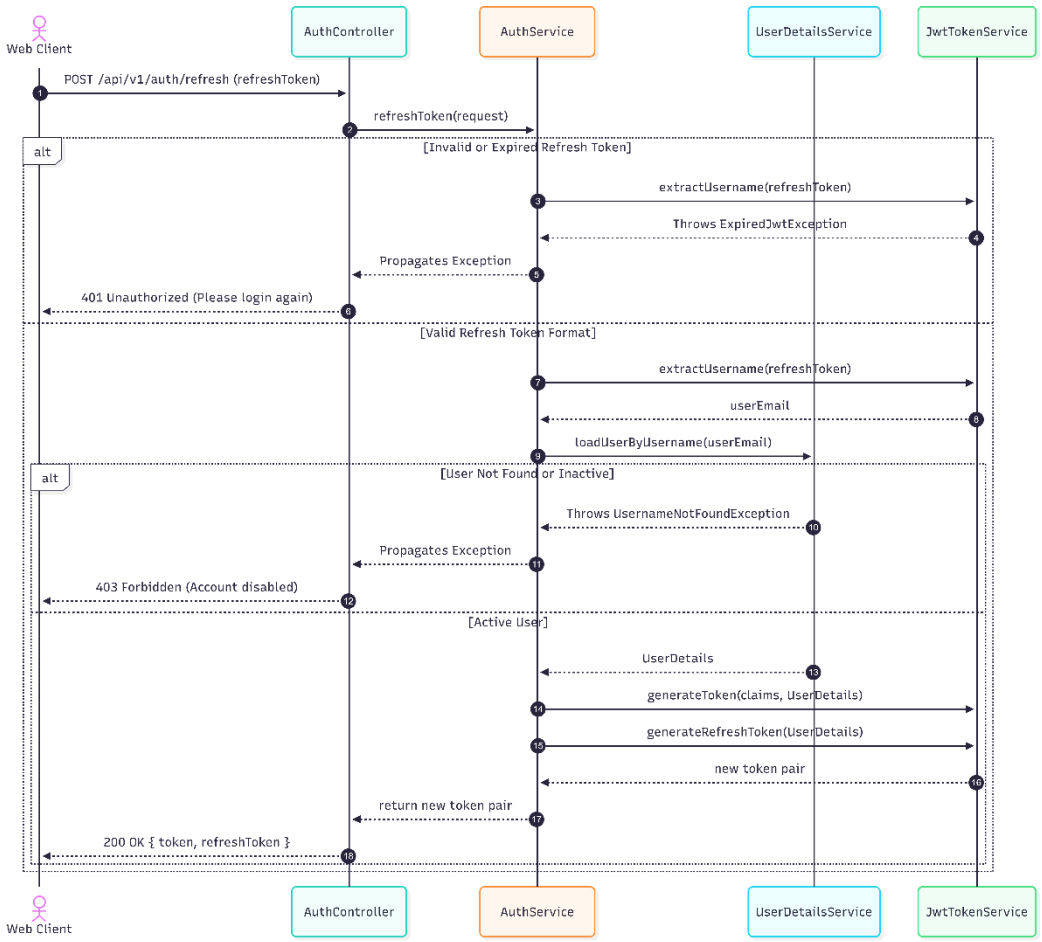

Figura 16 - Diagrama de Sequência: Refresh Token

Anexo 3 – Arquitetura de Sistema e Lógica

Devido à dimensão e complexidade técnica dos artefactos apresentados infra, as versões originais em alta resolução podem ser consultadas no quadro *Trello*.

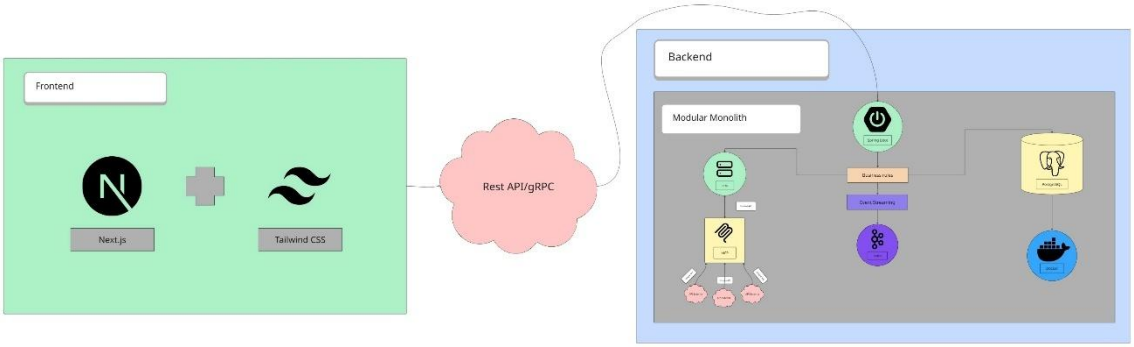

Figura 17 - Diagrama da Arquitetura Global do Sistema

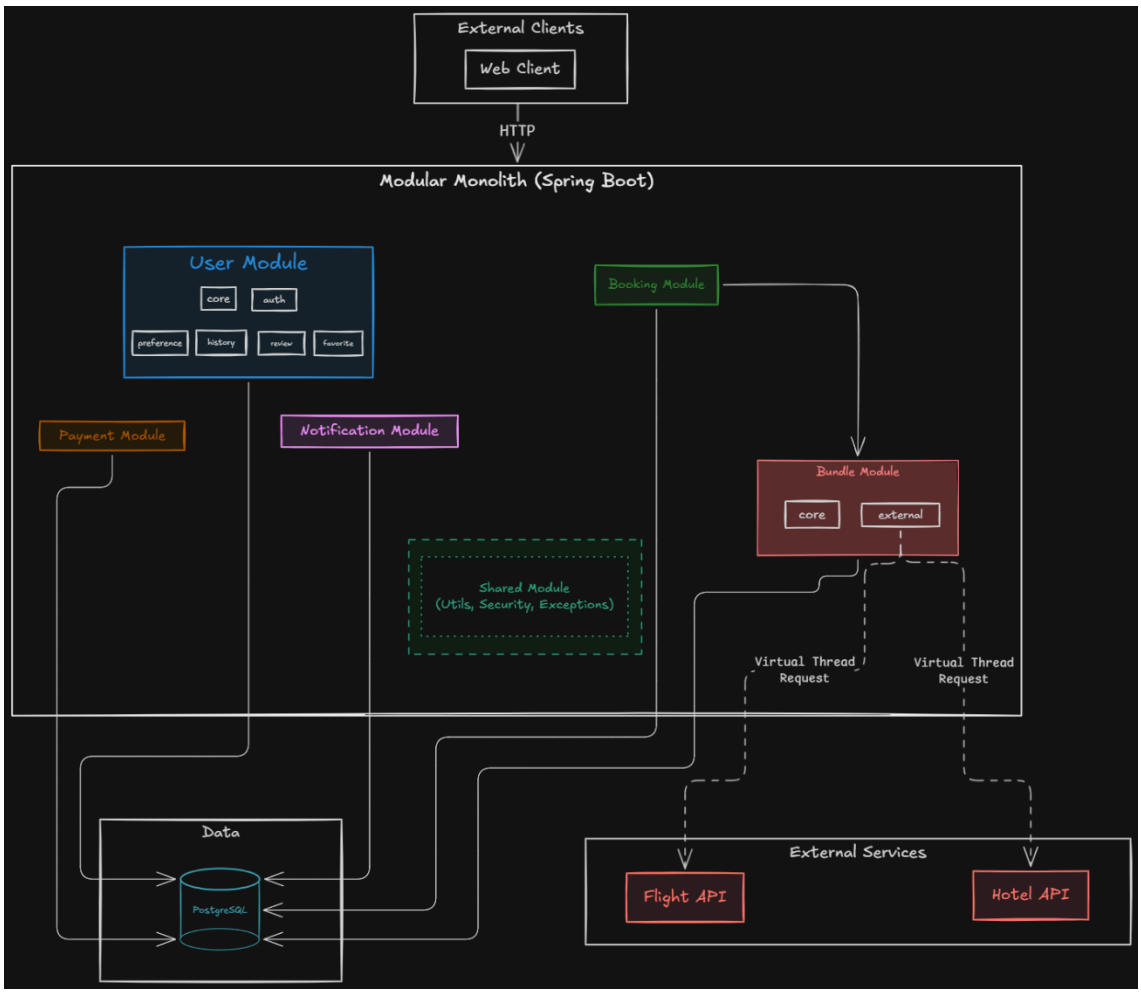

Figura 18 - Bounded context

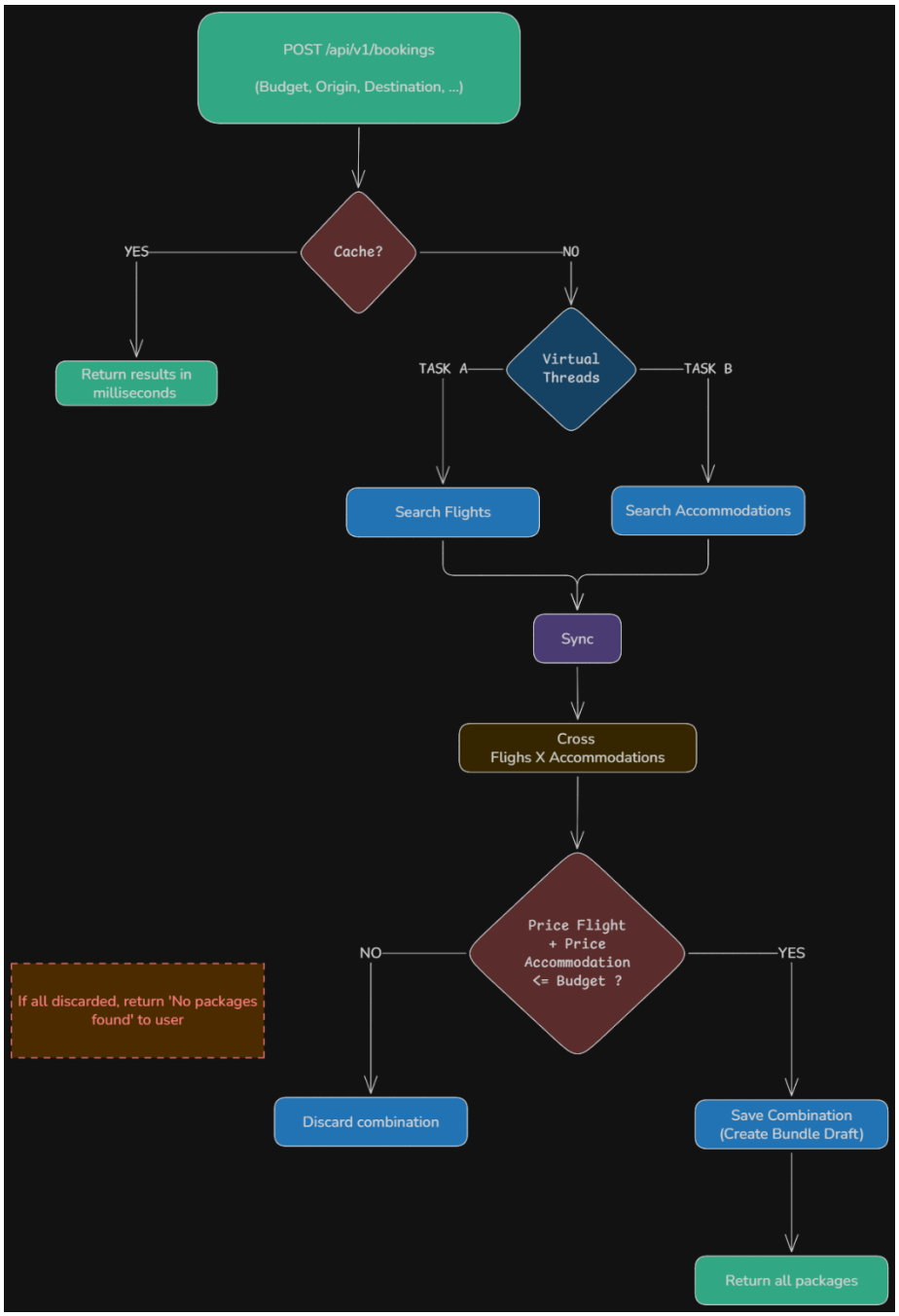

Figura 19 - DAG de agregação de pacotes

Anexo 4 – Protótipos de Interface (*Mockups*)

Devido à dimensão e complexidade técnica dos artefactos apresentados infra, as versões originais em alta resolução podem ser consultadas no quadro *Trello*.

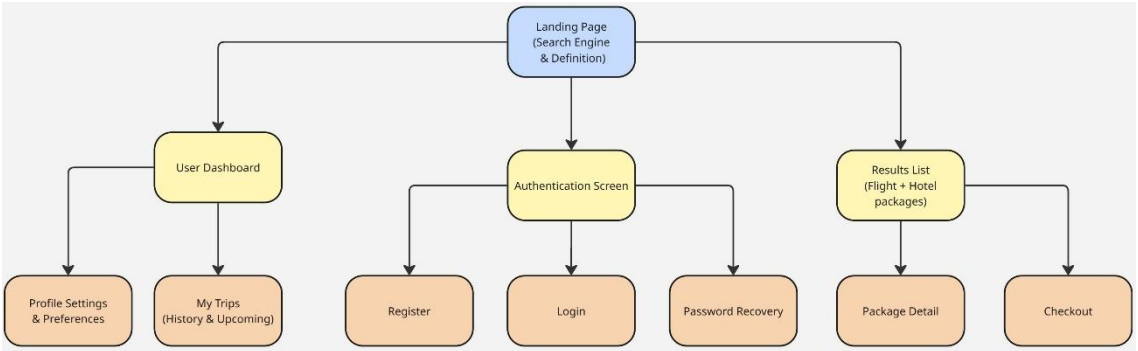

Figura 20 - Mapa Aplicacional

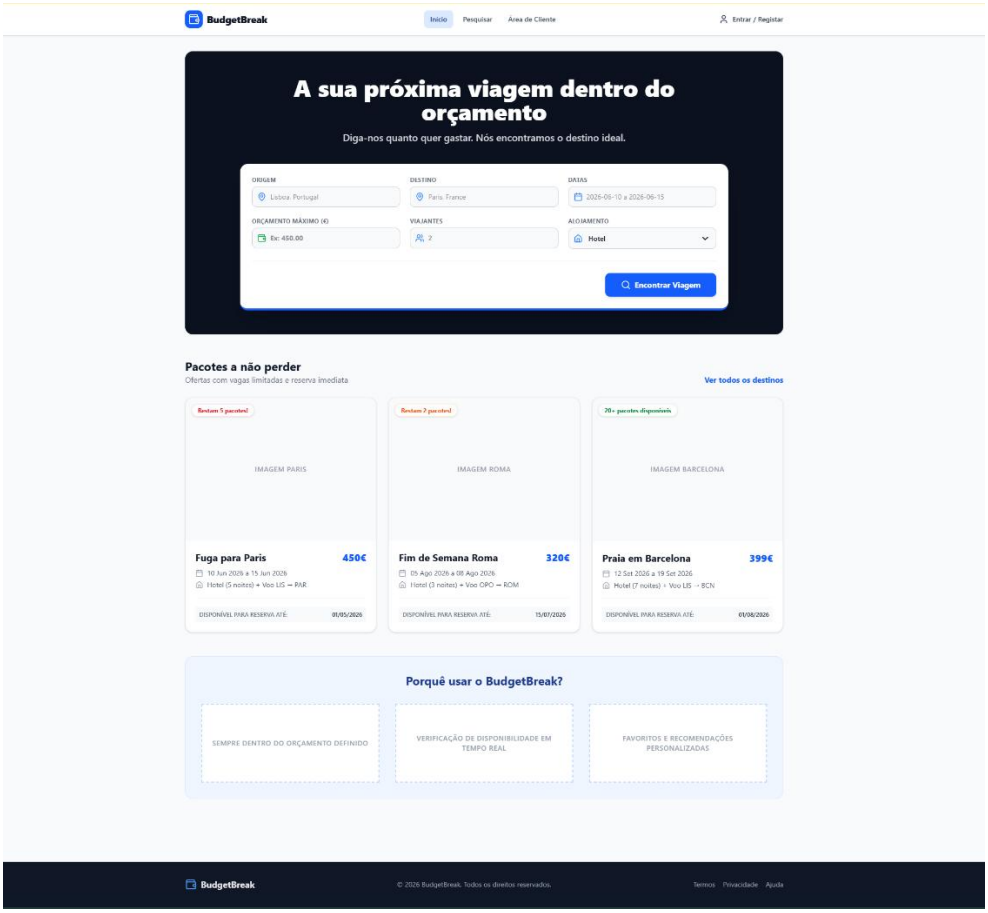

Figura 21 - Mockup: Landing Page

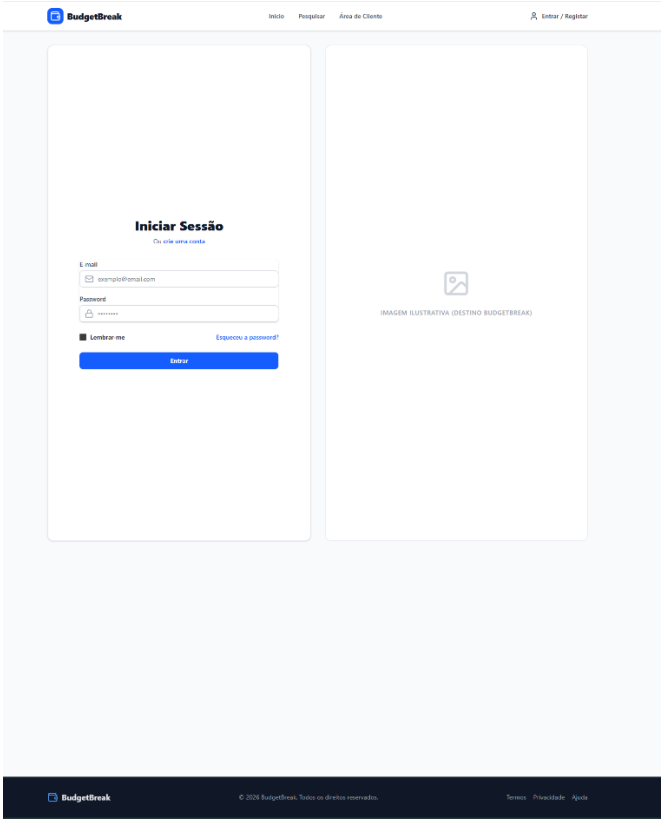

Figura 22 - Mockup: Login

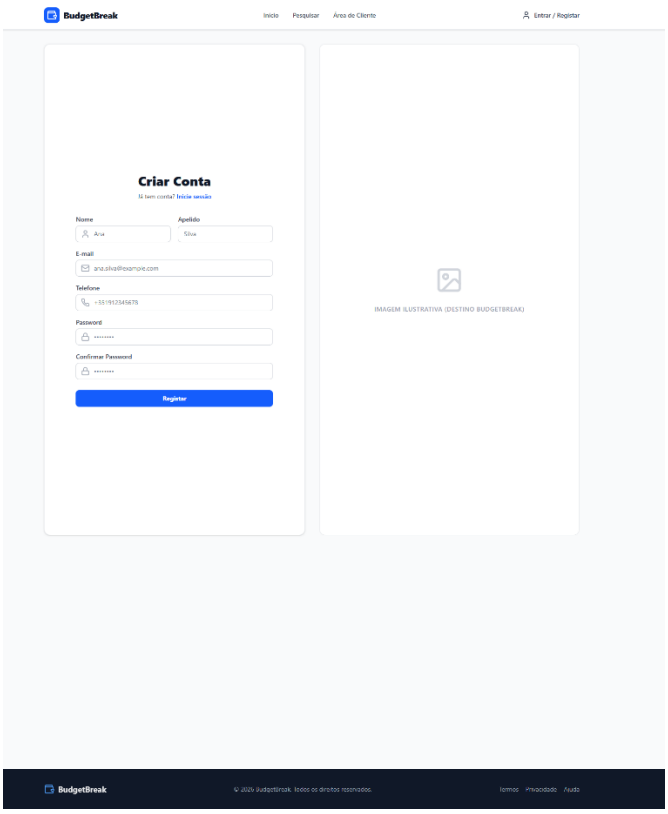

Figura 23 - Mockup: Register

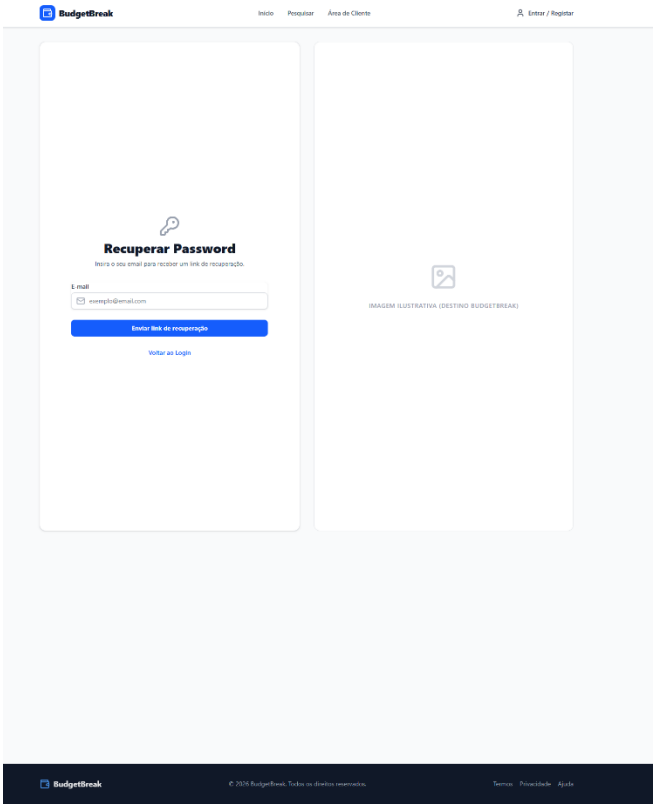

Figura 24 - Mockup: Reset Password

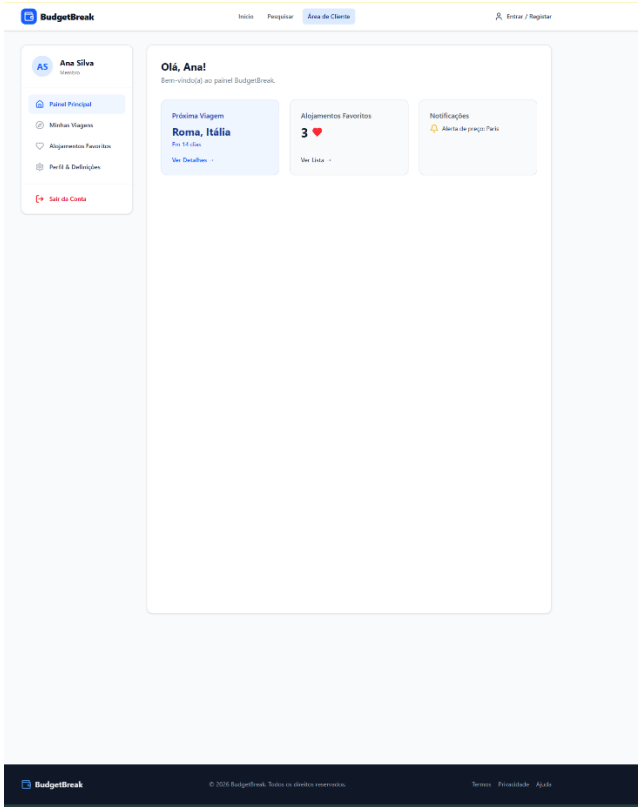

Figura 25 - Mockup: User Dashboard

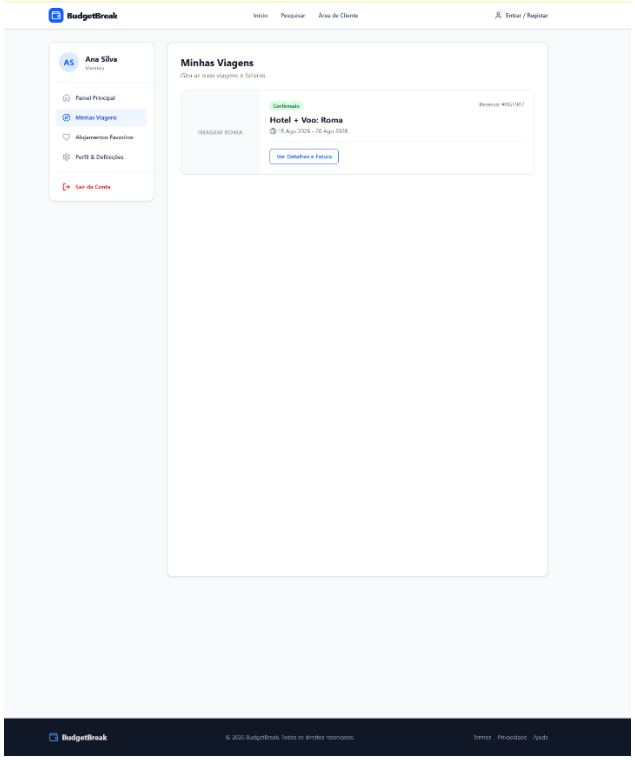

Figura 26 - Mockup: User Trips

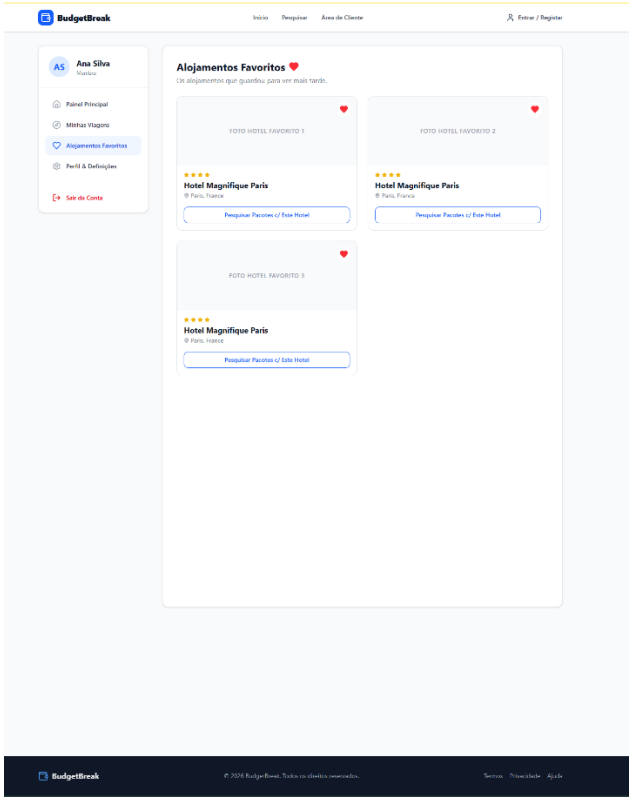

Figura 27 - Mockup: Favorite Accommodation

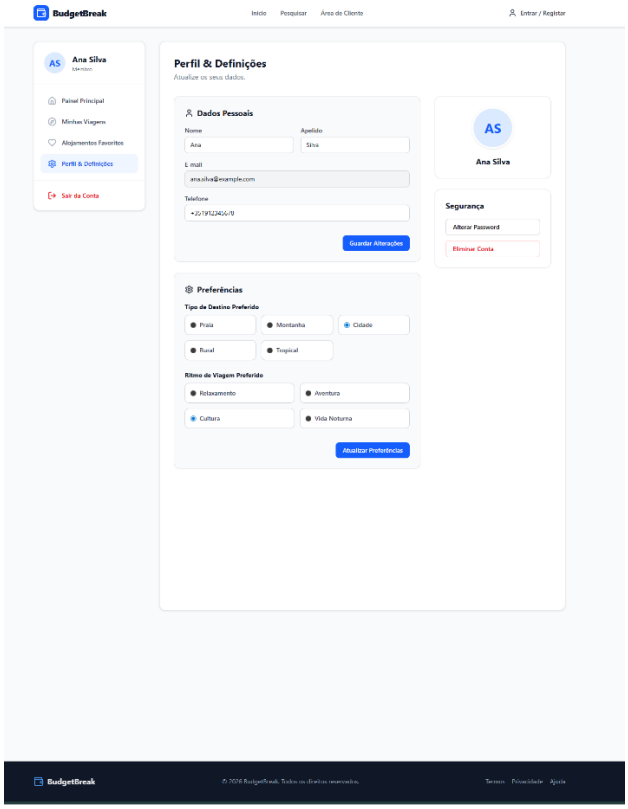

Figura 28 - Mockup: Profile settings & Preferences

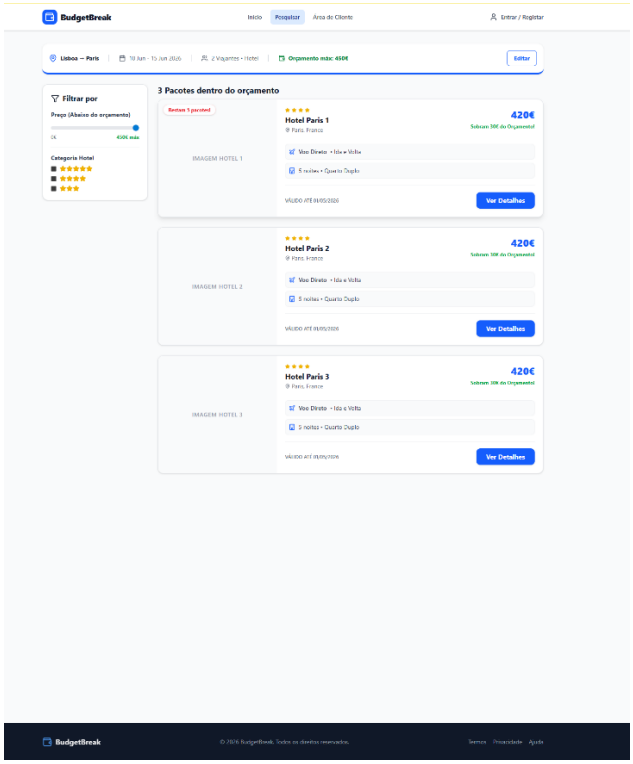

Figura 29 - Mockup: Package list

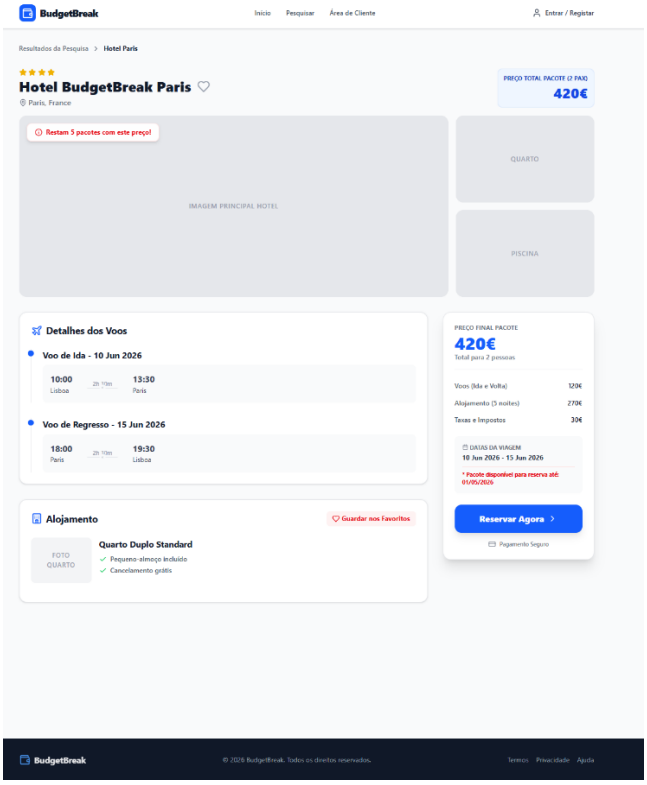

Figura 30 - Mockup: Package details

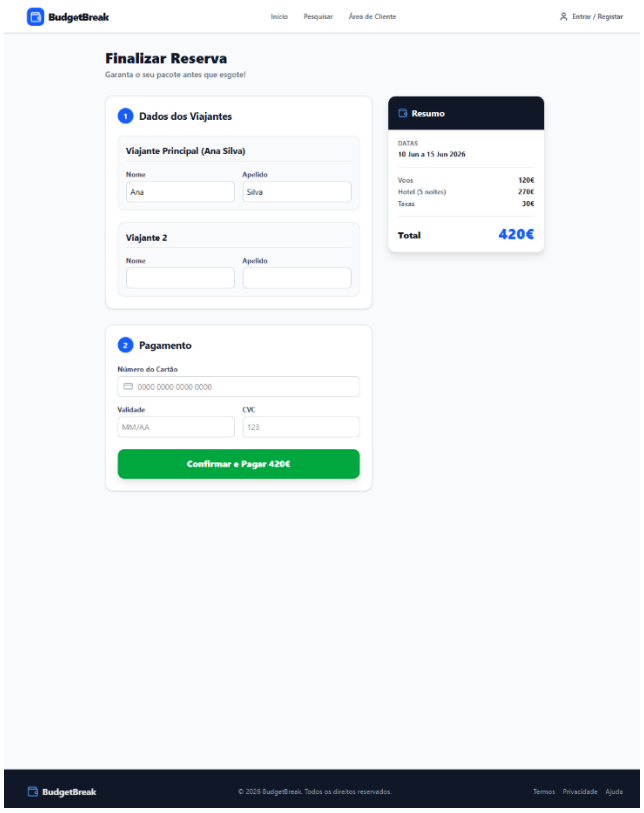

Figura 31 - Mockup: Checkout

Anexo 5 – Testes

application

application

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
shared		2%		0%	54	58	119	126	45	49	6	10
user		77%		71%	55	145	82	383	43	117	13	31
bundle		74%		77%	20	69	52	213	12	49	8	17
booking		91%		100%	6	40	14	115	6	37	2	10
notification		0%		n/a	4	4	4	4	4	4	4	4
payment		0%		n/a	4	4	4	4	4	4	4	4
Total	1,244 of 3,839	67%	43 of 120	64%	143	320	275	845	114	260	37	76

Figura 32 - Coverage: Application

com.pt.ulusofona.budgetbreak.user.core.api.controller

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
UserController		68%		n/a	3	8	6	19	3	8	0	1
RoleController		69%		n/a	2	6	2	8	2	6	0	1
PermissionController		100%		n/a	0	4	0	5	0	4	0	1
AccountController		100%		n/a	0	2	0	4	0	2	0	1
Total	46 of 193	76%	0 of 0	n/a	5	20	8	36	5	20	0	4

Figura 33 - Coverage: User Controller

com.pt.ulusofona.budgetbreak.user.core.internal.service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
UserServiceImpl		89%		95%	6	31	3	86	5	20	0	1
RoleServiceImpl		77%		62%	7	15	5	36	5	11	0	1
UserDetailServiceImpl		0%		n/a	2	2	2	2	2	2	1	1
PermissionServiceImpl		91%		100%	1	7	2	19	1	6	0	1
Total	119 of 746	84%	4 of 32	87%	16	55	12	143	13	39	1	4

Figura 34 - Coverage: User Service

com.pt.ulusofona.budgetbreak.user.preference.api.controller

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
PreferenceController		63%		n/a	2	5	4	13	2	5	0	1
Total	21 of 58	63%	0 of 0	n/a	2	5	4	13	2	5	0	1

Figura 35 - Coverage: Preference Controller

com.pt.ulusofona.budgetbreak.user.auth.api.controller

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
AuthController		87%		n/a	1	6	2	11	1	6	0	1
Total	8 of 63	87%	0 of 0	n/a	1	6	2	11	1	6	0	1

Figura 36 - Coverage: Authentication Controller

com.pt.ulusofona.budgetbreak.user.auth.internal.service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
AuthServiceImpl		84%		50%	6	12	9	50	0	5	0	1
Total	36 of 233	84%	7 of 14	50%	6	12	9	50	0	5	0	1

Figura 37 - Coverage: Authentication Service

com.pt.ulusofona.budgetbreak.booking.api.controller

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
BookingController		100%		n/a	0	6	0	13	0	6	0	1
Total	0 of 70	100%	0 of 0	n/a	0	6	0	13	0	6	0	1

Figura 38 – Coverage: Booking Controller

com.pt.ulusofona.budgetbreak.booking.internal.service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
BookingServiceImpl		100%		n/a	0	11	0	35	0	11	0	1
CalendarServiceImpl		100%		100%	0	11	0	26	0	8	0	1
Total	0 of 275	100%	0 of 6	100%	0	22	0	61	0	19	0	2

Figura 39 – Coverage: Booking Service

com.pt.ulusofona.budgetbreak.bundle.core.api.controller

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
BundleController		100%		n/a	0	4	0	11	0	4	0	1
Total	0 of 41	100%	0 of 0	n/a	0	4	0	11	0	4	0	1

Figura 40 – Coverage: Package Controller

com.pt.ulusofona.budgetbreak.bundle.core.internal.service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
BundleServiceImpl		98%		87%	4	33	3	100	0	17	0	1
Total	9 of 468	98%	4 of 32	87%	4	33	3	100	0	17	0	1

Figura 41 – Coverage: Package Service

Anexo 6 – Formulário de declaração de uso de ferramentas de Inteligência Artificial

1. Utilização de IA

[ ] Não foram utilizadas ferramentas de IA na realização deste trabalho.
[X] Foram utilizadas ferramentas de IA na realização deste trabalho.

---

2. Ferramentas utilizadas

Assinalar todas as que se aplicam.

Assistência geral à escrita, análise ou ideação

[ ] ChatGPT
[ ] Microsoft Copilot
[X] Gemini
[ ] Claude
[ ] Perplexity
[ ] Outras. Quais? _____

Assistência à programação / desenvolvimento

[X] GitHub Copilot

[ ] Claude

[ ] OpenAI Codex
[ ] Cursor
[ ] Tabnine
[ ] Amazon CodeWhisperer / Amazon Q
[ ] Outras. Quais? _____**Gemini**_____

Geração de imagem / design / multimédia

[ ] DALL·E
[ ] Midjourney
[ ] Stable Diffusion
[ ] Canva AI / Magic Design
[ ] Outras. Quais? _____**Figma**_____

Outros usos

[ ] Contexto: Ferramentas?

3. Fases do trabalho em que foi utilizada IA

[X] Planeamento do trabalho
[ ] Pesquisa exploratória / levantamento inicial de informação
[ ] Documentação técnica
[ ] Redação do relatório
[X] Desenho / modelação / arquitetura
[ ] Design / prototipagem / interface
[X] Geração de código
[X] Revisão / refatoração / debugging de código
[X] Criação de testes / casos de teste
[ ] Análise de resultados
[ ] Preparação de apresentação ou materiais auxiliares
[ ] Outros. Quais? _____

---

4. Tipo de utilização

Descrever sucintamente como a IA foi utilizada.

Exemplos: brainstorming, estruturação de secções, revisão linguística, sugestão de arquitetura, geração de exemplos, explicação de conceitos, geração parcial de código, correção de erros, criação de casos de teste, apoio ao design.

No desenvolvimento deste projeto, a IA nunca foi utilizada como ferramenta que nos suplanta, mas como uma ferramenta de apoio. O seu uso foi, mais prominente em sugestão de arquitetura (pelo elevado grau de dificuldade do projeto em questão, surgiram dúvidas em relação à melhor "*approach*" em relação à arquitetura), parcial geração de código (sempre e primeiramente mais em conceitos mencionados pelo professor orientador, desconhecidos em matéria lecionada, e por isso como motor de aprendizagem acelerada). Também utilizada para enquadrar este projeto no "*timeframe*" apertado, a correção de possíveis erros, e sobretudo na criação de casos de teste.

---

5. Partes do trabalho afetadas

Indicar as secções, componentes, módulos, ficheiros, entregáveis ou atividades que foram influenciados pelo uso de IA.

A IA funcionou como uma ferramenta de apoio e mitigação de tempo, impactando de forma geral. Ainda assim, as partes do trabalho com imposto mais direto incluem e estruturação arquitetural do sistema, as secções de código onde foram implementados os novos conceitos sugeridos pelo orientador, as áreas sujeitas a

depuração e correção de erros ao longo do desenvolvimento e, de forma bastante significativa, componentes de teste.

6. Exemplos de *prompt*

Inserir exemplos de *prompt*, diferenciando por âmbito (enquadrado na questão 2) e fase (enquadrado na questão 4)

Prompt 1 (Análise Estrutural/Inicio de desenvolvimento de logica): gostava de saber como dentro do bundle posso juntar as 2 apis vindas das mock apis (accommodation-service e flight-service), e juntar ambos utilizando virtual treads para depois poder trabalhar com os dados.

Prompt 2 (Análise Estrutural): Ao fazer um CustomUserDetails adicionei a a variavel do id para usar na manipulacao de dados no sistema, facilita o uso e é seguro. O utilizador é autenticado por um email e password, é identificado pelo um token que aponta diretamente para o email. Para vias de segurança entre modulos e identificacao do utilizador numa reserva, entre outros ... é necessario anexar o id pq se for alterado o email desse utilizador vai perder tudo oq tinha feito anteriormente. Isto tem algum erro de estrutura, desempenho, segurança ? É que tenho duvidas se usar este metodo do wrapper para identificar dentro do sistema seja uma má pratica

7. Validação, revisão e intervenção dos autores

Descrever que verificação, revisão, correção, adaptação ou reescrita foi realizada pelos autores.

Nota: se a IA tiver sido usada em código, testes, scripts, modelos, consultas, configurações ou outros artefactos técnicos, deve ser indicado de que forma os autores validaram o funcionamento e confirmaram a sua compreensão.

Em implementação de novos componentes e módulos no código, na ligação entre os mesmos e os que já existiam, por vezes ocorriam erros. A IA foi utilizada para situações destas para correção e adaptação do código que implementámos e que divergia ligeiramente do suposto. Para a criação de testes, e já mencionado anteriormente ajudou bastante a enquadrar no "*timeframe*" da entrega. No fim, com a análise em grupo e com o professor orientador, foram feitas várias validações para verificar que a implementação e as sugestões da IA eram corretas e não cegamente aceitar a sua versão. Se ainda existisse alguma dúvida ou suspeita por parte da solução proposta pela IA, a decisão era revertida e seguia-se um caminho por métodos de pesquisa sem recurso à tecnologia em questão.

8. Grau de utilização

[] Residual
[X] Moderado
[ ] Extensivo

[ ] Utilização homogénea
[X] Grau de uso diferenciado por fase ou componente de trabalho

Descrever sucintamente os diferentes usos.

Para áreas de criação de testes, análise estrutural, debugging e correção de erros, geração parcial de código para contexto específico.

9. Trabalhos em parceria

Protecção de dados confidenciais e recursos proprietários de parceiros

[X] O trabalho foi realizado em parceria com entidade externa ao DEISI

No caso da resposta anterior ser verdadeira, responder às seguintes questões:

[X] O parceiro tem regras para restringir submissão de dados

[ ] As submissões validam aplicação de regras de tratamento de dados

[ ] Foram implementados mecanismos para restringir a partilha de recursos proprietários

10. Declaração de responsabilidade

Ao assinarem a presente declaração, os autores declaram que:

- a informação acima é verdadeira e reflete o uso efetivo de ferramentas de IA na realização do trabalho;
- compreendem que a IA não substitui autoria nem responsabilidade académica;
- verificaram a validaram e veracidade das referências bibliográficas incluídas no relatório
- assumem integralmente a responsabilidade técnica, científica, ética e académica por todo o conteúdo submetido, incluindo texto, código, modelos, testes, imagens, diagramas e restantes artefactos entregues.

---

11. Identificação dos autores

Nome(s): Henrique Valente Marçalo | Bernardo António Agostinho Jerónimo

Número(s): a22302483 | a22402766

Data: 07/04/2026

Assinatura(s): *Henrique Marçalo* | *Bernardo Jerónimo*

Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso