



Personalização de Recomendações Turísticas através de Mineração de Padrões e Dados de Contexto

LUÍS CARLOS FERNANDES PINTO

Setembro de 2025

Personalização de Recomendações Turísticas através de Mineração de Padrões e Dados de Contexto

Luís Carlos Fernandes Pinto

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Engenharia de Dados**

**Orientador: Professora Patrícia Alves
Co-Orientador: Professora Goreti Marreiros**

Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim. As exceções estão explicitamente reconhecidas na secção “Considerações Éticas” do primeiro capítulo. Esta secção também declara como as ferramentas de IA foram utilizadas e para que finalidade.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, Porto, 28 de setembro de 2025

Resumo

Os sistemas de recomendação são ferramentas concebidas para apoiar os utilizadores nas suas escolhas, tornando-as progressivamente mais personalizadas. Com a evolução destes sistemas, as recomendações alcançam cada vez mais níveis superiores de precisão e coerência. Contudo, quando utilizados em recomendações de grupos, surge um problema decorrente do facto de os diferentes membros apresentarem preferências distintas, sendo, por isso, necessários métodos mais avançados que conciliem preferências e resolvam potenciais conflitos.

O trabalho realizado incidiu sobre a melhoria de um sistema de recomendação para grupos (GRS) de turismo, Grouplanner, em desenvolvimento no Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e Desenvolvimento (GECAD) com o objetivo de resolver os problemas de heterogeneidade de preferências em grupos de turismo. Esta aplicação é suportada por uma arquitetura baseada em cinco microserviços, através da qual são disponibilizadas funcionalidades como a gestão de utilizadores, a definição de preferências e a geração de recomendações.

As funcionalidades adicionadas permitem a melhoria das recomendações com base nos perfis dos utilizadores, através da extração de padrões de comportamento. Para tal, foram integrados três algoritmos distintos de mineração de padrões frequentes — *Apriori*, *FP-Growth* e *Eclat* — possibilitando a comparação entre diferentes abordagens e a escolha dinâmica da técnica mais adequada por meio de configuração. A extração destas regras foi incorporada no serviço multiagente (MAMS), para serem consideradas no motor de recomendações pelo motor de recomendação (REMS), permitindo a geração de sugestões mais personalizadas.

A metodologia proposta foi concebida considerando não apenas os dados demográficos declarados dos utilizadores, mas também fatores como medos, limitações físicas, avaliações dos POI e dados de contexto, o que resulta em recomendações mais personalizadas e adequadas ao perfil dos turistas.

Foram realizados testes para avaliar o impacto da escolha do algoritmo de mineração de padrões frequentes e do número de utilizadores na qualidade das recomendações geradas e no desempenho computacional. Os resultados obtidos demonstraram que o algoritmo *FP-Growth* apresentou o melhor desempenho neste contexto.

Estes resultados também demonstraram que a utilização de regras de associação contribuiu para melhorar a personalização das recomendações, promovendo experiências mais relevantes para os utilizadores.

Palavras-chave: Sistema de Recomendação, Ponto de Interesse Turístico, Regra de Associação, Mineração de Padrões Frequentes, *Apriori*, *Eclat*, *FP-Growth*, Dados de Contexto

Abstract

Recommendation systems are widely used to personalize user experiences. While modern systems can provide accurate recommendations for individuals, applying the same methods to groups is challenging due to differences in member preferences, requiring more advanced approaches to reconcile conflicts.

This dissertation extends a pre-existing group-based tourist POI recommendation system, GrouPlanner, which is built on a microservices architecture. The enhancements introduced leverage user profiles and behavioral patterns to improve recommendation quality. Three frequent pattern mining algorithms — *Apriori*, *FP-Growth*, and *Eclat* — were integrated to enable comparison and dynamic selection of the most suitable technique. Extracted rules were incorporated into the multi-agent service (MAMS) and applied to the recommendation engine (REMS) to generate more personalized suggestions.

The methodology considers not only demographic data but also factors such as fears, physical limitations, POI evaluations, and contextual information, resulting in recommendations better suited to diverse scenarios. Experiments were conducted to evaluate the impact of the chosen algorithm (*Apriori*, *FP-Growth* or *Eclat*) and the number of users on recommendation quality and computational performance. The results demonstrated that the *FP-Growth* algorithm achieved the best performance in this context.

The results also demonstrate that using association rules enhances personalization, delivering more relevant experiences for users.

Keywords: Recommendation System, Points of Interest (POI), Association Rule, Frequent Pattern Mining, *Apriori*, *FP-Growth*, *Eclat*, Contextual Data

Agradecimentos

Gostaria de expressar o meu profundo agradecimento a todos aqueles que me apoiaram ao longo desta jornada e que estiveram sempre disponíveis para me ajudar nesta caminhada longa e desafiante.

Em primeiro lugar, agradeço ao ISEP, instituição que me acolheu e me proporcionou as bases necessárias para a realização deste trabalho, bem como ao GECAD pela oportunidade de estágio e pela partilha de conhecimento.

Deixo também um agradecimento especial à minha orientadora, Professora Patrícia Alves, e à minha coorientadora, Professora Goreti Marreiros, pelo acompanhamento próximo, pela orientação constante e pelo incentivo fundamental ao longo de todo o projeto.

À minha família, agradeço pela paciência, carinho e encorajamento incondicional, que foram essenciais para ultrapassar os momentos mais exigentes deste percurso.

A todos os meus amigos, àqueles que fiz no ISEP e todos os outros de igual forma, deixo um sincero obrigado pela amizade, disponibilidade e apoio nos momentos em que mais precisei.

Por fim, quero reconhecer todos os meus colegas que partilharam comigo esta jornada, em especial a minha equipa de projeto, pelo companheirismo e colaboração.

A todos, um muito obrigado!

Índice

Lista de Figuras	xv
Lista de Tabelas	xvii
Lista de Código	xix
Lista de Acrónimos	xxi
1 Introdução	1
1.1 Contexto	1
1.2 Descrição do Problema	2
1.3 Objetivos e Questões de Investigação	3
1.4 Considerações Éticas	3
1.5 Estrutura do Documento	4
1.6 Planeamento do Trabalho	5
2 Conceitos e Estado da Arte	7
2.1 Conceitos	7
2.1.1 Sistemas de Recomendação	7
2.1.1.1 Sistemas Baseados em Conteúdo	8
2.1.1.2 Sistemas Colaborativos	9
2.1.1.3 Sistemas Híbridos	9
2.1.1.4 Context-Aware Recommender Systems	10
2.1.1.5 Limitações dos Sistemas de Recomendação	11
2.1.2 Sistemas de Recomendação para Grupos	12
2.1.2.1 Limitações dos Sistemas de Recomendação para Grupos	13
2.1.3 Modelo Big Five e Personalização de Recomendações	14
2.1.4 Regras de Associação	15
2.2 Revisão da Literatura	15
2.2.1 Critérios de Inclusão e Exclusão	16
2.2.2 Fontes de Informação	16
2.2.3 Termos de Pesquisa	17
2.2.4 Processo de Seleção	17
2.3 Abordagens para Mineração de Regras de Associação	17
2.3.1 Algoritmo Apriori	17
2.3.2 Algoritmo Eclat	20
2.3.3 Algoritmo FP-Growth	22
2.4 Trabalhos Relacionados	25
2.4.1 Mineração de Regras em Sistemas Reais	25
2.4.2 Recomendações em Sistemas Reais	26

3	Conceptualização e Design da Solução	29
3.1	Arquitetura do GrouPlanner	29
3.2	Domínio	32
3.3	Atores do Sistema	34
3.4	Requisitos Funcionais	35
3.5	Recolha dos Dados	38
3.5.1	Descrição dos Atributos	38
3.5.2	Caracterização da Amostra	41
3.5.3	Definição de Parâmetros para Geração de Regras	45
3.6	Desenho da Solução	46
3.6.1	Mecanismo de Geração de Regras de Associação	46
3.6.2	Integração dos Dados de Contexto	51
3.6.3	Adaptação Dinâmica a Condições Meteorológicas Adversas	52
3.6.4	Previsão de Condições Meteorológicas Adversas para Excursões	53
3.6.5	Adaptação Dinâmica à Qualidade do Ar e Nível de Ruído	54
4	Implementação da Solução	57
4.1	Mineração de Regras de Associação	57
4.1.1	Algoritmo Apriori	59
4.1.2	Algoritmo Eclat	60
4.1.3	Algoritmo FP-Growth	61
4.1.4	Extração de Regras a Partir de Conjuntos Frequentes	62
4.1.5	Exportação das Regras	63
4.2	Geração de Recomendações	64
4.2.1	Recomendações Individuais	64
4.2.2	Recomendações para Grupos	68
4.2.3	Integração de Contexto nas Recomendações	69
5	Resultados e Análise	75
5.1	Regras de Associação	75
5.2	Recomendações de Ponto de Interesse (POI)	76
5.3	Ajuste de Recomendações Baseadas no Contexto	77
5.4	Cenários de Teste	78
5.4.1	Variação de Parâmetros e Utilizadores	79
5.4.2	Variação de Algoritmo e Utilizadores	80
5.5	Metodologia de Avaliação dos Resultados	80
5.6	Avaliação das Regras de Associação	80
5.6.1	Tempo de Execução e Memória	81
5.6.2	Qualidade das Regras de Associação	82
5.6.3	Precisão e Taxa de Acerto	83
5.7	Avaliação de Recomendações	84
5.7.1	Cobertura das Recomendações	84
5.7.2	Matriz de Confusão	85
5.7.3	Efeito da Contextualização	86
5.8	Discussão e Resposta às Questões de Pesquisa	87
6	Conclusões	89
6.1	Objetivos Concretizados	89
6.2	Limitações	90

6.3	Trabalho Futuro	90
6.4	Considerações Finais	90
	Referências	93

Lista de Figuras

2.1	Taxonomia dos sistemas de recomendação	8
2.2	Fluxograma do algoritmo <i>Apriori</i>	19
2.3	Fluxograma do algoritmo <i>Eclat</i>	22
2.4	Estrutura <i>FP-Tree</i>	23
3.1	Arquitetura de microserviços do Grouplanner	30
3.2	Diagrama de vista lógica	31
3.3	Diagrama de vista física	31
3.4	Excerto do modelo de domínio do <i>Multi-Agent Microservice</i> (MAMS)	33
3.5	Diagrama de casos de uso	35
3.6	Gráfico de distribuição etária dos participantes do questionário	41
3.7	Gráfico de distribuição dos participantes segundo o nível de educação	42
3.8	Gráfico de distribuição dos participantes segundo a dimensão de personalidade de conscienciosidade	42
3.9	Gráficos da distribuição das avaliações dos participantes por categoria de turismo	44
3.10	Diagrama de sequência da geração de conjuntos frequentes pelo algoritmo <i>FP-Growth</i>	47
3.11	Diagrama de sequência de geração de regras de associação	48
3.12	Diagrama de sequência para recomendações individuais	49
3.13	Diagrama de sequência para recomendações para grupos	50
3.14	Diagrama de sequência para integração de dados de contexto	51
3.15	Fluxograma de adaptação dinâmica a condições meteorológicas adversas	52
3.16	Fluxograma de decisão para previsão de condições meteorológicas adversas para excursões	53
3.17	Fluxograma de adaptação dinâmica à qualidade do ar e nível de ruído	55
4.1	Fluxo de aplicação das regras de associação às preferências	66
5.1	Lista de recomendações individuais na interface móvel da aplicação	76
5.2	Lista de recomendações para subgrupo na interface móvel da aplicação	76
5.3	Notificação do <i>Virtual Pet</i> a sugerir ajuste da atividade do utilizador devido a previsão de chuva	78
5.4	Gráfico de comparação do tempo de execução dos algoritmos <i>Apriori</i> , <i>Eclat</i> e <i>FP-Growth</i> para diferentes números de utilizadores	81
5.5	Gráfico de comparação do consumo de memória dos algoritmos <i>Apriori</i> , <i>Eclat</i> e <i>FP-Growth</i> para diferentes números de utilizadores	82
5.6	Matriz de confusão	86

Lista de Tabelas

1.1	Cronograma	5
2.1	Métodos de filtragem híbrida em sistemas de recomendação	10
2.2	Variáveis de contexto em sistemas de recomendação	10
2.3	Estratégias de agregação de preferências em sistemas de recomendação em grupo	13
2.4	Critérios de inclusão e exclusão para revisão da literatura	16
2.5	Exemplo de lista de transações de uma mercearia	18
2.6	Conjuntos de um item candidatos	18
2.7	Conjuntos de dois itens frequentes candidatos	18
2.8	Conjuntos de três itens candidatos	19
2.9	Regras de associação a partir dos conjuntos frequentes gerados com <i>Apriori</i>	19
2.10	<i>TID-lists</i> dos itens individuais	20
2.11	<i>TID-lists</i> dos conjuntos de dois itens	20
2.12	<i>TID-list</i> do conjunto de três itens	21
2.13	Regras de associação a partir dos conjuntos frequentes gerados com <i>Eclat</i>	21
2.14	Tabela de frequências	23
2.15	Transações ordenadas por frequência dos itens	23
2.16	Padrões condicionais da <i>FP-Tree</i>	24
2.17	Padrões frequentes extraídos dos prefixos	24
2.18	Regras de Associação a partir dos conjuntos frequentes gerados com <i>FP-Growth</i>	24
3.1	Requisitos funcionais	36
4.1	Matriz de categorias de turismo e respetivas avaliações de um utilizador	67
4.2	Exemplo de subdivisão de um grupo em subgrupos pelo MAMS	68
4.3	Índice <i>QualAr</i>	72
5.1	Parâmetros considerados para cada subconjunto de regras de associação	79
5.2	Cenários de teste com diferentes números de utilizadores e parâmetros de regras	79
5.3	Cenários de teste com diferentes números de utilizadores e algoritmo utilizado	80
5.4	Tempo de execução e memória utilizada para os diferentes algoritmos e número de utilizadores	81
5.5	Valores médios de confiança e desvio padrão das regras de associação para diferentes algoritmos e número de utilizadores	83
5.6	Precisão média, utilizadores com acertos e taxa de acerto das recomendações por número de utilizadores e algoritmo	84
5.7	Resumo da precisão média e taxa de acerto dos algoritmos	84
5.8	Utilizadores com recomendações e cobertura para diferentes números de utilizadores e algoritmos	85

Lista de Código

4.1	Classe <i>TransactionFilterOptions</i>	58
4.2	<i>Apriori</i> em clusters e utilizadores globais	59
4.3	Núcleo do algoritmo <i>Apriori</i>	60
4.4	Núcleo do algoritmo <i>Eclat</i>	61
4.5	Núcleo do algoritmo <i>FP-Growth</i>	61
4.6	Geração de regras a partir dos conjuntos frequentes por <i>cluster</i>	62
4.7	Estrutura do ficheiro <i>JavaScript Object Notation</i> (JSON) para gerar regras de associação	63
4.8	Estrutura do ficheiro JSON com as regras de associação	64
4.9	Avaliação das condições meteorológicas para os POI	70
4.10	Estrutura do ficheiro JSON enviado ao <i>Frontend</i> para gerar notificação de condições adversas	71
4.11	Avaliação da qualidade do ar e ruído para os POI	73
5.1	Exemplo de geração de regras de associação	76
5.2	Exemplo de mensagem JSON para ajuste de recomendações	77

Lista de Acrónimos

APA	Agência Portuguesa do Ambiente.
API	<i>Application Programming Interface.</i>
ATT	Acelerar e Transformar Turismo.
BFI	<i>Big Five Inventory.</i>
CARS	<i>Context-Aware Recommender Systems.</i>
CSV	<i>Comma-Separated Values.</i>
DEI	Departamento de Engenharia Informática.
DIMEI	Dissertação.
DP	Desvio-Padrão.
GECAD	Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e Desenvolvimento.
GRS	Sistema de Recomendação para Grupos.
IA	Inteligência Artificial.
IPMA	Instituto Português do Mar e da Atmosfera.
ISEP	Instituto Superior de Engenharia do Porto.
JSON	<i>JavaScript Object Notation.</i>
MAMS	<i>Multi-Agent Microservice.</i>
POI	Ponto de Interesse.
POIMS	<i>Point of Interest Microservice.</i>
PREPD	Preparação para Dissertação.
REMS	<i>Recommendation Engine Microservice.</i>
RGPD	Regulamento Geral sobre a Proteção de Dados.
RS	Sistema de Recomendação.
SNMS	<i>Social Network Microservice.</i>
UMMS	<i>User Management Microservice.</i>
XML	<i>Extensible Markup Language.</i>

Capítulo 1

Introdução

Neste capítulo é feito o enquadramento inicial da dissertação. São apresentados o contexto e o tema de estudo, clarificado o problema que se pretende abordar, definidos os objetivos e formuladas as perguntas de investigação. Por último, descreve-se a organização do documento e ainda o planeamento do trabalho.

1.1 Contexto

No âmbito da unidade curricular de Dissertação (DIMEI) do Mestrado em Engenharia Informática, com especialização em Engenharia de Dados, do Instituto Superior de Engenharia do Porto (ISEP), este projeto teve como objetivo melhorar as recomendações de Ponto de Interesse (POI) num protótipo de Sistema de Recomendação para Grupos (GRS) no contexto turístico.

O GrouPlanner [1] dá continuidade ao projeto Acelerar e Transformar Turismo (ATT) [2], cujo objetivo é tirar partido dos avanços tecnológicos para impulsionar a indústria do turismo, aumentando a capacidade de resposta e a eficiência operacional do setor.

Este GRS de turismo está atualmente em desenvolvimento no Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e Desenvolvimento (GECAD), também sediado no ISEP, cuja missão é promover a investigação científica e a inovação para a incorporação de inteligência em sistemas complexos de engenharia e computação [3].

O sistema é disponibilizado como uma aplicação móvel cliente, que funciona como *frontend* em ambiente *Android*, suportada por uma arquitetura de microserviços desenvolvida em .NET e C#, que constitui o *backend*. Atualmente, é composto por cinco microserviços, hospedados numa máquina virtual do GECAD, com as respetivas bases de dados alojadas no Departamento de Engenharia Informática (DEI).

O GrouPlanner foi concebido para enfrentar desafios comuns em GRS, como a diversidade de preferências entre utilizadores e a necessidade de satisfazer todos os membros de um grupo, especialmente no contexto turístico [4].

Para lidar com a diversidade de perfis e preferências dentro de um grupo, o GrouPlanner integra um mecanismo baseado na avaliação de dimensões de personalidade dos utilizadores. Cada utilizador interage com jogos sérios e questionários implícitos, permitindo extrair características de personalidade sem depender de históricos extensivos de avaliações. Estes dados são processados pelo *Multi-Agent Microservice* (MAMS), que cria *clusters* dinâmicos de utilizadores com perfis semelhantes em tempo real. Este agrupamento possibilita gerar

recomendações mais personalizadas, mesmo para novos utilizadores, mitigando o problema de *cold-start*, e garante coerência e satisfação entre os membros de grupos heterogêneos.

Este microserviço, MAMS, utiliza uma técnica de *d-means* para agrupar os utilizadores com base nas semelhanças de personalidade e interesses, permitindo que os grupos sejam criados de forma adaptativa e em tempo real [5]. A abordagem assegura que cada recomendação de POI seja adequada ao perfil coletivo do grupo, equilibrando preferências individuais e promovendo uma experiência turística mais satisfatória e coerente [5].

Além disso, o sistema já integra um mecanismo de regras de associação, calculadas pelo algoritmo *Apriori*, que identifica padrões entre os utilizadores e contribui para aprimorar a qualidade das recomendações [5].

O *Recommendation Engine Microservice* (REMS) é responsável por gerar a lista de POI a recomendar ao utilizador ou ao grupo, analisando os seus perfis. Existe ainda uma ligação a uma *Application Programming Interface* (API) externa de dados meteorológicos, permitindo ajustar as recomendações de acordo com as condições climáticas previstas.

Os restantes microserviços incluem os módulos de gestão de utilizadores, gestão de POI e uma rede social interna, que permite o envio de mensagens entre utilizadores e subgrupos, facilitando a interação dentro do grupo. Estes componentes complementam o GRS, garantindo uma experiência integrada e coerente para grupos turísticos.

Adicionalmente, o GrouPlanner integra elementos de gamificação, como avatares, desafios, pontos e recompensas, incluindo um *Virtual Pet* que interage com o utilizador de forma lúdica, promovendo a participação ativa. Este mecanismo permite não apenas aumentar a motivação, mas também recolher dados de forma implícita sobre as escolhas e interesses do utilizador, que são posteriormente usados para personalizar as recomendações de POI [6].

A utilização de jogos sérios constitui uma abordagem menos intrusiva do que os questionários tradicionais, permitindo avaliar implicitamente as dimensões de personalidade, melhorar a experiência do utilizador e incentivar a sua participação [6].

1.2 Descrição do Problema

Os Sistema de Recomendação (RS) tornaram-se ferramentas essenciais para ajudar os utilizadores a tomar decisões em diversos domínios [4]. Os sistemas mais modernos são capazes de gerar recomendações consistentes e satisfatórias para cada utilizador individual. Contudo, dentro de um grupo, estas ferramentas sentem dificuldade visto que as preferências e as características de cada indivíduo são variadas [4].

Os GRS foram construídos como ferramenta para combater a heterogeneidade e conflitos de preferências dentro de um grupo de utilizadores [4]. Estes sistemas têm em consideração as preferências de cada membro do grupo de modo a tornar cada experiência satisfatória e agradável.

Contudo, os GRS também enfrentam desafios associados ao custo computacional, ao tempo de processamento e à intrusão necessária para gerar recomendações e criar perfis de utilizador.

Os algoritmos adicionais, *Eclat* e *FP-Growth*, foram implementados para otimizar os custos computacionais, permitindo comparar diferentes abordagens e selecionar dinamicamente a técnica mais adequada, substituindo o uso exclusivo do *Apriori* no sistema inicial.

Além disso, foi necessária a migração do motor de recomendação, desenvolvido no REMS, e a aplicação das regras de associação geradas para melhorar a qualidade das recomendações.

Também foi necessário atualizar a ligação à aplicação externa de dados meteorológicos e criar uma nova ligação a uma aplicação externa que fornece informações sobre ruído e qualidade do ar, expandindo a consideração de dados de contexto nas recomendações.

1.3 Objetivos e Questões de Investigação

Este projeto teve como objetivo aprimorar as recomendações de POI no protótipo de GRS para turismo. Procurou-se melhorar a estratégia de identificação de padrões baseada nos perfis dos utilizadores. Para além disso, integraram-se informações adicionais, como qualidade do ar e níveis de ruído, de modo a otimizar a experiência turística e fornecer recomendações mais precisas e personalizadas.

Para isso definiram-se os seguintes objetivos:

1. Estudar e analisar a estratégia previamente implementada e avaliar o seu desempenho na geração de regras de associação;
2. Explorar outras estratégias ou algoritmos, avaliar o seu desempenho e comparar os resultados com a estratégia implementada;
3. Melhorar/Otimizar a estratégia de recomendação usando diferentes algoritmos de regras de associação;
4. Personalizar as recomendações existentes acrescentando qualidade do ar e nível de ruído;

Tendo em conta os objetivos definidos, colocaram-se então as seguintes questões de investigação:

- **Questão 1:** Como se compara o desempenho e a eficácia do algoritmo *Apriori* face a outras técnicas de extração de regras de associação, como o *FP-Growth* e o *Eclat*, no contexto deste sistema de recomendação?
- **Questão 2:** De que forma a utilização de regras de associação na geração de recomendações influencia a personalização das sugestões e a coerência das recomendações em grupos de utilizadores?
- **Questão 3:** Qual é o impacto da integração de dados contextuais, como condições meteorológicas, qualidade do ar e ruído, no funcionamento do motor de recomendação e na adaptação dinâmica das sugestões?

1.4 Considerações Éticas

O motor de recomendações baseia-se na análise de dados dos utilizadores para gerar sugestões personalizadas. Reconhecendo que estas informações podem ser sensíveis, é fundamental assegurar que não sejam comprometidas. Assim, os utilizadores devem ser sempre informados da finalidade da recolha dos dados, garantindo a conformidade ética e legal, em particular com o Regulamento Geral sobre a Proteção de Dados (RGPD) [7].

Ao registar-se no GRS, o utilizador é informado sobre a recolha de dados e deve fornecer o seu consentimento explícito.

Adicionalmente, o algoritmo de recomendação foi concebido para evitar qualquer tipo de discriminação, não permitindo que características como género, etnia ou localização geográfica resultem em sugestões indevidas ou injustas.

Além disso, a utilização de variáveis contextuais, como localização em tempo real, implica cuidados adicionais com a privacidade e proteção de dados. Estas informações são altamente sensíveis e requerem mecanismos de consentimento explícito, anonimização e segurança reforçada. A transparência quanto à finalidade da recolha e processamento destes dados é essencial para garantir a confiança e aceitação do sistema pelos utilizadores, equilibrando a personalização das recomendações com o respeito pelos direitos individuais e conformidade legal, nomeadamente com o RGPD.

Este cuidado ético reforça a confiança no sistema e garante que o desenvolvimento tecnológico seja acompanhado por responsabilidade social e legal, alinhando a inovação com a proteção dos direitos dos utilizadores.

Para o desenvolvimento desta dissertação, foram utilizadas ferramentas de Inteligência Artificial (IA), como modelos de linguagem avançados, para apoiar a redação e revisão do texto. Estas ferramentas auxiliaram, por exemplo, na verificação de erros ortográficos, na tradução de excertos de texto quando necessário e na clarificação de conceitos ou explicação de determinados assuntos.

É importante salientar que o trabalho crítico, a análise dos resultados e a produção do conteúdo científico foram realizados integralmente pelo autor, garantindo a integridade académica da dissertação.

1.5 Estrutura do Documento

Esta secção tem como objetivo apresentar a estrutura em que o documento se encontra organizado, descrevendo os diferentes capítulos e os respetivos conteúdos. O documento está dividido em seis principais capítulos.

O Capítulo 2 apresenta os conceitos fundamentais relacionados com o tema e discutidos trabalhos relevantes, realizando-se uma revisão da literatura.

O Capítulo 3 apresenta a conceptualização e o design da solução proposta. São descritos a arquitetura do sistema *GrouPlanner*, os requisitos funcionais, o sistema de recomendação e a recolha dos dados utilizados. Este capítulo termina com o desenho da integração das regras de associação e dos dados de contexto.

O Capítulo 4 foca-se na implementação da solução, detalhando os algoritmos de mineração de regras de associação e o processo de geração de recomendações com os dados de contexto.

O Capítulo 5 apresenta os resultados obtidos com a implementação, incluindo a integração do sistema com o *frontend* da aplicação. Depois são analisados criticamente os resultados, considerando diferentes cenários de teste com variação de parâmetros, algoritmos e número de utilizadores. São ainda avaliados o desempenho, a qualidade das regras e as recomendações, recorrendo a métricas específicas. O capítulo conclui com a resposta às questões de investigação.

Por fim, o Capítulo 6 sintetiza as conclusões do trabalho, destacando os objetivos atingidos, as limitações identificadas, as propostas para trabalho futuro e as considerações finais.

1.6 Planeamento do Trabalho

No presente projeto, o desenvolvimento seguiu uma abordagem ágil com a integração da metodologia *Scrum* [8]. Esta metodologia permitiu que o processo fosse mais flexível e iterativo, pois permitiu a adaptação rápida caso existissem mudanças de requisitos [8].

O calendário foi dividido em períodos de uma semana, denominados de *sprints*. Em cada semana realizou-se uma reunião com toda a equipa de trabalho, incluindo o orientador, onde se expôs o trabalho realizado e se definiu o trabalho a realizar no *sprint* seguinte. Esta reunião desempenhou, então, os papéis de *Sprint Review* e de *Sprint Planning*.

Para além desta, ocorreram reuniões adicionais ao longo do *sprint* para discutir aspetos ou outras questões relacionadas com o projeto. Estas reuniões foram realizadas no formato online ou presencial, conforme a necessidade da equipa.

Como ferramenta de gestão de tarefas, recorreu-se a um documento *Excel*, no qual cada membro da equipa possuía uma página individual com as tarefas a si atribuídas. Para além disso, utilizou-se a funcionalidade *Issues* do *GitHub* [9], permitindo gerir o progresso das tarefas através de categorias como *Backlog*, *To Do*, *In Progress*, *Pending/Blocked*, *Testing* e *Done*.

O cronograma da Tabela 1.1 é uma abordagem gráfica que permite verificar como as tarefas foram organizadas ao longo do tempo. Para facilitar a interpretação, este está organizado por meses, contemplando todas as fases do desenvolvimento.

Tabela 1.1: Cronograma do projeto.

Mês	2024			2025								
	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set
Identificação do Problema	■	■										
Objetivos		■										
Contexto/Estado da Arte		■	■	■								
Análise de Código				■	■							
Implementação					■	■	■	■				
Testes e Validação								■	■	■		
Resultados/Conclusões											■	■
Redação da Dissertação	■	■	■	■	■	■	■	■	■	■	■	■

Devido a uma primeira entrega realizada em janeiro, no âmbito da unidade curricular de Preparação para Dissertação (PREPD), deu-se grande ênfase às tarefas relacionadas com a contextualização, conceitos e estado da arte.

Após essa entrega, a prioridade passou a ser o desenvolvimento das tarefas de design, implementação, testes, validação e análise dos resultados.

Capítulo 2

Conceitos e Estado da Arte

Este capítulo tem como objetivo expor os conceitos e apresentar uma análise do estado da arte relativamente a técnicas e trabalhos relevantes para o desenvolvimento desta dissertação.

Primeiro introduz-se os conceitos relacionados com sistemas de recomendação individuais e para grupos, sistemas *context-aware*, dimensões de personalidade e regras de associação.

De seguida, descreve-se o processo de revisão da literatura, incluindo os critérios de inclusão e exclusão, as fontes de informação, as palavras-chave e processo de seleção dos estudos.

Por fim, apresentam-se os trabalhos relevantes identificados na literatura com foco em sistemas de recomendação para o ramo de turismo e na comparações diferentes estratégias de mineração de regras de associação.

2.1 Conceitos

Esta secção tem como objetivo introduzir os principais conceitos que fundamentam o trabalho desenvolvido nesta dissertação.

São apresentados os tipos de sistemas de recomendação, com destaque para os métodos baseados em conteúdo, filtragem colaborativa e abordagens híbridas, bem como as suas limitações. É também abordada a forma como alguns sistemas utilizam dados de contexto para afinar as suas recomendações.

Depois, é abordado o conceito de sistemas de recomendação para grupos, considerando os desafios associados à agregação de preferências múltiplas.

A seguir, é apresentada a teoria da personalidade, com ênfase no modelo *Big Five* descrevendo as cinco dimensões e discutindo o seu papel na adaptação de sistemas de recomendação a perfis individuais de utilizador.

Por fim, é explorado o conceito de regras de associação, dado o seu papel central na abordagem proposta.

2.1.1 Sistemas de Recomendação

Com o crescimento das plataformas digitais, nomeadamente, comércio eletrónico, plataformas de *streaming*, redes sociais e turismo, os utilizadores são cada vez mais expostos a um volume exponencial de informação [10].

Os sistemas de recomendação são ferramentas computacionais que procuram mitigar a sobrecarga de informação, fornecendo aos utilizadores sugestões personalizadas com base nos seus interesses ou comportamentos anteriores. Esta estratégia permite reduzir o tempo de pesquisa, mas também personalizar o serviço e aumentar a conexão do utilizador [10].

Logo, a recolha de *feedback* é uma das partes essenciais de um sistema de recomendação e pode ser feita de duas maneiras distintas [11]:

- **Métodos Explícitos:** Através de um mecanismo de avaliação (por exemplo, avaliar entre 1 e 5 estrelas) ou de um questionário que pergunte diretamente a satisfação do utilizador. Por norma, esta abordagem retorna resultados mais fiéis, mas é mais invasiva e requer a participação ativa do utilizador;
- **Métodos Implícitos:** Através de um mecanismo passivo que verifica a interação do utilizador com um item (por exemplo, número de cliques ou tempo de visualização de um item). Esta abordagem é mais *user-friendly*, mas requer muitos dados e tempo para obter *feedback* de qualidade.

Por norma, os sistemas de recomendação acabam por recolher o *feedback* através da utilização de ambos os métodos em paralelo.

Tendo em conta a técnica de filtragem de recomendação utilizada, é possível classificar os sistemas de recomendação em três tipos distintos: Sistemas baseados em Conteúdo, Sistemas Colaborativos ou Sistemas Híbridos. A Figura 2.1 apresenta a classificação dos sistemas de recomendação tendo em conta o tipo de filtragem que utilizam.

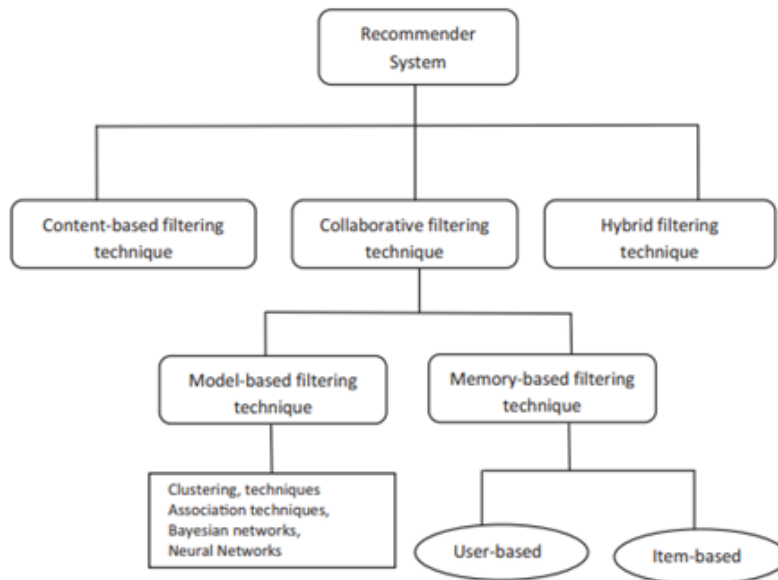


Figura 2.1: Taxonomia dos sistemas de recomendação, retirado de [10].

2.1.1.1 Sistemas Baseados em Conteúdo

Tal como o nome indica, estes sistemas fazem a análise dos itens e das suas características, agrupando-os em categorias. Por exemplo, para um livro as características serão o autor, género literário, editora, etc.

Caso o utilizador faça uma avaliação positiva a um item, o sistema agrega outros itens daquela categoria de modo a construir um perfil do utilizador. Posteriormente, o sistema recomenda itens que estejam mais relacionados com os itens avaliados positivamente [12].

Esta abordagem permite fazer recomendações sem requerer detalhes de outros utilizadores, pois cada perfil é específico a cada indivíduo, o que garante maior segurança e privacidade dos dados do utilizador. Para além disso, esta estratégia é capaz de se adaptar dinamicamente às mudanças de preferência do utilizador, o que é importante tendo em conta que estas podem alterar-se com o tempo.

Estes sistemas conseguem ainda recomendar itens mesmo que estes não possuam avaliações, bastando ter características que o algoritmo seja capaz de comparar. Por outro lado, esta abordagem é limitada ao conhecimento profundo das características dos itens e nem sempre essa informação pode estar disponível [12].

2.1.1.2 Sistemas Colaborativos

As abordagens colaborativas funcionam através da análise da similaridade entre utilizadores. Estes sistemas constroem vizinhanças, ou seja, grupos de utilizadores com padrões de preferências semelhantes. As recomendações são geradas com base nos itens mais bem avaliados dentro desses grupos [12].

Estas abordagens podem ser subdivididas em dois subtipos principais [13]:

- **Baseadas em Memória:** Utilizam diretamente a matriz de utilidade (matriz de avaliações dos utilizadores) para gerar previsões, identificando padrões entre perfis semelhantes;
- **Baseadas em Modelo:** Recorrem a algoritmos de mineração de dados ou de *machine learning* para construir um modelo preditivo. Este modelo abstrai-se da matriz de utilidade completa, permitindo realizar recomendações a utilizadores que ainda não constam nessa matriz, mitigando problemas como o *cold-start* e desafios de escalabilidade.

No entanto, devido à necessidade de partilhar preferências entre utilizadores, este tipo de sistema levanta preocupações relacionadas com a privacidade dos dados.

Acresce ainda a limitação associada à escassez de informação sobre itens recentemente introduzidos no sistema. Como estes não possuem avaliações, torna-se impossível recomendá-los, uma vez que, contrariamente aos sistemas baseados em conteúdo, as abordagens colaborativas não consideram características intrínsecas dos itens.

Em contrapartida, quando o sistema dispõe de uma quantidade significativa de *feedback*, as abordagens colaborativas tendem a superar os sistemas baseados em conteúdo, proporcionando recomendações mais diversas. A sua eficácia está fortemente dependente da capacidade do algoritmo em identificar, com precisão, as vizinhanças relevantes de utilizadores[12, 13].

2.1.1.3 Sistemas Híbridos

Tal como o nome indica, esta abordagem aplica diferentes métodos de filtragem de recomendações com o intuito de combater as lacunas resultantes da sua utilização individual [12].

A Tabela 2.1 compila um conjunto de métodos híbridos que procuram superar as limitações dos sistemas de recomendação tradicionais através da combinação de abordagens distintas:

Tabela 2.1: Métodos de filtragem híbrida em sistemas de recomendação, adaptado de [12].

Abordagem	Descrição
<i>Meta-level</i>	Um modelo pré-treinado é utilizado como entrada para outro sistema de recomendação.
<i>Feature combination</i>	As características de um sistema de recomendação são incorporadas noutra.
<i>Feature augmentation</i>	O resultado de um modelo é utilizado como entrada para outro.
<i>Mixed hybridization</i>	As saídas de diferentes sistemas de recomendação são combinadas e o resultado agregado é utilizado como recomendação.
<i>Cascade hybridization</i>	Um sistema melhora os resultados de outro.
<i>Switching hybridization</i>	O sistema seleciona dinamicamente o modelo de recomendação a utilizar consoante o contexto ou necessidade atual.
<i>Weighted hybridization</i>	As classificações de diferentes técnicas são agregadas para calcular uma única recomendação.

2.1.1.4 Context-Aware Recommender Systems

Tal como os sistemas referidos anteriormente, os *Context-Aware Recommender Systems* (CARS) avaliam o histórico de preferências e comportamentos de um utilizador, mas podem ainda considerar outras variáveis contextuais que influenciam as escolhas dos utilizadores num determinado momento [14].

As variáveis contextuais estão descritas na Tabela 2.2, sendo exemplos típicos a localização, hora do dia e o clima:

Tabela 2.2: Variáveis de contexto em sistemas de recomendação, adaptado de [14].

Variável	Descrição
Localização	Onde o utilizador se encontra (ex.: cidade, país, coordenadas GPS)
Hora	Hora do dia ou dia da semana (ex.: manhã, fim de semana)
Clima	Condições meteorológicas (ex.: sol, chuva, temperatura)
Companhia	Se o utilizador está sozinho, com amigos, família, etc.
Dispositivo	Tipo de dispositivo usado (ex.: <i>smartphone</i> , <i>desktop</i>)
Atividade atual	O que o utilizador está a fazer (ex.: a viajar, a trabalhar)
Estado emocional	Humor ou estado emocional reportado ou inferido

A incorporação destas variáveis permite uma personalização mais fina das recomendações, ajustando os resultados não apenas às preferências globais do utilizador, mas também às suas necessidades momentâneas. Por exemplo, num contexto climático, um utilizador pode preferir atividades ao ar livre num dia de sol, mas optar por atrações cobertas em dias de chuva [14].

Apesar das vantagens evidentes da utilização de variáveis contextuais, a sua recolha e processamento levantam questões relativas a privacidade e proteção de dados. Informações como localização em tempo real, estado emocional ou companhia podem ser altamente sensíveis, exigindo mecanismos de consentimento explícito, anonimização e segurança reforçada.

Além disso, a perceção dos utilizadores sobre o uso dos seus dados pode afetar diretamente a aceitação e confiança no sistema de recomendação. Assim, o desenvolvimento de sistemas sensíveis ao contexto deve equilibrar a personalização com garantias éticas e legais de proteção da privacidade.

2.1.1.5 Limitações dos Sistemas de Recomendação

Apesar dos avanços dos sistemas de recomendação, subsistem algumas limitações recorrentes que podem comprometer a sua eficácia em determinados cenários. Compreender estes obstáculos e explorar formas de os mitigar é essencial para desenvolver sistemas mais robustos e eficazes.

Cold-Start

Introduzido anteriormente, o problema de *cold-start* consiste na dificuldade dos sistemas de recomendação em fornecer sugestões adequadas na ausência de dados prévios [12]. Isto acontece frequentemente com utilizadores recentemente inseridos ou com a introdução de itens que ainda não receberam qualquer tipo de avaliação. A natureza estatística e preditiva dos algoritmos faz com que a falta de informação seja uma barreira à personalização [12].

Para colmatar esta limitação, é comum solicitar aos utilizadores que façam algumas avaliações iniciais de itens, escolham categorias preferidas, ou forneçam dados como idade, género ou localização, que podem ajudar a personalizar as sugestões iniciais [12].

O GrouPlanner integra uma análise da personalidade dos subgrupos para prever as preferências das categorias turísticas. Assim, como os membros de um subgrupo apresentam personalidades muito semelhantes e, conseqüentemente, preferências semelhantes, é possível determinar implicitamente as preferências iniciais dos turistas, permitindo gerar recomendações iniciais sem necessidade de interação prévia com a aplicação, negociação ou fornecimento de *feedback* [5].

Sparsity

Também referido anteriormente, consiste na ausência de dados suficientes num conjunto de avaliações, especialmente quando os utilizadores ativos avaliam apenas um pequeno número de itens. Esta limitação compromete a exatidão das recomendações geradas [12].

Técnicas como filtragem demográfica, decomposição por valores singulares e abordagens colaborativas baseadas em modelos são frequentemente adotadas para superar este problema [12].

O GrouPlanner, ou mais concretamente o MAMS, utiliza um algoritmo de *clustering* dinâmico que agrega turistas com personalidades semelhantes em *clusters* em tempo real à

medida que os utilizadores se registam na aplicação. Esta estratégia permite diminuir o número de *outliers* ao atribuir membros a *clusters* existentes ou a criar novos *clusters* quando faz sentido. Este processo incremental permite que o utilizador seja imediatamente incluído num *cluster* com preferências e interesses semelhantes [5].

Escalabilidade

Com o aumento constante do número de utilizadores e itens, os sistemas de recomendação têm de lidar com grandes volumes de informação. Este crescimento pode causar problemas de desempenho, nomeadamente nos recursos computacionais necessários para manter a qualidade das recomendações[12].

Para lidar com esta situação, muitas vezes recorre-se a estratégias que simplificam os dados, como a redução de dimensionalidade ou a organização dos utilizadores em grupos menores[12].

No caso do GrouPlanner, a escalabilidade é abordada através do algoritmo de *clustering* dinâmico *d-means*, que organiza os turistas em *clusters* de forma incremental à medida que os novos utilizadores se registam. Esta abordagem evita a necessidade de recalculá-los todos os centróides ou reprocessar todos os dados existentes sempre que há um novo utilizador, reduzindo significativamente a carga computacional. Além disso, a criação e atualização em tempo real dos *clusters* permitem que o sistema suporte um número crescente de utilizadores sem comprometer a qualidade das recomendações, mantendo a eficiência e a precisão mesmo em grupos grandes ou heterogéneos [5].

Shilling Attack

Este problema acontece quando um utilizador cria múltiplas identidades falsas para enviar avaliações enganosas, com o intuito de manipular a popularidade dos produtos. Este tipo de ataque compromete a precisão do sistema de recomendação [12].

A solução passa pela deteção precoce destes perfis fraudulentos e pela eliminação das suas avaliações do sistema [12].

2.1.2 Sistemas de Recomendação para Grupos

Os sistemas de recomendação para grupos, à semelhança dos sistemas de recomendação tradicionais, são projetados para fornecer sugestões aos utilizadores. No entanto, neste caso, estes sistemas são desenhados para fornecer sugestões a conjuntos de utilizadores em vez de apenas a um único indivíduo.

Estes sistemas têm de ser capazes de gerar preferências tendo em conta um inventário que inclua as preferências de todos os membros do grupo. Os métodos utilizados para gerar recomendações individuais não são capazes de lidar com a complexidade e a dinâmica dos grandes grupos [4], [5].

Existem algumas estratégias para agregar as opiniões individuais de modo a alcançar recomendações satisfatórias para todos os membros do grupo. A Tabela 2.3 apresenta as estratégias mais comuns [15]:

2.1. Conceitos

Tabela 2.3: Estratégias de agregação de preferências em sistemas de recomendação para grupos, adaptado de [15].

Estratégia	Explicação
Votação por Pluralidade	O item com mais votos é o escolhido.
Média	Faz média das avaliações individuais.
Multiplicativa	Multiplica as avaliações individuais.
Contagem de Borda	Utiliza uma estratégia de pontos tendo em conta a lista de preferência individual. O item no final da lista recebe zero pontos, o item acima desse recebe um ponto, e assim sucessivamente.
Regra de <i>Copeland</i>	Faz o cálculo da diferença entre a frequência das vezes que o item ganha com as vezes que o item perde.
Votação por Aprovação	Conta o número de pessoas que deram uma avaliação superior a um nível predefinido de aprovação.
Menor Sofrimento	Escolhe o item tendo em conta a menor avaliação individual.
Maior Prazer	Escolhe o item tendo em conta a maior avaliação individual.
Média Sem Sofrimento	Faz a média das avaliações individuais após eliminar os itens que estejam abaixo de um nível predefinido de aprovação.

Como se pode observar, existem múltiplas formas de combinar as preferências individuais para chegar a uma decisão coletiva. A escolha da estratégia mais apropriada depende do tipo de aplicação e dos objetivos específicos do sistema de recomendação. No caso do GrouPlanner, a estratégia adotada é a média das avaliações individuais [5].

2.1.2.1 Limitações dos Sistemas de Recomendação para Grupos

Os sistemas de recomendação para grupos também sofrem das limitações dos sistemas individuais. Contudo, existem outros problemas que são mais evidentes quando estes são aplicados a grupos. É importante perceber estes problemas e explorar formas de os mitigar.

Fairness

A *fairness* refere-se à capacidade do sistema em tratar todos os membros do grupo de forma equitativa, garantindo que as preferências individuais não sejam ignoradas ou sobrepostas sistematicamente [16]. Este problema surge frequentemente quando se agregam preferências de forma simplista, como por *Average* ou *Plurality Voting* (referidos na Tabela 2.3), o que pode levar a situações em que os mesmos utilizadores são constantemente beneficiados, enquanto outros raramente veem os seus interesses refletidos. Esta falta de equilíbrio pode afetar a aceitação do sistema e reduzir o envolvimento dos utilizadores, especialmente em grupos com grande diversidade de perfis [16].

Para mitigar este problema, têm sido desenvolvidos métodos de agregação que promovem justiça distributiva entre os membros do grupo. Estratégias como o *Least Misery* consideram o impacto das recomendações sobre cada utilizador. O uso de métricas específicas de *fairness* também pode ajudar a monitorizar e ajustar o comportamento do sistema [16].

Tendência

Ocorre quando os sistemas de recomendação tendem a sugerir itens que são amplamente populares entre os utilizadores, em detrimento de itens menos conhecidos mas potencialmente mais relevantes para um grupo específico. Esta limitação é particularmente problemática em contextos de grupo, onde a diversidade de preferências pode ser sacrificada em favor de itens genéricos. A preferência por itens populares pode levar a uma experiência redundante e à estagnação de descobertas, reduzindo a personalização e a utilidade do sistema [17].

É comum aplicar mecanismos de penalização de popularidade ou reponderação dos *scores* de recomendação. Técnicas como *re-ranking* com foco em diversidade e descoberta ajudam a promover itens menos populares, mas que se alinham melhor com os interesses combinados do grupo [17].

Heterogeneidade de Personalidades

A heterogeneidade torna difícil encontrar recomendações que agradem simultaneamente a todos os membros. Perfis muito distintos podem gerar conflitos nas preferências, levando a uma redução na satisfação global. Além disso, indivíduos mais passivos podem adaptar-se a decisões dominadas por outros, o que não reflete uma experiência justa ou representativa [4].

Técnicas de *clustering* de utilizadores dentro do grupo ou a criação de subgrupos com maior afinidade têm sido exploradas para melhorar a coerência das recomendações. O GrouPlanner é exatamente um exemplo da aplicação desta estratégia, criando subgrupos de turistas com personalidades e interesses semelhantes e utilizando o algoritmo de *clustering* dinâmico *d-means*. Desta forma, cada subgrupo recebe recomendações mais coerentes e personalizadas, reduzindo conflitos e aumentando a satisfação global dos membros do grupo [5].

2.1.3 Modelo Big Five e Personalização de Recomendações

A personalidade é um conjunto relativamente estável de características psicológicas que influenciam o comportamento, preferências e decisões dos indivíduos [18]. A compreensão destas dimensões pode ser explorada para aprimorar sistemas de recomendação, permitindo que sugestões de produtos, serviços ou atividades sejam mais alinhadas com o perfil do utilizador [4].

O modelo *Big Five Inventory* (BFI)[18] divide a personalidade em cinco dimensões:

- **Abertura à experiência (Openness):** Refere-se à apreciação por novas experiências, ideias e curiosidade intelectual. Indivíduos com mais abertura tendem a procurar atividades culturais e intelectualmente estimulantes;
- **Conscienciosidade (Conscientiousness):** Relaciona-se com a autodisciplina, organização e orientação para objetivos. Pessoas com alta conscienciosidade geralmente preferem atividades estruturadas e planeadas;
- **Extroversão (Extraversion):** Reflete o grau de sociabilidade, assertividade e energia. Indivíduos extrovertidos tendem a procurar ambientes sociais e atividades dinâmicas;
- **Agradabilidade (Agreeableness):** Indica a tendência para ser compassivo, cooperativo e confiável. Estas pessoas geralmente preferem interações harmoniosas e atividades colaborativas;

- **Neuroticismo (Neuroticism)**: Refere-se à propensão para experimentar emoções negativas, como ansiedade e irritabilidade. Indivíduos com alto neuroticismo podem preferir ambientes tranquilos e previsíveis.

A incorporação destas dimensões na personalização de sistemas de recomendação permite adaptar sugestões de atividades turísticas às preferências individuais dos utilizadores, não apenas com base em escolhas passadas, mas também considerando tendências comportamentais e emocionais [18].

2.1.4 Regras de Associação

As regras de associação são uma técnica de mineração de dados que permite descobrir padrões e relações entre itens em grandes bases de dados [19]. Ao contrário da filtragem colaborativa, que procura preferências individuais, as regras procuram preferências globais ou partilhadas entre todos os utilizadores [19].

Uma regra de associação é composta por dois elementos denominados antecedente e consequente. O antecedente corresponde ao item encontrado no conjunto de dados. O consequente é o item que normalmente ocorre em combinação com o antecedente. Para medir a força destas regras são utilizadas métricas [19]:

- **Suporte**: indica a frequência em que um conjunto de itens ocorre nas transações;

$$\text{Suporte}(X \rightarrow Y) = \frac{\text{Freq}(X \cup Y)}{N}$$

- **Confiança**: mede a probabilidade de o consequente acontecer quando o antecedente já ocorreu;

$$\text{Confiança}(X \rightarrow Y) = \frac{\text{Freq}(X \cup Y)}{\text{Freq}(X)}$$

- **Lift**: ajusta a confiança tendo em conta a probabilidade de o consequente ocorrer independentemente.

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Confiança}(X \rightarrow Y)}{\text{Suporte}(Y)}$$

2.2 Revisão da Literatura

O objetivo da revisão da literatura é identificar e analisar estudos relevantes para os temas abordados na dissertação, de forma a estabelecer uma base sólida para o seu enquadramento teórico e técnico.

Para garantir a qualidade e abrangência desta revisão, definiram-se previamente os critérios de inclusão e exclusão, as fontes de informação a consultar, as palavras-chave a utilizar e o processo de seleção a adotar.

2.2.1 Critérios de Inclusão e Exclusão

Estes critérios ajudam a filtrar os estudos que são mais relevantes para o projeto. Os critérios de inclusão são critérios que devem ser obrigatoriamente abordados pelos estudos que são incluídos na revisão. Já os de exclusão, são exatamente o oposto, não podem coincidir com os estudos. Quando combinados, estes critérios ajudam a construir uma bibliografia sólida selecionando o melhor conteúdo de uma lista abundante de fontes.

A Tabela 2.4 apresenta os critérios definidos para a revisão da literatura.

Tabela 2.4: Critérios de inclusão e exclusão para revisão da literatura.

Critério	Descrição
Inclusão	Estudos que abordam sistemas de recomendação, individuais ou para grupos.
	Trabalhos que discutem algoritmos de mineração de regras de associação (<i>Apriori</i> , <i>FP-Growth</i> , <i>Eclat</i>).
	Pesquisas que exploram a aplicação de regras de associação em sistemas de recomendação.
	Estudos publicados em conferências, revistas científicas, dissertações e capítulos de livros.
	Trabalhos que apresentam metodologias, implementações, avaliações experimentais ou análises teóricas relevantes.
Exclusão	Estudos sem texto completo disponível.
	Trabalhos duplicados ou com conteúdo redundante.
	Publicações não relacionadas diretamente com os temas centrais.
	Artigos sem revisão por pares ou com qualidade duvidosa.
	Artigos que não sejam escritos em inglês ou português.
	Estudos anteriores a 2015, exceto caso faça sentido no contexto.

Observa-se que a análise foi limitada a uma janela temporal de dez (10) anos. No entanto, foram incluídos alguns trabalhos mais antigos, especialmente aqueles que apresentam os primeiros desenvolvimentos dos algoritmos *Apriori*, *Eclat* e *FP-Growth*, dada a sua relevância para este contexto.

2.2.2 Fontes de Informação

Para aceder aos estudos, foram utilizadas bases de dados científicas de referência, nomeadamente *IEEE Xplore* [20], *ACM Digital Library* [21], *Google Scholar* [22] e *ScienceDirect* [23]. Estes repositórios contêm um vasto conjunto de artigos, conferências e publicações científicas relevantes nas áreas de sistemas de recomendação e mineração de dados, permitindo uma pesquisa abrangente e atualizada sobre os temas abordados.

2.2.3 Termos de Pesquisa

Os termos de pesquisa utilizados para encontrar os estudos relevantes para o caso foram os seguintes:

- Sistemas de Recomendação;
- Sistemas de Recomendação para Grupos;
- Sistemas de Recomendação *Context-Aware*;
- Regras de Associação;
- Mineração de Dados;
- Algoritmo *Apriori*;
- Algoritmo *Eclat*;
- Algoritmo *FP-Growth*;
- Turismo.

2.2.4 Processo de Seleção

Com estes parâmetros definidos, o processo de seleção seguiu os seguintes passos:

- **Recolha Inicial:** Foram aplicados os termos de pesquisa nas fontes mencionadas, resultando num total de 880 estudos inicialmente recolhidos;
- **Aplicação dos critérios de inclusão e exclusão:** Os estudos foram filtrados de acordo com os critérios previamente estabelecidos, eliminando-se aqueles que não atendiam aos requisitos de relevância, atualidade e qualidade. Após esta etapa, restaram 57 resultados.
- **Seleção Refinada:** Procedeu-se à leitura do resumo (*abstract*) e da introdução de cada estudo para confirmar que abordavam os temas centrais da dissertação. Esta fase reduziu o número de resultados para 38.
- **Seleção Final:** Por fim, foi feita uma leitura integral e criteriosa dos estudos selecionados, originando a lista definitiva de trabalhos que serviram de base para o enquadramento teórico e técnico do projeto, totalizando 33 resultados.

Este procedimento assegurou a construção de uma base bibliográfica sólida, focada nos avanços mais relevantes e atuais para o tema em análise.

2.3 Abordagens para Mineração de Regras de Associação

Para extrair essas regras de forma eficiente, diversas abordagens têm sido desenvolvidas, as quais são apresentadas e analisadas detalhadamente nas próximas secções.

2.3.1 Algoritmo Apriori

O algoritmo *Apriori* foi introduzido por Agrawal e Srikan em 1994 [24] e permanece como um dos métodos mais utilizados para a extração de regras de associação. Tal como o

nome indica, “Apriori” deve-se ao facto do algoritmo se basear em conhecimento prévio dos conjuntos de itens frequentes.

Este algoritmo afirma que se um conjunto de dados é frequente então todos os seus subconjuntos também o são. Seguindo o mesmo raciocínio, caso o conjunto seja pouco frequente, todos os seus superconjuntos também o são [24].

De modo a ilustrar o funcionamento do algoritmo, é utilizado o exemplo clássico das transações de uma mercearia, enumeradas na Tabela 2.5:

Tabela 2.5: Exemplo de lista de transações de uma mercearia.

TID	Itens
T1	Pão, Leite, Manteiga
T2	Pão, Leite
T3	Pão, Manteiga
T4	Leite
T5	Pão, Leite, Manteiga

Assumindo um limite mínimo de 70% de confiança e 50% de suporte, o algoritmo executa as seguintes etapas [25]:

1. **Gerar os conjuntos frequentes de um item:** Calcula-se a frequência com que cada item individual aparece nas transações. Apenas os itens que atingem ou ultrapassam o limiar de suporte ($\geq 50\%$) são considerados para as etapas seguintes;

Tabela 2.6: Conjuntos de um item candidatos.

Item	Contagem de Suporte	Suporte (%)
Pão	4	80%
Leite	4	80%
Manteiga	3	60%

2. **Gerar os conjuntos frequentes de dois itens:** São feitas todas as combinações possíveis entre os conjuntos de um item, e calcula-se novamente o suporte.

Tabela 2.7: Conjuntos de dois itens frequentes candidatos.

Itens	Contagem de Suporte	Suporte (%)
Pão, Leite	3	60%
Pão, Manteiga	3	60%
Leite, Manteiga	2	40%

Neste caso, apenas os conjunto {Pão, Leite} e {Pão, Manteiga} atingem o limiar de suporte e, portanto, passam à etapa seguinte.

3. **Gerar os conjuntos frequentes de três itens:** Seguindo a mesma lógica, combina-se o conjunto frequente anterior para gerar combinações de três itens.

2.3. Abordagens para Mineração de Regras de Associação

Tabela 2.8: Conjuntos de três itens candidatos.

Itens	Contagem de Suporte	Suporte (%)
Pão, Leite, Manteiga	1	20%

Como o conjunto não atinge o limiar de suporte, é ignorado.

4. **Gerar as regras de associação:** Com base nos conjuntos frequentes gerados, criam-se regras de associação, e calcula-se a confiança de cada uma. Apenas as regras que atingem ou ultrapassam o limiar de confiança ($\geq 70\%$) são consideradas válidas.

Tabela 2.9: Regras de associação a partir dos conjuntos frequentes gerados com *Apriori*.

Regra	Suporte do Conjunto	Suporte do Antecedente	Confiança (%)	Validação
Pão \rightarrow Manteiga	3	4	75	Aprovada
Manteiga \rightarrow Pão	3	3	100	Aprovada
Pão \rightarrow Leite	3	4	75	Aprovada

De acordo com as regras de associação extraídas na Tabela 2.9, observa-se que um cliente que compra manteiga também compra pão, e que, caso compre pão, há uma alta probabilidade de que também adquira leite ou manteiga.

No exemplo apresentado, o algoritmo termina execução após os conjuntos de tamanho três, mas em casos reais, o número de iterações pode ser maior, dependendo da complexidade e diversidade dos dados [24].

O processo de geração dos conjuntos frequentes repete-se iterativamente e exaustivamente, aumentando o tamanho dos conjuntos de itens em cada etapa, até que não existam mais conjuntos que atendam ao limiar mínimo de suporte. O fluxograma da Figura 2.2 ilustra de forma esquemática este processo iterativo e sua lógica [24].

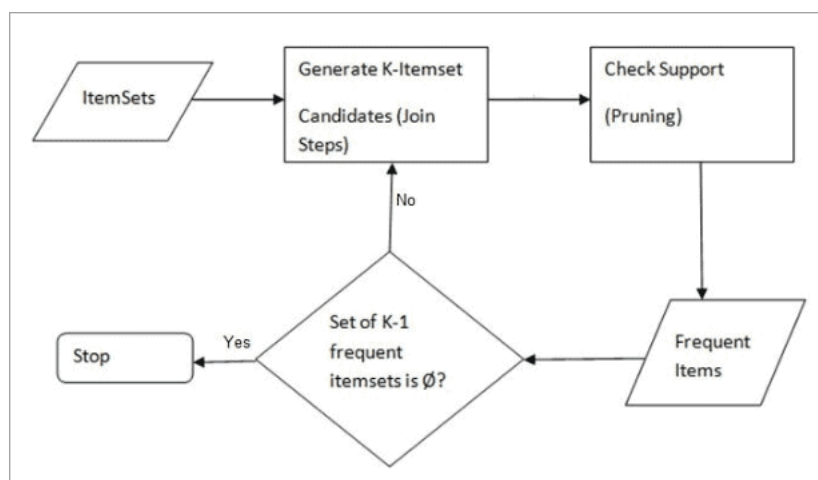


Figura 2.2: Fluxograma do algoritmo *Apriori*, retirado de [24].

O facto de o algoritmo gerar exaustivamente todos os conjuntos frequentes torna-o pouco eficiente para grandes volumes de dados. Este processo pode-se tornar computacionalmente caro e intensivo, limitando a sua escalabilidade em contextos com conjuntos de dados muito extensos.

2.3.2 Algoritmo Eclat

O algoritmo *Eclat* ou *Equivalent Class Transformation* foi introduzido por Zaki et al. em 1997 [26]. Surgiu como uma alternativa ao *Apriori* no sentido de mudar a eficácia da geração dos conjuntos frequentes.

Apesar do mesmo objetivo de encontrar padrões frequentes em bases de dados, aplica uma forma diferente de processamento dos dados. Enquanto o *Apriori* aplica uma abordagem **horizontal**, o *Eclat* adota uma abordagem **vertical** baseada em listas de identificadores de transações (*TID-Lists*).

A ideia central é transformar os dados da forma habitual (transações com itens) para uma representação onde cada item é associado à lista de transações em que ocorre. A interseção entre essas listas permite calcular rapidamente o suporte de combinações de itens.

De modo a ilustrar o funcionamento do algoritmo, é utilizado o mesmo exemplo das transações de uma mercearia, enumeradas na Tabela 2.5.

Assumindo de igual forma um limite mínimo de 70% de confiança e 50% de suporte, o algoritmo executa as seguintes etapas [27]:

1. **Construir as listas de TID para cada item:** Consiste em gerar a lista de transações para cada item;

Tabela 2.10: *TID-lists* dos itens individuais.

Item	TID-list	Suporte (%)
Pão	{T1, T2, T3, T5}	80%
Leite	{T1, T2, T4, T5}	80%
Manteiga	{T1, T3, T5}	60%

2. **Gerar os conjuntos frequentes de dois itens:** São gerados os conjuntos frequentes de tamanho dois através da interseção de *TIDs*.

Tabela 2.11: *TID-lists* dos conjuntos de dois itens.

Conjunto de Itens	TID-list	Suporte (%)
Pão, Leite	{T1, T2, T5}	60%
Pão, Manteiga	{T1, T3, T5}	60%
Leite, Manteiga	{T1, T5}	40%

Neste caso, o conjunto {Leite, Manteiga} não passa a fase seguinte pois não atinge o limiar de suporte.

3. **Gerar os conjuntos frequentes de três itens:** Seguindo a mesma lógica, são gerados os conjuntos frequentes de tamanho três através da interseção de *TIDs*.

2.3. Abordagens para Mineração de Regras de Associação

Tabela 2.12: *TID-list* do conjunto de três itens.

Conjunto de Itens	TID-list	Suporte (%)
Pão, Leite, Manteiga	{T1, T5}	40%

Como o conjunto não atinge o limiar de suporte, é ignorado.

4. **Gerar as regras de associação:** Com base nos conjuntos frequentes gerados, criam-se regras de associação, e calcula-se a confiança de cada uma. Apenas as regras que atingem ou ultrapassam o limiar de confiança ($\geq 70\%$) são consideradas válidas.

Tabela 2.13: Regras de associação a partir dos conjuntos frequentes gerados com *Eclat*.

Regra	Suporte do Conjunto	Suporte do Antecedente	Confiança (%)	Validação
Pão \rightarrow Manteiga	3	4	75	Aprovada
Manteiga \rightarrow Pão	3	3	100	Aprovada
Pão \rightarrow Leite	3	4	75	Aprovada

Através da análise da Tabela 2.13 verifica-se que as regras de associação extraídas são as mesmas do *Apriori*, o que faz sentido visto que ambos os algoritmos analisaram o mesmo conjunto de dados [26].

Tal como no *Apriori*, neste contexto o algoritmo termina após a geração dos conjuntos de tamanho três. No entanto, em contextos reais, o número de iterações pode ser significativamente superior, dependendo da complexidade e diversidade dos dados, tal como demonstrado no fluxograma da Figura 2.3.

O algoritmo *Eclat* apresenta vantagens em termos de desempenho para conjuntos de dados mais densos, uma vez que evita a geração exaustiva de candidatos e permite operações de interseção mais rápidas. Contudo, a manutenção das *TID-lists* também se pode tornar custosa em casos com elevado volume de transações, o que pode impactar a escalabilidade da solução [27, 28].

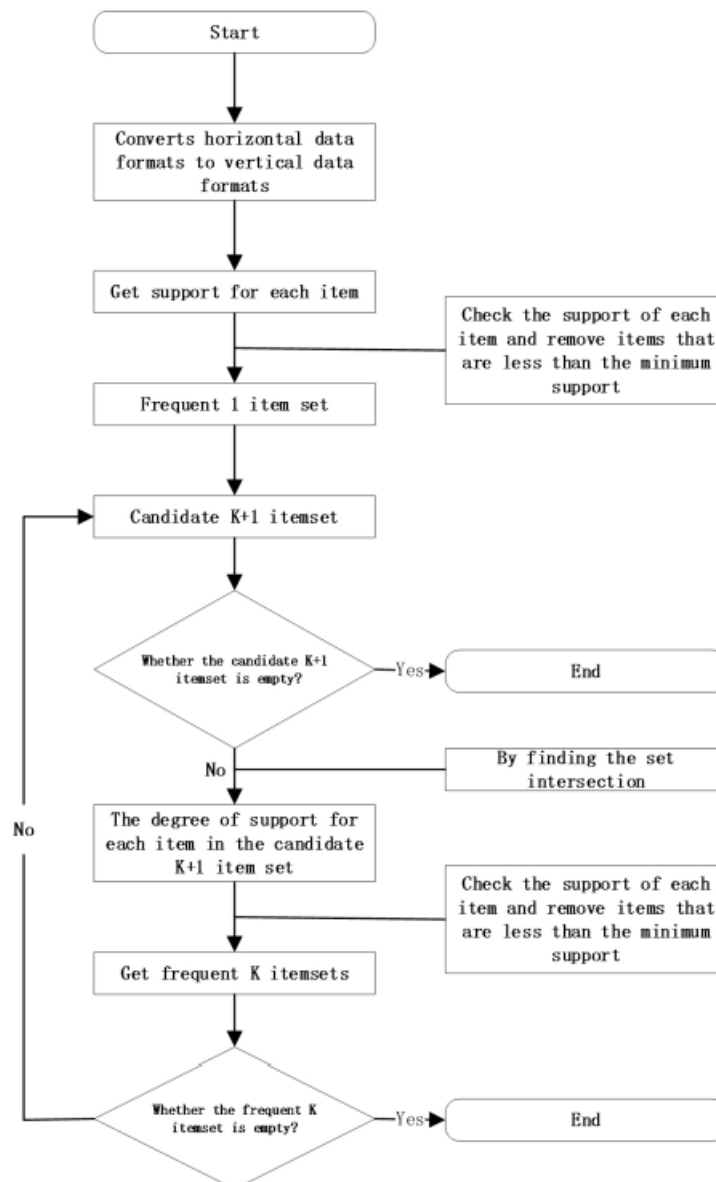


Figura 2.3: Fluxograma do algoritmo *Eclat*, retirado de [28].

2.3.3 Algoritmo FP-Growth

O algoritmo *FP-Growth* ou *Frequent Pattern Growth* foi introduzido por Han et al. no ano de 2000 [29]. Surgiu como uma outra abordagem que não necessita de gerar candidatos explicitamente.

Como esperado, este algoritmo também partilha do mesmo objetivo de encontrar padrões frequentes em transações de dados. Contudo, este constrói uma estrutura em árvore onde caminhos comuns partilham prefixos, de modo a evitar enumerar e verificar cada conjunto de dados individualmente [29].

A estrutura de uma *FP-Tree* permite extrair os conjuntos frequentes através da projeção de caminhos condicionais, reduzindo significativamente o custo computacional, especialmente em bases de dados densas ou com itens recorrentes.

2.3. Abordagens para Mineração de Regras de Associação

Para ilustrar o funcionamento do algoritmo, é utilizado novamente o exemplo das transações da Tabela 2.5, assumindo um limiar mínimo de 70% de confiança e 50% de suporte. O algoritmo segue as seguintes etapas [30]:

1. **Ordenar os itens por frequência numa tabela:** Esta tabela mostra quantas vezes cada item aparece, usada para posteriormente construir a árvore;

Tabela 2.14: Tabela de frequências.

Item	Contagem de Suporte	Suporte (%)
Pão	4	80%
Leite	4	80%
Manteiga	3	60%

2. **Ordenar as transações pelos itens mais frequentes:** Após obter a tabela de frequência dos itens, o próximo passo é ordenar os itens dentro de cada transação por ordem decrescente da sua frequência global. Esta ordenação é essencial para garantir que a *FP-Tree* maximize a compressão dos dados, ao agrupar prefixos comuns;

Tabela 2.15: Transações ordenadas por frequência dos itens.

Transação	Itens Ordenados
T1	Pão, Leite, Manteiga
T2	Pão, Leite
T3	Pão, Manteiga
T4	Leite
T5	Pão, Leite, Manteiga

3. **Construção da *FP-Tree*:** Com as transações ordenadas constrói-se a estrutura em árvore. Por definição, o nó raiz da árvore é nulo e cada transação é colocada sequencialmente respeitando a ordem dos itens ordenados previamente. Verifica-se se já existe um nó filho com o item correspondente ao que se pretende adicionar. Se existir, incrementa-se o contador, caso não exista, cria-se um novo nó com contador com valor igual a um;

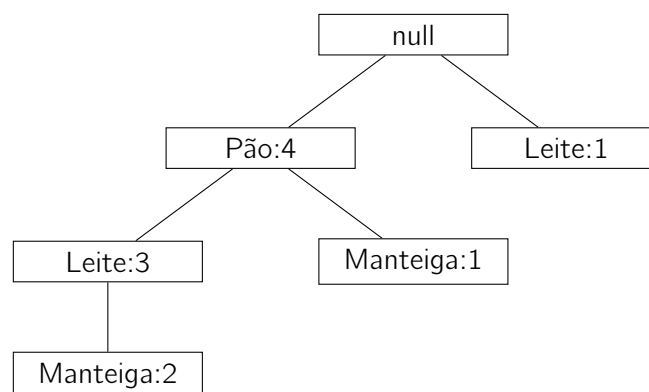


Figura 2.4: Estrutura *FP-Tree*.

4. **Construir padrões condicionais:** A partir da *FP-Tree* são extraídos os caminhos que terminam em cada item. Estes caminhos representam os padrões condicionais utilizados para gerar subconjuntos frequentes de forma recursiva. A tabela seguinte resume os prefixos encontrados para cada item relevante, acompanhados da respetiva contagem de suporte;

Tabela 2.16: Padrões condicionais da *FP-Tree*.

Item	Padrões Condicionais (Prefixos)
Manteiga	{Pão, Leite}:2; {Pão}:1
Leite	{Pão}:3

5. **Extrair padrões frequentes a partir dos prefixos:** Para cada prefixo extraem-se os conjuntos de itens frequentes (com suporte mínimo). Estes conjuntos são combinados com o item base para formar os padrões finais.

Tabela 2.17: Padrões frequentes extraídos dos prefixos.

Padrão Frequente	Contagem de Suporte	Suporte (%)
{Pão, Leite, Manteiga}	2	40%
{Pão, Manteiga}	3	60%
{Pão, Leite}	3	60%

6. **Gerar as regras de associação:** Com base nos conjuntos frequentes gerados, criam-se regras de associação, e calcula-se a confiança de cada uma. Apenas as regras que atingem ou ultrapassam o limiar de confiança ($\geq 70\%$) são consideradas válidas.

Tabela 2.18: Regras de Associação a partir dos conjuntos frequentes gerados com *FP-Growth*.

Regra	Suporte do Conjunto	Suporte do Antecedente	Confiança (%)	Validação
Pão \rightarrow Manteiga	3	4	75	Aprovada
Manteiga \rightarrow Pão	3	3	100	Aprovada
Pão \rightarrow Leite	3	4	75	Aprovada

Através da análise da Tabela 2.18 verifica-se, mais uma vez, que as regras de associação extraídas são as mesmas do *Apriori* e do *Eclat*, o que faz sentido visto que todos os algoritmos analisaram o mesmo conjunto de dados [26].

Como o *FP-Growth* evita a geração de todos os candidatos e explora a base de dados através de uma árvore, esta abordagem mostra-se capaz de lidar melhor com conjuntos de dados mais densos. No entanto, existem cenários em que a construção e manutenção da *FP-Tree* pode exigir mais memória, nomeadamente quando existe uma alta diversidade de padrões [30].

2.4 Trabalhos Relacionados

O objetivo desta secção é apresentar os trabalhos relevantes relacionados com os pilares centrais desta dissertação. Através da revisão da literatura, é possível compreender o estado da arte, identificar abordagens existentes, analisar as suas vantagens e limitações, e reconhecer lacunas que fundamentam a proposta deste trabalho.

Num primeiro momento, são analisados estudos que aplicam algoritmos de mineração de regras de associação, como o *Apriori*, *FP-Growth* e *Eclat*, com foco na sua aplicação a sistemas de recomendação. Esta análise inclui comparações de desempenho, contextos de utilização e resultados obtidos.

De seguida, são explorados trabalhos relacionados com sistemas de recomendação, incluindo abordagens que incorporam regras de associação e/ou dados de contexto, evidenciando o seu impacto na personalização e eficácia das recomendações.

2.4.1 Mineração de Regras em Sistemas Reais

De modo a avaliar a eficácia computacional e a escalabilidade em diferentes conjuntos de dados, Heaton [31] aplicou os três algoritmos a diferentes conjuntos de dados. Os resultados indicam que, embora o *Apriori* seja mais intuitivo e simples, apresenta tempos de execução significativamente superiores aos restantes, especialmente em grandes bases de dados. O *FP-Growth* destaca-se pela menor utilização de memória devido à sua estrutura em árvore, mas o *Eclat* também mostra um bom desempenho.

O autor afirma que não existe uma solução universalmente superior - a escolha do algoritmo deve considerar o tamanho e natureza do conjunto de dados, bem como a disponibilidade de recursos computacionais [31].

O estudo de Nurbojatmiko et al. [32] aplicou os algoritmos *Eclat* e *FP-Growth* para analisar padrões na escolha de cursos universitários por estudantes utilizando dados transacionais sobre interesses, talentos e perspetivas de carreira.

Os resultados indicaram que ambos os algoritmos apresentaram desempenhos semelhantes - o *Eclat* gerou regras mais simples e interpretáveis, enquanto o *FP-Growth* forneceu maior nível de detalhe e cobertura [32].

Já o estudo de Srinadh [33] comparou os algoritmos *Apriori*, *FP-Growth* e *Eclat* em diferentes conjuntos de dados, avaliando métricas como o tempo de execução, número de regras extraídas, uso de memória e desempenho com diferentes configurações de suporte e confiança. O número de conjuntos frequentes identificados foi o mesmo entre os algoritmos, o que era expectável dado que foram aplicados aos mesmos dados.

No entanto, observaram-se diferenças relevantes no desempenho: o *Apriori* apresentou os tempos médios de execução mais elevados, enquanto o *Eclat* se destacou pela sua eficiência, com resultados muito próximos dos obtidos pelo *FP-Growth*. Contudo, no que diz respeito ao uso de memória, verificou-se o oposto — o *Eclat* registou os picos mais elevados, seguido pelo *Apriori*, enquanto o *FP-Growth* se destacou por consumir significativamente menos memória. O objetivo do autor era analisar qual dos algoritmos era mais eficiente em termos de tempo de execução, tendo concluído que o *Eclat* se destacava nesse aspeto. Contudo, foi também salientado que, caso fossem considerados outros fatores — como o uso de memória — a conclusão poderia ser diferente [33].

Como os resultados variam entre autores e tipos de dados, um dos objetivos deste trabalho foi verificar esses diferentes aspectos no contexto do GrouPlanner, de modo a avaliar a eficiência relativa dos três algoritmos (*Apriori*, *Eclat* e *FP-Growth*) em termos de tempo de execução, consumo de memória e qualidade das regras de associação geradas para o GRS em desenvolvimento.

2.4.2 Recomendações em Sistemas Reais

Sminov et al.[34] estudaram o impacto de integrar dados de contexto, como localização e clima, na precisão das sugestões de um sistema de recomendação para grupos de turismo móvel. O modelo combina filtragem colaborativa e pré-filtragem baseada no contexto com o objetivo de adaptar as recomendações às condições ambientais e às necessidades do grupo.

A arquitetura modular do sistema facilita a sua implementação e adaptação a diversos casos de uso, destacando-se pela flexibilidade e capacidade de resposta às mudanças no contexto. O estudo também aborda os desafios na coleta e processamento de dados contextuais, realçando a importância de técnicas que suportem ambientes dinâmicos e variados [34].

Tao et al.[35] implementaram um sistema de recomendação que combina as preferências individuais com a popularidade dos pontos turísticos. A popularidade é usada como uma métrica que reflete tanto os gostos pessoais como as tendências gerais do setor. O principal objetivo é maximizar a satisfação coletiva do grupo, tendo em conta a diversidade de interesses entre os seus membros.

Esta abordagem destaca-se pela capacidade de melhorar a relevância das recomendações em contextos de decisão coletiva. De acordo com os testes realizados pelos autores, o método demonstrou melhorias significativas na experiência dos utilizadores. O estudo sublinha ainda os desafios associados à agregação de preferências em grupo e reforça a importância de estratégias que promovam a justiça e o consenso nas recomendações [35].

Wan [36] propôs um sistema de recomendação para turismo que, embora gere apenas recomendações individuais, integra mineração de regras de associação e filtragem colaborativa. Para mitigar problemas relacionados com a sobrecarga informacional no domínio turístico, a arquitetura combina dados de transações colaborativas com os padrões extraídos por associação, com o objetivo de melhorar a precisão das sugestões.

Esta abordagem destaca-se pela fórmula híbrida de filtragem colaborativa otimizada e associação de dados, projetada especificamente para recomendar itinerários e atrações mesmo em cenários turísticos dinâmicos e densos. O sistema proposto visa melhorar a relevância das sugestões, eliminando informação redundante e maximizando a adaptabilidade ao contexto do utilizador [36].

Yang et al.[37] apresenta um sistema de recomendação de produtos cerâmicos baseado em regras de associação extraídas a partir de dados personalizados de transações. O principal objetivo é identificar padrões de compra e preferências dos consumidores, permitindo gerar recomendações mais direcionadas e relevantes para cada utilizador. A arquitetura do sistema combina mineração de regras de associação com análise de dados históricos, de forma a capturar tendências de consumo específicas e promover produtos que têm maior probabilidade de serem adquiridos pelos clientes.

Esta abordagem destaca-se pela capacidade de personalização baseada em comportamentos individuais, oferecendo recomendações que vão além das tendências gerais do mercado. Os resultados do estudo demonstram que o uso de regras de associação pode aumentar a

precisão das recomendações, mas também evidenciam limitações, como a necessidade de um volume mínimo de dados para extrair padrões significativos e o risco de gerar sugestões redundantes quando os perfis dos utilizadores são muito homogêneos. O sistema mostra-se eficaz para otimizar a experiência do consumidor e apoiar decisões de compra, especialmente em contextos com grande variedade de produtos [37].

Zhou [38] propôs um sistema de recomendação para compras online baseado no algoritmo *Apriori*, utilizando regras de associação para identificar padrões frequentes de compra entre os utilizadores. O objetivo é gerar sugestões personalizadas de produtos com base nas transações anteriores, de forma a aumentar a relevância das recomendações e melhorar a experiência de compra. A metodologia combina a análise de dados históricos com a extração de regras de associação, permitindo ao sistema prever produtos que têm maior probabilidade de interesse para cada utilizador.

Esta abordagem destaca-se pela simplicidade e eficácia do algoritmo *Apriori* em cenários com grande volume de transações, facilitando a identificação de padrões frequentes de compra. Contudo, o estudo também aponta limitações, como o elevado custo computacional para bases de dados muito grandes e a dificuldade em lidar com novos utilizadores ou produtos (problema de *cold-start*). Apesar disso, os resultados indicam que a aplicação de regras de associação aumenta a precisão das recomendações, contribuindo para decisões de compra mais informadas e para a satisfação do utilizador [38].

Kongpeng et al. [39] desenvolveram um sistema de recomendação de destinos turísticos baseado em algoritmos de *machine learning*. O estudo recorreu a um questionário com 481 participantes para recolher preferências de viagem no período pós-pandemia e testou diferentes algoritmos, incluindo *Random Forest*, *Support Vector Machine*, *K-Nearest Neighbors*, *Adaptive Boosting* e redes neuronais artificiais. O objetivo foi identificar o método mais eficaz para prever as escolhas de viagem, analisando ainda os efeitos marginais das variáveis que mais influenciam as preferências dos utilizadores.

Esta abordagem destaca-se pela capacidade de integrar dados contextuais, como mudanças de comportamento após a pandemia, com técnicas de *machine learning* avançadas, permitindo assim recomendações mais adaptadas ao perfil do utilizador. Apesar de não explorar regras de associação ou estratégias colaborativas, os resultados obtidos demonstram ganhos relevantes em termos de precisão e capacidade preditiva. O estudo mostra ainda que diferentes algoritmos apresentam desempenhos distintos consoante a natureza dos dados, sublinhando a importância da escolha do modelo mais adequado ao contexto turístico [39].

Após a análise de diferentes abordagens aplicadas a sistemas de recomendação, importa destacar o GrouPlanner, um sistema já existente que serve como base para este trabalho. O GrouPlanner diferencia-se por abordar diretamente o problema das recomendações em grupo, integrando fatores como as preferências turísticas individuais, as dimensões de personalidade e a heterogeneidade entre os membros [5]. Para tal, recorre a técnicas de *clustering* dinâmico para agrupar utilizadores com perfis semelhantes em tempo real, mitigando problemas como o *cold-start* e a dispersão de preferências [5].

Neste contexto, as regras de associação são utilizadas para identificar padrões frequentes de preferências e comportamentos, permitindo afinar as recomendações com base em correlações observadas entre diferentes categorias turísticas [5]. Desta forma, contribuem para melhorar a personalização e aumentar a relevância das sugestões apresentadas, reforçando a experiência coletiva dos utilizadores [5].

Capítulo 3

Conceptualização e Design da Solução

O presente capítulo descreve a conceptualização e o design da solução desenvolvida para este projeto, a otimização de um sistema de recomendação de pontos de interesse turísticos, individual e em grupo. O objetivo é detalhar o processo seguido para atingir os objetivos definidos anteriormente.

Inicia-se com a apresentação da aplicação, domínio e dos atores envolvidos, seguindo-se a definição dos requisitos essenciais que suportam a lógica de recomendação. A secção dedicada ao desenho da solução descreve as interações entre os serviços e a forma como os dados dos utilizadores e os dados externos são processados para gerar recomendações personalizadas. Finalmente, o capítulo aborda a integração de dados de contexto, incluindo meteorologia, qualidade do ar e nível de ruído, explicando como estes fatores influenciam as recomendações e a experiência do utilizador.

3.1 Arquitetura do GrouPlanner

Antes do desenvolvimento deste trabalho, o GrouPlanner já existia e foi construído com tecnologia *.NET*, uma *framework* amplamente utilizada para criar serviços web e aplicações multiplataforma [5].

O *User Management Microservice* (UMMS) é o responsável por gerir e armazenar os dados dos utilizadores registados. Estes dados incluem informações demográficas, dimensões de personalidade, limitações, medos, escolhas de atrações turísticas, preferências de viagem, registos de POI visitados e não visitados, avaliações de POI e outros dados de perfil [5].

Já o *Point of Interest Microservice* (POIMS) é responsável pela gestão de todos os POI que podem ser sugeridos aos utilizadores. Este microserviço utiliza informações obtidas através da API da *TripAdvisor* [40] para garantir que os dados de cada POI se mantenham atualizados [41].

O MAMS cria agentes no seu ambiente sempre que um utilizador se regista ou quando uma nova excursão é formada. Cada agente é modelado de acordo com o perfil do respetivo utilizador. Este serviço é responsável por gerar *clusters* dinâmicos de turistas com personalidades semelhantes e, no caso de grupos de excursão, subdividi-los em subgrupos consoante as características individuais [5].

Adicionalmente, o MAMS armazena em ambiente as regras de associação extraídas, que influenciam diretamente o processo de recomendação [5].

Em conjunto com o REMS, este é um dos serviços mais relevantes para o presente projeto, uma vez que assegura a organização prévia dos utilizadores em subgrupos homogêneos, o que constitui uma base sólida para a aplicação de técnicas de filtragem colaborativa na fase de geração das recomendações [5].

Para determinar os melhores POI a recomendar, o REMS considera as preferências de atrações turísticas, limitações, fobias, regras identificadas pelo MAMS e dados de contexto, como meteorologia, qualidade do ar e níveis de ruído.

Na versão anterior da aplicação, a informação meteorológica era disponibilizada pela API de clima da Sistrade [42], que, durante o desenvolvimento deste projeto, foi substituída pela API do Instituto Português do Mar e da Atmosfera (IPMA) [43]. Relativamente à qualidade do ar e aos níveis de ruído, estes foram conceitos novos introduzidos no sistema, sendo necessária a integração com as respetivas API da Porto Digital [44, 45]. O processo é descrito mais detalhadamente no Subcapítulo 4.2.3.

Existe ainda o *Social Network Microservice (SNMS)*, responsável pela organização dos *chats* de grupo e subgrupo, garantindo que as mensagens entre grupos sejam acessíveis e apresentadas corretamente na interface do utilizador [41].

Por fim, a aplicação móvel, acessível ao cliente, funciona num ambiente *Android*. Esta aplicação dispõe de diversas funcionalidades, jogos de personalidade, criação de grupos, listas de recomendações, sistema de troféus, entre outros. Além disso, o dispositivo fornece informações essenciais, como a geolocalização do utilizador, que permitem integrar dados de contexto e personalizar as recomendações [41]. Existe ainda uma conexão com a API *Google Places* [46] que ajuda com o histórico dos POI visitados do utilizador.

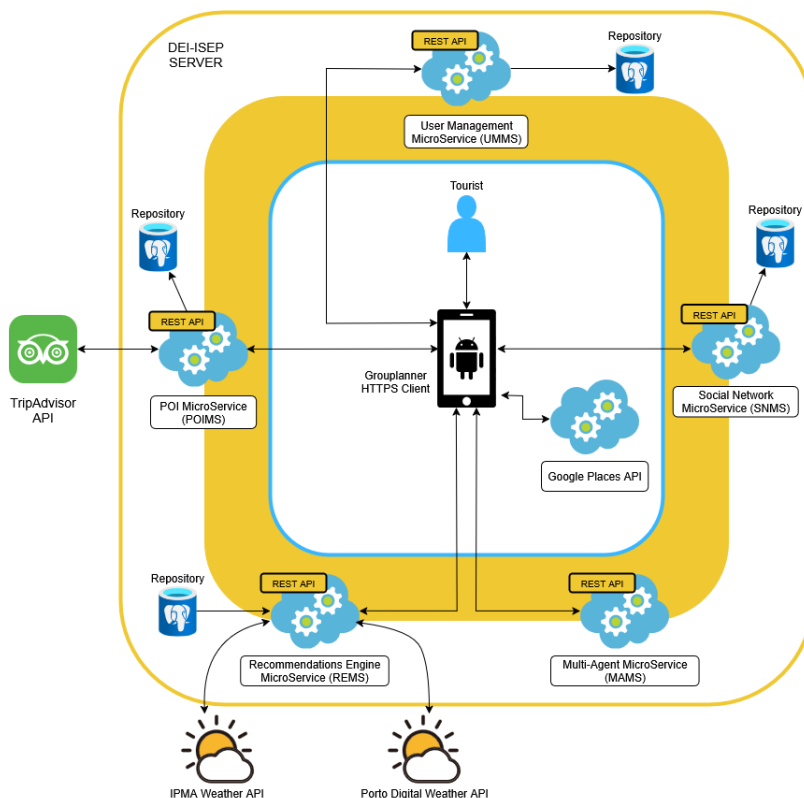


Figura 3.1: Arquitetura de microserviços do Grouplanner, adaptado de [41].

3.1. Arquitetura do GrouPlanner

Para facilitar a compreensão da arquitetura global da aplicação, apresentam-se dois diagramas: o diagrama adaptado de Alves et al. [41] (Figura 3.1), que ilustra a arquitetura geral da aplicação, e o diagrama de vista lógica (Figura 3.2), que detalha os principais módulos, microserviços e fluxos de informação.

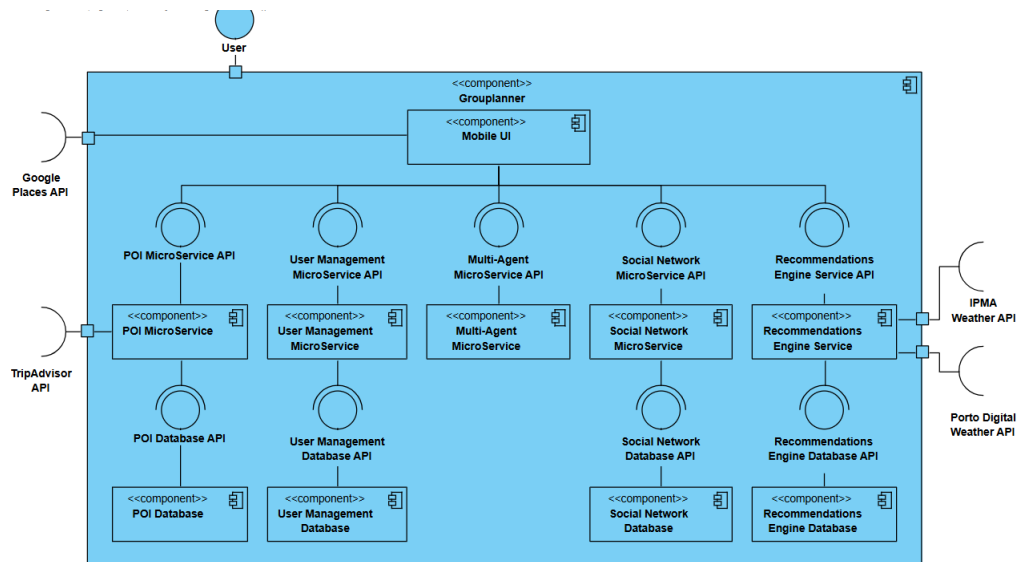


Figura 3.2: Diagrama de vista lógica.

Complementarmente, a Figura 3.3 apresenta o diagrama de vista física, que ilustra a distribuição real dos componentes da aplicação, incluindo servidores, bases de dados e interfaces externas. Esta visão permite compreender onde cada módulo é executado e como os diferentes elementos comunicam entre si, garantindo que a aplicação seja escalável, eficiente e resiliente.

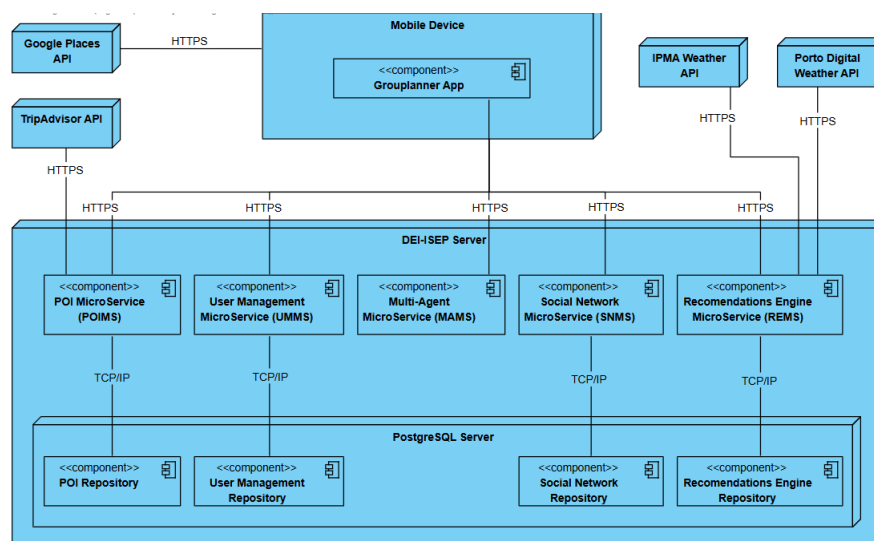


Figura 3.3: Diagrama de vista física.

3.2 Domínio

De modo a enquadrar melhor o problema em estudo, é necessário compreender o domínio. O objetivo da apresentação do modelo de domínio é identificar as entidades principais, bem como as relações entre elas, fornecendo uma visão conceptual do contexto em que a aplicação se insere.

A Figura 3.4 apresenta o excerto do modelo relevante para este projeto. Desde o início do desenvolvimento, o modelo sofreu algumas alterações de modo a acomodar contributos de outros colaboradores, contudo, no âmbito específico deste trabalho foram apenas realizadas pequenas modificações como renomear ou reorganizar variáveis.

O utilizador (**User**) constitui a entidade central do sistema. Para além dos atributos básicos de identificação, possui ainda características adicionais que permitem representar o seu perfil de forma mais completa, nomeadamente as limitações (**Limitation**), os medos (**Fear**) e as preferências turísticas (**TravelPreferences**). Adicionalmente, cada utilizador encontra-se associado à sua cidade de residência (**City**).

As entidades referidas são fundamentais, uma vez que moldam o perfil do utilizador e permitem a construção de recomendações mais adequadas. Para além destas, destaca-se a entidade personalidade (**Personality**), que incorpora as cinco dimensões da personalidade conhecidas do modelo *Big Five* [18]: abertura à experiência (**Openness**), conscienciosidade (**Conscientiousness**), extroversão (**Extraversion**), agradabilidade (**Agreeableness**) e neuroticismo (**Neuroticism**) [47].

Para além disso, os utilizadores podem ser organizados em conjuntos (**Clusters**), definidos com base nas suas características de personalidade (**Personality**). Estes conjuntos (**Clusters**) permitem agrupar utilizadores com perfis semelhantes, e podem ser associados a regras (**Rules**), geradas a partir dos atributos dos utilizadores, que orientam o processo de recomendação de **POI** de forma mais personalizada.

Para finalizar, os utilizadores (**Users**) podem integrar grupos de excursão (**Group**). Estes grupos podem, por sua vez, ser subdivididos em subgrupos (**Subgroup**), de acordo com a personalidade e as preferências dos membros do respetivo subgrupo. Esta estrutura permite ao sistema considerar tanto características individuais como coletivas na geração de recomendações (**POIRecommendation**) personalizadas.

3.3 Atores do Sistema

Os atores são as entidades responsáveis por desempenhar os casos de uso. É possível que um mesmo ator esteja associado a um ou mais casos de uso. Estes podem ser pessoas ou sistemas, como hardware ou software, que interagem diretamente com o sistema.

No GrouPlanner foram identificados os seguintes atores:

- **Turista:** assume o papel central, pois é através da sua interação que as funcionalidades ganham sentido.

Ao preencher um questionário de personalidade, o utilizador fornece dados essenciais que permitem ao sistema compreender melhor os seus interesses, preferências e preocupações de viagem [4].

Com base nessa informação, o **Turista** pode então solicitar recomendações personalizadas de pontos de interesse, que refletem os seus gostos específicos.

Tem a possibilidade de avaliar atrações turísticas, ação que contribui para refinar a qualidade das recomendações, beneficiando tanto o próprio como outros utilizadores com perfis semelhantes.

Pode integrar grupos de excursão onde pode comunicar e partilhar experiências com os restantes participantes através de uma funcionalidade de *chat*;

- **Turista Líder do Grupo:** tem acesso às mesmas funcionalidades do **Turista** comum, mas, tal como o nome indica, o seu papel é de líder.

Cabe-lhe a criação e gestão de grupos de excursão, bem como a definição das excursões que estes realizam. Para além de coordenar a estrutura do grupo, este ator tem ainda a possibilidade de solicitar recomendações coletivas, construídas com base nas preferências e perfis de todos os membros desse grupo;

- **Sistema:** assume também o papel de ator, pois não se limita a receber dados - processa-os e age com base neles.

Cabe-lhe as tarefas de organizar os **Turistas** em *clusters*, identificar padrões através de regras de associação e gerar recomendações que podem ser solicitadas individualmente ou em grupo;

- **Administrador do Sistema:** possui funções avançadas de gestão.

Para além de supervisionar a aplicação, este pode intervir diretamente no processo de geração de regras de associação, selecionando o algoritmo a aplicar e ajustando critérios como suporte, confiança ou ainda os parâmetros relacionados com os utilizadores, assegurando assim maior flexibilidade no comportamento do **Sistema**.

Com a identificação e descrição dos atores do sistema, torna-se possível compreender quem interage com a aplicação. A Figura 3.5 apresenta o Diagrama de Casos de Uso, resumindo visualmente as interações entre os atores e os casos de uso relevantes para o desenvolvimento deste projeto. Este diagrama inclui casos de uso já implementados no sistema, bem como outros considerados relevantes para o contexto da dissertação, mesmo que não tenham sido efetivamente desenvolvidos por este trabalho.

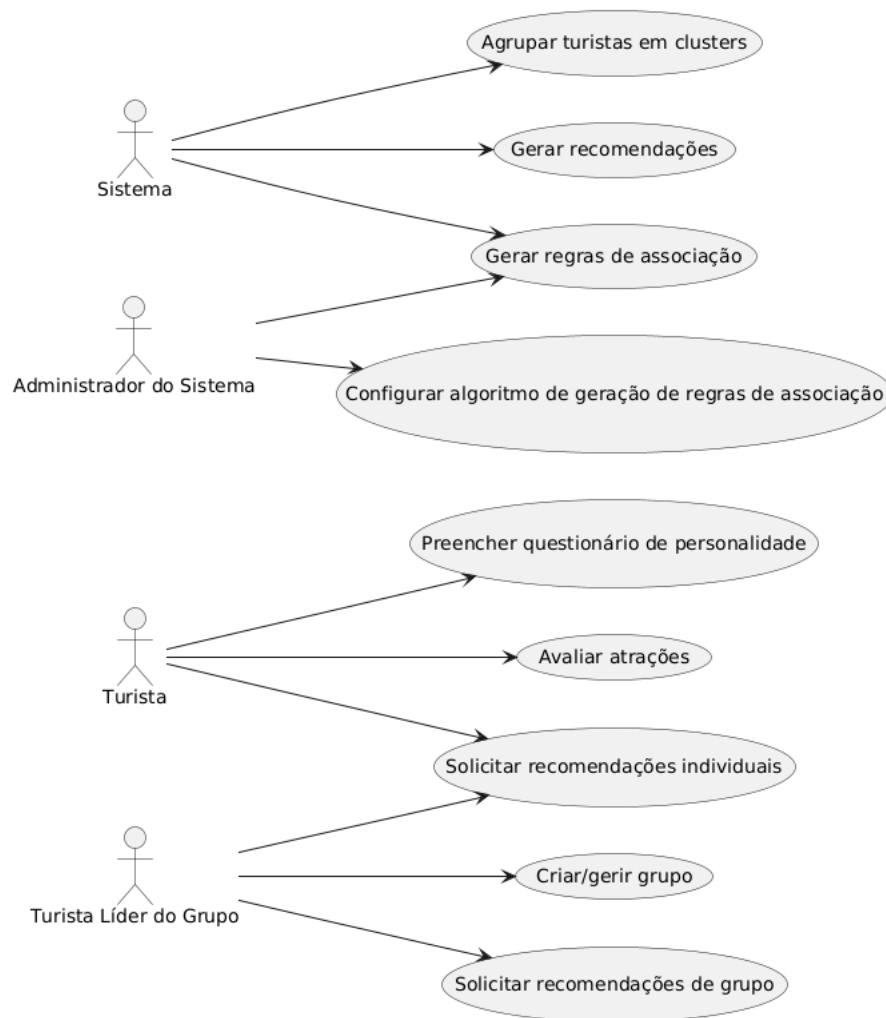


Figura 3.5: Diagrama de casos de uso.

3.4 Requisitos Funcionais

Após a identificação e descrição dos atores do sistema, é importante definir os requisitos funcionais que orientam a implementação das funcionalidades relevantes para este projeto. A Tabela 3.1 apresenta as *user stories* que foram definidas no âmbito deste trabalho, refletindo as ações que o sistema deve realizar para gerar regras de associação e fornecer recomendações de pontos de interesse, considerando ou não o contexto.

As primeiras três *user stories* (US-101 a US-103) focam na geração de regras de associação, incluindo a extração de padrões a partir dos perfis e preferências dos utilizadores, a configuração dos parâmetros e a exportação dos resultados para análise.

As *user stories* seguintes (US-201 e US-202) detalham o processo de recomendação de POI, tanto de forma genérica, baseada nas regras extraídas, como de forma contextualizada, considerando fatores ambientais como condições meteorológicas, nível de ruído e qualidade do ar.

Tabela 3.1: Requisitos funcionais.

ID	Requisito Funcional
US-101 Geração de Regras de Associação	Como Sistema , quero gerar regras de associação a partir dos perfis e preferências dos utilizadores, para identificar padrões.
US-102 Configuração de Parâmetros	Como Administrador do Sistema , quero poder configurar limites de suporte e confiança, bem como selecionar quais atributos dos utilizadores considerar, de forma a personalizar o processo de geração de regras.
US-103 Exportação de Regras	Como Sistema , quero exportar as regras geradas em formato legível (CSV, XML ou JSON), para análise e reutilização.
US-201 Geração de Recomendações	Como Sistema , quero recomendar POI a utilizadores individuais ou grupos considerando as regras de associação extraídas, os perfis e as preferências de cada utilizador.
US-202 Geração de Recomendações Contextualizadas	Como Sistema , quero adaptar as recomendações de POI, para utilizadores individuais ou grupos, tendo em conta o contexto (como condições meteorológicas, nível de ruído ou qualidade do ar), garantindo que as sugestões sejam adequadas à situação.

A seguir, apresentam-se as *user stories* identificadas com mais detalhe, incluindo a respetiva descrição, critérios de aceitação e prioridade:

US-101 Geração de Regras de Associação

- **Descrição:** O sistema deve gerar regras de associação a partir dos perfis e preferências dos utilizadores, identificando padrões frequentes e correlações entre atributos. Deve suportar múltiplos algoritmos de mineração (Apriori, FP-Growth e Eclat).
- **CrITÉrios de Aceitação:** As regras geradas devem refletir os padrões mais frequentes dentro do conjunto de dados de utilizadores, garantindo que cada algoritmo utilizado produza resultados consistentes e corretos. As regras podem ser associadas a *clusters* específicos ou globais, dependendo dos limites definidos.
- **Prioridade:** M (*Must*) - essencial para o funcionamento do sistema de recomendação.

US-102 Configuração de Parâmetros

- **Descrição:** O administrador do sistema deve ser capaz de configurar os parâmetros de mineração de regras de associação, nomeadamente os valores mínimos de suporte e confiança, bem como a seleção dos atributos dos utilizadores que serão considerados no processo (por exemplo, preferências turísticas, limitações ou medos). Esta funcionalidade assegura flexibilidade e personalização na extração de padrões, possibilitando ajustar os resultados às necessidades específicas de cada análise.
- **CrITÉrios de Aceitação:** O sistema deve ler corretamente os parâmetros definidos no ficheiro *appsettings.json* e aplicar os valores de suporte, confiança e atributos selecionados no processo de mineração. Alterações realizadas no ficheiro devem refletir-se

automaticamente em novas execuções do sistema, sem comprometer regras já previamente geradas.

- **Prioridade:** M (*Must*) - essencial para o funcionamento do sistema de recomendação.

US-103 Exportação de Regras

- **Descrição:** O sistema deve permitir a exportação das regras de associação geradas em formatos legíveis e reutilizáveis (como *Comma-Separated Values (CSV)*, *Extensible Markup Language (XML)* ou *JavaScript Object Notation (JSON)*). Esta funcionalidade garante que os resultados obtidos possam ser analisados externamente, documentados ou integrados noutros sistemas. Para além disso, os ficheiros exportados podem ser utilizados em processos de teste de qualidade e avaliação de desempenho, permitindo verificar a consistência e eficiência dos algoritmos de mineração de regras.
- **CrITÉrios de Aceitação:** O sistema deve disponibilizar a exportação das regras num ficheiro no formato especificado, contendo os elementos essenciais de cada regra (antecedente, conseqüente e confiança). O ficheiro gerado deve poder ser lido e processado em ferramentas externas de análise de dados ou manipulação de texto.
- **Prioridade:** S (*Should*) - funcionalidade importante para a flexibilidade e personalização do sistema, mas não crítica para o funcionamento básico.

US-201 Geração de Recomendações

- **Descrição:** O sistema deve recomendar POI a utilizadores individuais ou grupos, tendo por base as regras de associação previamente extraídas, os perfis e as preferências de cada utilizador. Esta funcionalidade constitui o núcleo do sistema de recomendação, permitindo gerar sugestões personalizadas que refletem tanto os interesses declarados pelos utilizadores como os padrões identificados pela sua personalidade.
- **CrITÉrios de Aceitação:** As recomendações devem apresentar POI alinhados com os perfis e preferências dos utilizadores, utilizando as regras de associação como suporte ao processo de decisão. Cada utilizador ou grupo deve receber um conjunto de recomendações relevantes e consistentes. O sistema deve assegurar que as recomendações não incluam POI já visitados.
- **Prioridade:** M (*Must*) - essencial para o funcionamento do sistema de recomendação.

US-202 Geração de Recomendações Contextualizadas

- **Descrição:** O sistema deve adaptar as recomendações de POI a utilizadores individuais ou grupos, tendo em conta fatores contextuais, tais como condições meteorológicas, nível de ruído ou qualidade do ar. Esta funcionalidade complementa a geração de recomendações base, garantindo que as sugestões sejam adequadas ao momento e às circunstâncias específicas do utilizador.
- **CrITÉrios de Aceitação:** As recomendações devem refletir não apenas as preferências e perfis dos utilizadores, mas também as condições do ambiente no momento da consulta. O sistema deve filtrar ou priorizar POI que sejam compatíveis com o contexto identificado (por exemplo, evitar atividades ao ar livre em caso de chuva intensa). As recomendações devem manter consistência, evitando contradições entre preferências e contexto, e devem poder ser reproduzidas com base nas mesmas condições de entrada.

- **Prioridade:** S (*Should*) - funcionalidade importante para a flexibilidade e personalização do sistema, mas não crítica para o funcionamento básico.

3.5 Recolha dos Dados

O sistema de recomendação assenta sobre um conjunto de dados previamente recolhido durante um estudo de dois modelos de previsão de preferências turísticas com base na personalidade, documentados no trabalho de Alves et al.[4].

Do estudo realizado, resultou um *dataset* de mil e trinta e cinco (1035) participantes, abrangendo diferentes áreas de formação e faixas etárias.

Para implementar os algoritmos das regras de associação, foi necessário determinar previamente quais os dados demográficos a utilizar do *dataset*. Esta análise inclui a descrição dos atributos recolhidos, o papel de cada atributo no contexto do sistema de recomendação, bem como a caracterização geral da amostra de utilizadores e das distribuições das suas preferências.

3.5.1 Descrição dos Atributos

O *dataset* utilizado encontrava-se no formato CSV e continha os seguintes atributos para cada utilizador:

Identificação

Informações relacionadas com a identificação única e organização dos registos dos participantes (variáveis meramente técnicas, sem impacto direto na análise do estudo).

- **Timestamp:** data e hora de submissão do questionário;
- **ID:** identificador único do participante.

Demográficos

Informações relativas ao perfil demográfico dos participantes, utilizadas para caracterizar a amostra e possibilitar análises segmentadas.

- **age:** idade do participante (valor numérico positivo);
- **gender:** género do participante (feminino, masculino);
- **sexualOrientation:** orientação sexual;
- **nationality:** nacionalidade declarada;
- **country:** país de residência;
- **maritalStatus:** estado civil (solteiro(a), numa relação, casado(a), outro);
- **children:** se o participante tem filhos (sim ou não);
- **liveWith:** situação de coabitação (pais, parceiro(a) e filhos, parceiro(a), amigos, sozinho(a), filhos, outro);
- **education:** nível de escolaridade atingido (ensino secundário, licenciatura, mestrado, doutoramento, outro);

- **formationArea:** área de formação principal (engenharia e tecnologia, ciências médicas e da saúde, ciências naturais, ciências Exatas, humanidades, ciências Sociais, outra, nenhuma);
- **professionalSituation:** situação profissional atual (empregado(a), estudante, trabalhador estudante, trabalhador(a) por conta própria);
- **liquidIncome:** rendimento líquido mensal (não aplicável, < 650, entre 650 e 1000, entre 1001 e 2000, > 2000 €);
- **religion:** religião ou crença declarada (sim, não mas acredita em um ser superior, ateu, agnóstico(a), outro).

Condições de saúde e limitações

Informações relacionadas com aspetos de saúde ou restrições pessoais que podem influenciar as preferências e possibilidades de viagem dos participantes.

- **allergyChronicdiseaseDisability:** resposta binária (sim/não) que indica se o participante possui alergias, doenças crónicas ou alguma deficiência;
- **limitations:** lista textual onde o participante especifica as limitações existentes (cadeira de rodas, perda de visão, surdez, problemas cardíacos).

Informações para a viagem

Informações relacionadas com o contexto da viagem, nomeadamente o tipo de companhia com que o participante costuma viajar.

- **travelCompanions:** descreve os companheiros de viagem habituais do participante (sozinho, com família, com amigos, com companheiro).

Personalidade - BFI

Informações relacionadas com as características de personalidade dos participantes, avaliadas através do BFI Estas variáveis definem as dimensões de personalidade dos participantes.

- **Avg-Extraversion:** pontuação média das respostas do participante às questões relacionados com a dimensão extravertido da personalidade;
- **Avg-Agreeableness:** pontuação média das respostas do participante às questões relacionados com a dimensão agradabilidade da personalidade;
- **Avg-Conscientiousness:** pontuação média das respostas do participante às questões relacionados com a dimensão conscienciosidade da personalidade;
- **Avg-Neuroticism:** pontuação média das respostas do participante às questões relacionados com a dimensão neuroticismo da personalidade;
- **Avg-Openness:** pontuação média das respostas do participante às questões relacionados com a dimensão abertura à experiência da personalidade;

Preferências e fobias

Informações relacionadas com medos, fobias e preferências individuais que podem condicionar a escolha de atividades turísticas.

- **fearsPhobias:** lista de medos ou fobias reportadas pelo participante (nenhum, alturas, claustrofobia, agorafobia, outro);

- **weatherPref**: preferências relativas às condições meteorológicas durante a viagem (ex.: clima quente, ameno, frio).

Atrações turísticas

Este atributo corresponde a um conjunto de sessenta e oito (68) itens, formulados de modo a capturar preferências turísticas específicas com maior nível de detalhe. Cada item é avaliado numa escala de 1 (menor interesse) a 7 (maior interesse), permitindo quantificar o grau de preferência em diferentes contextos de lazer e turismo.

Alguns exemplos destes itens incluem:

- Ir a um festival gastronómico (comida e/ou bebidas);
- Assistir a um fenómeno natural (e.g., erupção vulcânica ou aurora boreal);
- Assistir a uma celebração religiosa;
- Visitar as cidades/vilas históricas do destino;
- Visitar um oceanário;
- Visitar cavernas/grutas/vulcões;

Pontuação para as categorias de turismo

As pontuações foram calculadas no âmbito do estudo de Alves et al. [4], com base nas respostas atribuídas às diversas atrações turísticas. São variáveis medidas indiretamente (latentes) e descritas em seguida. As respetivas pontuações encontram-se normalizadas num intervalo entre zero e um.

- **F1-normalized**: pontuação atribuída à categoria **Adrenaline Activities**: atividades de aventura intensa, como desportos radicais aquáticos, motorizados, subaquáticos, escalada, *bungee jumping*, paraquedismo, esqui, entre outros;
- **F2-normalized**: pontuação atribuída à categoria **Wild Nature Activities**: atividades de contacto com a natureza, como caminhadas, montanhismo, visitas a reservas naturais, florestas ou montanhas;
- **F3-normalized**: pontuação atribuída à categoria **Party, Music & Nightlife**: participação em eventos culturais e recreativos, como discotecas, bares com música ao vivo, desfiles temáticos, festivais de cinema, música ou dança;
- **F4-normalized**: pontuação atribuída à categoria **Sun, Water & Sand**: atividades de lazer e relaxamento na praia ou piscina, banhos de sol e férias em ilhas;
- **F5-normalized**: pontuação atribuída à categoria **Museums, Boat Trips & Viewpoints**: visitas a museus históricos ou científicos, passeios de barco, apreciação de paisagens em miradouros naturais ou urbanos;
- **F6-normalized**: pontuação atribuída à categoria **Theme & Animal Parks**: atividades de lazer familiar, incluindo idas a parques aquáticos, feiras, zoológicos ou parques temáticos;
- **F7-normalized**: pontuação atribuída à categoria **Cultural Heritage**: exploração do património histórico e cultural, como castelos, mosteiros, igrejas, catedrais e cidades/vilas históricas;

- **F8-normalized:** pontuação atribuída à categoria **Sports & Games:** participação ou assistência em atividades desportivas, incluindo caça, pesca ou eventos competitivos como jogos de futebol;
- **F9-normalized:** pontuação atribuída à categoria **Gastronomy Events:** experiências gastronómicas, como festivais, roteiros de comida típica ou *gourmet*, e provas de vinhos;
- **F10-normalized:** pontuação atribuída à categoria **Health & Well-being:** práticas relacionadas com o relaxamento, como idas a SPAs, centros de beleza, estâncias termais ou tratamentos de saúde e bem-estar;
- **F11-normalized:** pontuação atribuída à categoria **Natural Phenomena:** observação de fenómenos naturais (erupções vulcânicas, auroras boreais, quedas de água), visitas a grutas, cavernas ou vulcões.

3.5.2 Caracterização da Amostra

Após a análise da distribuição etária dos participantes, observa-se que mais de metade dos participantes (54%) se concentra entre faixas dos vinte (20) e trinta e cinco (35) anos. No entanto, destaca-se que o grupo etário entre os dezoito (18) e os vinte e cinco (25) anos apresenta o maior número de participantes, conforme ilustrado na Figura 3.6.

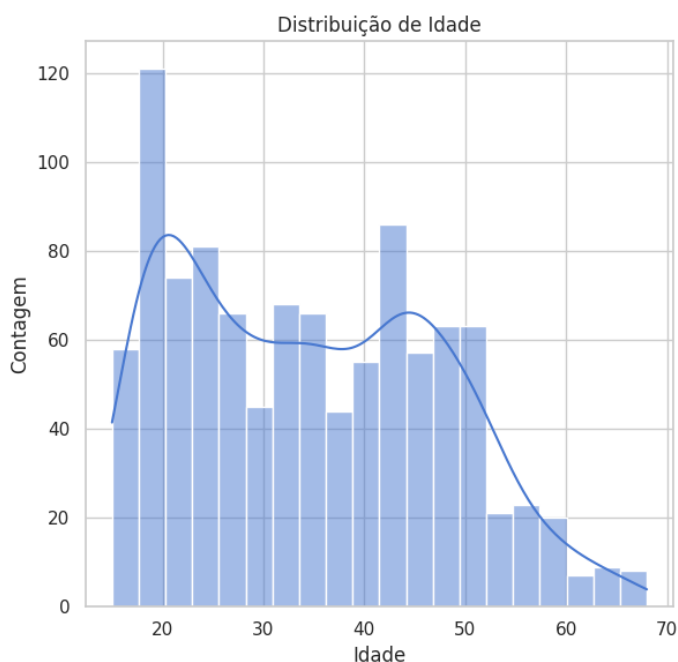


Figura 3.6: Gráfico de distribuição etária dos participantes do questionário.

No que diz respeito ao género, observa-se uma predominância do sexo feminino (74,3%), face ao masculino (25,7%).

Em relação ao nível de educação, verifica-se que a maioria dos participantes possui licenciatura (3–5 anos), seguida por Ensino Secundário e, em terceiro lugar, mestrado. As restantes categorias apresentam valores mais reduzidos, mas ainda assim relevantes para o contexto, como se observa na Figura 3.7.

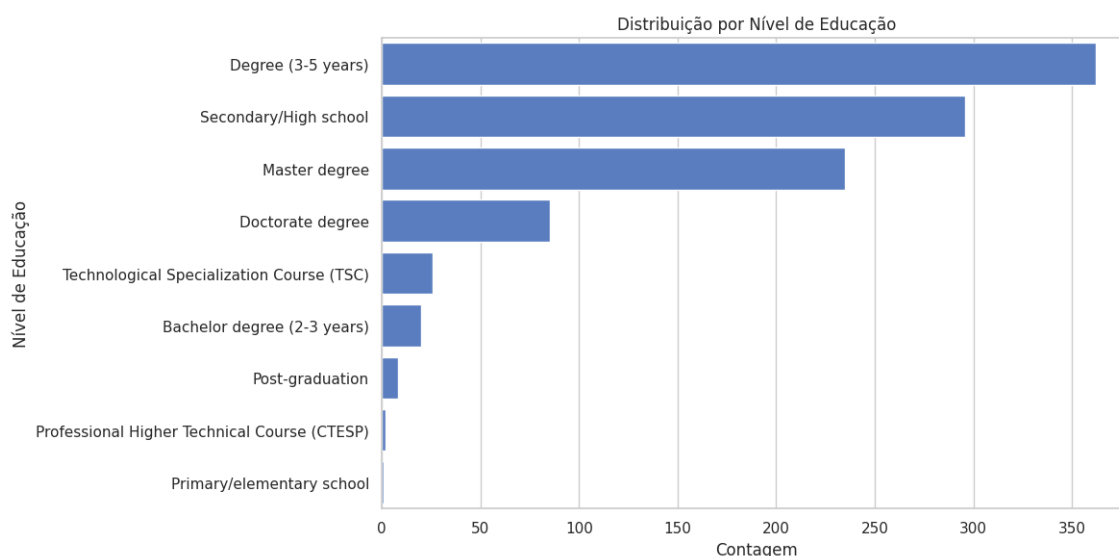


Figura 3.7: Gráfico de distribuição dos participantes segundo o nível de educação.

Os gráficos de distribuição para as dimensões de personalidade (BFI) mostram padrões típicos. As dimensões de abertura à experiência, conscienciosidade e agradabilidade apresentam ligeira assimetria positiva, enquanto extraversão e neuroticismo estão próximas do ponto médio, isto mostra enviesamento devido à deseabilidade social. No geral, as cinco dimensões seguem aproximadamente uma distribuição normal, sendo a dimensão de conscienciosidade o que mais se desvia dessa normalidade, como ilustrado na Figura 3.8.

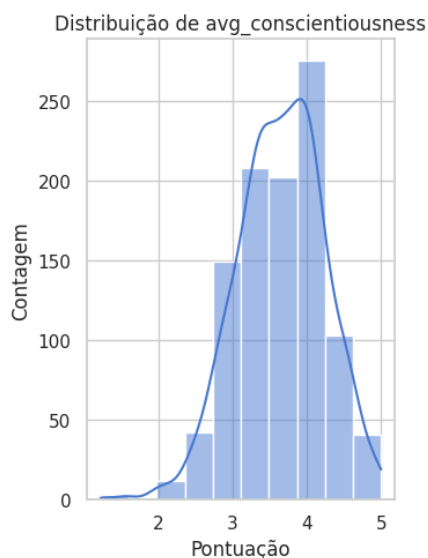


Figura 3.8: Gráfico de distribuição dos participantes segundo a dimensão de personalidade de conscienciosidade.

Verifica-se que o perfil mais comum pertence a indivíduos que se encontram numa relação (54%), sem filhos (63%) e com ensino superior (86%), predominantemente na área de Engenharia e Tecnologia.

Para as avaliações das categorias de turismo, verifica-se que o comportamento dos participantes varia consoante a categoria. Todos os gráficos correspondentes encontram-se representados na Figura 3.9.

- **Adrenaline Activities:** Distribuição relativamente simétrica, com algumas avaliações negativas, indicando que, embora haja interesse, existe também uma parcela significativa de utilizadores que não se identifica com atividades de adrenalina;
- **Wild Nature Activities:** Forte tendência para valores elevados, sugerindo uma preferência clara pela interação com a natureza e atividades ao ar livre mais intensas;
- **Party, Music & Nightlife:** Distribuição centrada refletindo uma apreciação generalizada deste tipo de atividades sociais e culturais;
- **Sun, Water & Sand:** Predominância de valores altos, evidenciando que atividades relacionadas com praia, mar e lazer aquático são populares entre os participantes;
- **Museums, Boat Trips & View-Points:** Distribuição bimodal, sugerindo a existência de dois subgrupos distintos: um com interesse moderado e outro com interesse elevado, mas com uma tendência maior para valores altos, indicando que uma parte significativa valoriza turismo cultural e experiências panorâmicas;
- **Theme & Animal Parks:** Distribuição simétrica, indicando uma aceitação equilibrada: alguns utilizadores têm forte interesse, enquanto outros mostram menor preferência;
- **Cultural Heritage:** Distribuição fortemente inclinada para valores altos, evidenciando que a maioria dos utilizadores valoriza visitas a locais históricos e património cultural;
- **Sports & Games:** Distribuição com muitos valores negativos, indicando que esta categoria não é muito popular entre os participantes;
- **Gastronomy Events:** Maioria das avaliações positivas, refletindo um interesse consistente por experiências gastronómicas e eventos culinários;
- **Health & Well-being:** Distribuição simétrica, com a maioria dos participantes a atribuir avaliações moderadas, sugerindo interesse equilibrado por atividades de saúde e bem-estar;
- **Natural Phenomena:** Tendência clara para valores positivos, indicando que os utilizadores apreciam atividades relacionadas com observação de fenómenos naturais e paisagens naturais.

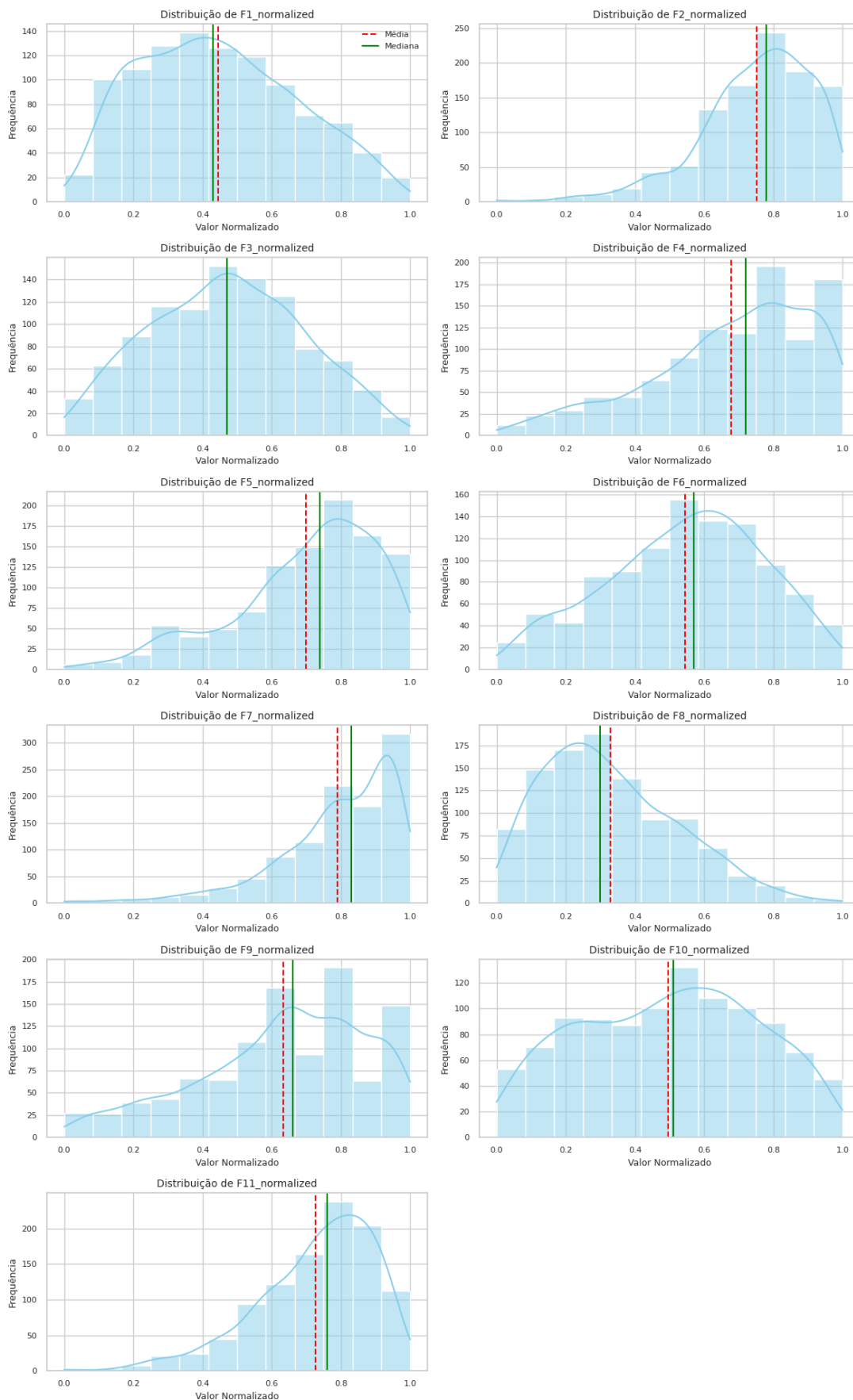


Figura 3.9: Gráficos da distribuição das avaliações dos participantes por categoria de turismo.

3.5.3 Definição de Parâmetros para Geração de Regras

Para gerar as regras de associação foram selecionados atributos que permitem capturar características relevantes dos participantes e das suas preferências turísticas.

Os atributos escolhidos incluem:

- **Data de nascimento (idade);**
- **Género;**
- **Estado civil;**
- **Presença de filhos;**
- **Situação de coabitação;**
- **Religião;**
- **Companhia habitual em viagens;**
- **Nível de escolaridade;**
- **Área de formação;**
- **Situação profissional;**
- **Rendimento líquido;**
- **Pontuação para as categorias de turismo;**
- **Limitações;**
- **Medos/fobias;**

A inclusão destes atributos visa assegurar que as regras de associação reflitam padrões significativos entre características demográficas, sociais e psicológicas, bem como preferências e restrições individuais, permitindo gerar recomendações mais personalizadas e alinhadas com o perfil de cada utilizador.

Os restantes atributos não foram incluídos porque não apresentam relevância direta para a identificação de padrões de preferência turística, sendo em muitos casos redundantes ou pouco informativos. Dessa forma, a sua inclusão não contribuiria para a geração de regras de associação significativas nem para a personalização das recomendações.

3.6 Desenho da Solução

Como já referido, no ponto central desta dissertação encontra-se o sistema de recomendação, responsável por oferecer sugestões de POI a turistas, tanto individualmente quanto em grupo.

Para descrever o sistema de recomendação, é útil recorrer aos conceitos discutidos no Capítulo 2 e adaptá-los a este contexto específico, garantindo que a solução proposta se apoie nas melhores práticas identificadas.

Em relação à recolha de feedback, o sistema utiliza os dois métodos:

- **Métodos Explícitos:** utiliza a classificação direta dos utilizadores aos POI visitados e sugeridos nas listas de recomendação;
- **Métodos Implícitos:** utiliza a personalidade para calcular as preferências do utilizador. Os jogos sérios integrados ajustam dinamicamente os valores dos parâmetros de personalidade de cada utilizador, refinando o perfil de interesses ao longo do tempo.

Para além disso, o sistema é híbrido, combinando técnicas de filtragem baseadas em **conteúdo** e **colaborativas**.

As técnicas baseadas em conteúdo utilizaram os atributos dos POI e os perfis dos utilizadores, permitindo recomendar POI semelhantes àqueles que o utilizador já classificou ou demonstra preferência. No âmbito deste trabalho, essa semelhança é determinada principalmente pela **categoria turística** a que pertencem os POI.

As técnicas de filtragem colaborativa identificam padrões de comportamento entre utilizadores, reconhecendo semelhanças de perfil e preferências para gerar recomendações personalizadas com base na **experiência de outros utilizadores**.

As regras de associação reforçam esta componente colaborativa, ao minerar padrões entre os utilizadores e destacar relações relevantes.

Combinando estas características, o sistema fornece recomendações mais precisas e adaptadas, aproveitando a informação explícita e implícita dos utilizadores e atenuando as limitações inerentes a cada técnica individual.

3.6.1 Mecanismo de Geração de Regras de Associação

Começando pela mineração das regras de associação, o processo inicia-se quando o *RecommendationController* invoca o *AssociationRulesService* com as configurações, que incluem suporte, confiança, limite por *cluster* e tipo de algoritmo.

O *AssociationRulesService* escolhe dinamicamente qual serviço de mineração utilizar (*AprioriService*, *EclatService* ou *FP-GrowthService*) com base no parâmetro *AlgorithmType*.

Dependendo do algoritmo selecionado, cada um dos três algoritmos possui um serviço independente responsável por gerar os conjuntos frequentes de acordo com a sua lógica específica. Na Figura 3.10 é apresentado o exemplo do serviço para o algoritmo *FP-Growth*.

3.6. Desenho da Solução

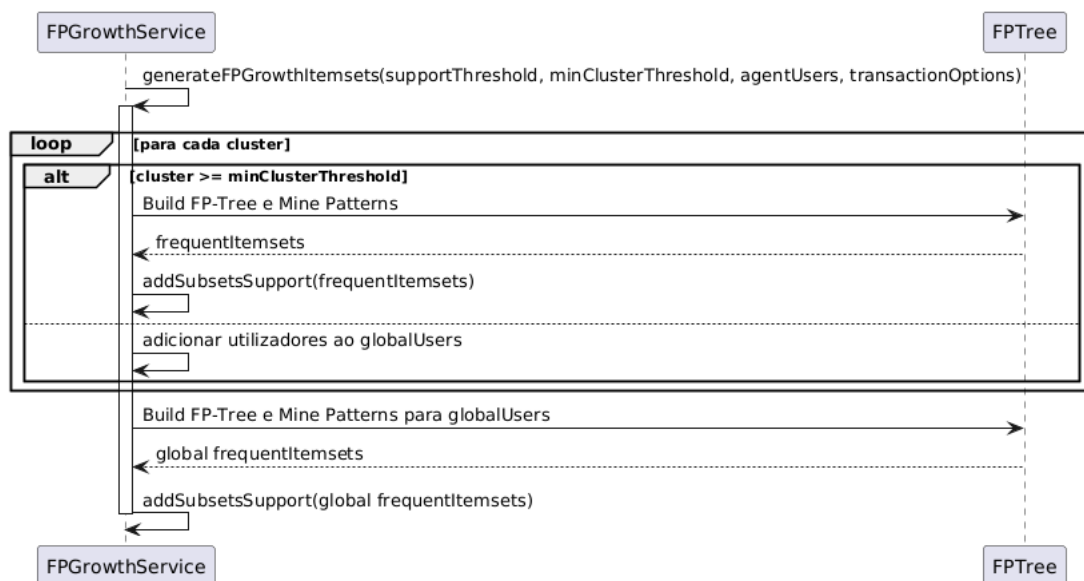


Figura 3.10: Diagrama de sequência da geração de conjuntos frequentes pelo algoritmo *FP-Growth*.

O processo inicia-se quando o *FPGrowthService* invoca o método *generateFPGrowthItemsets*, recebendo como parâmetros o suporte mínimo, o limite mínimo de utilizadores por *cluster*, a lista de utilizadores e as opções de filtragem de transações. Por defeito, o suporte mínimo está definido como 75%, o limite mínimo de utilizadores é de dez (10) e todos os tipos de transação encontram-se ativados.

Para cada *cluster*, verifica-se se o número de utilizadores é superior ao limite mínimo. Caso afirmativo, constrói-se uma *FP-Tree* e procede-se à mineração de padrões frequentes, aplicando posteriormente o cálculo de suporte dos subconjuntos. Se o *cluster* não atingir o limiar mínimo, os utilizadores são adicionados à lista de *globalUsers*.

Após processar todos os *clusters*, é gerada a *FP-Tree* e os padrões frequentes para os *globalUsers*, calculando também os subconjuntos correspondentes. O resultado final consiste no conjunto de itens frequentes por *cluster* e para os utilizadores globais, que é utilizado pelo serviço de regras de associação.

Em seguida, o *AssociationRulesService* cria os objetos *RecommendationRule* a partir desses conjuntos frequentes e devolve ao *RecommendationController* a lista final de regras de associação, pronta para ser aplicada às preferências dos utilizadores.

Este processo é ilustrado pelo diagrama de sequência da Figura 3.11.

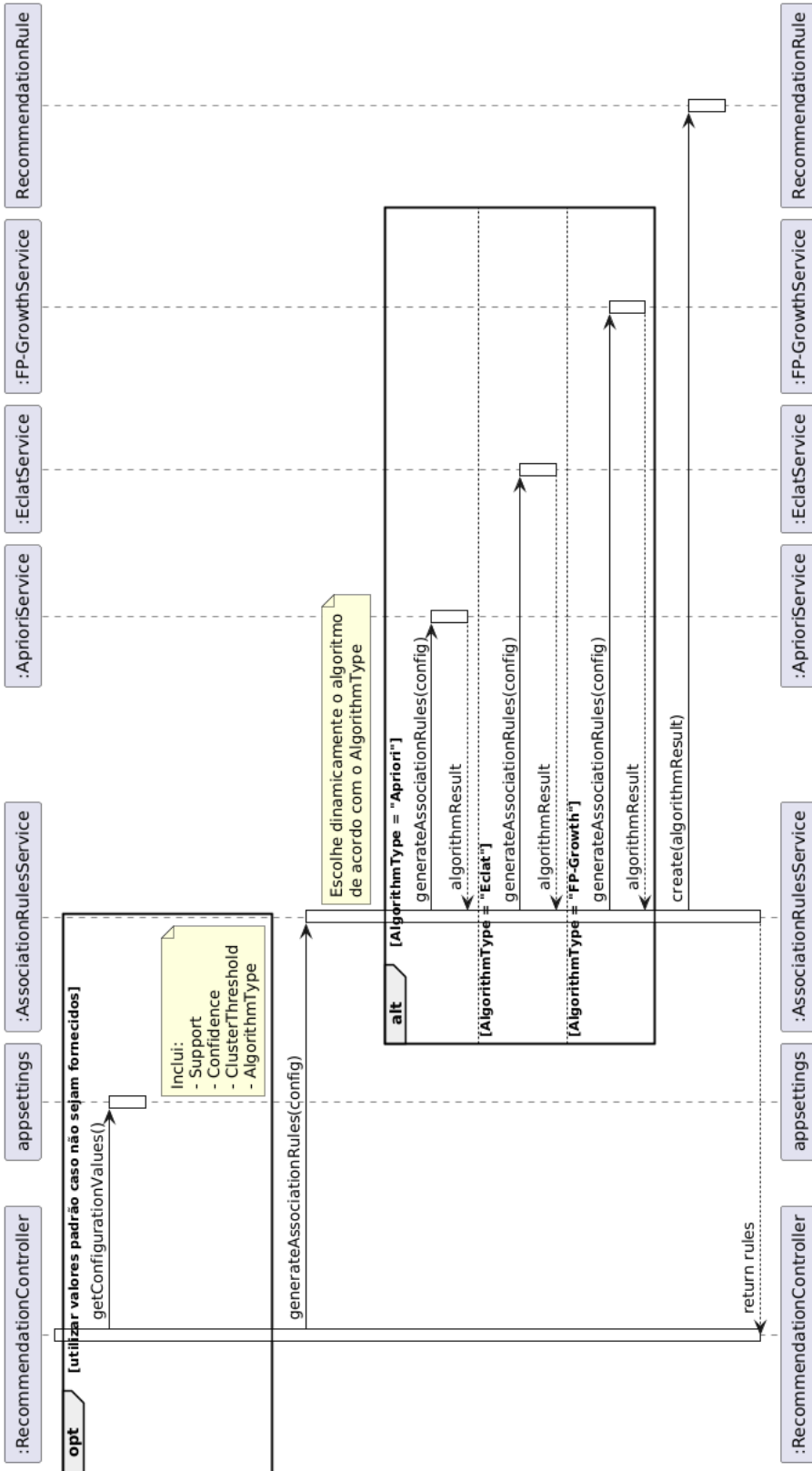


Figura 3.11: Diagrama de sequência de geração de regras de associação.

3.6. Desenho da Solução

Para as recomendações individuais, o processo inicia-se com a autenticação do Turista na aplicação, seguida pelo seu pedido de recomendações de POI. Em seguida, o REMS utiliza os dados do utilizador fornecidos pelo UMMS, os dados dos POI provenientes do POIMS e as regras de associação calculadas pelo MAMS para gerar a lista de recomendações. Esta lista é então retornada para a aplicação, permitindo que o utilizador visualize e interaja com as sugestões. O fluxo detalhado encontra-se ilustrado no diagrama da Figura 3.12.

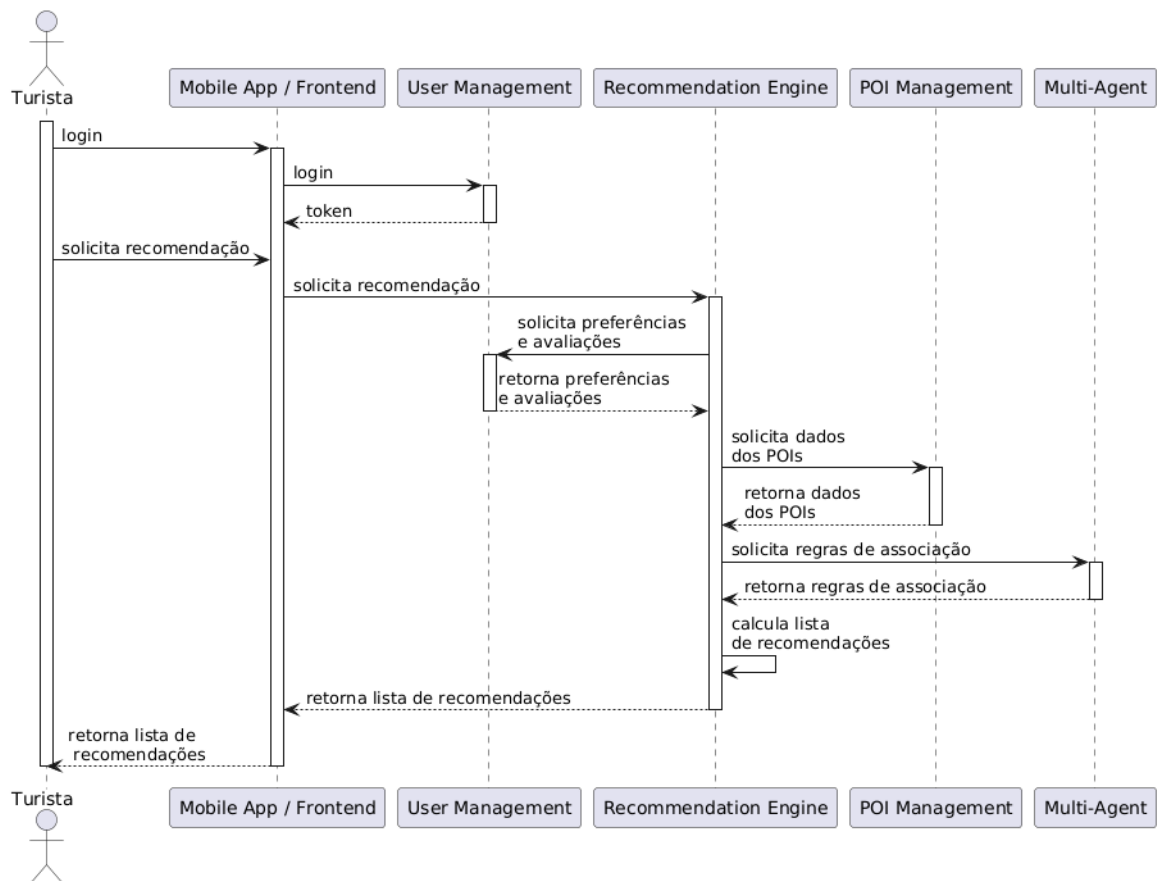


Figura 3.12: Diagrama de sequência para recomendações individuais.

No caso das recomendações em grupo, o processo inicia-se com o Turista Líder a autenticar-se na aplicação e a criar o grupo, utilizando o SNMS. O MAMS é então acionado para atualizar os agentes do grupo, os *clusters*, bem como a subdivisão em subgrupos. Posteriormente, o REMS utiliza estes dados agregados, os dados dos POI fornecidos pelo POIMS e as regras de associação calculadas pelo MAMS para gerar a lista de recomendações adaptadas ao grupo. Esta lista é devolvida à aplicação, permitindo que os membros do grupo visualizem e explorem as sugestões de forma colaborativa. O fluxo detalhado encontra-se ilustrado no diagrama da Figura 3.13.

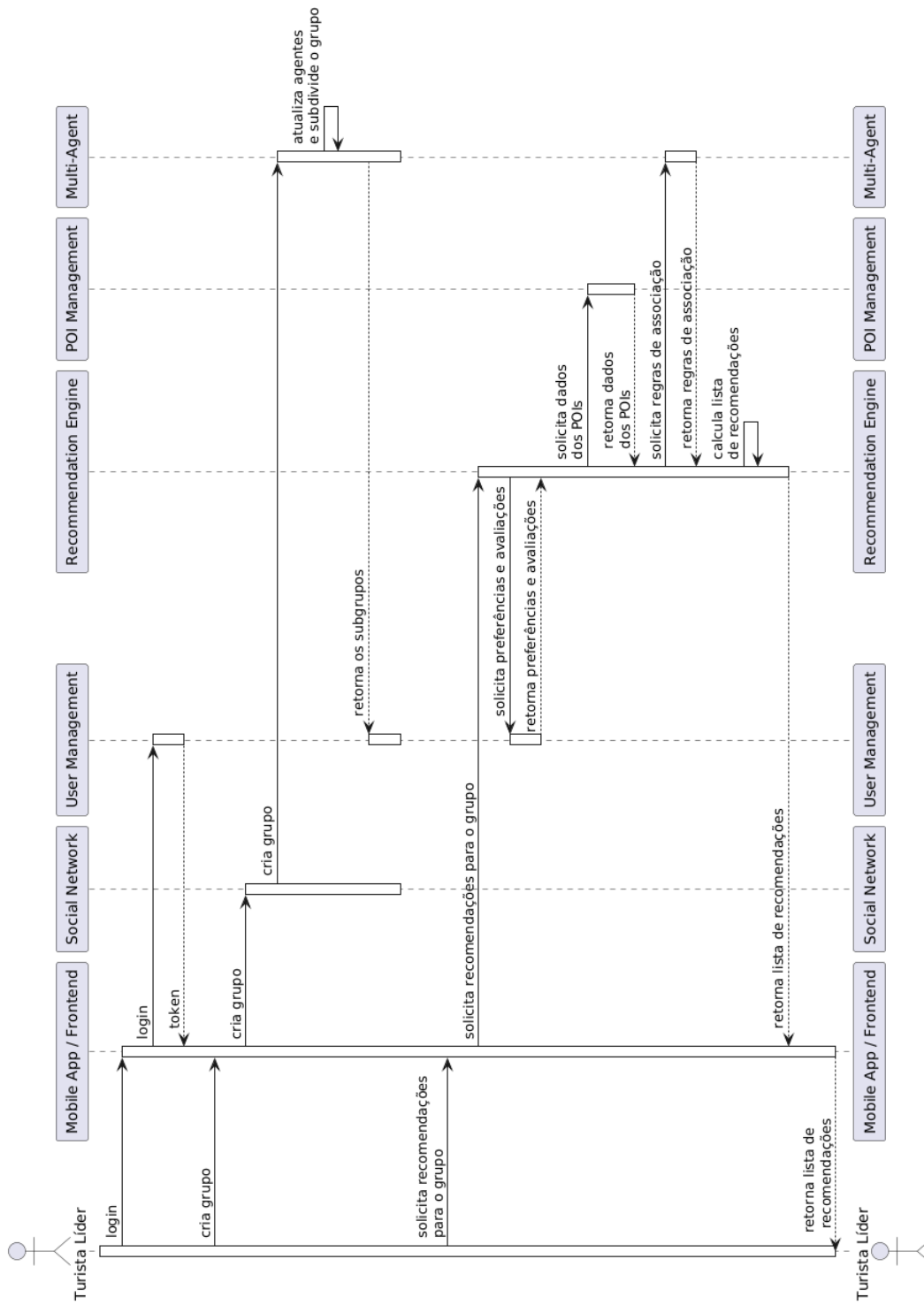


Figura 3.13: Diagrama de sequência para recomendações para grupos.

3.6.2 Integração dos Dados de Contexto

Os dados de contexto foram incluídos no GRS com o objetivo de aumentar o realismo e a aplicabilidade prática das recomendações apresentadas ao utilizador.

Enquanto as pontuações atribuídas às categorias de turismo e as características individuais fornecem uma base sólida para identificar potenciais POI a recomendar, os fatores contextuais acrescentam um nível adicional de profundidade ao processo de recomendação.

No âmbito do turismo, optou-se pela integração de variáveis como condições meteorológicas, qualidade do ar e nível de ruído, uma vez que estas influenciam diretamente a viabilidade e a atratividade de determinadas atividades, permitindo ajustar as recomendações às circunstâncias concretas em que a experiência ocorre.

Ou seja, a integração de dados de contexto foi concebida não como uma substituição das preferências individuais, mas como um complemento que assegura recomendações mais realistas, adaptativas e alinhadas com as condições reais de utilização.

Estes dados são obtidos através de API externas, tal como detalhado no fluxo do diagrama da Figura 3.14.

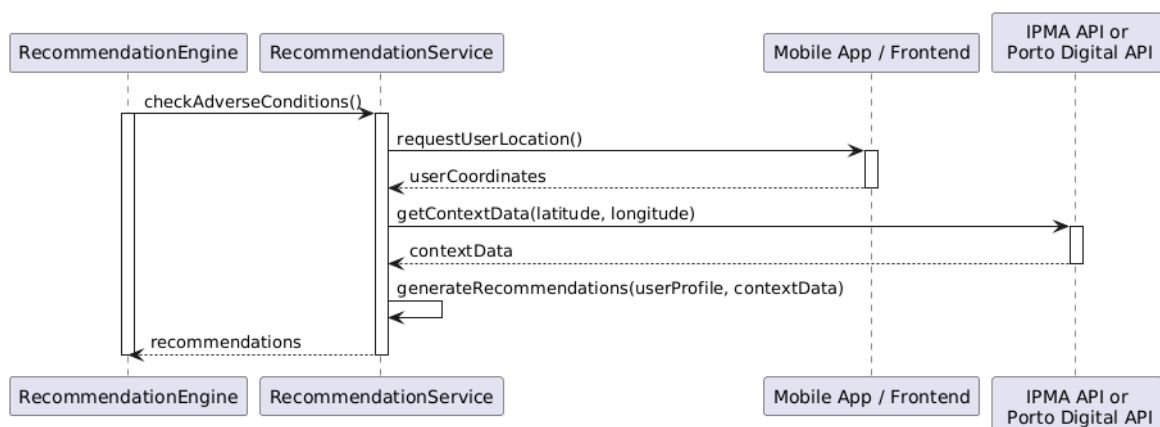


Figura 3.14: Diagrama de sequência para integração de dados de contexto.

No entanto, é essencial compreender de que forma são aplicados no processo de recomendação. Para esse efeito, foram definidos os cenários em que estes fatores contextuais devem ser integrados na aplicação.

3.6.3 Adaptação Dinâmica a Condições Meteorológicas Adversas

Esta funcionalidade tem como objetivo **adaptar dinamicamente as recomendações às condições meteorológicas**, garantindo a relevância e a utilidade das sugestões apresentadas.

Considerando a localização do dispositivo em **tempo real**, caso o utilizador se encontre no exterior e exista previsão de condições meteorológicas adversas (chuva, tempestade, vento forte, entre outras) para o intervalo de tempo considerado por defeito, o sistema notifica o utilizador e apresenta POI interiores como alternativa.

Se não existirem POI interiores no seu top-3 categorias de turismo preferidas, é apresentada uma lista geral de alternativas.

Independentemente do cenário, a decisão final sobre aceitar ou não a sugestão adaptada permanece sempre com o utilizador, garantindo a autonomia e a personalização da experiência.

Os dados relevantes obtidos pela API do IPMA [43] são os de temperatura, probabilidade de precipitação, velocidade do vento ou ainda avisos meteorológicos, que incluem o tipo de aviso e o nível de alerta (amarelo, laranja ou vermelho) [43].

O processo encontra-se representado graficamente pelo fluxograma da Figura 3.15.

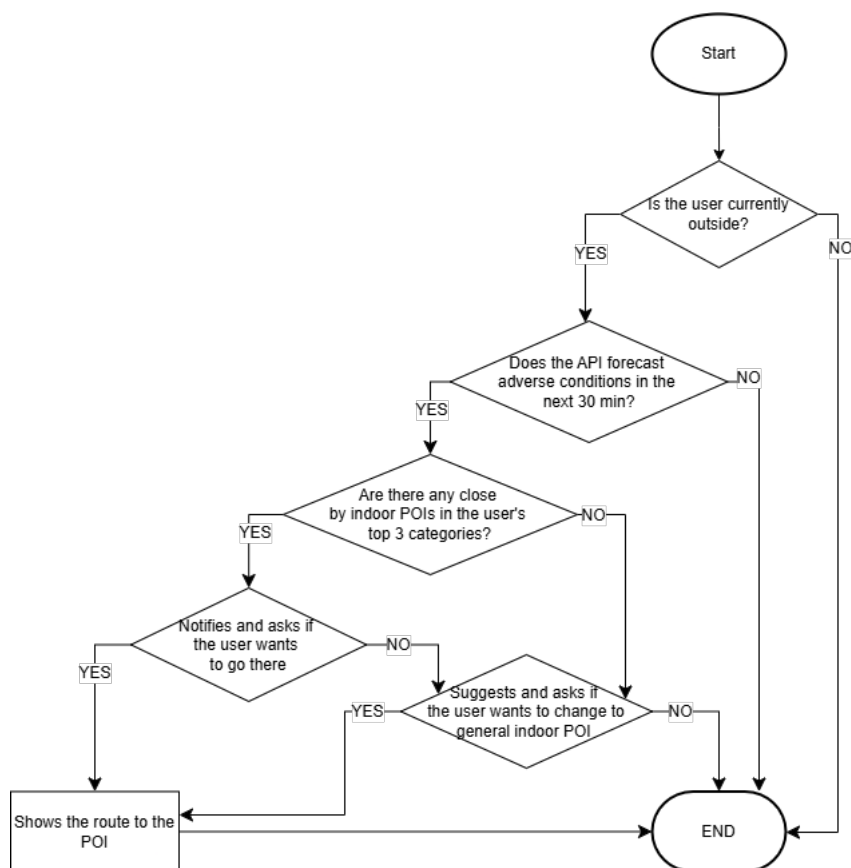


Figura 3.15: Fluxograma de adaptação dinâmica a condições meteorológicas adversas.

3.6.4 Previsão de Condições Meteorológicas Adversas para Excursões

Este processo segue a lógica do anterior. No GRS, as viagens, individuais ou em grupo, são agendadas para uma data escolhida pelo utilizador. O objetivo desta funcionalidade é **garantir que as recomendações permanecem relevantes e viáveis mesmo perante condições meteorológicas adversas**.

Caso seja prevista a ocorrência de condições desfavoráveis nessa data, o sistema procede à verificação da lista de POI recomendados. Se forem identificados POI classificados como **exteriores**, o utilizador é notificado.

Este processo encontra-se representado graficamente pelo fluxograma da Figura 3.16.

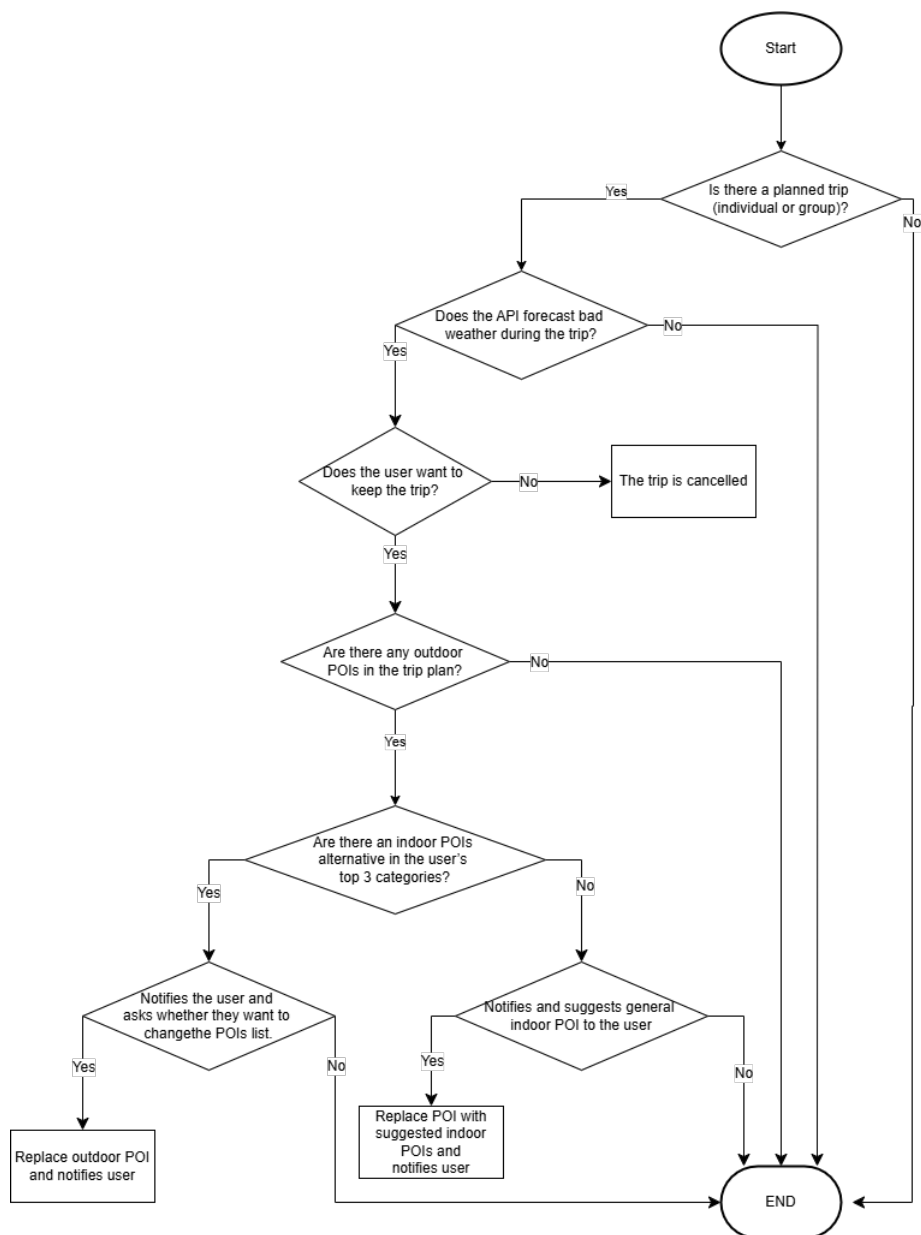


Figura 3.16: Fluxograma de decisão para previsão de condições meteorológicas adversas para excursões.

Se existirem POI **interiores** dentro do seu top-3 categorias de turismo preferidas, é-lhe sugerida a substituição. Caso não existam POI interiores nesse conjunto, é apresentada uma lista alternativa geral.

Tal como no caso anterior, a decisão final de manter ou alterar a lista de POI cabe sempre ao utilizador.

As condições desfavoráveis são avaliadas com base nos atributos relevantes também definidos no caso anterior, obtidos pela API do IPMA [43].

3.6.5 Adaptação Dinâmica à Qualidade do Ar e Nível de Ruído

Esta funcionalidade também tem como objetivo **adaptar dinamicamente as recomendações às condições ambientais**, garantindo a segurança, conforto e relevância das sugestões apresentadas.

Considerando a localização do dispositivo em tempo real, caso se verifique que a qualidade do ar ou os níveis de ruído são prejudiciais, o sistema procede à adaptação da lista de POI recomendados.

O sistema procura POI fora da área considerada prejudicial e que pertençam ao top-3 categorias de turismo preferidas do utilizador. O utilizador é notificado e questionado sobre se deseja alterar o seu percurso. Caso não existam POI adequados dentro do top-3 categorias de turismo preferidas, é apresentada uma lista alternativa geral.

Independentemente do cenário, a decisão final sobre aceitar ou não a sugestão adaptada permanece sempre com o utilizador, garantindo a autonomia e a personalização da experiência.

Os dados relevantes obtidos pela API da Porto Digital [44, 45] incluem as concentrações dos diferentes parâmetros no ar (O_2 , SO_2 , NO_2 , PM_{10} e $PM_{2.5}$), bem como os níveis de ruído obtidos em decibéis.

O processo encontra-se representado graficamente pelo fluxograma da Figura 3.17.

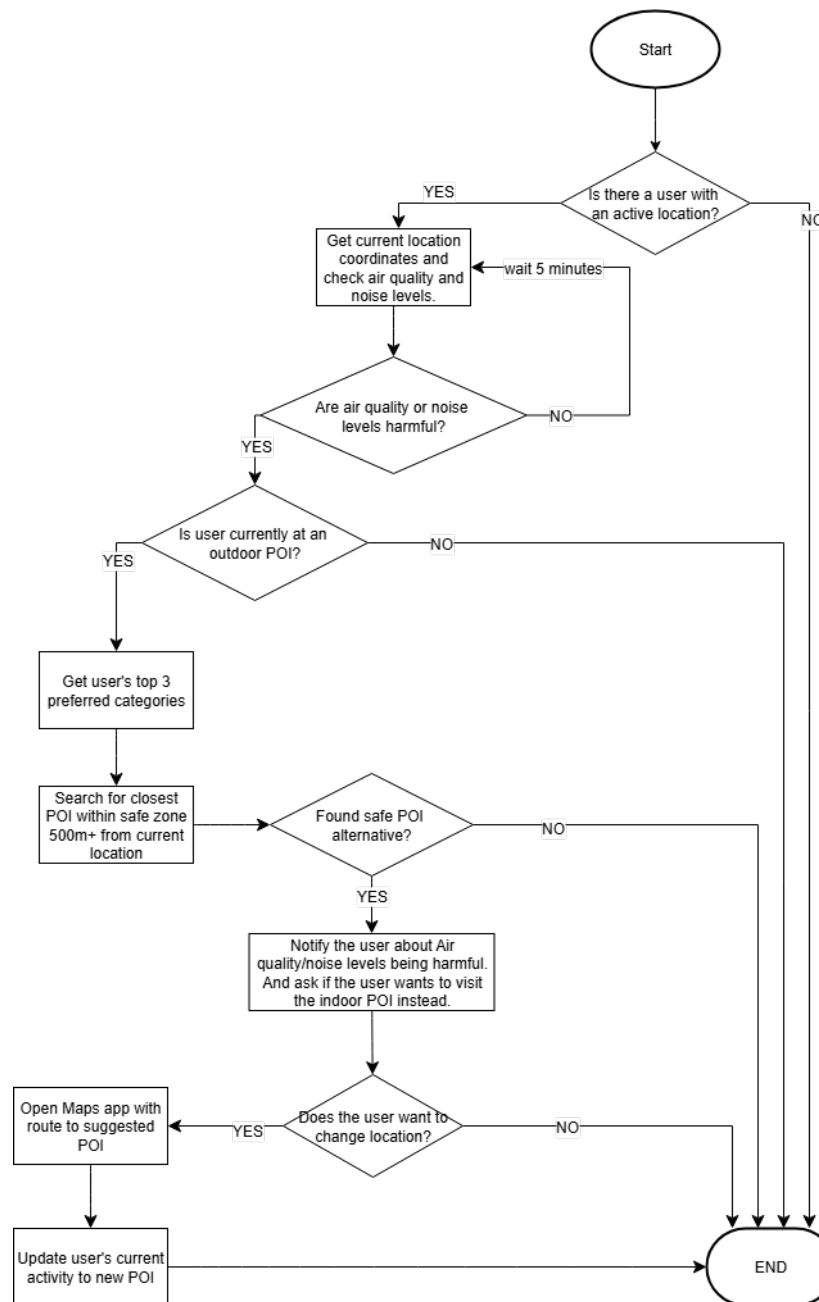


Figura 3.17: Fluxograma de adaptação dinâmica à qualidade do ar e nível de ruído.

Capítulo 4

Implementação da Solução

O objetivo deste capítulo é descrever o processo de implementação, abrangendo tanto a mineração de regras de associação como a geração de recomendações baseadas nos novos dados de contexto mencionados no capítulo anterior.

Inicialmente, apresenta-se a implementação dos algoritmos *Apriori*, *Eclat* e *FP-Growth* para a geração de regras de associação.

De seguida, detalha-se a lógica de recomendação, distinguindo recomendações individuais e para grupos, verificando também o ajuste ao contexto ambiental e situacional, evidenciando como o sistema assegura sugestões relevantes e consistentes para cada utilizador ou grupo.

4.1 Mineração de Regras de Associação

O mecanismo de mineração de regras de associação foi implementado no MAMS. Como os agentes são atualizados em conjunto com o UMMS, num intervalo regular de 24 horas, tanto as informações dos utilizadores como a composição dos *clusters* refletem sempre o estado mais recente. Em consequência, as regras de associação extraídas acompanham automaticamente estas atualizações, garantindo que o sistema dispõe de conhecimento continuamente ajustado à realidade atual dos perfis de utilizadores.

Antes de aplicar qualquer um dos algoritmos de descoberta de padrões, foi criada a classe de apoio, demonstrada no Código 4.1, que permite selecionar os parâmetros a incluir na geração das regras, o que é particularmente relevante para a US-102.

Esta classe possibilita ativar ou desativar individualmente os seguintes parâmetros:

- **Informações Demográficas:** data de nascimento, género, estado civil, filhos, composição familiar, religião, companheiros de viagem, nível de educação, área de formação, situação profissional, rendimento líquido mensal;
- **Preferências de Turismo:** pontuação atribuída às categorias de atrações turísticas, com possibilidade de definir um limite mínimo para considerar uma preferência (por exemplo, 0,6 numa escala de 0 a 1);
- **Limitações e Fobias:** lista de limitações e lista de fobias;

```

1 public class TransactionFilterOptions
2 {
3     public bool IncludeBirthdayBin { get; set; } = true;
4     public bool IncludeGender { get; set; } = true;
5     public bool IncludeCivilState { get; set; } = true;
6     public bool IncludeHasChildren { get; set; } = true;
7     public bool IncludeLiveWith { get; set; } = true;
8     public bool IncludeReligion { get; set; } = true;
9     public bool IncludeTravelCompanion { get; set; } = true;
10    public bool IncludeEducationLevel { get; set; } = true;
11    public bool IncludeFormationArea { get; set; } = true;
12    public bool IncludeProfessionalSituation { get; set; } = true;
13    public bool IncludeIncome { get; set; } = true;
14
15    public bool IncludeAttractionPreferences { get; set; } = true;
16    public double AttractionPreferenceThreshold { get; set; } = 0.6;
17
18    public bool IncludeLimitations { get; set; } = true;
19    public bool IncludeFears { get; set; } = true;
20 }

```

Código 4.1: Classe *TransactionFilterOptions*.

Adicionalmente, os valores de suporte e confiança foram implementados de forma a poderem ser configurados a partir da aplicação cliente. Para tal, no ficheiro *appsettings.json* do MAMS foram criadas as opções *Support* e *Confidence*. Neste mesmo ficheiro é também possível especificar o algoritmo a utilizar através da variável *AlgorithmType*. Este mecanismo centralizado permite ao administrador do sistema ajustar de forma rápida e eficiente o algoritmo de regras e respetivas configurações.

Antes de aplicar qualquer algoritmo de mineração de regras de associação (*Apriori*, *Eclat* ou *FP-Growth*), foi definida uma função para cada algoritmo, com lógica de processamento equivalente: extrair conjuntos frequentes em *clusters* de utilizadores e globalmente para aqueles pertencentes a *clusters* demasiado pequenos. A função para o algoritmo *Apriori*, por exemplo, é *AprioriClusters(users, suporteMin, minClusterThreshold)*, cujo pseudocódigo é apresentado no Código 4.2.

As variáveis utilizadas são:

- *clusters*: lista de *clusters* de utilizadores calculados previamente pelo MAMS;
- *clusterItemsets*: lista onde são armazenados os conjuntos frequentes extraídos de cada *cluster*, juntamente com o seu centroide;
- *globalUsers*: conjunto de utilizadores pertencentes a *clusters* demasiado pequenos para análise independente;
- *globalItemsets*: conjuntos frequentes extraídos do grupo agregado de utilizadores globais.

Esta função recebe como entrada o conjunto de utilizadores já agrupados pelo MAMS (*users*), um valor mínimo de suporte (*suporteMin*) e um limite mínimo de elementos por *cluster* (*minClusterThreshold*).

O objetivo da função é gerar os conjuntos frequentes em dois níveis: dentro de cada *cluster* que tenha dimensão suficiente; e de forma global, agregando os utilizadores pertencentes a *clusters* demasiado pequenos.

4.1. Mineração de Regras de Associação

O resultado retornado consiste em dois conjuntos: *clusterItemsets*, que contém os conjuntos frequentes de cada *cluster* (associados ao respetivo centroide), e *globalItemsets*, que agrega os padrões encontrados no conjunto global de utilizadores.

```
1 FUNCAO AprioriClusters(users , suporteMin , minClusterThreshold):
2   clusters <- calculados pelo MAMS
3   clusterItemsets <- []
4   globalUsers <- []
5
6   PARA cada cluster em clusters:
7     SE tamanho(cluster) >= minClusterThreshold:
8       itemsets <- Apriori(cluster.transactions , suporteMin)
9       adicionar {cluster.centroid , itemsets} a clusterItemsets
10    SENAO:
11      adicionar utilizadores de cluster a globalUsers
12
13   globalItemsets <- Apriori(globalUsers.transactions , suporteMin)
14   RETORNAR {clusterItemsets , globalItemsets}
```

Código 4.2: *Apriori* em clusters e utilizadores globais.

Por fim, cada um dos três algoritmos (*Apriori*, *FP-Growth* e *Eclat*) possui um serviço individual dentro do MAMS, no qual a respetiva lógica foi implementada, garantindo modularidade e flexibilidade na escolha do método de extração de padrões. Estes algoritmos foram adaptados para processar *clusters* de utilizadores, bem como utilizadores que pertençam a *clusters* que não atingem o limite mínimo predefinido de elementos. A implementação destes algoritmos é detalhada na secção seguinte.

4.1.1 Algoritmo Apriori

O algoritmo *Apriori* foi definido na função *Apriori(transactions, suporteMin)*, cujo pseudo-código é apresentado no Código 4.3.

As variáveis utilizadas são:

- *k*: tamanho dos subconjuntos (*itemsets*) considerados em cada iteração;
- *frequentItemsets*: lista global que armazena os *itemsets* frequentes descobertos ao longo do processo;
- *candidatos*: conjunto de subconjuntos candidatos a frequentes, gerados a partir dos itens anteriores;
- *frequentSets*: subconjuntos que satisfazem o valor mínimo de suporte na iteração corrente.

Esta função recebe como entrada o conjunto de transações (*transactions*), que representam as características dos utilizadores, e um valor mínimo de suporte (*suporteMin*), que define o limiar para considerar um conjunto como frequente.

O núcleo do algoritmo gera candidatos de diferentes tamanhos, conta quantas transações contêm cada conjunto de itens e filtra aqueles que não atingem o suporte mínimo. Este processo é repetido até não existirem novos conjuntos frequentes. Por fim, calcula-se o suporte de todos os subconjuntos de cada conjunto frequente, informação necessária para a geração posterior das regras de associação.

O resultado final retornado consiste no conjunto de todos os *frequentItemsets*.

```

1 FUNCAO Apriori(transactions , suporteMin):
2   k <- 1
3   frequentItemsets <- []
4
5   ENQUANTO houver candidatos:
6     candidatos <- gerar subconjuntos de tamanho k
7     frequentSets <- filtrar candidatos com suporte >= suporteMin
8
9     SE frequentSets vazio: parar
10    SENAO:
11      frequentItemsets <- frequentSets
12      k <- k + 1
13
14   calcular suporte dos subconjuntos de cada itemset
15   RETORNAR frequentItemsets

```

Código 4.3: Núcleo do algoritmo *Apriori*.

4.1.2 Algoritmo Eclat

Para o algoritmo *Eclat*, foi definida a função *Eclat(transactions, suporteMin)*, cujo pseudo-código é apresentado no Código 4.4.

As variáveis utilizadas são:

- *k*: tamanho dos subconjuntos (*itemsets*) considerados em cada iteração;
- *itens*: conjunto dos itens únicos presentes nas transações;
- *frequentItemsets*: lista que armazena os conjuntos frequentes descobertos;
- *candidatos*: subconjuntos candidatos a frequentes, gerados a partir dos itens atuais;
- *frequentSets*: subconjuntos que satisfazem o valor mínimo de suporte na iteração corrente;
- *support*: valor do suporte calculado para cada subconjunto.

Esta função recebe como entrada o conjunto de transações (*transactions*), que representam as características dos utilizadores, e um valor mínimo de suporte (*suporteMin*), que define o limiar para considerar um conjunto como frequente.

Na base do algoritmo, são construídas listas verticais de transações para cada item, que são intersecadas para gerar conjuntos de itens de tamanho crescente e calcular o suporte de cada conjunto. Apenas os conjuntos que atingem o suporte mínimo são mantidos. Por fim, é calculado o suporte de todos os subconjuntos de cada conjunto frequente, informação necessária para a geração posterior das regras de associação.

O resultado final retornado consiste no conjunto de todos os *frequentItemsets*, que poderão ser utilizados posteriormente para a geração de regras de associação.

4.1. Mineração de Regras de Associação

```
1 FUNCAO Eclat(transactions , suporteMin):
2   k <- 1
3   itens <- itens unicos de transactions
4   frequentItemsets <- []
5
6   ENQUANTO existirem candidatos:
7     candidatos <- gerar subconjuntos de tamanho k de itens
8     frequentSets <- []
9     PARA cada candidato em candidatos:
10      suporte <- calcular suporte(candidato , transactions)
11      SE suporte >= suporteMin:
12        adicionar candidato a frequentSets
13      SE frequentSets vazio:
14        PARAR
15      frequentItemsets <- frequentSets
16      itens <- itens unicos em frequentSets
17      k <- k + 1
18
19   PARA cada itemset em frequentItemsets:
20     gerar subconjuntos nao vazios
21     calcular suporte de cada subconjunto
22
23   RETORNAR frequentItemsets
```

Código 4.4: Núcleo do algoritmo *Eclat*.

4.1.3 Algoritmo FP-Growth

O algoritmo FP-Growth constrói uma *FP-Tree* que representa de forma compacta todos os itens presentes nas transações. A mineração é realizada percorrendo a árvore, extraindo os conjuntos frequentes a partir dos caminhos e calculando o suporte de cada conjunto. Apenas os conjuntos que atingem o suporte mínimo são mantidos.

Para mineração da árvore, foi definida a função *MineFPTree(transactions, suporteMin)*, cujo pseudocódigo é apresentado no Código 4.5.

As variáveis utilizadas são:

- *transactions*: conjunto de transações que representam as características dos utilizadores;
- *frequentItemsets*: lista que armazena os conjuntos frequentes descobertos;
- *supportMin*: valor mínimo de suporte utilizado para filtrar os conjuntos frequentes.

O resultado final retornado consiste no conjunto de todos os *frequentItemsets*, que poderão ser utilizados posteriormente para a geração de regras de associação.

```
1 FUNCAO MineFPTree(transactions , suporteMin):
2   calcular frequencia de cada item
3   remover itens abaixo do suporte minimo
4   construir FP-Tree a partir das transactions ordenadas
5   percorrer a FP-Tree para extrair todos os itemsets frequentes
6   RETORNAR frequentItemsets
```

Código 4.5: Núcleo do algoritmo *FP-Growth*.

4.1.4 Extração de Regras a Partir de Conjuntos Frequentes

Apesar das divergências, cada um dos algoritmos termina a execução e retorna duas listas de conjuntos frequentes por si calculadas: uma para os *clusters* e outra para os utilizadores globais. Com estas listas, é possível calcular as regras de associação, conforme definido na US-101, determinando os antecedentes e consequentes de cada conjunto de itens e o seu nível de confiança.

A geração de regras por *cluster* é feita percorrendo cada conjunto frequente de cada *cluster* e considerando cada um dos seus subconjuntos. Para cada subconjunto, calcula-se o consequente (os itens que não pertencem ao subconjunto) e a confiança da regra, obtida pela razão entre o suporte do *itemset* completo e o suporte do subconjunto. Apenas as regras cuja confiança atinja ou ultrapasse o limiar predefinido são mantidas. O Código 4.6 ilustra a lógica de geração das regras a partir dos conjuntos frequentes por *cluster*.

```

1 private static List<AssociationRule> generateAssociationRulesCluster(
2     List<ClusterItemset> clusterItemsets, double confidenceThreshold)
3     {
4         var associationRules = new List<AssociationRule>();
5
6         // Iterate through each cluster's frequent itemsets
7         foreach (var clusterItemset in clusterItemsets)
8         {
9             foreach (var itemsetInfo in clusterItemset.frequentItemsets)
10            {
11                var itemset = itemsetInfo.Itemset;
12                var subsetsSupport = itemsetInfo.SubsetsSupport;
13
14                // Generate rules by iterating through subsets of the itemset
15                foreach (var (antecedent, supportAntecedent) in subsets)
16                {
17                    var consequent = itemset.Except(antecedent).ToList();
18
19                    var supportItemset = itemsetInfo.Support;
20                    var confidence = supportItemset / supportAntecedent;
21
22                    // Add the rule if it meets the confidence threshold
23                    if (confidence >= confidenceThreshold)
24                    {
25                        associationRules.Add(new AssociationRule
26                        {
27                            centroid = clusterItemset.clusterCentroid,
28                            antecedent = antecedent,
29                            consequent = consequent,
30                            confidence = confidence
31                        });
32                    }
33                }
34            }
35        }
36        return associationRules;
37    }
38 }

```

Código 4.6: Geração de regras a partir dos conjuntos frequentes por *Cluster*.

O mesmo processo aplica-se aos utilizadores globais, garantindo que regras de associação também sejam extraídas para aqueles que não pertencem a *clusters* suficientemente grandes. A diferença é que, neste caso, a regra é gravada sem associar um valor de *centroid*, visto que não se aplica a um *cluster* específico.

4.1.5 Exportação das Regras

As regras de associação devem ser atualizadas diariamente, de acordo com um intervalo pré-definido de 24 horas. Caso seja necessário forçar a geração de regras fora desse intervalo, foi criado um *endpoint* **POST /generateAssociationRules** no MAMS, que recebe um JSON com a configuração desejada, equivalente ao apresentado no Código 4.7. Desta forma, é possível alterar os valores de suporte, confiança, limite de utilizadores por *cluster* e ativar/desativar as diferentes características dos utilizadores. Na ausência deste ficheiro de configuração, a geração das regras é realizada normalmente, utilizando os valores pré-definidos.

```
1 {
2   "Support": 0.2,
3   "Confidence": 0.6,
4   "MinClusterThreshold": 3,
5   "AlgorithmType": "Apriori",
6   "TransactionOptions": {
7     "IncludeBirthdayBin": true,
8     "IncludeGender": true,
9     "IncludeCivilState": true,
10    "IncludeHasChildren": true,
11    "IncludeLiveWith": true,
12    "IncludeReligion": true,
13    "IncludeTravelCompanion": true,
14    "IncludeEducationLevel": true,
15    "IncludeFormationArea": true,
16    "IncludeProfessionalSituation": true,
17    "IncludeIncome": true,
18    "IncludeAttractionPreferences": true,
19    "AttractionPreferenceThreshold": 0.6,
20    "IncludeLimitations": true,
21    "IncludeFears": true
22  }
23 }
```

Código 4.7: Estrutura do ficheiro JSON para gerar regras de associação.

Por serem tão voláteis, persistir as regras na base de dados não seria adequado. Em vez disso, como o MAMS disponibiliza um ambiente para armazenamento de informação, as regras são guardadas diretamente nesse ambiente. Quando solicitado pelo REMS, o MAMS é capaz de enviar as regras para que sejam utilizadas na construção das recomendações.

Contudo, para verificar o funcionamento dos algoritmos e ainda para futuros testes de qualidade, em concordância com a US-103, foi criado um método que gera um ficheiro JSON que inclui o nome do algoritmo utilizado para a mineração, as regras de *cluster* associadas ao mesmo, e as regras globais. Cada regra é ainda dividida em antecedente e conseqüente, e tem associado a si o valor de confiança. O ficheiro segue a estrutura detalhada pelo Código 4.8.

```

1 {
2   "AlgorithmUsed": "ALGORITHM_NAME" ,
3   "ByCluster": {
4     "CLUSTER_ID_1": [
5       "{ITEM_A, ITEM_B} => {ITEM_C}    CONFIDENCE_VALUE" ,
6       "{ITEM_D, ITEM_E} => {ITEM_F}    CONFIDENCE_VALUE"
7     ],
8     "CLUSTER_ID_2": [
9       "{ITEM_G} => {ITEM_H}    CONFIDENCE_VALUE"
10    ]
11  },
12  "Global": [
13    "{ITEM_I, ITEM_J} => {ITEM_K}    CONFIDENCE_VALUE"
14  ]
15 }

```

Código 4.8: Estrutura do ficheiro JSON com as regras de associação.

Como o REMS também necessita aceder às regras para gerar recomendações, foi adotado o formato JSON, compatível com os outros serviços do sistema. Para tal, foi criado um *endpoint* **GET /getAssociationRules** no MAMS, através do qual o REMS pode consultar as regras de associação.

4.2 Geração de Recomendações

O algoritmo de recomendação estava implementado no seu próprio microserviço, denominado REMS. O objetivo principal deste serviço é sugerir POI adequados, tanto para utilizadores individuais como para grupos de turismo.

Para tal, o REMS integra informação proveniente dos restantes microserviços, em conformidade com o que já foi apresentado nos diagramas das Figuras 3.12 e 3.13 no capítulo anterior.

No início do projeto, o código encontrava-se em processo de migração para o servidor do GECAD. Paralelamente, foram realizadas melhorias de desempenho, tendo sido necessário apoiar na implementação destas alterações. As otimizações incidiram principalmente na integração das regras de associação para influenciar as recomendações, assim como na gestão dos dados de contexto.

4.2.1 Recomendações Individuais

O processo de recomendação envolve diferentes métodos que são explicados de seguida.

O primeiro passo consiste em obter os medos, limitações e preferências turísticas do utilizador. As preferências turísticas correspondem à pontuação prevista para as onze categorias de turismo, listadas anteriormente na secção "**Pontuação para as categorias de turismo**", no Subcapítulo 3.5.

Estas preferências são estimadas com base nas suas dimensões de personalidade, utilizando modelos de equações estruturais descritos em Alves et al [4].

Cada categoria F_j tem associada uma função de preferência do tipo:

$$\text{Preferencia}_{u,F_j} = \sum_{t \in T} \beta_{F_j,t} \cdot P_{u,t}$$

onde $P_{u,t}$ representa o valor da dimensão de personalidade t do utilizador u (extraversão, agradabilidade, conscienciosidade, neuroticismo, abertura à experiência) e $\beta_{F_j,t}$ indica o peso de cada dimensão na preferência pela categoria F_j . Através deste cálculo, obtêm-se as preferências **preliminares** do utilizador.

Quando se verifica que o utilizador é novo, isto é, num cenário de *cold-start*, o sistema recorre a uma *flag* para detetar se todas as suas **preferências apresentam valor nulo**. Se assim for, as preferências preliminares do novo utilizador são **estimadas a partir da média das preferências do cluster** em que este é integrado.

Posteriormente, com as regras de associação fornecidas pelo MAMS, é feito um ajuste aos valores das preferências:

1. Selecionam-se as regras do *cluster* do utilizador (caso não existam, utilizam-se as regras globais);
2. Filtram-se apenas as regras do tipo "LIKES" ou "DISLIKES_ATTRACTION_TYPE", que estão relacionadas com as preferências turísticas;
3. Cada regra é aplicada caso aconteça um dos seguintes cenários: o utilizador esteja sinalizado como *cold-start*; a regra não possua antecedente; ou caso todos os antecedentes das regras estejam presentes nas preferências **preliminares** do utilizador;
4. Caso seja "LIKES", a avaliação da categoria aumenta, fazendo um valor ponderado entre o valor existente e o valor de confiança da regra; caso seja "DISLIKES", o mesmo processo é aplicado mas com o sinal invertido;
5. O processo termina com uma matriz atualizada das categorias de turismo e das respetivas pontuações, ordenada de forma decrescente, refletindo a relevância de cada categoria para o utilizador. A estrutura desta matriz segue a apresentada na Tabela 4.1, apresentada mais à frente.

Este processo também se encontra ilustrado pelo fluxograma da Figura 4.1.

O fator de ponderação foi definido como 30%. Este valor foi escolhido de forma heurística, de modo a equilibrar a influência das preferências originais do utilizador com as regras de associação extraídas. Um **valor demasiado baixo reduziria o impacto das regras**, limitando o contributo da componente colaborativa no processo de recomendação. Por outro lado, um **valor demasiado elevado faria com que as regras dominassem as preferências preliminares**, levando a recomendações menos personalizadas.

Assim, o valor de 30% garante que as regras influenciam as preferências do utilizador sem sobrepor totalmente o seu perfil inicial, funcionando como um compromisso entre informação explícita e inferida. No entanto, devido à centralização, este parâmetro pode facilmente ser atualizado caso necessário.

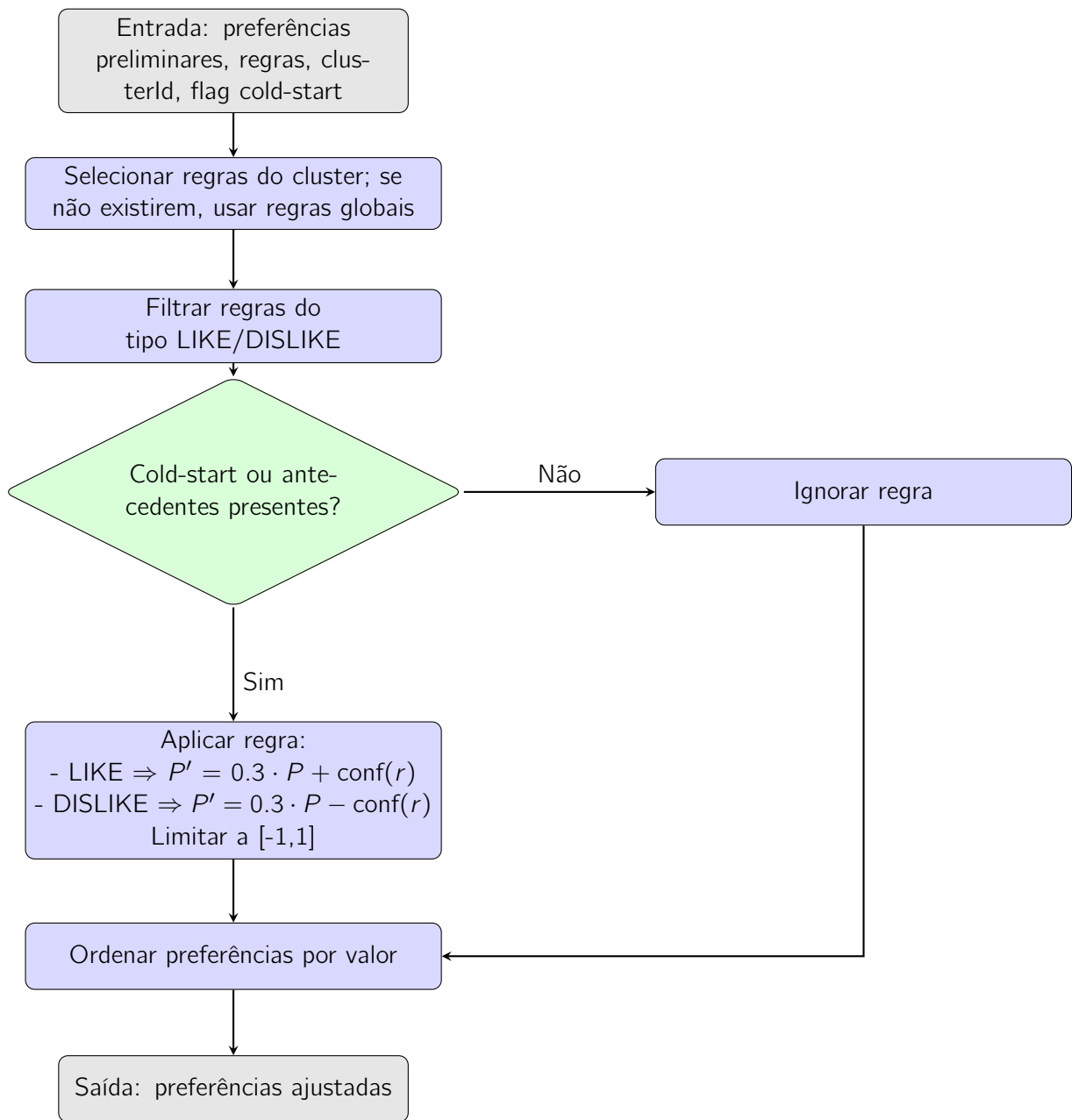


Figura 4.1: Fluxo de aplicação das regras de associação às preferências.

Com a matriz equivalente à da Tabela 4.1 e com base na fórmula definida por Alves et al. [4], calcula-se quantos POI de cada categoria devem ser recomendados, utilizando uma abordagem proporcional baseada nas pontuações de preferência do utilizador. As categorias com pontuações mais altas recebem mais recomendações, enquanto as menos preferidas recebem menos. O método garante que o total de recomendações seja igual ao número de POI pré-definido ($N_{POI} = 10$), arredondando os valores e distribuindo os restantes de forma justa.

4.2. Geração de Recomendações

A fórmula utilizada é a seguinte:

$$NF_j = \text{round.up} \left(\frac{APC_q SgF_j}{\sum_{k=1}^3 APC_q SgF_k} \cdot N_{POI} \right)$$

onde NF_j representa o número de POI a recomendar para a categoria F_j , $APC_q SgF_j$ é a pontuação prevista para o top-3 categorias de turismo preferidas do utilizador, e N_{POI} é o número total de POI a recomendar.

As categorias com pontuações mais altas recebem mais recomendações, enquanto as menos preferidas recebem menos. O método garante que o total de recomendações seja igual ao número de POI pré-definido (dez) para a recomendação.

Tabela 4.1: Matriz de categorias de turismo e respetivas avaliações de um utilizador.

Categoria	Avaliação
Sun, Water & Sand	0,94
Theme & Animal Parks	0,83
Adrenaline Activities	0,81
Wild Nature Activities	0,79
Natural Phenomena	0,79
Gastronomy Events	0,78
Health & Well-Being	0,76
Cultural Heritage	0,75
Museums, Boat trips & Viewpoints	0,72
Sports & Games	0,71
Party, Music & Nightlife	0,32

Depois de identificar todos os POI elegíveis a partir do POIMS, aplica-se um processo de filtragem que remove aqueles que **colidam com os medos ou limitações do utilizador, bem como os POI já visitados ou avaliados negativamente**. Esta etapa corresponde a uma **filtragem baseada em restrições**, garantindo que apenas POI adequados às preferências, limitações e histórico do utilizador avancem para a fase de recomendação.

Entre estes, são favorecidos aqueles com **maior média de avaliação e número de avaliações**, assegurando recomendações de qualidade e alinhadas às preferências globais. Nesta seleção final, são considerados tanto os atributos dos POI como as avaliações de outros utilizadores, incorporando assim os princípios da **filtragem baseada em conteúdo e colaborativa**.

No final, é ainda gerada uma lista alternativa de recomendações, com o objetivo de apoiar a adaptação ao contexto, conforme detalhado mais à frente. Esta lista corresponde à

continuação da lista original de recomendações, que foi truncada para o número de POI pré-definido (dez). Como a lista se encontra ordenada por ordem decrescente de relevância, esta abordagem garante consistência e aproveitamento lógico das recomendações excedentes.

O REMS tem acesso a base de dados própria, onde estas listas são persistidas juntamente com o identificador do utilizador e a compatibilidade de cada recomendação.

4.2.2 Recomendações para Grupos

O algoritmo de recomendação para grupos aplica a lógica utilizada nas recomendações individuais, mas considera as preferências agregadas de múltiplos utilizadores.

Os grupos são formados pelo turista líder e não existem restrições quanto ao número de membros, nem quanto à participação de utilizadores com limitações específicas.

Embora o sistema não controle diretamente a formação dos grupos, este pode proceder à sua subdivisão com base nos *clusters* previamente calculados, ou seja, agrupando utilizadores com perfis e preferências semelhantes. Aproveitando a capacidade do MAMS de subdividir grupos em subgrupos, demonstrada na Tabela 4.2, é possível segmentar estes grupos em porções menores, nas quais as características dos membros são mais homogêneas.

Tabela 4.2: Exemplo de subdivisão de um grupo em subgrupos pelo MAMS.

Grupo	Utilizador	Cluster	Subgrupo
Grupo 1	User 1	Cluster A	Subgrupo 1
Grupo 1	User 2	Cluster A	Subgrupo 1
Grupo 1	User 3	Cluster B	Subgrupo 2
Grupo 1	User 4	Cluster B	Subgrupo 2
Grupo 1	User 5	Cluster C	Subgrupo 3

Como demonstrado no Capítulo 2 pela Tabela 2.3, existem várias técnicas para calcular preferências num grupo. Como mencionado por Alves et al.[4], o MAMS utiliza a **média aritmética**, garantindo que cada utilizador contribui de forma igual para o perfil agregado do subgrupo.

É utilizada uma média incremental para facilitar o cálculo, conforme demonstrado na equação seguinte, sendo que o resultado é equivalente ao obtido com a média convencional:

$$\text{categoryAverage}_{n+1} = \frac{(\text{categoryAverage}_n \cdot n) + \text{newUserScore}}{n + 1}$$

Desta forma, como a média é aplicada a um subgrupo mais pequeno, existe uma maior hipótese de que o perfil agregado do subgrupo reflita com maior precisão as preferências individuais de cada membro. Esta abordagem permite que a recomendação seja mais personalizada, mesmo em contexto de grupo, e **reduz o risco de diluição das preferências** quando o grupo é grande ou heterogêneo.

Os medos e limitações são adicionados a uma lista completa, para depois serem filtrados — esta é a forma mais justa de não excluir qualquer membro.

O restante processo é equivalente ao descrito para as recomendações individuais. Cada subgrupo funciona como um indivíduo, e o número de iterações é igual ao número total

de subgrupos do grupo. Por exemplo, considerando a Tabela 4.2, as recomendações seriam calculadas para o Subgrupo 1, Subgrupo 2 e Subgrupo 3, utilizando a média das preferências dos membros de cada subgrupo. Desta forma, cada porção do grupo recebe sugestões alinhadas às características agregadas dos seus membros.

No que diz respeito às regras de associação, uma vez que cada subgrupo pertence a um *cluster*, as regras utilizadas para ponderar as recomendações correspondem às associadas a esse *cluster*. Caso não existam regras específicas para o *cluster*, são então aplicadas as regras globais

No final, com as recomendações de cada subgrupo geradas, estas são armazenadas na base de dados, associadas ao respetivo subgrupo.

4.2.3 Integração de Contexto nas Recomendações

Aquando da geração das recomendações, tanto para indivíduos como para grupos, é criada uma **lista alternativa de recomendações**. O objetivo desta lista é funcionar como um *fallback*, caso as recomendações originais não sejam adequadas, tendo em conta os dados de contexto.

A implementação da parte de *frontend*, incluindo notificações, gamificação e integração da localização do dispositivo, ficou a cargo de outros colaboradores.

Esta integração está alinhada com os fluxogramas das Figuras 3.15, 3.16 e 3.17, adicionando contextualização ambiental às recomendações e melhorando significativamente a experiência do utilizador ao evitar sugestões inadequadas devido a condições adversas.

Meteorologia

Para obter o contexto climático, o sistema liga-se à API do IPMA que fornece informações sobre meteorologia e avisos meteorológicos através de dois *endpoints* principais:

Endpoint 1: https://api.ipma.pt/open-data/forecast/warnings/warnings_www.json

O retorno é um ficheiro JSON com avisos específicos para cada região, contendo:

- **text:** texto descritivo do aviso (preenchido apenas para avisos amarelo, laranja ou vermelho);
- **awarenessTypeName:** tipo de aviso (ex: trovoada, agitação marítima, precipitação, vento, nevoeiro, neve, tempo frio, tempo quente);
- **awarenessLevelID:** nível do aviso (verde, amarelo, laranja, vermelho);
- **startTime** e **endTime:** datas de início e fim do aviso;
- **idAreaAviso:** identificador da área geográfica do aviso.

Endpoint 2: <https://api.ipma.pt/open-data/forecast/meteorology/cities/daily/{globalIdLocal}.json>

O retorno diário inclui previsões específicas para cada cidade ou região, como:

- **forecastDate:** data da previsão;
- **dataUpdate:** data de atualização do ficheiro (taxa horária);
- **tMin, tMax:** temperatura mínima e máxima diária;

- **predWindDir, classWindSpeed**: direção e intensidade do vento;
- **probPrecipita, classPrecInt**: probabilidade e intensidade da precipitação;
- **idWeatherType**: código do tipo de tempo;
- **latitude, longitude, globalIdLocal**: informações de localização.

Quando é **criada uma excursão**, são fornecidos dois parâmetros adicionais: a data de início e a data de fim da viagem.

Com base nestes parâmetros e nas coordenadas atribuídas a cada POI, o processo de recomendação inclui uma etapa adicional de avaliação das condições meteorológicas. São verificados avisos meteorológicos de nível amarelo, laranja ou vermelho, bem como uma probabilidade de precipitação superior a 60% para os POI recomendados, tal como demonstrado pelo Código 4.9.

```

1 foreach (var poi in recommendedPois)
2 {
3     var weatherAlert = IPMA.GetWeatherAlert(poi.Latitude, poi.Longitude,
4         startDate, endDate);
5     var dailyForecast = IPMA.GetDailyForecast(poi.Latitude, poi.
6         Longitude, startDate, endDate);
7
8     bool hasAlert = weatherAlert.Level == "yellow" ||
9         weatherAlert.Level == "orange" ||
10        weatherAlert.Level == "red";
11
12    bool highPrecipitation = dailyForecast.ProbabilityOfPrecipitation >
13    60;
14
15    if (hasAlert || highPrecipitation)
16    {
17        poi.IsFlaggedForWeather = true;
18        poi.AlternativePois = GetIndoorAlternatives(poi, top3Categories)
19    }
20 }

```

Código 4.9: Avaliação das condições meteorológicas para os POI.

Caso alguma destas condições se verifique, o utilizador é informado e disponibiliza-se a **lista alternativa de recomendações**, permitindo-lhe ajustar as sugestões. Para excursões programadas com vários dias de antecedência, a verificação é realizada a cada 24 horas, de modo a incluir possíveis alterações nas previsões meteorológicas.

É enviado ao *frontend* um JSON contendo os dados necessários para exibir a notificação ao utilizador. A estrutura desta mensagem é apresentada no Código 4.10.

Com base na decisão do utilizador, o *frontend* envia de volta um JSON que atualiza, ou mantém inalterada, a lista de recomendações, refletindo a escolha efetuada pelo utilizador.

Caso **não** exista uma excursão programada, mas o **utilizador se encontre no exterior**, o sistema utiliza as coordenadas fornecidas pelo dispositivo para consultar a API e verificar se existem condições meteorológicas adversas no local durante os próximos 30 minutos.

```
1 {
2   "poild": <POI_ID>,
3   "name": "<POI_NAME>",
4   "latitude": <LATITUDE>,
5   "longitude": <LONGITUDE>,
6   "contextAlert": {
7     "level": "<ALERT_LEVEL>",
8     "type": "<ALERT_TYPE>", //ex:"precipitation", "airQuality", "noise"
9     "startTime": "<START_TIMESTAMP>",
10    "endTime": "<END_TIMESTAMP>"
11  },
12  "precipitationProbability": <PROBABILITY_VALUE>,
13  "alternativePois": [
14    {
15      "poild": <ALTERNATIVE_POI_ID_1>,
16      "name": "<ALTERNATIVE_POI_NAME_1>",
17      "type": "<TYPE_1>"
18    },
19    {
20      "poild": <ALTERNATIVE_POI_ID_2>,
21      "name": "<ALTERNATIVE_POI_NAME_2>",
22      "type": "<TYPE_2>"
23    },
24    {
25      "poild": <ALTERNATIVE_POI_ID_3>,
26      "name": "<ALTERNATIVE_POI_NAME_3>",
27      "type": "<TYPE_3>"
28    }
29  ]
30 }
```

Código 4.10: Estrutura do ficheiro JSON enviado ao *Frontend* para gerar notificação de condições adversas.

Se forem detetadas condições desfavoráveis, é enviada uma mensagem em formato JSON, estruturada de forma semelhante ao Código 4.10, contendo os POI do top-3 categorias de turismo preferidas do utilizador, mas apenas aqueles de tipo interior.

Nível de Ruído e Qualidade do Ar

Estes dois atributos exigiram a integração com a API da Porto Digital [44, 45], que utiliza coordenadas para retornar a qualidade do ar e o nível de ruído da zona através de dois *endpoints* principais:

Endpoint 3: <https://broker.firmware.urbanplatform.portodigital.pt/v2/entities/?type=AirQualityObserved>

O retorno é um ficheiro JSON com a qualidade do ar observada em diferentes pontos da cidade do Porto, contendo:

- **id:** identificador único do sensor/entidade;
- **type:** tipo da entidade (sempre `AirQualityObserved`);
- **co, no2, o3, pm10, pm25:** valores medidos dos diferentes poluentes (em $\mu\text{g}/\text{m}^3$);
- **dateObserved:** data e hora da observação;
- **location:** coordenadas geográficas do sensor em formato GeoJSON.

Endpoint 4: <https://broker.firmware.urbanplatform.portodigital.pt/v2/entities/?type=NoiseLevelObserved>

O retorno é um ficheiro JSON com medições de ruído em diferentes pontos da cidade do Porto, contendo:

- **id:** identificador único do sensor/entidade;
- **type:** tipo da entidade (sempre `NoiseLevelObserved`);
- **LAeq:** nível de ruído medido em decibéis (dB);
- **dateObserved, dateObservedFrom, dateObservedTo:** período da medição;
- **location:** coordenadas geográficas do sensor em formato GeoJSON.

Para definir os limites da qualidade do ar, foi adotado o índice *QualAr*[48] proposto pela Agência Portuguesa do Ambiente (APA) . Este índice divide a qualidade do ar em cinco categorias, com base em intervalos de concentração (expressos em $\mu\text{g}/\text{m}^3$), tal como apresentado na Tabela 4.3.

Tabela 4.3: Índice *QualAr*, retirado de [48].

Classificação	PM ₁₀	PM _{2,5}	NO ₂	O ₃	SO ₂
Muito Bom	0-20	0-10	0-40	0-80	0-100
Bom	21-35	11-20	41-100	81-100	101-200
Médio	36-50	21-25	101-200	101-180	201-350
Fraco	51-100	26-50	201-400	181-240	351-500
Mau	101-1200	51-800	401-1000	241-600	501-1250

Tendo em conta os intervalos do índice, utilizadores mais sensíveis podem começar a ser afetados a partir da classificação **Média**. Por esse motivo, este foi o limite adotado para a implementação.

Também de acordo com a APA, o limite de nível de ruído não deve ultrapassar os 50 dB. Valores acima deste limiar podem ser considerados prejudiciais para a saúde e para o bem-estar [49].

Com os limites definidos, a estratégia é semelhante às condições meteorológicas, caso o **utilizador se encontre no exterior**, o sistema utiliza as coordenadas fornecidas pelo dispositivo para consultar a API e verificar a qualidade do ar e nível de ruído a cada 5 minutos, tal como demonstrado pelo Código 4.11

4.2. Geração de Recomendações

```
1 foreach (var poi in recommendedPois)
2 {
3     var airQuality = PD.GetLatestAirQuality(poi.Latitude, poi.Longitude)
4     ;
5     var noiseLevel = PD.GetLatestNoise(poi.Latitude, poi.Longitude);
6
7     double pm25 = airQuality.PM25;
8     double pm10 = airQuality.PM10;
9     double no2 = airQuality.NO2;
10    double o3 = airQuality.O3;
11
12    double noise = noiseLevel.LAeq;
13
14    bool poorAir = pm25 > 25 || pm10 > 50 || no2 > 200 || o3 > 180;
15    bool highNoise = noise > 50;
16
17    if (poorAir || highNoise)
18    {
19        poi.IsFlaggedForEnvironment = true;
20        poi.AlternativePois = GetIndoorAlternatives(poi, top3Categories)
21    }
22 }
```

Código 4.11: Avaliação da qualidade do ar e ruído para os POI.

Se forem detetadas condições desfavoráveis, é enviada uma mensagem em formato JSON, estruturada de forma equivalente ao Código 4.10, contendo os POI do top-3 categorias turísticas preferidas do utilizador, mas apenas aqueles de tipo interior.

Capítulo 5

Resultados e Análise

Este capítulo apresenta os resultados obtidos pelo sistema e a sua integração na aplicação. São mostradas as regras de associação, as listas de recomendações individuais e para grupos, bem como as notificações que indicam a necessidade de ajustar as recomendações devido a condições adversas. O objetivo é fornecer uma visão completa do comportamento do sistema, servindo de base para a análise detalhada dos resultados.

Em seguida, é realizada essa análise. São descritos os cenários de teste considerados e a metodologia utilizada para avaliar o desempenho dos diferentes algoritmos de extração de regras de associação, assim como o funcionamento do motor de recomendação. A avaliação divide-se em duas partes principais: a análise das regras de associação, considerando métricas como tempo de execução, utilização de memória, qualidade das regras e taxa de acerto; e a análise das recomendações, que aborda a cobertura, a coerência das escolhas e o impacto da integração de dados de contexto.

Por fim, é feita uma discussão crítica dos resultados, respondendo às questões de investigação definidas inicialmente.

5.1 Regras de Associação

Como descrito nos capítulos anteriores, as regras de associação são geradas a partir dos parâmetros dos utilizadores, recorrendo a um dos três algoritmos disponíveis.

O Código 5.1 apresenta um excerto de um ficheiro exemplo de regras conseguidas de uma iteração do sistema.

Para qualquer iteração, é possível verificar o algoritmo utilizado, as regras para os diferentes *clusters* e as regras globais, sendo cada uma seguida do valor da sua confiança.

A estrutura apresentada facilita a interpretação pelo sistema. Contudo, as regras podem ser representadas de forma mais intuitiva. Por exemplo, no Código 5.1, observa-se que um utilizador do *cluster* "user_108", que aprecia as categorias *Cultural Heritage* e *Natural Phenomena*, apresenta uma elevada probabilidade de também gostar de *Wild Nature Activities*.

```

1 {
2   "AlgorithmUsed": "Apriori",
3   "ByCluster": {
4     "user_108": [
5       "{Cultural Heritage (LIKE), Natural Phenomena (LIKE)} => {Wild
6       Nature Activities (LIKE)} 0.96",
7       "{Wild Nature Activities (LIKE), none (LIMITATION)} => {Natural
8       Phenomena (LIKE)} 1.00",
9       "{Natural Phenomena (LIKE), none (LIMITATION)} => {Wild Nature
10      Activities (LIKE)} 0.96"
11     ],
12     "user_1008": [
13       "{Cultural Heritage (LIKE)} => {Sports & Games (DISLIKE)} 0.95"
14     ]
15   },
16   "Global": []
17 }

```

Código 5.1: Exemplo de geração de regras de associação.

5.2 Recomendações de POI

As recomendações são persistidas no formato de listas de POI, acompanhadas da respetiva compatibilidade. Nas Figuras 5.1 e 5.2 apresenta-se um exemplo da interface móvel da aplicação, onde o utilizador pode visualizar as listas de recomendações individuais e para subgrupos, respetivamente. Além disso, estas páginas permitem ao utilizador avaliar o seu interesse em cada POI, numa escala de uma a cinco estrelas (1–5).

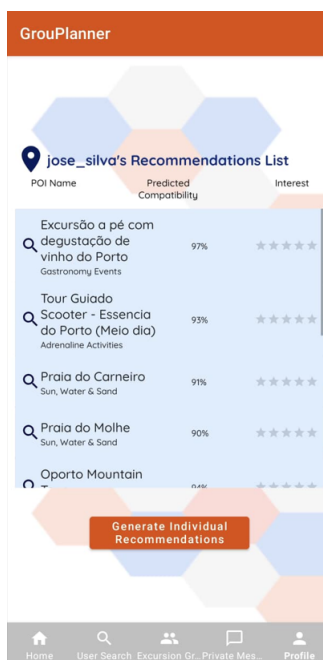


Figura 5.1: Lista de recomendações individuais na interface móvel da aplicação.



Figura 5.2: Lista de recomendações para grupos na interface móvel da aplicação.

5.3 Ajuste de Recomendações Baseadas no Contexto

O ajuste de recomendações baseadas em contexto tem como objetivo adaptar dinamicamente as sugestões de POI às condições externas e às restrições do utilizador. Para tal, o sistema gera mensagens em formato JSON, que encapsulam não só a informação dos POI recomendados, mas também os alertas de contexto e possíveis alternativas.

No exemplo do Código 5.2, segue o exemplo do POI **Cais da Ribeira** que é inicialmente sugerido ao utilizador. Contudo, devido à previsão de precipitação, é gerado um alerta de contexto com nível amarelo, indicando que as condições meteorológicas poderão afetar negativamente a experiência ao ar livre.

Neste cenário, o sistema apresenta automaticamente um conjunto de alternativas interiores próximas, como o restaurante **À Bolina**, a **Casa do Fado** e o **Museu do Vinho do Porto**, garantindo que a experiência turística se mantém adequada mesmo em condições adversas.

```
1 {
2   "poild": 73,
3   "name": "Cais da Ribeira",
4   "latitude": 41,1406402306194,
5   "longitude": -8,61118221820829,
6   "contextAlert": {
7     "level": "yellow",
8     "type": "precipitation",
9     "startTime": "2025-08-17T08:00:00",
10    "endTime": "2025-08-17T18:00:00"
11  },
12  "alternativePois": [
13    {
14      "poild": 5,
15      "name": "A Bolina",
16      "type": "indoor"
17    },
18    {
19      "poild": 6,
20      "name": "A Casa do Fado",
21      "type": "indoor"
22    },
23    {
24      "poild": 42,
25      "name": "Museu do Vinho do Porto",
26      "type": "indoor"
27    }
28  ]
29 }
```

Código 5.2: Exemplo de mensagem JSON enviada ao *frontend* para ajuste de recomendações com base num alerta amarelo de precipitação.

A variável *type* deste ficheiro pode assumir os valores *precipitation*, *noise* ou *airQuality*, permitindo adaptar a mensagem ao tipo de condição adversa identificada.

Esta mensagem constitui a base para a geração de notificações apresentadas pelo *Virtual Pet* da aplicação. Por exemplo, na Figura 5.3 é ilustrado o momento em que a mascote questiona o utilizador sobre a possibilidade de alterar a sua atividade para um POI interior, devido à previsão de chuva.

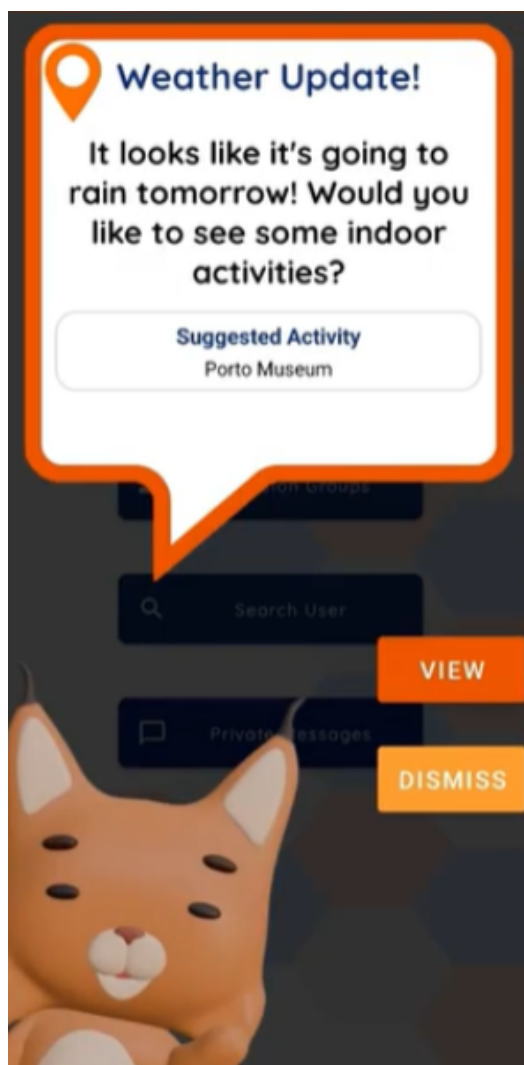


Figura 5.3: Notificação do *Virtual Pet* a sugerir ajuste da atividade do utilizador devido a previsão de chuva.

A decisão final cabe sempre ao utilizador, pelo que o POI recomendado apenas é atualizado caso este seleccione efetivamente uma das alternativas sugeridas.

5.4 Cenários de Teste

Para testar as componentes implementadas — nomeadamente a geração de regras de associação e de recomendações — definiram-se vários cenários de teste.

Os resultados apresentados correspondem a recomendações individuais. As recomendações para grupos revelaram padrões semelhantes, uma vez que, como referido anteriormente, estas são geradas a partir dos subgrupos formados pelo MAMS, tratando cada subgrupo como um indivíduo. O MAMS realiza a subdivisão dos grupos com o objetivo de **homogeneizar personalidades e preferências**, permitindo recomendações mais consistentes e alinhadas aos interesses de cada subgrupo.

5.4.1 Variação de Parâmetros e Utilizadores

O objetivo consistiu em compreender o impacto do número de utilizadores na plataforma, bem como do conjunto de parâmetros considerados na geração das regras.

Como a aplicação permite alterar os parâmetros utilizados na geração das regras, foram definidos quatro (4) cenários: sem regras, regras de base, regras sem preferências turísticas e regras com todos os parâmetros. Estes estão ilustrados na Tabela 5.1.

Tabela 5.1: Parâmetros considerados para cada subconjunto de regras de associação.

Parâmetro	None	Base	ExceptAttr	NoFilter
Idade		X	X	X
Género		X	X	X
Estado civil			X	X
Filhos			X	X
Escolaridade			X	X
Área de Formação			X	X
Situação Profissional			X	X
Com quem vive			X	X
Rendimento Líquido			X	X
Religião			X	X
Companheiros de viagem			X	X
Limitações		X	X	X
Medos/Fobias		X	X	X
Preferências Turísticas				X

Os primeiros dezasseis (16) cenários de teste combinam diferentes números de utilizadores com os distintos tipos de regras definidos acima, conforme ilustrado na Tabela 5.2.

Tabela 5.2: Cenários de teste com diferentes números de utilizadores e parâmetros de regras.

Nº Utilizadores	None	Base	ExceptAttr	NoFilter
35	X	X	X	X
200	X	X	X	X
500	X	X	X	X
1035	X	X	X	X

A seleção destes números de utilizadores permite avaliar o desempenho do sistema em diferentes escalas, desde pequenos grupos até uma base mais representativa da população. O número máximo de mil e trinta e cinco (1035) utilizadores foi definido como limite, correspondendo ao ficheiro completo de dados fornecido.

Cada subconjunto de regras reflete combinações distintas de filtros e parâmetros, permitindo analisar como a inclusão ou exclusão de determinados atributos influencia a geração de regras e, consequentemente, as recomendações produzidas.

5.4.2 Variação de Algoritmo e Utilizadores

Para complementar o exposto anteriormente e avaliar a performance do motor de recomendação, a precisão foi analisada ao variar os algoritmos de geração de regras e o número de utilizadores considerados. Nesta fase, as regras foram geradas utilizando todos os atributos disponíveis — designadas acima como **Rules Without Filters** — de modo a assegurar que cada algoritmo operasse com a informação completa sobre os utilizadores.

Tal como demonstrado pela Tabela 5.3, cada cenário combina um dos três algoritmos disponíveis — *Apriori*, *Eclat* e *FP-Growth* — com um dos quatro subconjuntos de utilizadores (35, 200, 500 e 1035).

Tabela 5.3: Cenários de teste com diferentes números de utilizadores e algoritmo utilizado.

Nº Utilizadores	Apriori	Eclat	FP-Growth
35	X	X	X
200	X	X	X
500	X	X	X
1035	X	X	X

5.5 Metodologia de Avaliação dos Resultados

Com os cenários definidos, procedeu-se à recolha dos resultados. Cada cenário foi executado na aplicação e, através de métodos auxiliares, os resultados foram registados em ficheiros *Excel*. Foram gerados dois tipos distintos de ficheiros:

1. **Regras:** contém apenas uma folha com duas tabelas distintas. A primeira apresenta a classificação real do utilizador em cada categoria turística, equivalente à Tabela 4.1. A segunda inclui o número de utilizadores, o tipo de regra aplicado segundo a Tabela 5.1, a lista de POI recomendados, a respetiva categoria e o tempo de execução;
2. **Recomendações:** contém várias folhas, incluindo os dados de todos os utilizadores, as classificações reais de cada utilizador por categoria turística, a lista de regras de associação geradas e as recomendações produzidas para cada utilizador.

Posteriormente, os ficheiros foram importados para um ambiente *Python* utilizando *Jupyter Notebook*, permitindo a análise e visualização dos resultados de forma sistemática. Este método facilitou o cálculo de estatísticas descritivas, comparações entre algoritmos e cenários, bem como a preparação de gráficos e tabelas que ilustram a performance do motor de recomendação sob diferentes condições.

5.6 Avaliação das Regras de Associação

As regras de associação foram geradas com todos os parâmetros ativados - **Rules Without Filters** —, pois apenas nesse cenário se verificaram alterações significativas nas recomendações. Como esperado, a desativação de alguns parâmetros resultou em redução do tempo de execução e do consumo de memória.

Nesta secção, é avaliado o desempenho dos algoritmos considerando métricas como tempo de execução, consumo de memória e qualidade das regras.

As comparações entre os algoritmos permitem identificar vantagens e limitações de cada abordagem na geração de regras.

5.6.1 Tempo de Execução e Memória

A estratégia passou por fazer a geração de regras para os diferentes números de utilizadores e os três algoritmos. Os resultados obtidos encontram-se expostos na Tabela 5.4.

Tabela 5.4: Tempo de execução (T, ms) e memória utilizada (M, MB) para os diferentes algoritmos e números de utilizadores.

Algoritmo	35 Utilizadores		200 Utilizadores		500 Utilizadores		1035 Utilizadores	
	T	M	T	M	T	M	T	M
Apriori	7393	124	6625	247	7125	380	7219	459
Eclat	6720	97	5975	124	6184	244	6578	297
FP-Growth	6632	85	5824	102	6092	187	6387	209

Para uma visualização mais ilustrativa, foram ainda construídos os gráficos das Figuras 5.4 e 5.5.

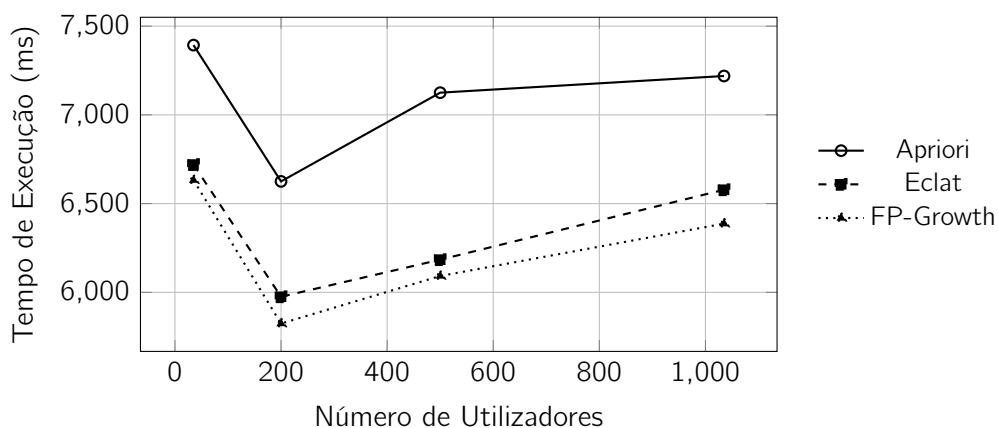


Figura 5.4: Gráfico de comparação do tempo de execução dos algoritmos *Apriori*, *Eclat* e *FP-Growth* para diferentes números de utilizadores.

O gráfico da Figura 5.4 mostra uma descida irregular entre os trinta e cinco (35) e os duzentos (200) utilizadores. Esta anomalia pode dever-se a variações na configuração do ambiente de execução, otimizações internas do algoritmo ou diferenças na distribuição dos dados entre os testes. Apesar desta irregularidade inicial, observa-se que, de forma geral, o tempo de execução aumenta com o número de utilizadores.

Por sua vez, o gráfico da Figura 5.5 mostra que o consumo de memória aumenta proporcionalmente com o número de utilizadores, como era de esperar. Nos casos do *Eclat* e do *FP-Growth*, este crescimento é menos acentuado, especialmente ao passar de trinta e cinco (35) para duzentos (200) utilizadores.

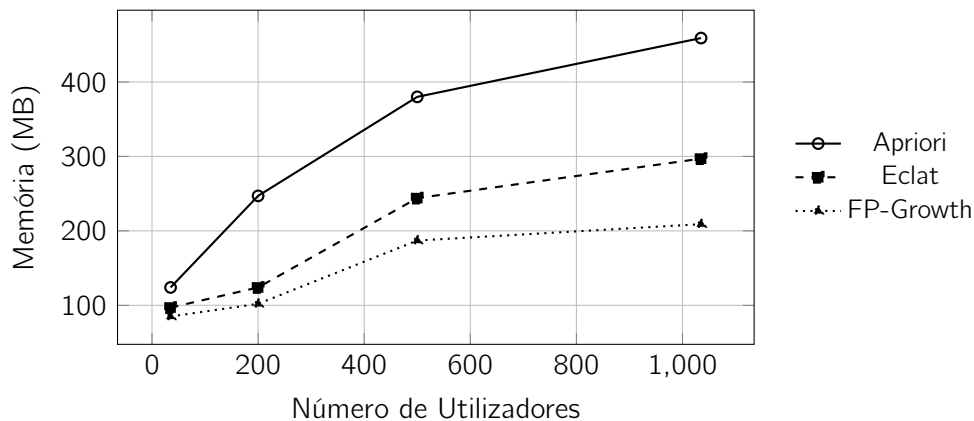


Figura 5.5: Gráfico de comparação do consumo de memória dos algoritmos *Apriori*, *Eclat* e *FP-Growth* para diferentes números de utilizadores.

Verifica-se que o *Apriori* é o mais custoso, tanto em termos de tempo como de memória, devido à necessidade de gerar todos os conjuntos candidatos e calcular a frequência de cada conjunto frequente.

O *Eclat* apresenta melhor desempenho, com tempos de execução e consumo de memória menores, graças à sua abordagem vertical, que permite calcular o suporte diretamente nos conjuntos de transações. Durante a **Revisão da Literatura**, no Capítulo 2, foi mencionado que este algoritmo consumia mais memória, mas apresentava menor tempo de execução [33], o que, no entanto, não se verificou neste caso. Contudo, convém salientar que a amostra utilizada também é relativamente reduzida, o que pode influenciar estes resultados.

Ainda assim, o *FP-Growth* revela-se o mais eficiente, especialmente em termos de memória, ao utilizar a *FP-Tree* para compactar a informação das transações e evitar a geração de candidatos.

Para *datasets* pequenos, as diferenças entre os algoritmos são moderadas, mas, para *datasets* maiores, o *Apriori* consome significativamente mais tempo e memória.

Concluindo, estes resultados indicam que o *FP-Growth* é a opção mais eficiente para grandes volumes de dados, o *Eclat* constitui uma alternativa razoável, e o *Apriori* é recomendado apenas para *datasets* pequenos ou quando se pretende simplicidade na implementação.

5.6.2 Qualidade das Regras de Associação

Para as regras de associação geradas, verifica-se que as regras globais são mais comuns quando o número de utilizadores é mais reduzido. As regras associadas a *clusters* são mais comuns para os números de utilizadores maiores. Isto significa que, à medida que a dimensão da amostra cresce, o sistema consegue identificar padrões mais específicos e segmentados, deixando de depender apenas de tendências globais.

Este comportamento é esperado, uma vez que conjuntos de dados mais extensos aumentam a diversidade de combinações possíveis e permitem detetar relações que não seriam visíveis em grupos menores.

Para avaliar a qualidade das regras de associação geradas, foram calculadas a confiança média e o Desvio-Padrão (DP) das regras obtidas para diferentes os algoritmos (*Apriori*,

5.6. Avaliação das Regras de Associação

Eclat e *FP-Growth*) e diferentes números de utilizadores. A Tabela 5.5 demonstra os valores obtidos.

Tabela 5.5: Valores médios de confiança (C, %) e desvio padrão (DP, %) das regras de associação para diferentes algoritmos e números de utilizadores.

Algoritmo	35 Utilizadores		200 Utilizadores		500 Utilizadores		1035 Utilizadores	
	C	DP	C	DP	C	DP	C	DP
Apriori	96,8	±1,1	100	0	99,8	±0,6	99,7	±1,3
Eclat	96,8	±1,1	100	0	98,8	±1,5	98,9	±1,4
FP-Growth	96,7	±0,2	100	0	100	0	98,7	±1,3

Através da análise da Tabela 5.5, verifica-se que, independentemente do cenário, os valores de confiança se situam acima dos 96,7%. Isto indica que as regras geradas são confiáveis, mesmo com diferentes números de utilizadores ou diferentes algoritmos.

Observa-se que, para um número reduzido de utilizadores (35), todos os algoritmos apresentam médias de confiança próximas, entre 96,7% e 96,8%, com DP relativamente baixos, o que demonstra estabilidade na qualidade das regras mesmo em pequenos conjuntos de dados.

Com o aumento do número de utilizadores para duzentos (200), a confiança média atinge 100% para todos os algoritmos, refletindo que, a partir de uma escala intermédia, o sistema consegue gerar regras de associação com máxima confiança, independentemente do método escolhido.

Para escalas maiores (500 e 1035 utilizadores), a confiança média permanece elevada (entre 98,7% e 99,8%), embora o DP aumente ligeiramente, indicando uma maior variação nos valores individuais de confiança, o que é esperado dado o aumento da complexidade e diversidade dos dados.

Comparando os algoritmos, o *FP-Growth* apresenta um desempenho ligeiramente mais consistente em termos de DP para trinta e cinco (35) utilizadores, enquanto *Apriori* atinge a confiança mais elevada média (99,8%) para 500 utilizadores. O *Eclat* mantém-se muito próximo dos demais também com variações mínimas.

Em síntese, os resultados sugerem que a escolha do algoritmo tem impacto limitado na confiança média das regras quando os conjuntos de dados são grandes, enquanto o número de utilizadores influencia ligeiramente a estabilidade (DP) das regras geradas.

5.6.3 Precisão e Taxa de Acerto

Para verificar qual dos três algoritmos influencia positivamente as recomendações, foram ainda avaliados os seguintes parâmetros:

1. **Precisão:** indica a proporção de recomendações corretas em relação ao total de recomendações feitas a cada utilizador. Valores mais altos significam que a maioria das recomendações é relevante;
2. **Taxa de Acerto:** indica a proporção de utilizadores que receberam pelo menos uma recomendação correta. Valores mais altos significam que mais utilizadores foram efetivamente contemplados pelas recomendações.

Os valores obtidos foram organizados na Tabela 5.6.

Tabela 5.6: Precisão média (P, %), utilizadores com acertos (UA) e taxa de acerto (TA, %) das recomendações por número de utilizadores e algoritmo.

Algoritmo	35 Utilizadores			200 Utilizadores			500 Utilizadores			1035 Utilizadores		
	P	UA	TA	P	UA	TA	P	UA	TA	P	UA	TA
Nenhum	-	-	-	-	-	-	-	-	-	89,3	987	95,4
Apriori	95,2	35	100	98,3	199	99,5	93,4	476	95,2	94,6	994	96,0
Eclat	95,2	35	100	98,3	199	99,5	93,4	476	95,2	94,6	994	96,0
FP-Growth	98,1	35	100	98,7	200	100	96,9	493	98,6	95,4	1004	97,0

Verifica-se, desde logo, que, independentemente do algoritmo e do número de utilizadores, os valores de precisão são elevados, superiores a 95%. Observa-se também que qualquer uma das estratégias de geração de regras apresenta desempenho superior, em termos de precisão e taxa de acerto, comparativamente à ausência de regras.

Observa-se também que os algoritmos *Apriori* e *Eclat* apresentam exatamente os mesmos valores. O desempenho de ambos tende a diminuir ligeiramente à medida que o número de utilizadores aumenta.

Por fim, o *FP-Growth* destaca-se positivamente, obtendo o maior número de utilizadores com acertos, assim como a maior precisão média e taxa de acerto. Este algoritmo consegue manter uma taxa de acerto elevada mesmo em cenários com muitos utilizadores, evidenciando a sua robustez na manutenção da qualidade das recomendações.

Pode se verificar o resumo dos valores na Tabela 5.7.

Tabela 5.7: Resumo da precisão média e taxa de acerto dos algoritmos.

Algoritmo	Precisão Média (%)	DP (%)	Taxa Acerto (%)	DP (%)
Nenhum	89,3	±0,17	95,4	±0,15
Apriori	95,4	±0,21	97,9	±0,22
Eclat	95,4	±0,21	97,9	±0,22
FP-Growth	97,3	±0,15	99,1	±0,11

5.7 Avaliação de Recomendações

Nesta secção, procede-se à análise detalhada das recomendações geradas pelo sistema, considerando métricas de cobertura, precisão e coerência temática. Serão apresentados os resultados obtidos, bem como uma avaliação crítica do desempenho das recomendações em diferentes cenários e condições.

5.7.1 Cobertura das Recomendações

Foi necessário verificar se todos os utilizadores podiam receber recomendações como esperado. A Tabela 5.8 apresenta a percentagem de cobertura das recomendações para diferentes números de utilizadores e algoritmos.

Tabela 5.8: Utilizadores com recomendações (UA) e cobertura (C, %) para diferentes números de utilizadores e algoritmos.

Algoritmo	35 Users		200 Users		500 Users		1035 Users	
	UA	C	UA	C	UA	C	UA	C
Apriori	35	100	200	100	499	99,8	1027	99,23
Eclat	35	100	200	100	499	99,8	1027	99,23
FP-Growth	35	100	200	100	499	99,8	1027	99,23

Verifica-se que a cobertura não é influenciada pelo algoritmo escolhido, uma vez que os valores são iguais para os três, mas sim pelo número de utilizadores.

Mesmo assim, os valores de cobertura mantêm-se elevados, e a grande maioria dos utilizadores recebe recomendações, indicando que o sistema é capaz de suportar grandes volumes de utilizadores.

5.7.2 Matriz de Confusão

Anteriormente, no Subcapítulo 5.6.3, já foram verificadas a precisão e a taxa de acerto das recomendações, tendo-se concluído que, com o algoritmo *FP-Growth*, obtiveram-se os melhores valores destas métricas utilizadas para avaliar as recomendações.

As métricas atingem valores bastante altos, indicando que o sistema está a gerar recomendações com sucesso. Contudo, é importante analisar quais as categorias mais acertadas ou menos precisas, de forma a compreender se o sistema apresenta alguma dificuldade específica em relação a alguma delas.

Para este efeito, foi construída uma matriz de confusão com base nas regras geradas pelo *FP-Growth* para os mil e trinta e cinco (1035) utilizadores. O objetivo desta matriz é identificar em que classes o sistema de recomendação apresenta maiores dificuldades.

A matriz, representada na Figura 5.6, mostra que a diagonal concentra os valores mais elevados, revelando que o sistema é consistente na geração das recomendações, acertando na maioria das situações dentro da mesma categoria.

Entre todas as categorias, sobressai de forma evidente a de *Cultural Heritage*, cuja frequência é significativamente superior às restantes, refletindo a sua predominância no conjunto de preferências analisadas.

Verifica-se também que os valores fora da diagonal apresentam uma frequência bastante reduzida, o que sugere que o sistema tende a gerar recomendações consistentes dentro de cada categoria.

Este comportamento pode ser interpretado de duas formas: por um lado, confirma a precisão do algoritmo em manter a coerência temática das recomendações; por outro, revela uma limitação na capacidade de explorar relações transversais entre diferentes tipos de atividades.

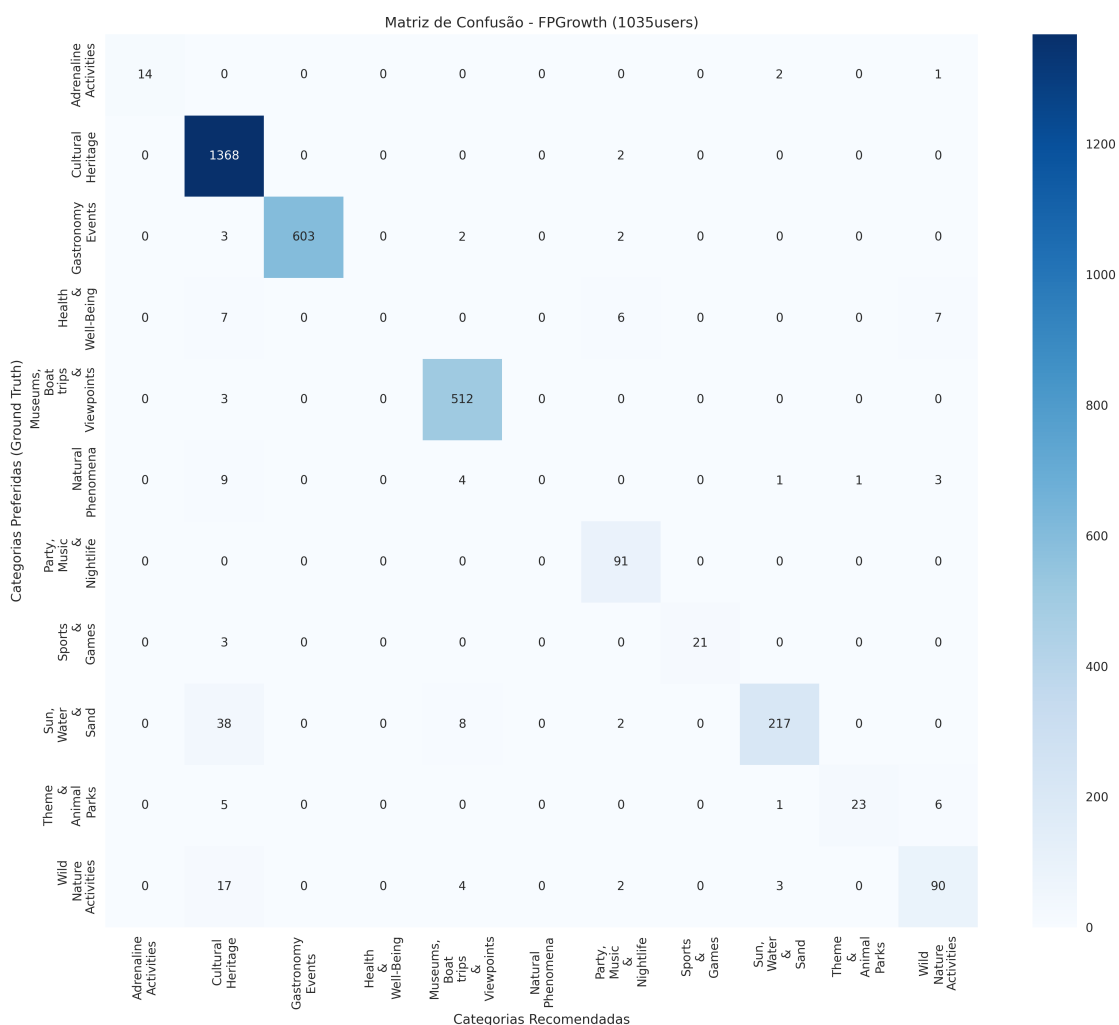


Figura 5.6: Matriz de confusão.

Ou seja, embora categorias como *Gastronomy & Events*, *Museums, Boat Trips & Viewpoints* e *Sun, Water & Sand* apresentem também valores significativos, a predominância de *Cultural Heritage* evidencia um possível problema de **tendência** do sistema para esta classe, já discutido no Subcapítulo 2.1.2.1. Esta preferência já tinha sido observada no Subcapítulo 3.5.2, através da análise do gráfico de distribuição das avaliações na Figura 3.9.

Para determinar se esta **tendência** constitui de facto um problema, seria necessário analisar novos dados de utilizador e, caso se confirme a sua existência, aplicar estratégias de mitigação, como mecanismos de *re-ranking* que promovam categorias menos populares.

5.7.3 Efeito da Contextualização

A ligação com as API externas foi bem-sucedida, contudo, verificou-se que a responsável por disponibilizar os dados relativos à qualidade do ar e dos níveis de ruído nem sempre se encontra funcional. Este é um problema alheio tanto a este projeto como ao próprio GECAD.

Apesar desta limitação, foram simulados cenários em que as notificações eram ativadas. Verificou-se que, perante a previsão de condições adversas, a aplicação notifica o utilizador e envia POI alternativos.

Em todos os cenários simulados como condições adversas, os POI classificados como exteriores foram substituídos por POI do tipo interior, tal como esperado.

À semelhança do que foi referido anteriormente, esta funcionalidade deverá ser avaliada de forma quantitativa em ambiente experimental, onde será possível estudar o impacto da contextualização na satisfação real dos utilizadores.

5.8 Discussão e Resposta às Questões de Pesquisa

Para concluir a análise dos resultados, é necessário dar resposta às questões de investigação definidas no Subcapítulo 1.3.

O objetivo é articular as evidências recolhidas nos testes e experiências realizadas com os objetivos delineados inicialmente, de forma a validar o contributo do trabalho desenvolvido.

As respostas que se seguem sintetizam os principais resultados obtidos, procurando avaliar o desempenho dos algoritmos de extração de regras de associação, a influência das recomendações na personalização e coerência das sugestões, bem como o impacto da integração de dados contextuais no motor de recomendação.

Questão 1: Como se compara o desempenho e a eficácia do algoritmo *Apriori* face a outras técnicas de extração de regras de associação, como o *FP-Growth* e o *Eclat*, no contexto deste sistema de recomendação?

A comparação entre os algoritmos de extração de regras de associação permitiu identificar diferenças relevantes em termos de desempenho e eficiência.

Como demonstrado na Tabela 5.4, o algoritmo *Apriori* revelou-se o menos eficiente, sobretudo devido à elevada exigência computacional em memória e tempo de execução, causada pela geração exaustiva de candidatos. O *FP-Growth*, por sua vez, apresentou melhores resultados, conseguindo extrair regras de forma mais rápida e eficiente. O *Eclat*, embora tenha apresentado algumas vantagens em cenários específicos com conjuntos de dados mais pequenos, mostrou-se menos eficiente do que o *FP-Growth* no contexto dos testes realizados.

As métricas de precisão e taxa de acerto, ilustradas na Tabela 5.7, permitiram avaliar a eficácia dos algoritmos. Todos alcançaram valores elevados, mas o *FP-Growth* destacou-se, obtendo uma vantagem consistente relativamente aos restantes.

Assim, no contexto avaliado, conclui-se que o *FP-Growth* é o algoritmo mais indicado para suportar a geração de regras de associação, constituindo um bom equilíbrio entre desempenho e qualidade.

Questão 2: De que forma a utilização de regras de associação na geração de recomendações influencia a personalização das sugestões e a coerência das recomendações em grupos de utilizadores?

As regras de associação foram introduzidas no sistema com o objetivo de identificar padrões de preferência entre utilizadores, os quais contribuem para a personalização das recomendações.

Como verificado na Tabela 5.7, os valores de precisão e taxa de acerto aumentaram, independentemente do algoritmo utilizado para gerar as regras, indicando que esta estratégia cumpriu o seu propósito.

Por fim, a análise da matriz de confusão demonstra que a maioria das recomendações acerta nas categorias de interesse dos utilizadores, evidenciando uma coerência temática consistente.

Questão 3: Qual é o impacto da integração de dados contextuais, como condições meteorológicas, qualidade do ar e ruído, no funcionamento do motor de recomendação e na adaptação dinâmica das sugestões?

A integração de dados contextuais no motor de recomendação foi implementada com o objetivo de permitir a adaptação dinâmica das sugestões de acordo com fatores externos, como condições meteorológicas, qualidade do ar ou nível de ruído.

Como referido, durante os testes, verificou-se que a ligação com as API externas foi bem-sucedida, embora algumas fontes, nomeadamente as que fornecem os dados de qualidade do ar e nível de ruído, nem sempre estivessem disponíveis.

Apesar destas limitações, foram simulados cenários em que notificações e ajustes de recomendações eram acionados. Observou-se que, quando condições adversas eram observadas, o sistema fornecia alternativas de POI do tipo interior, cumprindo o comportamento esperado. Em todos os cenários simulados, a contextualização das recomendações mostrou-se funcional e coerente.

No entanto, uma análise quantitativa mais robusta só será possível em ambiente experimental, com utilizadores reais, permitindo avaliar o impacto da contextualização na satisfação e experiência de utilização.

Capítulo 6

Conclusões

Os resultados obtidos neste trabalho permitiram responder às questões de investigação com sucesso e de forma rigorosa, demonstrando a eficiência das estratégias implementadas para a personalização de recomendações turísticas em grupos.

A análise dos diferentes algoritmos de extração de regras de associação (*Apriori*, *Eclat* e *FP-Growth*), combinada com a integração de dados contextuais, demonstrou o impacto destas abordagens na precisão, cobertura e coerência das recomendações.

De forma mais específica, foi possível observar como a incorporação de perfis de personalidade e dados de contexto influencia positivamente a capacidade do sistema de fornecer recomendações adaptadas aos utilizadores e aos grupos, respondendo assim às questões de investigação propostas no início deste trabalho.

A análise detalhada dos cenários de teste evidenciou que a abordagem adotada consegue mitigar limitações comuns em sistemas de recomendação, como a heterogeneidade de personalidades, e oferece uma base sólida para a personalização de experiências turísticas.

A síntese destes resultados fornece, portanto, uma visão científica sobre o desempenho do GrouPlanner, destacando não apenas os sucessos alcançados, mas também as áreas em que melhorias adicionais poderão ser exploradas em trabalhos futuros.

6.1 Objetivos Concretizados

De modo a verificar se os objetivos delineados no Subcapítulo 1.3 foram alcançados, apresenta-se a seguir um resumo.

O objetivo relacionado com a análise da estratégia previamente implementada no projeto, de forma a planear as melhorias futuras, foi concluído com sucesso.

Quanto ao objetivo de explorar novas estratégias ou algoritmos de extração de regras de associação, após a revisão da literatura, foram implementados os algoritmos *Eclat* e *FP-Growth*. A análise comparativa com o *Apriori* utilizou métricas como confiança, precisão e taxa de acerto. O *FP-Growth* demonstrou ser o mais eficiente, equilibrando melhor o desempenho computacional e a qualidade das regras. O *Eclat* apresentou vantagens apenas em conjuntos de dados menores.

O objetivo de melhorar e otimizar a estratégia de recomendação foi atingido, com o algoritmo refeito, incorporando as regras de associação e dados de contexto, apresentando cobertura, precisão e taxas de acerto satisfatórias.

No que diz respeito à integração de dados contextuais ao motor de recomendações, incluindo condições meteorológicas, qualidade do ar e níveis de ruído, este objetivo foi alcançado, com a integração das API externas e a lógica de adaptação dinâmica a funcionar corretamente. As métricas de avaliação só poderão ser analisadas quando a aplicação estiver em ambiente experimental, avaliando a satisfação dos utilizadores reais.

6.2 Limitações

Numa fase inicial de integração, o MAMS constituiu uma limitação, sobretudo devido à falta de conhecimento prévio sobre o conceito e ao esforço necessário para compreender a comunicação deste microserviço com os restantes componentes do sistema.

Outra limitação relevante prende-se com a dimensão da amostra utilizada: mil e trinta e cinco (1035) participantes. Embora se trate de dados reais, a sua quantidade é relativamente reduzida para suportar generalizações mais amplas. Além disso, a análise dos dados e os resultados evidenciaram uma concentração excessiva nas avaliações de determinadas categorias, em particular em *Cultural Heritage*, revelando uma tendência que pode comprometer a diversidade das recomendações.

Por fim, embora a integração dos dados contextuais tenha sido realizada com sucesso, verificou-se que as medições de qualidade do ar e de níveis de ruído nem sempre estavam disponíveis ou corretas. Este problema foi identificado e reportado aos responsáveis pela API externa, mas ainda assim representa uma limitação prática à plena utilização destes dados no sistema de recomendação.

6.3 Trabalho Futuro

Relativamente a trabalhos futuros, como já foi destacado anteriormente, torna-se essencial avaliar o desempenho dos diferentes algoritmos e a integração dos dados de contexto, através de uma análise da satisfação efetiva de utilizadores reais.

Espera-se também um aumento significativo do número de utilizadores, o que permitirá avaliar a escalabilidade do sistema e compreender se a tendência observada para a categoria *Cultural Heritage* constitui, de facto, uma limitação relevante ou se é apenas um reflexo da amostra utilizada.

No que diz respeito à extração de regras, poderá ser pertinente explorar métodos mais avançados, como modelos de *machine learning* ou redes neuronais, que poderão oferecer maior capacidade de generalização e adaptação.

Por fim, caso a integração de dados contextuais se revele satisfatório, poderá ser vantajoso alargar o leque de variáveis consideradas, incluindo fluxos turísticos em tempo real, eventos locais ou até a ocorrência de focos de incêndio (particularmente comuns nos verões em Portugal). Desta forma, o sistema poderá fornecer recomendações ainda mais adaptativas e personalizadas, alinhadas com a realidade do utilizador no momento.

6.4 Considerações Finais

Em suma, este trabalho permitiu aprimorar o motor de recomendações do protótipo de GRS para turismo, respondendo aos objetivos inicialmente traçados.

A exploração de diferentes algoritmos de extração de regras, aliada à incorporação de dados contextuais, contribuiu para recomendações mais precisas e adaptadas às necessidades dos utilizadores. Ainda que tenham sido identificadas limitações relevantes, como a dimensão reduzida da amostra ou a predominância de determinadas categorias, o sistema mostrou-se capaz de evoluir de forma consistente.

O projeto não só consolidou a compreensão sobre os desafios inerentes à recomendação personalizada em turismo, como também demonstrou o potencial de integrar dados externos dinâmicos para enriquecer a experiência do utilizador.

O trabalho desenvolvido abre assim caminho para investigações futuras, que poderão expandir o sistema em termos de escalabilidade, diversidade de variáveis contextuais e validação em cenários reais de utilização.

Deste modo, considera-se que os contributos alcançados oferecem uma base sólida para o desenvolvimento de soluções mais robustas, inteligentes e ajustadas ao setor do turismo, reforçando a importância da personalização como fator diferenciador na experiência turística.

Referências

- [1] *GrouPlanner*. <https://www.gecad.isep.ipp.pt/grouplanner/index.html>. Accessed: 2025-08-13. GECAD - ISEP.
- [2] *ATT – gead*. https://www.gecad.isep.ipp.pt/portfolio/_att/. Accessed: 2025-08-13. GECAD - ISEP.
- [3] *gecad – Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development*. url: <https://www.gecad.isep.ipp.pt/>.
- [4] Patrícia Alves, Helena Martins, Pedro Saraiva, João Carneiro, Paulo Novais e Goreti Marreiros. «Group recommender systems for tourism: how does personality predict preferences for attractions, travel motivations, preferences and concerns?» Em: *User Modeling and User-Adapted Interaction* (2023). doi: 10.1007/s11257-023-09361-2. url: <https://doi.org/10.1007/s11257-023-09361-2>.
- [5] Patrícia Alves, André Martins, Francisco Negrão, Paulo Novais, Ana Almeida e Goreti Marreiros. «Are heterogeneity and conflicting preferences no longer a problem? Personality-based dynamic clustering for group recommender systems». Em: (2024). doi: 10.1016/j.eswa.2024.124812. url: <https://doi.org/10.1016/j.eswa.2024.124812>.
- [6] Patrícia Alves, Joana Neto, Jorge Lima, José Silva, Goreti Marreiros e Luís Conceição. «Travel Together, Play Together: Gamifying a Group Recommender System for Tourism». Em: 25 (2025). doi: 10.1145/3705328.3759344. url: <https://doi.org/10.1145/3705328.3759344>.
- [7] *Regulamento - 2016/679 - EN - GDPR - EUR-Lex*. url: <https://eur-lex.europa.eu/legal-content/PT/TXT/?uri=CELEX%3A32016R0679>.
- [8] *Scrum Guide | Scrum Guides*. url: <https://scrumguides.org/scrum-guide.html>.
- [9] *GitHub Issues · Project planning for developers*. url: <https://github.com/features/issues>.
- [10] F. O. Isinkaye, Y. O. Folajimi e B. A. Ojokoh. «Recommendation systems: Principles, methods and evaluation». Em: *Egyptian Informatics Journal* 16 (3 nov. de 2015), pp. 261–273. issn: 1110-8665. doi: 10.1016/J.EIJ.2015.06.005. url: <https://www.sciencedirect.com/science/article/pii/S1110866515000341>.
- [11] Matevž Kunaver e Tomaž Požrl. «Diversity in recommender systems – A survey». Em: *Knowledge-Based Systems* 123 (mai. de 2017), pp. 154–162. issn: 0950-7051. doi: 10.1016/J.KNOSYS.2017.02.009. url: <https://www.sciencedirect.com/science/article/pii/S0950705117300680>.
- [12] «A systematic review and research perspective on recommender systems». Em: *Journal of Big Data* 9 (1 dez. de 2022), pp. 1–36. issn: 21961115. doi: 10.1186/S40537-022-00592-5/TABLES/2. url: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00592-5>.
- [13] P. H. Aditya, I. Budi e Q. Munajat. «A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT X». Em: *2016 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2016*

- (mar. de 2017), pp. 303–308. doi: 10.1109/ICACISIS.2016.7872755. url: https://www.researchgate.net/publication/314668222_A_comparative_analysis_of_memory-based_and_model-based_collaborative_filtering_on_the_implementation_of_recommender_system_for_E-commerce_in_Indonesia_A_case_study_PT_X.
- [14] Jiahui Lin e Xiaowen Liu. «Context-awareness-based Intelligent Recommendation Method of Tourism Service Resources». Em: (2024). doi: 10.1145/3711129.3711131. url: <https://doi.org/10.1145/3711129.3711131>.
- [15] Ludovico Boratto, Gianni Fenu e Pier Luigi Pau. «Design Criteria to Model Groups in Big Data Scenarios: Algorithms and Best Practices». Em: (2015).
- [16] Mesut Kaya, Derek Bridge e Nava Tintarev. «Ensuring Fairness in Group Recommendations by Rank-Sensitive Balancing of Relevance». Em: *RecSys 2020 - 14th ACM Conference on Recommender Systems 20* (set. de 2020), pp. 101–110. doi: 10.1145/3383313.3412232; SUBPAGE: STRING: ABSTRACT; CSUBTYPE: STRING: CONFERENCE. url: <https://dl.acm.org/doi/pdf/10.1145/3383313.3412232>.
- [17] Emre Yalcin e Alper Bilge. «Investigating and counteracting popularity bias in group recommendations». Em: *Information Processing & Management* 58 (5 set. de 2021), p. 102608. issn: 0306-4573. doi: 10.1016/J.IPM.2021.102608. url: <https://www.sciencedirect.com/science/article/pii/S0306457321001047>.
- [18] Christopher J. Soto e Joshua J. Jackson. «Five-Factor Model of Personality». Em: *Psychology* (fev. de 2013). doi: 10.1093/OB0/9780199828340-0120. url: https://www.researchgate.net/publication/264476432_Five-Factor_Model_of_Personality.
- [19] Eyad Kannout, Hung Son Nguyen e Marek Grzegorowski. «Speeding Up Recommender Systems Using Association Rules». Em: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13758 LNAI (nov. de 2022), pp. 167–179. doi: 10.1007/978-3-031-21967-2_14. url: <http://arxiv.org/abs/2211.08799>http://dx.doi.org/10.1007/978-3-031-21967-2_14.
- [20] *IEEE Xplore*. url: <https://ieeexplore.ieee.org/Xplore/home.jsp>.
- [21] *ACM Digital Library*. url: <https://dl.acm.org/>.
- [22] *Google Acadêmico*. url: <https://scholar.google.com/>.
- [23] *ScienceDirect.com | Science, health and medical journals, full text articles and books*. url: <https://www.sciencedirect.com/>.
- [24] «Fast Algorithms for Mining Association Rules». Em: (1994).
- [25] Sruthy. *Apriori Algorithm in Data Mining: Implementation With Examples*. Abr. de 2025. url: <https://www.softwaretestinghelp.com/Apriori-Algorithm/>.
- [26] M J Zaki, S Parthasarathy, M Ogihara e W Li. «New Algorithms for Fast Discovery of Association Rules *». Em: (1997). url: www.aaai.org.
- [27] Better Life. *Eclat Algorithm in Machine Learning*. Set. de 2023. url: <https://quality-life.medium.com/eclat-algorithm-in-machine-learning-fe07d33fcc5b>.
- [28] «Association analysis of Equipment Maintenance Data Based on Eclat Algorithm». Em: *ACM International Conference Proceeding Series* (jan. de 2024), pp. 200–204. doi: 10.1145/3672758.3672791. url: <https://dl.acm.org/doi/pdf/10.1145/3672758.3672791>.
- [29] Jiawei Han, Jian Pei e Yiwon Yin. «Mining frequent patterns without candidate generation». Em: *SIGMOD Record (ACM Special Interest Group on Management of Data)* 29 (2 mai. de 2000), pp. 1–12. issn: 01635808. doi: 10.1145/335191.335372; TOPIC: TOPIC: ACMSIG>SIGMOD; JOURNAL: JOURNAL: SIGMOD; REQUESTEDJOURNAL:

- JOURNAL : SIGMOD ; SERIALTOPIC : TOPIC : ACM-PUBTYPE>NEWSLETTER ; PAGE : STRING : ARTICLE/CHAPTER. url: <https://dl.acm.org/doi/pdf/10.1145/335191.335372>.
- [30] Shutchapol Chopvitayakun, Watchareewan Jitsakul e Nuntaporn Aukkanit. «Analyzing Purchase Behavior Using FP Growth Technique to Find Association Rules». Em: *ACM International Conference Proceeding Series* (dez. de 2024), pp. 106–111. doi: 10.1145/3678610.3678618 / ASSET / 71A702C1 - 02D3 - 4C74 - 866D - 99722CBC5AD3 / ASSETS / IMAGES / LARGE / IMAGE4 . JPG. url: <https://dl.acm.org/doi/pdf/10.1145/3678610.3678618>.
- [31] Jeff Heaton. «Comparing Dataset Characteristics that Favor the Apriori, Eclat or FP-Growth Frequent Itemset Mining Algorithms». Em: (jan. de 2017).
- [32] Nurbojatmiko, Eri Rustamaji e Asep Fajar Firmansyah. «Analysis of study program selection patterns using FP-growth and eclat methods». Em: *ACM International Conference Proceeding Series 7* (nov. de 2018), pp. 401–407. doi: 10.1145/3282373.3282848. url: <https://dl.acm.org/doi/pdf/10.1145/3282373.3282848>.
- [33] V Srinadh. «Evaluation of Apriori, FP growth and Eclat association rule mining algorithms». Em: *International journal of health sciences* (abr. de 2022), pp. 7475–7485. issn: 2550-6978. doi: 10.53730/IJHS.V6NS2.6729.
- [34] A. V. Smirnov, N. G. Shilov, A. V. Ponomarev, A. M. Kashevnik e V. G. Parfenov. «Group context-aware recommendation systems». Em: *Scientific and Technical Information Processing* 41 (5 2014), pp. 325–334. issn: 19348118. doi: 10.3103/S0147688214050050. url: https://www.researchgate.net/publication/272426594_Group_context-aware_recommendation_systems.
- [35] Yuan Tao, Shuping Yuan, Huakai Cheng, Junyang Han, Min Bao e Xue Zhang. «A Group Tourism Recommendation Method Based on User Preference and Attraction Popularity». Em: *CISAI 2024* (2024). doi: 10.1145/3703187.3703194. url: <https://doi.org/10.1145/3703187.3703194>.
- [36] Yuhang Wan. «Tourism intelligent recommendation system based on big data mining». Em: *ACM International Conference Proceeding Series* (out. de 2022), pp. 1040–1044. doi: 10.1145/3573428.3573615 ; SUBPAGE : STRING : ABSTRACT ; CSUBTYPE : STRING : CONFERENCE. url: <https://dl.acm.org/doi/pdf/10.1145/3573428.3573615>.
- [37] Jianzhong Yang, Huirong Chen, Shengnan Xiong, Zhongqiang Yang e Yu Jiang. «Custom Data Mining Association Rules of Nixing Pottery Product Recommendation System». Em: (2020). doi: 10.1145/3443467.3443734. url: <https://doi.org/10.1145/3443467.3443734>.
- [38] Shuang Zhou. «Research on Online Shopping Recommendation Algorithm Based on Apriori Algorithm with Association Rules». Em: 7 (2024). doi: 10.1145/3687311.3687370. url: <https://doi.org/10.1145/3687311.3687370>.
- [39] Sumitra Kongpeng e Anantaporn Hanskunatai. «Tourist Destination Recommendation System based on Machine Learning». Em: 10 (2024). doi: 10.1145/3695220.3695229. url: <https://doi.org/10.1145/3695220.3695229>.
- [40] *TripAdvisor API Overview*. url: <https://tripadvisor-content-api.readme.io/reference/overview>.
- [41] Patrícia Alves, Domingos Gomes, Catarina Rodrigues, João Carneiro, Paulo Novais e Goretí Marreiros. «Grouplanner: A Group Recommender System for Tourism with Multi-agent MicroServices». Em: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13616 LNAI (2022), pp. 454–460. issn: 16113349. doi: 10.1007/978-3-031-18192-4_37.
- [42] *Sistrade Software de Gestão Empresarial | MIS|ERP|MES*. url: <https://www.sistrade.com/>.

-
- [43] *IPMA - api*. url: <https://api.ipma.pt/>.
- [44] *Registos de qualidade do ar (rede de sensores da Porto Digital) - Conjunto de Dados - Portal de Dados*. url: <https://opendata.porto.digital/dataset/urban-platform-air-quality>.
- [45] *Registos de monitorização de ruído (rede sensores da Porto Digital) - Conjunto de Dados - Portal de Dados*. url: <https://opendata.porto.digital/dataset/urban-platform-noise-level>.
- [46] *Documentação da Plataforma Google Maps | Places API | Google for Developers*. url: <https://developers.google.com/maps/documentation/places/web-service?hl=pt-br>.
- [47] Patrícia Alves, Helena Martins, Pedro Saraiva, João Carneiro, Paulo Novais e Goreti Marreiros. «Group recommender systems for tourism: how does personality predict preferences for attractions, travel motivations, preferences and concerns?» Em: *User Modeling and User-Adapted Interaction* 33 (5 nov. de 2023), pp. 1141–1210. issn: 15731391. doi: 10.1007/S11257-023-09361-2/TABLES/4. url: <https://link.springer.com/article/10.1007/s11257-023-09361-2>.
- [48] *Índices de qualidade do ar | Agência Portuguesa do Ambiente*. url: <https://apambiente.pt/ar-e-ruído/indices-de-qualidade-do-ar>.
- [49] *Ruído ambiente | Agência Portuguesa do Ambiente*. url: <https://apambiente.pt/ar-e-ruído/ruído-ambiente>.