INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




# Clustering Autónomo Otimizado para Navegação Inteligente com recurso a Inteligência Artificial

**LUCAS SANTOS GUIMARÃES**
Junho de 2025

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

# Clustering Autónomo Otimizado para Navegação Inteligente com recurso a Inteligência Artificial

**Lucas Santos Guimarães**

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

**Orientador: Marílio Cardoso**

# 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.

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

ISEP, Porto, 29 de junho de 2025

# Resumo

Com o crescimento exponencial do *e-commerce*, a organização eficiente dos catálogos de produtos tornou-se um desafio crucial para melhorar a experiência dos clientes. A personalização das recomendações de categorias e filtros assume um papel fundamental para facilitar a navegação e a descoberta de produtos relevantes, e assim aumentar a satisfação e a fidelização dos clientes. Este trabalho, desenvolvido para a Flamingo S.A., propõe uma solução inteligente que combina técnicas de análise de dados, aprendizagem automática e *clustering* para gerar recomendações personalizadas de categorias ou filtros, adaptadas ao perfil e histórico de interação de cada cliente.

Para isso, foram implementados algoritmos que exploram diferentes abordagens, desde a popularidade geral das categorias ou filtros, passando pela similaridade entre perfis de clientes, até à segmentação baseada em clusters. O sistema também incorpora métricas de avaliação que permitem medir a qualidade das sugestões de forma a avaliar a qualidade das recomendações. A *framework* desenvolvida foi testada com recurso a testes unitários e de integração, para assegurar a fiabilidade dos métodos e a correta interação entre os componentes.

Esta abordagem contribui para a melhoria da experiência de compra, por oferecer aos clientes recomendações mais acertadas e um acesso mais intuitivo aos produtos de seu interesse, o que pode potencialmente aumentar as taxas de conversão e a retenção na plataforma.

**Palavras-chave**: Inteligência Artificial, Aprendizagem automática, Personalização, Categorização Dinâmica, E-commerce, Experiência do cliente

# Abstract

With the exponential growth of e-commerce, the efficient organization of product catalogs has become a crucial challenge in enhancing customer experience. The personalization of category and filter recommendations plays a key role in facilitating navigation and the discovery of relevant products, thereby increasing customer satisfaction and loyalty. This work, developed for Flamingo S.A., proposes an intelligent solution that combines data analysis, machine learning, and clustering techniques to generate personalized category or filter recommendations tailored to each customer's profile and interaction history.

To achieve this, algorithms were implemented that explore different approaches, ranging from the overall popularity of categories or filters to customer profile similarity, and cluster-based segmentation. The system also incorporates evaluation metrics to assess the quality of the suggestions and the effectiveness of the recommendations. The developed framework was tested using unit and integration tests to ensure the reliability of the methods and the correct interaction between components.

This approach contributes to improving the shopping experience by offering customers more accurate recommendations and more intuitive access to products of interest, which can potentially increase conversion and retention rates on the platform.

**Keywords**: Artificial Intelligence, Machine Learning, Personalization, Dynamic Categorization, E-commerce, Customer Experience

# Agradecimentos

Gostaria de expressar o meu agradecimento à empresa Flamingo por me ter proporcionado a oportunidade de realizar este estágio. Foi uma experiência enriquecedora, que me permitiu aplicar os conhecimentos adquiridos ao longo do curso e desenvolver novas competências.

Gostaria de agradecer também ao meu supervisor na empresa, Rui Marques, pelo acompanhamento constante, pela partilha de conhecimento e pela disponibilidade ao longo de todo o estágio.

Por fim, gostaria de expressar o meu sincero agradecimento ao Professor Marílio Cardoso pela sua orientação, apoio e disponibilidade ao longo do meu estágio. A sua experiência, paciência e dedicação foram essenciais para o sucesso desta parte do projeto. A sua orientação não só me ajudou a superar os desafios enfrentados, mas também me motivou durante estes semestres. Agradeço por todo o conhecimento compartilhado. Foi uma honra trabalhar sob a sua orientação.

# Índice

**1 Introdução ..... 19**

1.1 Contexto ..... 19

1.2 Problema ..... 19

1.3 Motivação e objetivos ..... 20

1.4 Considerações éticas ..... 20

1.5 Metodologia ..... 21

1.6 Planeamento ..... 22

1.7 Estrutura do relatório ..... 23

**2 Estado da arte ..... 25**

2.1 Personalização em e-commerce ..... 25
- 2.1.1 Técnicas básicas ..... 26
- 2.1.2 Integração de inteligência artificial e *machine learning* ..... 26
- 2.1.3 Benefícios e desafios ..... 27

2.2 Segmentação de utilizadores ..... 27
- 2.2.1 Modelos de segmentação ..... 27
- 2.2.2 Integração de inteligência artificial e *machine learning* ..... 28
- 2.2.3 Benefícios e desafios ..... 28

2.3 Categorização dinâmica de produtos ..... 28
- 2.3.1 Tecnologias principais ..... 29
- 2.3.2 Benefícios e desafios ..... 29

2.4 Técnicas de clustering ..... 30
- 2.4.1 K-means ..... 31
- 2.4.2 Fuzzy c-means ..... 31
- 2.4.3 Gaussian Mixture ..... 32
- 2.4.4 Spectral Clustering ..... 32
- 2.4.5 DBSCAN ..... 32

2.5 Desafios relativos à quantidade de dados ..... 33
- 2.5.1 Cold Start ..... 33
- 2.5.2 Black / Grey sheep ..... 33
- 2.5.3 Sparcity ..... 33

2.6 Métricas para avaliação do impacto ..... 34
- 2.6.1 Taxa de cliques ..... 34
- 2.6.2 Taxa de conversão ..... 34
- 2.6.3 Valor médio ao pedido ..... 35
- 2.6.4 Retenção de utilizadores e taxa de recompra ..... 35
- 2.6.5 Taxa de abandono do carrinho ..... 35

2.7 Estudo de casos e aplicações reais ..... 36
- 2.7.1 Otimização com Modelos de Deep learning ..... 36
- 2.7.2 Segmentação Dinâmica com *Machine Learning* ..... 36

2.7.3 Estratégias de Priorização Personalizada ..... 36
2.7.4 Integração de AI em websites de e-commerce ..... 37
2.8 Tecnologias Existentes / Emergentes ..... 37
2.8.1 Python ..... 37
2.8.2 R ..... 39
2.8.3 Java ..... 39
2.8.4 Tecnologias Emergentes ..... 40
**3 Análise da solução ..... 43**
3.1 Domínio do problema ..... 43
3.2 Requisitos funcionais e não funcionais ..... 44
3.2.1 Requisitos funcionais ..... 44
3.2.2 Requisitos não funcionais ..... 45
3.2.3 Pressupostos ..... 45
**4 Design da solução ..... 47**
4.1 Nível 1 ..... 47
4.2 Nível 2 ..... 49
4.3 Nível 3 ..... 51
**5 Descrição da implementação ..... 55**
5.1 Requisitos do sistema ..... 56
5.2 Processo de obtenção de recomendações ..... 57
5.2.1 Pré-processamento dos dados ..... 57
5.2.2 Algoritmos de recomendação ..... 57
5.2.3 Avaliação do desempenho ..... 58
5.2.4 Cálculo de probabilidade e margem de erro ..... 59
5.3 Base de dados ..... 59
5.4 Interface gráfica ..... 60
5.5 Testes ..... 62
5.5.1 Testes unitários ..... 62
5.5.2 Testes de integração ..... 63
**6 Avaliação da solução ..... 65**
**7 Conclusão ..... 71**
7.1 Objetivos concretizados ..... 71
7.2 Limitações e trabalho futuro ..... 72
7.3 Apreciação final ..... 72
**Referências ..... 73**
**Anexo A - WBS ..... 77**

**Anexo B - Diagrama de Gantt ..... 78**

**Anexo C – Gráfico de Gantt Inputs ..... 81**

**Anexo D – Matriz de riscos ..... 85**

# Lista de Figuras

Figura 1 - Excerto do diagrama de Gantt ..... 22
Figura 2 - Filtragem Colaborativa vs Filtragem Baseada em Conteúdo [7] ..... 26
Figura 3 - Modelo de domínio ..... 43
Figura 4 - Diagrama de casos de uso ..... 48
Figura 5 - Nível 1 vista lógica ..... 48
Figura 6 - Nível 1 vista de processo ..... 49
Figura 7 - Nível 2 vista lógica ..... 49
Figura 8 - Nível 2 vista de implementação ..... 50
Figura 9 - Nível 2 vista física ..... 50
Figura 10 - Nível 2 vista processo ..... 51
Figura 11 - Nível 3 vista lógica ..... 52
Figura 12 - Nível 3 vista de implementação ..... 53
Figura 13 - Nível 3 vista de processo ..... 54
Figura 14 - Modelo relacional ..... 60
Figura 15 - Interface para obter as recomendações ..... 61
Figura 16 - Categorias interagidas ..... 61
Figura 17 - Output dos algoritmos ..... 61
Figura 18 - Recomendação final ..... 61
Figura 19 - Inputs para a geração de recomendações ..... 65
Figura 20 - Visualização dos outputs parte 1 ..... 66
Figura 21 - Visualização dos outputs parte 2 ..... 67
Figura 22 - WBS Preparação para a dissertação ..... 77
Figura 23 - WBS Dissertação ..... 77
Figura 24 - Diagrama de Gantt parte 1 ..... 78
Figura 25 - Diagrama de Gantt parte 2 ..... 79
Figura 26 - Diagrama de Gantt parte 3 ..... 79
Figura 27 - Diagrama de Gantt parte 4 ..... 80
Figura 28 - Diagrama de Gantt parte 5 ..... 80

# Lista de Equações

Equação 1 – K-means ..... 31
Equação 2 – Fuzzy c-means ..... 31
Equação 3 – Gaussian Mixture ..... 32
Equação 4 – Taxa de cliques ..... 34
Equação 5 – Taxa de conversão ..... 34
Equação 6 – Valor médio ao pedido ..... 35
Equação 7 – Taxa de retenção de clientes ..... 35
Equação 8 – Taxa de abandono do carrinho ..... 35
Equação 9 – Margem de erro ..... 59

# Lista de Tabelas

Tabela 1 - Requisitos funcionais ..... 44
Tabela 2 - Requisitos não funcionais ..... 45
Tabela 3 - Testes de integração ..... 63
Tabela 4 - F1-score dos algoritmos (clientes >10 interações) ..... 68
Tabela 5 - F1-score dos algoritmos (clientes >5 interações) ..... 68
Tabela 6 - F1-score dos algoritmos sem pré-processamento aplicado ..... 69
Tabela 7 - Tabela de inputs (parte 1) ..... 81
Tabela 8 - Tabela de inputs (parte 2) ..... 82
Tabela 9 - Tabela de inputs (parte 3) ..... 83
Tabela 10 - Tabela de inputs (parte 4) ..... 84
Tabela 11 - Matriz de riscos ..... 85

# Acrónimos e Símbolos

## Lista de Acrónimos

| | |
|---|---|
| **AI** | Inteligência artificial (do inglês *Artificial Inteligence*) |
| **AOV** | Valor médio ao pedido (do inglês *Average Order Value*) |
| **AutoML** | *Automated Machine Learning* |
| **BERT** | *Bidirectional Encoder Representations from Transformers* |
| **CBF** | Filtragem baseada em conteúdo (do inglês *Content-Based Filtering*) |
| **CF** | Filtragem colaborativa (do inglês *Collaborative Filtering*) |
| **CNN** | Redes neuronais convulsionais (do inglês *Convulsional Neural Networks*) |
| **CRR** | Taxa de Retenção de Clientes (do inglês *Customer Retention Rate*) |
| **CTR** | Taxa de cliques (do inglês *Click-Through Rate*) |
| **CVR** | Taxa de conversão (do inglês *Conversion Rate*) |
| **GNNs** | *Graph Neural Networks* |
| **GPDR** | Regulamento Geral sobre a Proteção de Dados (do inglês *General Data Protection Regulation*) |
| **GPT** | *Generative Pretrained Transformer* |
| **LIME** | *Local Interpretable Model-agnostic Explanations* |
| **LLM** | *Large Language Models* |
| **ML** | Aprendizagem automática (do inglês *Machine Learning*) |
| **NLP** | Processamento de linguagem natural (do inglês *Natural Language Processing*) |
| **NLTK** | *Natural Language Toolkit* |
| **PCA** | *Principal Component Analysis* |
| **RFM** | *Recency, Frequency and Monetary* |
| **RPR** | Taxa de Recompra (do inglês *Repeat Purchase Rate*) |
| **SHAP** | *Shapley Additive Explanations* |

**WBS** *Work Breakdown Structure*

**XAI** *Explainable AI*

# 1 Introdução

Este capítulo apresenta o tema de estudo desenvolvido no âmbito da unidade curricular Dissertação (DIMEI) do Mestrado de Engenharia Informática (MEI), lecionada no Instituto Superior de Engenharia do Porto (ISEP). Serão explorados o contexto e a problemática que deram origem a este projeto, bem como a definição dos objetivos a serem alcançados com a solução proposta, as considerações éticas associadas e uma descrição da organização adotada neste documento.

## 1.1 Contexto

A *Flamingo S.A.* é uma multinacional, fundada em 1976, com mais de 40 anos de experiência nos mercados de luxo, especializada na produção de joias, acessórios de moda e artigos de decoração. Reconhecida pela sua excelência na criação de peças exclusivas e de alta qualidade, a empresa é a maior fabricante ibérica de joias em prata e ouro, bem como de pratas decorativas. A *Flamingo S.A.* mantém filiais em Portugal, Espanha e Itália, e oferece a mais vasta gama de produtos no mercado português, abrangendo desde artigos de joalharia e adorno pessoal até peças decorativas [1].

Com um mercado em constante transformação e a crescente importância das plataformas digitais, a *Flamingo S.A*. expandiu a sua atuação através da *Flamingo Tech*, uma ramificação tecnológica dedicada ao desenvolvimento e gestão das soluções digitais da marca, incluindo o seu *website* de vendas. A *Flamingo Tech* desempenha um papel essencial na operação e na evolução do *e-commerce* da empresa, com o objetivo de garantir que a experiência de compra *online* seja fluida, intuitiva e personalizada para os clientes.

## 1.2 Problema

Apesar do sucesso da transformação digital, o crescimento contínuo do catálogo de produtos oferecidos pela Flamingo S.A. acarretou desafios significativos relacionados com a organização e a navegação no *website*. A necessidade de categorizar os produtos de forma eficiente, tendo em consideração as expectativas dos utilizadores, tornou-se um problema relevante. Questões

como a quantidade de categorias necessárias, a definição de hierarquias adequadas e a personalização das categorias para atender diferentes perfis de utilizadores representam obstáculos que limitam a experiência de navegação.

## 1.3 Motivação e objetivos

Os desafios descritos na secção 1.2 evidenciam a necessidade de desenvolver uma solução inovadora que melhore significativamente a organização e a experiência de navegação do site de *e-commerce* da *Flamingo S.A.*

Atualmente, a *Flamingo S.A.* possui uma vasta gama de produtos, mas a categorização e navegação no *website* nem sempre refletem a diversidade e as necessidades dos utilizadores. Isso cria obstáculos para os clientes na identificação de produtos relevantes e impacta negativamente a experiência de compra. A motivação para este projeto é, portanto, a criação de uma solução que utilize inteligência artificial para oferecer categorização otimizada, alinhada às preferências e comportamentos dos utilizadores.

Como metas, foram estabelecidos os seguintes objetivos principais para a solução:

1. Descoberta de Conhecimento de Temas Agregadores: Identificar, através de pesquisa e análise, as melhores práticas, ferramentas, tecnologias e abordagens para categorizar produtos de forma eficiente.

2. Personalização de Categorias: Desenvolver um sistema que permita que a categorização seja adaptada dinamicamente com base em fatores como popularidade dos produtos, localização geográfica dos utilizadores e ciclo de vida dos itens.

3. Definição de Priorização: Estruturar hierarquias entre categorias e filtros de forma personalizada, garantindo que as categorias ou filtros mais relevantes para cada cliente sejam facilmente acessíveis.

4. Testes e Validação de Cenários: Realizar testes demonstrativos com dados reais de utilização para validar a eficácia e usabilidade da solução.

5. Integração com o Sistema Atual: Garantir que a solução proposta possa ser facilmente integrada no *website* existente da *Flamingo Tech.*

## 1.4 Considerações éticas

As considerações éticas são os princípios, valores e diretrizes que orientam entidades na tomada de decisões e ações moralmente responsáveis. A aplicação de inteligência artificial na categorização de produtos deve ser conduzida de forma responsável, protegendo os direitos dos utilizadores e alinhando-se com as diretrizes éticas e legais aplicáveis [2]. No âmbito deste trabalho, destacam-se os seguintes pontos éticos:

1. Consentimento Informado: A *Flamingo Tech* garante que os dados utilizados para este projeto foram obtidos e tratados de acordo com as autorizações necessárias, exclusivamente para o desenvolvimento da solução proposta. Todos os dados utilizados

são de propriedade da empresa, e o projeto terá uso interno, sem impacto direto nos utilizadores externos durante a sua fase de desenvolvimento.

2. Confidencialidade e Privacidade dos Dados: Para proteger a privacidade dos clientes da *Flamingo S.A.*, qualquer dado pessoal ou informação desnecessária para o desenvolvimento da solução foi ignorado e descartado.

3. Qualidade e Confiabilidade dos Dados: Para assegurar que os algoritmos de inteligência artificial gerem resultados precisos, será realizada uma avaliação rigorosa da qualidade dos dados. Decisões baseadas em dados inadequados podem impactar negativamente a experiência dos utilizadores, por isso, qualquer limitação na qualidade das informações será comunicada de forma clara e transparente.

4. Evitação de Algoritmos viciosos: Uma consideração especial será dada à garantia de que os algoritmos desenvolvidos não introduzam ou reforcem preconceitos. Qualquer padrão de categorização deverá ser monitorizado para evitar discriminação ou segmentação injusta, promovendo uma experiência igualitária para todos os utilizadores.

5. Monitorização e Melhoria Contínua: Após a implementação da solução, a *Flamingo Tech* irá adotar mecanismos contínuos de monitorização e avaliação, assegurando que a ferramenta permaneça eficaz e relevante ao longo do tempo. Este processo permitirá a identificação de áreas de melhoria e a adaptação da solução às necessidades em constante evolução dos utilizadores e da empresa.

## 1.5 Metodologia

O desenvolvimento desta dissertação segue uma abordagem estruturada, dividida em várias fases, alinhadas com as práticas lecionadas na unidade curricular de DIMEI e adaptadas às especificidades do projeto da *Flamingo Tech*. Estas fases visam garantir uma organização eficiente e um progresso consistente, conforme descrito abaixo:

1. Análise: Pesquisa, levantamento de requisitos e escolha de tecnologias;

2. Desenho: Planeamento e definição da arquitetura da solução;

3. Implementação: Desenvolvimento da solução;

4. Testes: Análise da eficácia e usabilidade da solução;

5. Documentação: Escrita deste documento de forma contínua.

A fase de análise consiste na compreensão detalhada do problema, levantamento dos requisitos funcionais e não funcionais, e no estudo do estado da arte. Durante esta etapa, será realizada uma pesquisa sobre técnicas de categorização e personalização com inteligência artificial, bem como uma análise de ferramentas e tecnologias aplicáveis.

Na fase de desenho, serão definidas as tecnologias e a arquitetura da solução. Esta etapa incluirá o desenho de fluxos de dados.

A implementação será realizada com base no planeamento da fase anterior. Esta etapa será dividida em módulos, correspondendo aos objetivos definidos na secção 1.3. O desenvolvimento incluirá a criação de um protótipo funcional,

A fase de testes será realizada a par com a fase de implementação para garantir a qualidade e o cumprimento dos requisitos.

A documentação do projeto será elaborada ao longo de todas as fases, culminando na redação final da dissertação.

## 1.6 Planeamento

O planeamento deste projeto foi realizado com base em práticas de gestão de projetos, seguindo uma sequência estruturada de elaboração e documentação de elementos fundamentais. Primeiramente, foi elaborado o termo de abertura do projeto, ou *Project Charter*, que estabelece os principais objetivos, entregáveis, partes interessadas, além de definir a visão e a justificativa do projeto. De seguida, foi contruída a *Work Breakdown Structure* (WBS). Este passo teve como objetivo detalhar todas as atividades necessárias ao projeto, subdividindo os entregáveis em componentes menores e mais práticos (Anexo 1).

Com base na WBS, como mostra a Figura 1, foi desenvolvido um planeamento geral (com mais detalhe no Anexo 2). Este documento inclui um cronograma detalhado que integra prazos, recursos e as dependências entre as tarefas (Anexo 3). Este cronograma levou em consideração a complexidade das atividades e as minhas capacidades de desenvolvimento, assegurando a gestão eficiente do tempo e dos recursos disponíveis.

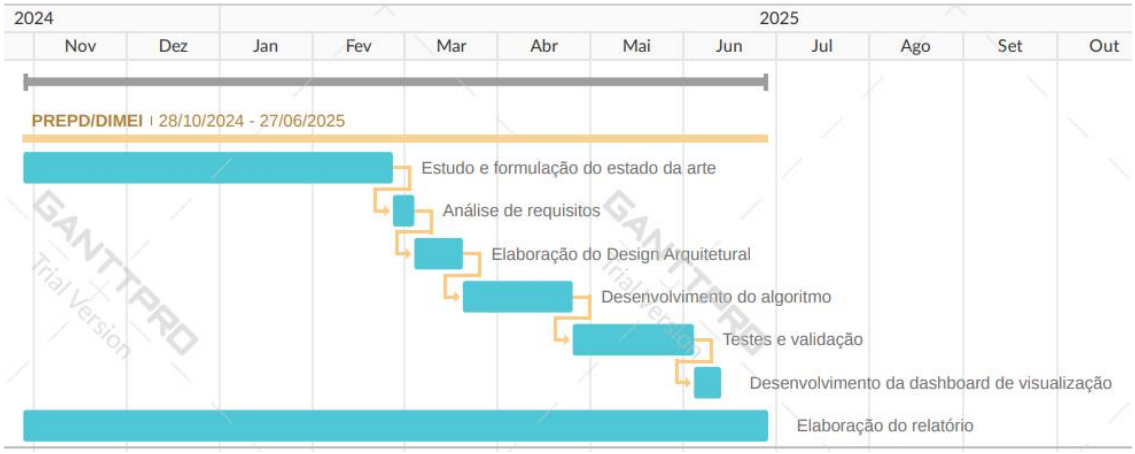


Figura 1 - Excerto do diagrama de Gantt

Foi realizado também um diagnóstico de competências que analisou diversas áreas, como trabalho em equipa, comunicação, gestão de stress e criatividade, entre outras. No entanto, decidi não incluir no projeto devido à sua pouca relevância prática para os objetivos específicos. Exemplos incluem "Capacidade para assumir riscos", cuja importância foi avaliada como baixa (1), e "Aprendizagem ao longo da vida", que também apresentou baixa prioridade (2). Essas competências, embora úteis em contextos gerais, não são consideradas prioritárias para este contexto específico.

Por fim, foi desenvolvida uma matriz de riscos (Anexo 4), com a finalidade de identificar, analisar e priorizar possíveis eventos que podem impactar o projeto. Cada risco foi avaliado de acordo com a probabilidade de ocorrência e o impacto, e foram definidos planos de resposta para mitigação ou contingência, conforme necessário.

## 1.7 Estrutura do relatório

Este relatório é composto por sete capítulos que abrangem desde a contextualização inicial até as conclusões finais do projeto.

No segundo capítulo, intitulado Estado da Arte, é realizada uma revisão detalhada das técnicas, ferramentas e algoritmos relevantes para a categorização automatizada de produtos e personalização em *e-commerce*. Além disso, são exploradas as abordagens mais recentes no uso de inteligência artificial aplicada a problemas similares.

No terceiro capítulo, intitulado Análise, é apresentada a relevância do projeto do ponto de vista do negócio e do utilizador. Este capítulo também inclui a definição dos requisitos funcionais e não funcionais, assim como uma análise detalhada dos desafios técnicos e éticos.

No quarto capítulo, intitulado Desenho da Solução, é descrito a arquitetura e o design da solução proposta.

No quinto capítulo, intitulado Implementação da Solução, são apresentados os detalhes do desenvolvimento do sistema, incluindo as ferramentas e técnicas utilizadas, os desafios enfrentados e os resultados parciais obtidos durante esta etapa.

No sexto capítulo, intitulado Avaliação da Solução, são descritos os métodos de validação e avaliação utilizados para medir a eficácia da solução. Este capítulo também apresenta os resultados obtidos e uma análise crítica do desempenho do sistema em cenários reais de uso.

No sétimo e último capítulo, intitulado Conclusões, são discutidos os principais resultados alcançados, a concretização dos objetivos estabelecidos e o impacto do projeto. Este capítulo também inclui sugestões para trabalhos futuros e possíveis melhorias na solução proposta.

# 2 Estado da arte

Este capítulo apresenta uma revisão abrangente das principais abordagens, técnicas e tecnologias relacionadas com a personalização e categorização no contexto de *e-commerce*. Esta análise explora o conhecimento e as inovações tecnológicas atuais que servem de apoio à organização dinâmica e eficiente de produtos num *website*.

Inicialmente, são discutidas as estratégias de personalização, destacando como algoritmos baseados em *machine learning* (ML) e inteligência artificial (AI) têm revolucionado a experiência do utilizador, por oferecer recomendações mais precisas e alinhadas às suas preferências. Em seguida, é explorada a segmentação de utilizadores, uma prática usada para entender as diversas necessidades e comportamentos dos utilizadores, que contribui para interações mais direcionadas e eficazes. O capítulo também discute o tema categorização dinâmica de produtos, por apresentar técnicas que utilizam processamento de linguagem natural (NLP), aprendizagem multimodal e modelos avançados de *deep learning* para ajustar categorias em tempo real. Por fim, são discutidas métricas de impacto utilizadas para avaliar a eficácia dessas estratégias.

## 2.1 Personalização em e-commerce

O crescimento exponencial do *e-commerce* intensificou a concorrência entre plataformas, o que levou as empresas a adotarem estratégias que garantam a retenção e satisfação dos clientes. A personalização no *e-commerce* é o ato de adaptar a experiência de compra às preferências, aos comportamentos e às necessidades individuais do cliente. Desde recomendações de produtos até interfaces adaptáveis, a personalização não apenas melhora a satisfação do utilizador, mas também promove a fidelização à marca e aumenta as taxas de conversão [3] [4].

A atual personalização aproveita algoritmos de ML, processamento de dados em tempo real e técnicas de filtragem colaborativa para prever com precisão as necessidades do utilizador. Estes sistemas têm como objetivo melhorar a experiência do utilizador, sugerindo produtos relevantes, otimizando os resultados da pesquisa e ajustando dinamicamente o conteúdo promocional [5] [6].

Os mecanismos de personalização são uma das ferramentas usadas no *e-commerce* para adaptar a experiência de compra às preferências e comportamentos dos utilizadores. Este

tópico explora as principais abordagens utilizadas, desde técnicas tradicionais, métodos avançados que integram AI e ML para melhorar a precisão e a relevância das recomendações. Além disso, são discutidos os benefícios proporcionados e os desafios enfrentados na implementação dessas estratégias.

### 2.1.1 Técnicas básicas

A personalização no *e-commerce* baseia-se principalmente em três técnicas fundamentais. Como demonstrado na Figura 2, a Filtragem Colaborativa (CF) recomenda produtos com base nos comportamentos e preferências semelhantes entre utilizadores, a Filtragem Baseada em Conteúdo (CBF) utiliza os atributos dos produtos para sugerir produtos similares aos já visualizados ou adquiridos e, para aumentar a precisão das recomendações, alguns sistemas adotam modelos híbridos que combinam CF e CBF [6].

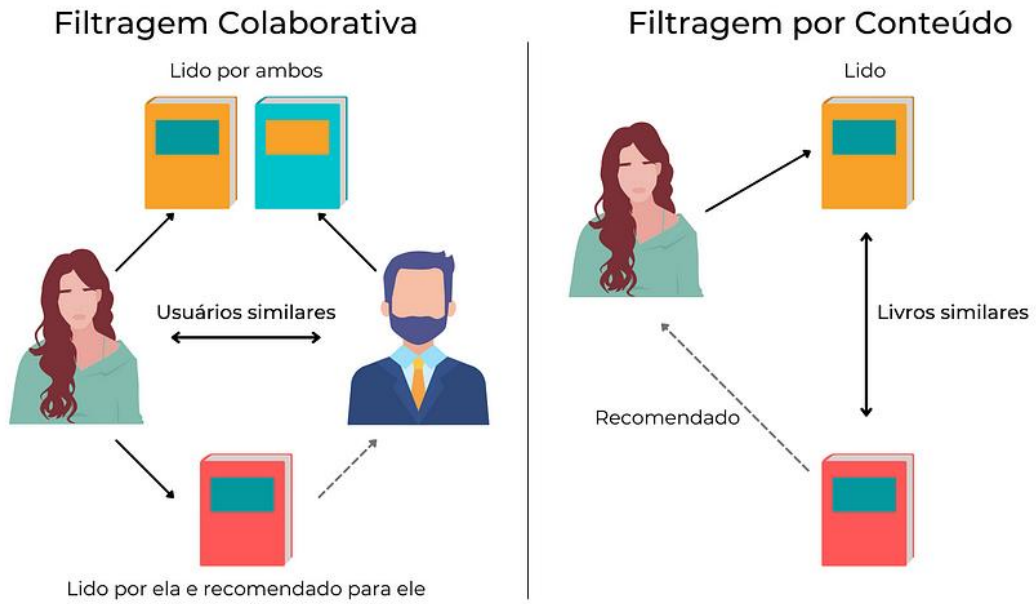


Figura 2 - Filtragem Colaborativa vs Filtragem Baseada em Conteúdo [7]

### 2.1.2 Integração de inteligência artificial e *machine learning*

A integração da AI e do ML no *e-commerce* permite a análise de grandes volumes de dados de forma a extrair informações úteis. Técnicas avançadas, como factoração de matrizes e redes neuronais convolucionais (CNN), são utilizadas para melhorar a precisão das recomendações. A factoração de matrizes decompõe grandes matrizes de interações entre utilizadores e produtos em componentes menores, o que permite identificar padrões latentes, como preferências implícitas e afinidades entre utilizadores e produtos. Já as CNNs são usadas para processar dados estruturados em formato de grade, como imagens ou descrições de produtos, extraindo características visuais e textuais relevantes para categorizações ou recomendações. Estas abordagens são particularmente eficazes em lidar com desafios como dados dispersos e preferências dinâmicas dos utilizadores [4] [6].

### 2.1.3 Benefícios e desafios

A personalização oferece vários benefícios, como uma maior retenção de clientes, uma vez que recomendações relevantes promovem a fidelidade, e taxas de conversão aprimoradas [5], devido a interfaces personalizadas que se alinham às necessidades específicas de cada utilizador. Contudo, ainda existem desafios significativos, como preocupações relacionadas à privacidade, imprecisões nos algoritmos e a complexidade técnica envolvida no processamento de dados em tempo real [3] [4].

## 2.2 Segmentação de utilizadores

A personalização no *e-commerce* revolucionou as compras *online*, o que permitiu que as empresas ofereçam experiências personalizadas que demonstram impacto na vida de cada utilizador individualmente. No entanto, a sua eficácia depende fortemente da compreensão dos diversos segmentos de clientes. Nesta secção, a segmentação de utilizadores aprofunda a categorização do público com base em atributos comportamentais, demográficos e psicográficos, refinando ainda mais as estratégias de personalização [3] [4].

No domínio competitivo do *e-commerce*, compreender a diversidade dos clientes é fundamental para a elaboração de estratégias de *marketing* eficazes e personalizadas. A segmentação de utilizadores consiste em dividir os clientes em grupos homogéneos por utilizar fatores como dados demográficos, histórico de compras e comportamentos online. Esta abordagem para além de ajudar as empresas a alocar recursos de forma eficiente, também aumenta a satisfação do cliente e a geração de lucro [8].

Os avanços no ML e na capacidade de análise de grandes porções de dados tornaram a segmentação mais dinâmica e precisa. Algoritmos como *clustering K-means* e árvores de decisão são bastante usados para identificar padrões em dados de consumidores, facilitando a previsão de comportamentos e preferências [9].

As técnicas e aplicações de segmentação são ferramentas utilizadas no *e-commerce* para compreender e classificar os clientes de forma mais eficaz. Este tópico aborda os principais modelos de segmentação, que incluem abordagens baseadas em dados comportamentais, psicográficos e demográficos, bem como a integração de AI e ML para aumentar a precisão e dinamismo do processo. Além disso, são apresentados os benefícios dessas práticas, como a personalização aprimorada e a maximização de resultados, bem como os desafios que ainda precisam ser enfrentados, como a qualidade dos dados e questões de privacidade.

### 2.2.1 Modelos de segmentação

Os modelos de segmentação utilizados em *e-commerce* são ferramentas usadas para compreender e classificar os clientes de uma forma mais eficaz. A Análise RFM *(Recency, Frequency, Monetary*) avalia três dimensões principais: tempo para o retorno, frequência de interações e valor monetário. Essa abordagem é útil para identificar os clientes mais valiosos, que mais contribuem para a receita de uma empresa, o que permite uma alocação mais eficiente dos recursos [9]. Já a segmentação comportamental concentra-se nas ações dos utilizadores, como hábitos de navegação e motivos de compra, ajudando a prever necessidades

futuras e personalizar ofertas com base em comportamentos reais [8]. A segmentação psicográfica e demográfica, por sua vez, vai além dos padrões de compra, combinando atributos de estilo de vida, interesses e informações demográficas para criar uma análise mais abrangente e orientada a campanhas específicas [9].

### 2.2.2 Integração de inteligência artificial e *machine learning*

A integração da AI e do ML tem revolucionado a segmentação de clientes, tornando-a mais precisa e dinâmica. Algoritmos de *clustering* como o *K-means* agrupam utilizadores com base na proximidade num espaço multidimensional, enquanto árvores de decisão e *random forest* são capazes de criar regras compreensíveis para a segmentação. Redes neuronais e técnicas de *deep learning* também têm sido utilizadas para detetar padrões mais complexos em dados não estruturados, como interações com o *website* ou dados textuais de avaliações de clientes [10].

### 2.2.3 Benefícios e desafios

Os benefícios dessas abordagens são significativos. Elas não apenas melhoram a segmentação de clientes, mas também permitem a criação de experiências personalizadas, aumentando a fidelidade e a satisfação do cliente. Campanhas de *marketing* tornam-se mais relevantes e eficazes, maximizando o retorno sobre o investimento. Outro benefício é a possibilidade de identificar novos nichos de mercado e expandir a base de clientes. Além disso, a análise aprofundada dos dados pode ajudar a prever comportamentos futuros, permitindo decisões estratégicas proativas [8] [9].

Apesar das vantagens, existem desafios importantes. A qualidade dos dados continua a ser um fator crítico, já que dados incompletos ou inconsistentes podem comprometer a eficácia dos modelos. As preocupações com a privacidade dos utilizadores também representam um obstáculo, especialmente com a introdução de regulamentações como o Regulamento Geral sobre a Proteção de Dados (GDPR). Por fim, o custo computacional do processamento de grandes volumes de dados é outro desafio, que exige infraestrutura adequada e investimento em tecnologia [10].

## 2.3 Categorização dinâmica de produtos

A segmentação de utilizadores ajuda a compreensão de diversos grupos de clientes, o que permite a implementação de estratégias direcionadas e interações personalizadas. No entanto, para aproveitar totalmente as informações da segmentação, a categorização dinâmica de produtos torna-se um aspeto relevante. Este passo na otimização do *e-commerce* garante que os agrupamentos de produtos evoluam em resposta ao comportamento do utilizador em tempo real, às tendências sazonais e à dinâmica do mercado. Ao integrar os resultados da segmentação em sistemas de categorização dinâmicos, as empresas podem criar exibições de produtos adaptáveis que melhoram a navegação e a descoberta, o que gera uma maior taxa de envolvimento e conversão [8] [9] [10].

No *e-commerce*, a categorização eficaz de produtos garante que os utilizadores possam encontrar rapidamente artigos de interesse e, ao mesmo tempo, que permite às plataformas gerir stocks extensos de forma eficiente. Os métodos tradicionais de categorização, que

dependem de hierarquias fixas, muitas vezes não conseguem adaptar-se à natureza dinâmica dos mercados *online*. A categorização dinâmica de produtos aborda estas limitações aproveitando o ML e o NLP para classificar automaticamente os produtos com base em critérios não estáticos, como as preferências do utilizador e as semelhanças semânticas.

A integração de técnicas multimodais avançadas, que combinam dados textuais e visuais, melhorou significativamente a precisão da categorização dinâmica. Esta abordagem é particularmente benéfica para grandes plataformas, onde a classificação manual é impraticável [11] [12].

As técnicas e aplicações de categorização são ferramentas utilizadas no *e-commerce* para compreender e classificar os clientes de forma mais eficaz. Este tópico aborda os principais modelos de segmentação, que incluem abordagens baseadas em dados comportamentais, psicográficos e demográficos, bem como a integração de AI e ML para aumentar a precisão e dinamismo do processo. Além disso, são apresentados os benefícios dessas práticas, como a personalização aprimorada e a maximização de resultados, bem como os desafios que ainda precisam ser enfrentados, como a qualidade dos dados e questões de privacidade.

### 2.3.1 Tecnologias principais

As tecnologias utilizadas na categorização dinâmica de produtos em e-commerce desempenham um papel importante na organização e personalização das experiências de compra. Entre as principais tecnologias, destaca-se o NLP, que analisa descrições de produtos e avaliações de utilizadores para refinar as categorias de forma dinâmica e precisa [11]. Outra tecnologia é a aprendizagem multimodal combina dados de texto e imagem e utiliza técnicas de fusão hierárquica para melhorar a precisão da classificação [13]. Ainda, modelos de *deep learning*, como o BERT e o GPT, também têm-se mostrado eficazes na compreensão semântica e na anotação automática de categorias, o que permite uma categorização mais contextual e relevante [12] [13].

### 2.3.2 Benefícios e desafios

A categorização dinâmica oferece alguns benefícios que contribuem diretamente para a experiência do utilizador e a eficiência operacional das plataformas de *e-commerce*. Um dos principais aspetos positivos é a pesquisa e navegação melhoradas: ao permitir que os utilizadores encontrem produtos de forma mais rápida e intuitiva, a experiência de compra torna-se mais satisfatória, o que pode resultar em maior fidelidade e taxas de conversão. Além disso, a personalização aprimorada alinha as categorias aos interesses e preferências individuais dos clientes, o que aumenta a relevância das recomendações e incentivando compras repetidas [12].

Outro benefício é a adaptabilidade do sistema. A categorização dinâmica ajusta automaticamente as classificações com base em tendências sazonais e comportamentos dos utilizadores. Por exemplo, durante feriados ou eventos especiais, categorias relevantes podem ser destacadas, enquanto termos de pesquisa populares ou mudanças nos históricos de compras ajudam a reorganizar os produtos de acordo com as necessidades em constante evolução dos clientes. Esta capacidade de adaptação não apenas melhora a experiência do

utilizador, mas também permite que o sistema responda rapidamente às mudanças no mercado [11].

Finalmente, a eficiência operacional é outro ponto de destaque. A automação dos processos de categorização reduz a necessidade de intervenções manuais, diminuindo custos e permitindo que as equipas de gestão foquem em outros assuntos. Isto é particularmente vantajoso em plataformas que lidam com vastos catálogos de produtos e dados dinâmicos [12].

Apesar das vantagens, a implementação de sistemas de categorização dinâmica enfrenta também alguns desafios. Um dos principais obstáculos é a qualidade dos dados. Dados incompletos, inconsistentes ou mal estruturados podem comprometer a precisão das classificações, o que reduz a eficácia das recomendações e por sua vez prejudica a experiência do utilizador. Garantir dados de alta qualidade requer um esforço contínuo em termos de limpeza, normalização e integração [12].

Outro desafio importante está relacionado às preocupações com privacidade. Com o aumento das regulamentações como o GDPR, as empresas precisam equilibrar o uso de dados pessoais para personalização com a necessidade de proteger as informações dos utilizadores. Violações de privacidade podem levar a multas significativas e danos à reputação, o que torna esta questão uma prioridade para qualquer implementação [12].

Além disso, o custo computacional é outro desafio, especialmente em plataformas que processam grandes volumes de dados em tempo real. Modelos avançados de *deep learning*, por exemplo, exigem uma infraestrutura robusta e recursos computacionais significativos, que nem todas as empresas estão prontas para suportar. Este investimento inicial pode ser uma barreira para organizações menores que buscam implementar soluções semelhantes [12].

Por fim, há o desafio de alinhar a categorização dinâmica aos objetivos de negócios. Sistemas automatizados nem sempre entendem as nuances estratégicas ou priorizam categorias que estejam alinhadas com metas comerciais específicas, o que exige ajustes manuais ou intervenções para garantir que a lógica automatizada esteja alinhada às prioridades da organização [12].

## 2.4 Técnicas de clustering

As técnicas de *clustering* têm se mostrado bastante úteis em diversas áreas do *e-commerce*, como personalização, segmentação de utilizadores e categorização de produtos. Por meio do *clustering*, é possível personalizar interações com clientes, criar grupos segmentados para campanhas mais direcionadas e organizar produtos de forma eficiente, contribuindo para um *e-commerce* mais inteligente e estratégico. O *clustering* consiste no agrupamento de conjuntos de dados em subgrupos, ou *clusters*, com base nas suas similaridades, sendo que cada subgrupo

representa um contexto diferente. Alguns métodos usados para esse efeito são o *k-means*[1], *fuzzy c-means*[2], *gaussian mixture*[3], *spectral clustering*[4] e *dbscan*[5].

### 2.4.1 K-means

O *K-means*, criado por Stuart Lloyd em 1957 [14], é baseado na simples observação de que o posicionamento ideal de um centro está no centroide do *cluster* associado e pode ser representado na Equação 1:

$$\sum_{i=1}^{C} \sum_{j=1}^{C_i} \left( \|x_i - v_j\| \right)^2$$

Equação 1 – K-means

Em que $\|x_i - v_j\|$ é a distância Euclidiana entre o ponto de dados $i$ e o cluster $j$ respetivamente, $C_i$ é o número de pontos de dados do $i$-ésimo *cluster* e $C$ o número de centros de *clusters*.

Dado k valor de centros $Z$, para cada centro $z \in Z$ , em que $V(z)$ indica a sua vizinhança, isto é, o conjunto de pontos de dados para os quais $z$ é o vizinho mais próximo. Cada etapa do algoritmo de Lloyd move-se em torno de cada centro $z$ para o centroide de $V(z)$ e, de seguida, atualiza $V(z)$ por recalculando a distância de cada ponto ao seu centroide mais próximo. Estes passos são repetidos até que alguma condição de convergência ocorra. Para pontos que estejam numa posição normal (não se apresentem equidistantes de dois centros), o algoritmo vai eventualmente convergir para o ponto que apresenta o mínimo local para a distorção. Contudo, o resultado não é necessariamente o mínimo global. Bradley et al. demonstraram como aplicar o *K-means* a grandes conjuntos de dados a partir de amostragem e poda (remoção de conexões de baixa relevância para o desempenho do algoritmo de forma a torná-lo mais eficiente e menos complexo) [15].

### 2.4.2 Fuzzy c-means

O Fuzzy c-means, proposto por Dunn em 1973 [16] e posteriormente aprimorado por Bezdek em 1981 [17], também é conhecido por *clustering* suave, e pode ser representado na Equação 2:

$$\sum_{i=1}^{c} \sum_{j=1}^{c_i} W_{ij}^{m} \left( \|x_i - v_j\| \right)^2$$

Equação 2 – Fuzzy c-means

[1] K-means: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
[2] Fuzzy c-means: https://www.mathworks.com/help/fuzzy/fuzzy-c-means-clustering.html
[3] Gaussian mixture: https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95
[4] Spectral clustering: https://towardsdatascience.com/spectral-clustering-aba2640c0d5b
[5] DBSCAN: https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556

O *Fuzzy c-means*, apresenta uma fórmula bastante similar à do *K-means*, acrescentando apenas $w_{ij}$, que tem apenas dois valores zero e um, e representa o valor de adesão ao cluster, e $m$ que é o grau de imprecisão que está compreendido entre zero e mais infinito.

Ao contrário do método *K-means*, que atribui a cada ponto nos dados um único *cluster*, o *Fuzzy c-means*, permite que este possa pertencer a todos os clusters, em que o seu grau de pertença a cada um irá variar entre 0 e 1, isto é, se um ponto nos dados está mais próximo do centro de um *cluster* o seu grau de pertença a esse cluster será maior. Este algoritmo apresenta uma complexidade relativamente elevada e, portanto, não é considerado escalável para grandes conjuntos de dados. Kolen et al. em 2002 propôs uma estratégia de acelerar o algoritmo, utilizando uma atualização combinada da matriz de pertinência e dos protótipos, o que torna a complexidade linear relativamente ao número de clusters [18].

### 2.4.3 Gaussian Mixture

O *Gaussian Mixture*, introduzido por Arthur Dempster et al. em 1977 [19], é um modelo de probabilidade bastante usado em aprendizagem não supervisionada, e pode ser representado como na Equação 3:

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)}$$

Equação 3 – Gaussian Mixture

Em que $\mu$ e $\sigma$ é a média e desvio padrão dos pontos de dados e $x$ os pontos de dados, respetivamente. Este modelo assume que todos os pontos de dados são gerados por uma mistura de um número finitos de distribuições gaussianas com parâmetros desconhecidos. Este modelo opera com K componentes gaussianos, onde cada componente possuí um vetor de média e uma matriz de covariância que representam a distribuição de probabilidade da variação dos dados [20].

### 2.4.4 Spectral Clustering

O *Spectral Clustering*, proposto inicialmente por Jain, Murty, e Flynn em 1999 [21], é baseado na autodecomposição de matrizes, o que normalmente obtém uma maior qualidade comparando com outros algoritmos de *clustering*. Enquanto outros algoritmos de *clustering* são baseados na geometria euclidiana e por causa disso apresentam certas limitações no formato dos seus clusters, o *Spectral Clustering* pode se adaptar a uma maior variedade de geometrias e detetar padrões não convexos e clusters que linearmente são não separáveis. Apesar da sua boa performance em termos de qualidade este método apresenta uma grande limitação, visto que é computacionalmente pesado quando usado em numa vasta gama de dados [22].

### 2.4.5 DBSCAN

Proposto por Martin Ester et al. em 1996 [23], é um algoritmo que foi projetado para descobrir os *clusters* com qualquer tipo de formato e tamanho. O algoritmo baseia-se no princípio de que para cada ponto de dados de um cluster, a sua vizinhança deve conter pelo menos um número mínimo de pontos de dados. Por outras palavras, o DBSCAN procura por *clusters* por verificar os vizinhos de cada ponto de dados do conjunto de dados. Se a vizinhança do ponto de dados

contiver mais do que o número mínimo fornecido pelo utilizador, um novo cluster é criado. O algoritmo termina assim que mais nenhum ponto de dados puder ser inserido em qualquer *cluster* [24].

## 2.5 Desafios relativos à quantidade de dados

Os sistemas de recomendação enfrentam desafios associados à quantidade e à qualidade dos dados disponíveis. Quando os dados são escassos, incompletos ou distribuídos de forma desigual, torna-se difícil gerar recomendações confiáveis e personalizadas. Situações como a introdução de novos utilizadores ou produtos, a existência de perfis de comportamento atípico ou a fraca densidade de interações são obstáculos para este tipo de sistemas. Neste ponto, são analisados três dos principais desafios relacionados com a insuficiência de dados: o problema do *Cold Start*, os casos de *Black/Grey Sheep* e a *Sparsity* (esparsidade), e são explicadas as suas causas, implicações e possíveis estratégias para os mitigar.

### 2.5.1 Cold Start

O problema de *Cold Start* ocorre quando o sistema de recomendação tem dificuldade em fornecer sugestões precisas devido à falta de dados iniciais. Este problema manifesta-se quando um novo utilizador registra-se no sistema sem histórico de interações, o que torna difícil prever as suas preferências, quando um novo produto é adicionado ao website, ou no lançamento inicial do sistema, quando há poucos utilizadores e produtos, o que resulta numa escassez geral de dados para gerar recomendações eficazes. Para mitigar o *Cold Start*, recorre-se a estratégias como a recolha de informações demográficas dos utilizadores, a solicitação de preferências iniciais ou a recomendação de produtos populares [25].

### 2.5.2 Black / Grey sheep

Os termos *Black Sheep* e *Gray Sheep* referem-se a utilizadores cujos comportamentos ou preferências diferem significativamente da maioria, o que apresenta desafios específicos para os sistemas de recomendação. *Black Sheep* (Ovelhas Negras) são utilizadores cujas preferências são tão únicas que o sistema de recomendação não consegue encontrar outros utilizadores semelhantes para gerar sugestões relevantes. *Gray Sheep* (Ovelhas Cinzentas) são utilizadores que têm algumas preferências iguais a outros utilizadores em algumas áreas, mas discordam noutras, tornando-os parcialmente semelhantes à maioria. Esta inconsistência dificulta a criação de recomendações precisas. Estes casos representam desafios porque os sistemas de recomendação, especialmente os baseados em filtragem colaborativa, dependem de padrões comuns entre utilizadores para prever preferências [26].

### 2.5.3 Sparsity

*Sparsity* refere-se à situação em que a relação de interações entre utilizadores e produtos é predominantemente composta por valores nulos. Isto significa que a maioria dos utilizadores interagiu apenas com uma pequena fração dos produtos disponíveis, o que resulta numa matriz esparsa. Esta escassez de dados pode dificultar a identificação de padrões e a geração de

recomendações precisas. Para lidar com este problema, a utilização de modelos híbridos que usam filtragem colaborativa e filtragem baseada em conteúdo, e a incorporação de dados adicionais, como informações demográficas ou contextuais, são empregues [27].

## 2.6 Métricas para avaliação do impacto

Avaliar o sucesso das estratégias no *e-commerce* é um processo multifacetado que depende da análise de várias métricas importantes. Estas métricas fornecem uma compreensão quantitativa e qualitativa do comportamento do utilizador e do desempenho operacional da companhia. As empresas quando utilizam métricas, tais como as que serão abordadas de seguida, estão a ajustar as suas estratégias para melhorar a experiência do utilizador, aumentar a receita e alcançar um crescimento sustentável. Esta secção investiga estas métricas, o seu cálculo, relevância e aplicações práticas na melhoria dos resultados do *e-commerce*.

### 2.6.1 Taxa de cliques

A Taxa de cliques (CTR) é uma métrica que avalia a eficácia das campanhas de *marketing* digital. Mede a percentagem de utilizadores que clicam num link, anúncio ou produto específico em comparação com o número total de pessoas que o visualizaram:

$$CTR = \frac{Número\ de\ cliques}{Número\ de\ visualizações} * 100$$ Equação 4 – Taxa de cliques

Uma CTR elevada (cerca de 2%) indica, geralmente, que a mensagem de *marketing* tem uma boa repercussão no público-alvo. Textos de anúncios eficazes, designs visualmente apelativos e segmentação personalizada aumentam significativamente a CTR. Ferramentas como os testes A/B permitem às empresas experimentar diferentes formatos, identificando a combinação que consegue o melhor desempenho. Investigações indicam que otimizar palavras-chave em campanhas de pesquisa paga e criar chamadas à ação apelativas estão entre as estratégias mais eficazes para aumentar o CTR. Além disso, plataformas como o *Google Ads* e o *Facebook Ads* fornecem dados granulares sobre o CTR, permitindo um refinamento contínuo [28] [29].

### 2.6.2 Taxa de conversão

A Taxa de conversão (CVR) vai além dos cliques para medir a percentagem de utilizadores que realizam uma ação desejada, como fazer uma compra ou subscrever uma *newsletter*. É calculado como:

$$CVR = \frac{Número\ de\ conversões}{Número\ de\ visitantes} * 100$$ Equação 5 – Taxa de conversão

Embora os *benchmarks* globais de CVR variem de acordo com o setor, a média para o *e-commerce* é de cerca de 5,2%. No entanto, as empresas de alto desempenho conseguem taxas de até 10% através de esforços de otimização direcionados. Os fatores que influenciam o CVR incluem o *design* do *website*, os tempos de carregamento, a facilidade de navegação e as opções

de pagamento. Além disso, foi demonstrado que a implementação de medidas de construção de confiança, tais como *gateways* de pagamento seguros e políticas de devolução transparentes, tem um impacto positivo no CVR [30] [31].

### 2.6.3 Valor médio ao pedido

O Valor médio ao pedido (AOV) é uma métrica crítica para compreender o comportamento de despesa do consumidor e maximizar a receita. É calculado como:

$$AOV = \frac{Receita\ total}{Quantidade\ de\ encomendas}$$

Equação 6 – Valor médio ao pedido

AOVs elevados podem indicar estratégias de *upsell* e vendas cruzadas bem-sucedidas. Por exemplo, as empresas utilizam frequentemente técnicas como agrupar produtos, oferecer descontos em compras de maior valor e promover limites de envio gratuito. Plataformas de comércio eletrónico como a *Amazon* e a *Shopify* demonstram como a otimização eficaz de AOV pode aumentar significativamente a rentabilidade global sem aumentar os custos de aquisição de clientes [30].

### 2.6.4 Retenção de utilizadores e taxa de recompra

A retenção de utilizadores é a base do sucesso comercial a longo prazo no *e-commerce*. Métricas como a Taxa de Retenção de Clientes (CRR) e a Taxa de Recompra (RPR) fornecem informações sobre a lealdade e satisfação da base de clientes. A CRR mede a percentagem de clientes recorrentes num determinado período, calculada como:

$$\frac{Nº\ de\ clientes\ recorrentes\ -\ Novos\ clientes}{Total\ de\ clientes\ no\ início} * 100$$

Equação 7 – Taxa de retenção de clientes

A RPR reflete a percentagem de clientes que fazem compras múltiplas num período. As estratégias de retenção envolvem frequentemente programas de fidelização, recomendações personalizadas de produtos e ofertas exclusivas para clientes recorrentes. Uma elevada taxa de retenção não só reduz o custo por aquisição, como também promove a lealdade à marca e o valor vitalício [28] [29].

### 2.6.5 Taxa de abandono do carrinho

O abandono do carrinho é um dos desafios mais críticos no *e-commerce*, com uma taxa média global de quase 70%. A métrica é calculada como:

$$\text{Taxa de Abandono de Carrinho} = \frac{1 - \text{Compras Concluídas}}{\text{Carrinhos Criados}} * 100$$

Equação 8 – Taxa de abandono do carrinho

As causas comuns incluem custos de envio inesperados, processos de checkout complicados e falta de opções de pagamento preferenciais. Para combater taxas elevadas, a personalização

tem sido usada para melhorar a experiência de compra, por oferecer produtos recomendados com base no histórico de navegação ou nas preferências do cliente, a personalização aumenta a adesão e a probabilidade de conversão do cliente, o que reduz as tendências de abandono dos carrinhos. Para além disso, oferecer incentivos personalizados, como descontos ou promoções direcionadas, também pode diminuir as frustrações e impulsionar a finalização da compra. [31] [32].

## 2.7 Estudo de casos e aplicações reais

A utilização de AI para melhorar a categorização e a experiência de navegação em sites de *e-commerce* tem ganhado destaque em estudos acadêmicos e aplicações industriais. Este capítulo explora casos reais que apresentam abordagens alinhadas com os objetivos da *Flamingo S.A.*

### 2.7.1 Otimização com Modelos de Deep learning

Uma abordagem que utiliza o modelo BERT (*Bidirectional Encoder Representations from Transformers*) foi apresentada por Xu et al. para análise semântica em sistemas de recomendação de *e-commerce*. A combinação do BERT com algoritmos de vizinhos mais próximos (*k nearest neighbors)* permitiu classificar produtos de forma contextual e oferecer recomendações baseadas no comportamento do utilizador. Segundo o autor, através de uma avaliação manual, a eficácia do sistema de recomendação é confirmada, e afirma que os produtos recomendados pelo sistema são altamente consistentes com as preferências de compra dos clientes [33].

### 2.7.2 Segmentação Dinâmica com *Machine Learning*

Um sistema de recomendação desenvolvido por Haque et al. combina algoritmos de aprendizado supervisionado, como *Random Forest* e Regressão Logística, para segmentar utilizadores em tempo real. Esta abordagem personalizou a organização de categorias de acordo com a localização geográfica, as preferências declaradas e o histórico de navegação de cada utilizador. Com isso, o modelo alcançou uma precisão de 99,6%, o que comprova que técnicas de ML ao serem bem aplicadas podem impactar positivamente a experiência de compra do cliente [34].

### 2.7.3 Estratégias de Priorização Personalizada

Um sistema hierárquico para plataformas de *e-commerce* que utiliza redes neuronais e *deep learning*, foi desenvolvido por por Zhang et al. com o objetivo de criar hierarquias entre categorias e produtos. O sistema analisa o ciclo de vida dos produtos para determinar a sua relevância em diferentes momentos, com o objetivo de garantir que os produtos de maior prioridade fossem destacados para os utilizadores com base em padrões de comportamento do utilizador e sazonalidade [35].

### 2.7.4 Integração de AI em websites de e-commerce

A *Amazon* utiliza ferramentas que ajustam dinamicamente as categorias de produtos na sua plataforma de *e-commerce*, personalizando ofertas com base no histórico de compras e na localização dos utilizadores. A implementação de AI permitiu melhorar a experiência de compra. A principal inovação é a personalização das recomendações de produtos, que agora são feitas por grandes categorias com base no histórico de compras do cliente. Isto facilita a navegação, agrupando os produtos de forma mais relevante para os interesses dos usuários, utilizando um modelo *Large Language Models* (LLM) para ajustar as recomendações de forma dinâmica [36].

## 2.8 Tecnologias Existentes / Emergentes

O uso de tecnologias avançadas em *e-commerce* tem se tornado fundamental para melhorar a personalização, a segmentação de utilizadores e a categorização de produtos. Diversas bibliotecas, *frameworks* e ferramentas em linguagens como *Python*[6], *R*[7], *Java*[8] e outras têm facilitado o desenvolvimento de soluções robustas e escaláveis. Esta secção explora algumas das tecnologias mais relevantes e emergentes para esses fins.

### 2.8.1 Python

*Python* destaca-se por ser uma das linguagens mais populares devido à sua simplicidade, extensibilidade. A linguagem oferece uma ampla gama de bibliotecas e *frameworks* que suportam tarefas como ML, análise de dados, NLP e manipulação de dados em larga escala.

#### 2.8.1.1 TensorFlow

A biblioteca *TensorFlow*[9] é uma plataforma de código aberto desenvolvida pelo Google para computação numérica e ML, utilizada no desenvolvimento de modelos avançados de *deep learning*. O *TensorFlow* suporta a criação de redes neuronais para tarefas como classificação, regressão, tradução de linguagem natural e recomendação de produtos. Um dos seus principais recursos é a capacidade de realizar cálculos em larga escala, o que permite o processamento eficiente de grandes volumes de dados. A biblioteca utiliza tensores (*arrays* multidimensionais) como base para as operações matemáticas o que permite o desenvolvimento de modelos customizados. No *e-commerce,* o *TensorFlow* é utilizado para criar representações vetoriais de produtos, que convertem informações como descrições, categorias e atributos em vetores numéricos. Esses vetores capturam relações semânticas entre os produtos, sendo utilizados para melhorar a categorização e oferecer recomendações personalizadas.

#### 2.8.1.2 PyTorch

*PyTorch*[10], desenvolvido pelo *Facebook AI Research*, é outra biblioteca de *deep learning* utilizada também devido à sua flexibilidade e facilidade de uso. A biblioteca destaca-se por

[6] Python: https://www.python.org
[7] R: https://www.r-project.org/about.html
[8] Java: https://www.java.com/en/download/help/whatis_java.html
[9] TensorFlow: https://www.tensorflow.org/learn?hl=pt-br
[10] PyTorch: https://pytorch.org/docs/stable/index.html

oferecer uma abordagem dinâmica para treinar de redes neuronais, o que permite que os gráficos computacionais sejam definidos e ajustados em tempo de execução. No contexto de *e-commerce*, *PyTorch* é capaz de criar sistemas de recomendação baseados em *deep learning* e categorizações inteligentes de produtos. Similar ao *TensorFlow*, o *PyTorch* permite que os desenvolvedores criem representações vetoriais otimizadas para produtos e utilizadores, o que ajuda na criação de filtros colaborativos e sistemas de recomendação baseados em conteúdo.

#### 2.8.1.3 Scikit-learn

A biblioteca *Scikit-learn*[11] é uma ferramenta de código aberto voltada para ML e análise de dados. Ela oferece suporte a algoritmos de aprendizagem, como regressão, classificação e *clustering*. No *e-commerce*, *Scikit-learn* é utilizada para segmentação de utilizadores e categorização de produtos. Por exemplo, o algoritmo *K-Means* pode ser usado para agrupar clientes com base em suas preferências de compra, por identificar *clusters* que compartilham interesses semelhantes. Além disso, *Scikit-learn* inclui ferramentas de pré-processamento, como normalização e codificação de variáveis categóricas, para preparar os dados antes da aplicação de algoritmos de ML.

#### 2.8.1.4 Pandas e NumPy

*Pandas*[12] e *NumPy*[13] são bibliotecas de análise e manipulação de dados. O *Pandas* fornece estruturas de dados como *DataFrames*, que permitem organizar e analisar grandes volumes de informações estruturadas, como registos de compras, inventários de produtos e comportamento de navegação dos utilizadores. Ele suporta operações como filtragem, agregação e transformação de dados, sendo uma ferramenta muito útil na fase de pré-processamento para ML. Já o *NumPy* é usado para cálculos matemáticos avançados, oferecendo suporte a *arrays* multidimensionais e operações vetorizadas. Isso o torna ideal para manipular grandes conjuntos de dados numéricos de forma eficiente. No *e-commerce*, estas bibliotecas são usadas para limpar e organizar dados brutos antes de aplicar os modelos.

#### 2.8.1.5 SpaCy e NLTK

No domínio do PLN, *SpaCy*[14] e *Natural Language Toolkit*[15] (NLTK) são bibliotecas utilizadas para analisar dados textuais, como avaliações de clientes e descrições de produtos. O *SpaCy* é usado pela sua eficiência em tarefas como extração de entidades nomeadas, análise sintática e tokenização. Ele pode ser usado para identificar palavras-chave em avaliações, permitindo a criação de categorias de produtos baseadas nas opiniões dos utilizadores. Já o NLTK oferece um conjunto de ferramentas linguísticas, incluindo suporte para análise de sentimentos e tradução de texto. Ele é útil para interpretar opiniões de clientes e detetar tendências, como a popularidade de determinados itens em diferentes regiões.

[11] Scikit-learn: https://scikit-learn.org/stable/modules/clustering.html#clustering
[12] Pandas: https://pandas.pydata.org/docs/
[13] NumPy: https://numpy.org/doc/stable/
[14] SpaCy: https://spacy.io/
[15] NLTK: https://www.nltk.org/_modules/nltk.html

### 2.8.2 R

A linguagem *R* é reconhecida pelas suas capacidades avançadas em análise estatística e visualização de dados. Com uma vasta gama de bibliotecas, *R* permite a realização de análises detalhadas e a criação de gráficos, que podem auxiliar a tomada de decisões no *e-commerce*.

#### 2.8.2.1 Caret

A biblioteca *Caret*[16] auxilia o processo de treino e avaliação de modelos de ML. Ela fornece uma interface para aplicar diferentes algoritmos de aprendizado supervisionado e realizar tarefas como pré-processamento de dados, seleção de variáveis e ajuste de hiperparâmetros. No contexto de *e-commerce*, o *Caret* pode ser usado para prever preferências dos clientes com base no histórico de compras, dados demográficos e comportamentos de navegação.

#### 2.8.2.2 Ggplot2

O ggplot2 [17] é uma das ferramentas de visualização de dados. Ele permite criar gráficos personalizados e informativos, que permite compreender o comportamento dos utilizadores e o desempenho das categorias de produtos. O ggplot2 pode ser utilizado para criar mapas de calor que mostram as áreas do site com maior interação do cliente ou gráficos de séries temporais para analisar tendências de vendas ao longo do tempo. O ggplot2 é capaz de ajudar a visualizar *clusters* de utilizadores com base em padrões de compra.

#### 2.8.2.3 Arules

A biblioteca *Arules*[18] é uma ferramenta especializada em mineração, utilizado para identificar padrões de compra e descobrir relações entre produtos em grandes volumes de dados transacionais. O *Arules* permite identificar produtos que são frequentemente comprados juntos. Essas informações são usadas depois para personalizar a categorização de produtos e otimizar recomendações.

### 2.8.3 Java

A linguagem Java é utilizada em aplicações empresariais devido à sua robustez, escalabilidade e capacidade de integrar diversas bibliotecas e *frameworks* de ML e análise de dados. Para o *e-commerce*, o *Java* destaca-se na criação de *backends* eficientes que suportam funcionalidades de personalização, segmentação de utilizadores e sistemas de categorização dinâmicos.

#### 2.8.3.1 Apache Mahout

O *Apache Mahout*[19] é um *framework* de código aberto desenvolvido para criar algoritmos de ML em larga escala. Ele é projetado para lidar com grandes volumes de dados. Ele é usado em aplicações de *e-commerce* que exigem sistemas de recomendação robustos. No contexto de personalização, o *Mahout* é utilizado para implementar filtros colaborativos e sistemas baseados em conteúdo. Por exemplo, ao analisar o histórico de navegação e compras de um

[16] Caret: https://cran.r-project.org/web/packages/caret/vignettes/caret.html
[17] Ggplot2: https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf
[18] Arules: https://cran.r-project.org/web/packages/arules/arules.pdf
[19] Apache Mahout: https://mahout.apache.org/documentation/users/

cliente, o *Mahout* é capaz de recomendar produtos que outros utilizadores com preferências similares adquiriram.

#### 2.8.3.2 Weka

O *Weka*[20] é uma ferramenta de análise de dados e ML, utilizada para *clustering*, regressão, classificação e mineração de dados. No *e-commerce*, o *Weka* pode ser aplicado para segmentação de utilizadores ao agrupar clientes com comportamentos de compra semelhantes. Por exemplo, por utilizar algoritmos como *K-Means* ou árvores de decisão, a ferramenta identifica padrões que ajudam a ajustar categorias de produtos com base nas preferências do utilizador. Outro benefício do *Weka* é a capacidade de pré-processar dados, por lidar com valores ausentes e normalizar atributos.

### 2.8.4 Tecnologias Emergentes

As tecnologias emergentes têm sido um motor de inovação no setor de *e-commerce*, quando se trata de personalização, segmentação de utilizadores e categorização de produtos. Modelos de ML e abordagens avançadas de AI permitem uma análise mais detalhada do comportamento do consumidor e criam oportunidades para melhorar a experiência de compra *online*. Tecnologias emergentes, como *transformers*, *Explainable AI* (XAI), *Graph Neural Networks* (GNN) e plataformas de *AutoML* (*Automated Machine Learning*), estão cada vez mais a ser incluídas no *e-commerce*.

#### 2.8.4.1 Transformers e Modelos Pré-treinados

Modelos como BERT[21] e *Generative Pretrained Transformer*[22] (GPT) têm revolucionado o campo do PLN, sendo aplicados em sistemas de recomendação e personalização de *e-commerce*. Esses modelos pré-treinados oferecem uma compreensão profunda e contextualizada das interações de texto, permitindo que os sistemas interpretem as preferências dos clientes e melhorem a relevância das recomendações de produtos. O BERT pode ser utilizado para realizar classificação semântica de produtos, onde os produtos são agrupados em categorias ajustadas às intenções dos clientes. Isto melhora a navegação e a precisão das recomendações, ajustando-se dinamicamente ao comportamento de compra dos utilizadores. Ao analisar o histórico de busca e as descrições dos produtos, o modelo pode entender a relação entre produtos e fornecer sugestões precisas, baseadas em contextos semânticos e preferências implícitas. Estes modelos são particularmente eficazes na análise de grandes volumes de dados não estruturados.

#### 2.8.4.2 Frameworks de Explainable AI

Com o aumento da complexidade dos sistemas de ML, tornou-se cada vez mais importante compreender como as decisões que são tomadas, especialmente em sistemas de recomendação de *e-commerce*. A XAI visa tornar os modelos de AI mais transparentes e compreensíveis para os humanos, oferecendo uma explicação de como as recomendações são geradas. Ferramentas como *Local Interpretable Model-agnostic Explanations*[23] (LIME) e *Shapley*

[20] Weka: https://www.cs.waikato.ac.nz/ml/weka/
[21] BERT: https://huggingface.co/transformers/v3.0.2/model_doc/bert.html
[22] GPT: https://platform.openai.com/docs/concepts
[23] LIME: https://lime-ml.readthedocs.io/en/latest/

*Additive Explanations*[24] (SHAP) são exemplos de *frameworks* de XAI que são usados para interpretar os resultados de modelos complexos. Isso não apenas serve para otimizar a eficácia dos sistemas de recomendação, mas também para aumentar a confiança dos consumidores e garantir a conformidade com regulamentos de transparência.

#### 2.8.4.3 Graph Neural Networks

As GNNs[25] são uma abordagem emergente em redes neuronais que são projetadas para lidar com dados em forma de grafos. Estes modelos são usados para mapear e explorar relacionamentos complexos entre produtos e clientes em plataformas de *e-commerce*. Ao representar produtos e utilizadores como nós em um grafo e suas interações como arestas, as GNNs podem identificar padrões e associações entre produtos, o que cria recomendações mais precisas e contextuais. Um sistema de recomendação baseado em GNN pode identificar grupos de produtos que estão frequentemente comprados juntos ou recomendar itens baseados no comportamento de clientes com preferências semelhantes. Para além disso, as GNNs têm a capacidade de aprender de maneira eficiente em grafos de grande escala, o que as torna apropriadas para plataformas de *e-commerce* com grandes catálogos de produtos e uma base de clientes diversificada.

[24] SHAP: https://shap.readthedocs.io/en/latest/
[25] GNN: https://www.datacamp.com/tutorial/comprehensive-introduction-graph-neural-networks-gnns-tutorial

# 3 Análise da solução

Esta secção tem como objetivo apresentar uma análise da solução proposta, enquadrando-a no contexto do domínio do problema e explica os principais requisitos a que esta deve responder. Para tal, é inicialmente descrito o modelo de domínio, que permite representar os elementos centrais envolvidos na solução, bem como as suas interações. Seguidamente, são identificados e discutidos os requisitos funcionais e não funcionais, essenciais para assegurar que a implementação satisfaz as necessidades e os critérios técnicos do sistema. Por fim, são enunciados os pressupostos considerados durante o processo de definição e planeamento da solução, os quais sustentam as decisões de conceção e desenvolvimento adotadas.

## 3.1 Domínio do problema

Nesta subsecção será examinado o modelo de domínio presente na Figura 3, que oferece uma visão abrangente dos conceitos e necessidades para a personalização das recomendações de categorias e filtros.

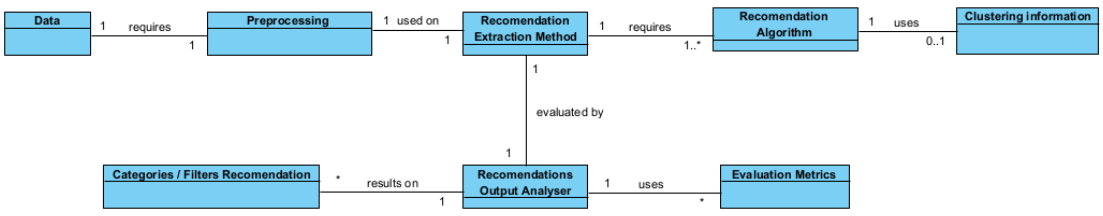


Figura 3 - Modelo de domínio

O processo inicial da personalização de recomendações de categorias e filtros, começa com a entidade *Data*, que no âmbito desde trabalho representará os dados fornecidos, pelos vários clientes da Flamingo durante a sua interação com o *website*, que foram armazenados numa base de dados. Este conjunto de dados é pré-processado (*Preprocessing*), onde são aplicados os seguintes passos: normalização dos dados e *Data Engineering*, onde são adicionadas novas características relativas aos dados, de forma a melhorar o desempenho dos algoritmos

aplicados. Após o pré-processamento, os algoritmos de recomendação são aplicados (*Recomendation Extraction Method*), onde serão aplicados três algoritmos, um baseado da popularidade das categorias e filtros, outro baseado na similaridade com outros clientes e ainda outro que recorre ao *clustering*, e, para o funcionamento deste método é necessário saber o número de grupos (ou *clusters*), por isso será usado um método que lhe permita calcular o número ótimo de grupos a serem criados (*Clustering Information*). Após a execução dos algoritmos, o seu *output* será avaliado de forma a decidir qual deles providencia a melhor personalização para o pretendido cliente (*Recomendations Output Analyser*). Esta entidade recorre a métricas (*Evaluation Metrics*) para proferir qual das recomendações é a mais ajustada. Por fim, a recomendação escolhida (*Categories/Filters Recomendation)* será disponibilizada ao cliente na sua próxima iteração no *website*.

## 3.2 Requisitos funcionais e não funcionais

Nesta subsecção serão apresentados os vários requisitos funcionais e não funcionais deste projeto, estando caracterizados de acordo com o modelo FURPS+[26]. Modelo este que usa diferentes atributos para caracterizar o software desenvolvido, sendo as suas categorias: funcionais (*Functionality*), usabilidade (*Usability*), confiabilidade (*Reliability*), desempenho (*Performance*), suporte (*Supportability*) e o "+" representando uma categoria adicional, criada quando o software não se encaixa nas categorias anteriores, podendo esta ser ambiental, legal, cultural ou outra.

### 3.2.1 Requisitos funcionais

Os requisitos funcionais são as características e funcionalidades que um sistema, software ou produto deve ter para atender às necessidades e expectativas do utilizador. A Tabela 1 contém os requisitos funcionais deste projeto, que estão enquadrados na categoria F (*Functionality*) do FURPS+.

Tabela 1 - Requisitos funcionais

| Requisito | Descrição Caso de Uso |
|---|---|
| **1: Pré-processar os dados obtidos** | UC1: Como analista de dados, quero que os dados sejam normalizados.<br>UC2: Como analista de dados, quero que os dados passem por um processo de *data engineering*. |
| **2: Criar algoritmos de personalização de categorias e filtros** | UC3: Como analista de dados, quero que seja criado um mecanismo de *cold start* para novos clientes.<br>UC4: Como analista de dados, quero que seja criado um algoritmo de popularidade.<br>UC5: Como analista de dados, quero que seja criado um algoritmo que recorra à similaridade com outros clientes.<br>UC6: Como analista de dados, quero que seja criado um algoritmo que recorra a *clustering*. |

[26] FURPS+: https://businessanalysttraininghyderabad.wordpress.com/2014/08/05/what-is-furps/

| Requisito | Descrição Caso de Uso |
|---|---|
| | UC7: Como analista de dados, quero que cada algoritmo apresente a percentagem de probabilidade e respetiva margem de erro para cada escolha de categoria/filtro. |
| **3: Criar métricas de avaliação dos algoritmos** | UC8: Como analista de dados, pretendo que a qualidade dos algoritmos seja testada. |
| **4: Guardar o resultado obtido** | UC9: Como analista de dados, pretendo que a melhor recomendação seja guardada. |

### 3.2.2 Requisitos não funcionais

Os requisitos não funcionais são os requisitos de um sistema de software que se referem a atributos de qualidade que não estão diretamente relacionados às funcionalidades do sistema, descrevendo assim como o sistema deve ser construído. A Tabela 2 contém todos os requisitos não funcionais para o desenvolvimento do projeto.

Tabela 2 - Requisitos não funcionais

| Requisito | Categoria |
|---|---|
| **1: A *framework* deverá ser desenvolvida em *Python*** | + Restrições de implementação |
| **2: A framework deverá disponibilizar os seus serviços como uma Application Programming Interface (API)** | + Restrições de Implementação<br>Usabilidade |
| **3: A ferramenta deverá ter uma arquitetura que possibilite a sua integração** | Suportabilidade<br>+ Restrições de Implementação |
| **4: Os dados serão recebidos através de uma base de dados** | + Restrições de Implementação |
| **5: A framework e documentação deve ser produzida em inglês** | Usabilidade<br>Suportabilidade |
| **6: Produção de um mapeamento de campos para o nome dos dados usados** | Usabilidade<br>Suportabilidade |

### 3.2.3 Pressupostos

O desenvolvimento desta *framework* conta com os seguintes pressupostos:

- O projeto realizado é para ser utilizado apenas pelo Flamingo, e, por esse motivo, tem em consideração as tecnologias lá utilizadas;
- O projeto já recebe os dados anonimizados, de forma a cumprir com os requisitos de proteção de dados;
- Os dados recebidos foram recolhidos corretamente, pelo que não possui dados indisponíveis;
- O projeto será implementado na rede interna do Flamingo, por esse motivo, requisitos de segurança não são levados em consideração;

- A solução apresentada é incremental, e, portanto, novos requisitos poderão ser integrados posteriormente.

# 4 Design da solução

Nesta secção estão presentes a análise e o desenho arquitetural da solução elaborada. Para o desenvolvimento do mesmo recorreu-se ao modelo *C4*[27] e também à visão-modelo 4+1[28], visto que este modelo permite representar a arquitetura do software com vários níveis de abstração. Este modelo começa no nível 1, que apresenta pouco detalhe, representando de forma macroscópica o trabalho, indo até o nível 4 que consiste na representação da implementação do código de cada componente representado, sendo o nível com mais detalhes. Para além disso, a visão-modelo complementa a anterior, visto que elabora 5 tipos de vistas: a vista de cenários, que descreve os casos de uso e os intervenientes; a vista lógica, que representa as várias funcionalidades que a aplicação desenvolvida contém; a vista de implementação, que demonstra a estrutura e organização do software; a vista física, que representa a conexão entre o *software* e o *hardware*; e, por fim, a vista de processos, que descreve as ações e o comportamento desempenhado pelo *software* durante a execução.

## 4.1 Nível 1

No nível 1 foram elaborados a vista de cenários, a vista lógica e a vista de processos, por serem as mais relevantes neste nível de abstração. Na Figura 4 apresentam-se os casos de uso do analista de dados na *framework*.

[27] C4: https://c4model.com
[28] Visão modelo: http://www.basef.com.br/old/uml/204-arquitetura-visao-modelo-41

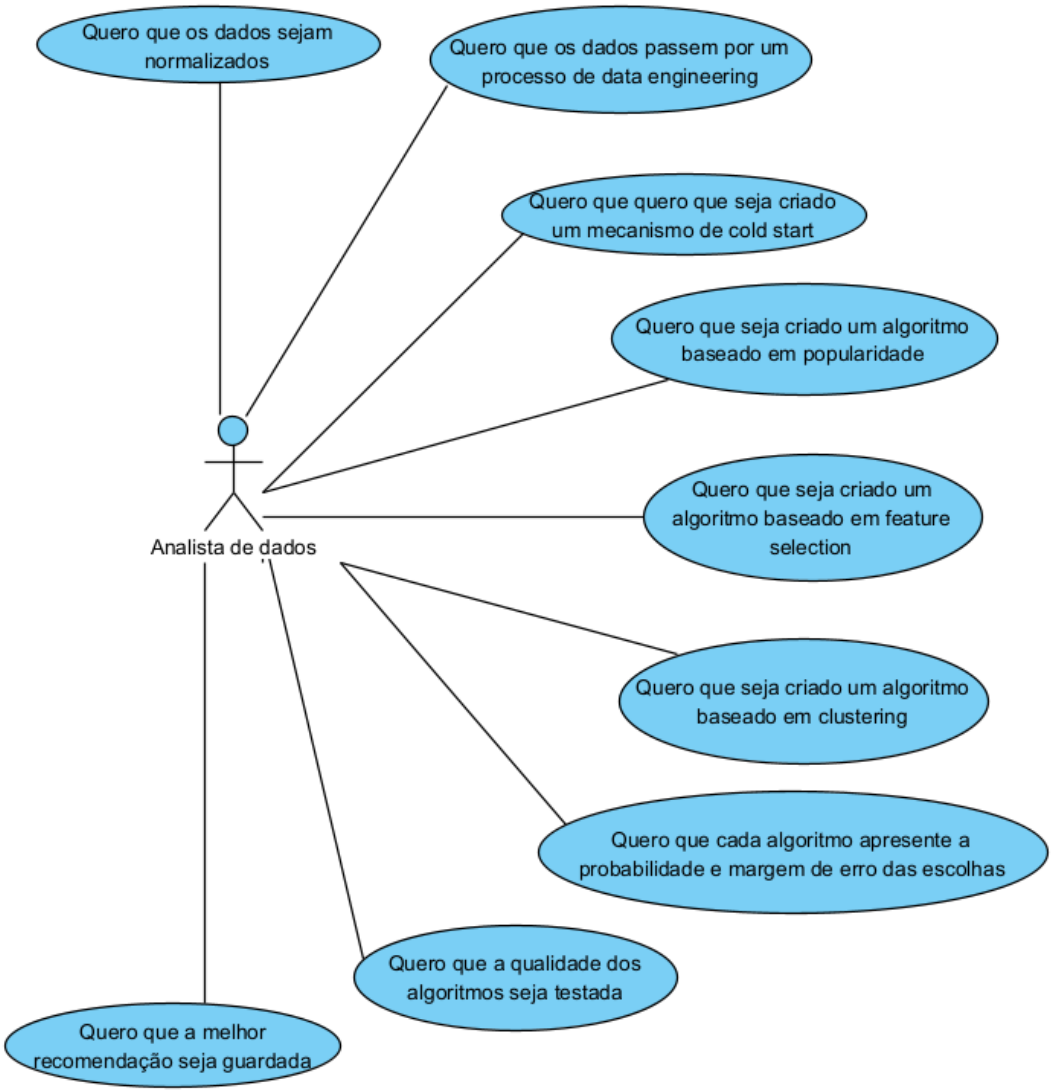


Figura 4 - Diagrama de casos de uso

A Figura 5 mostra como no futuro se pretende que a *framework* seja usada. O cliente da Flamingo S.A. ao interagir com o website da empresa (*Flamingo App*) fica com os seus dados de utilização guardados. Esses dados serão posteriormente disponibilizados ao componente *Framework CFP* que representa a *framework* desenvolvida. Esta tratará de os processar e devolver as recomendações ao componente *Flamingo App*.

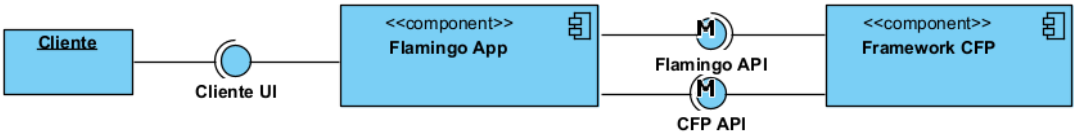


Figura 5 - Nível 1 vista lógica

A Figura 6, representa a única interação do analista de dados com o sistema durante a aquisição das categorias ou filtros de um certo utilizador.

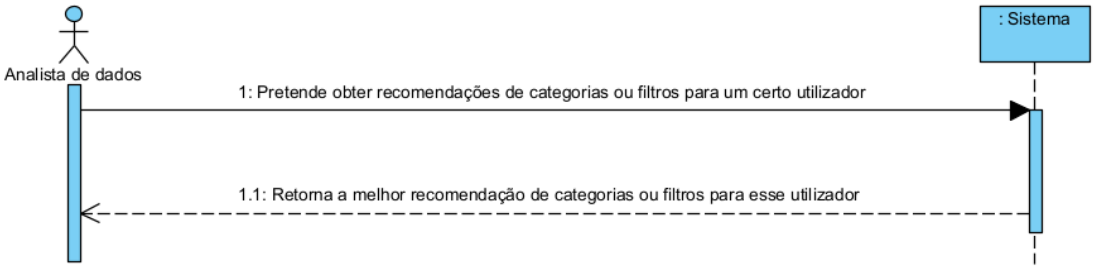


Figura 6 - Nível 1 vista de processo

## 4.2 Nível 2

No nível 2 de abstração foram desenhadas as vistas lógica, implementação, física e de processos. A Figura 7 mostra que existem dois módulos, o primeiro (*Módulo de personalização de categorias ou filtros*), acede à base de dados disponibilizada pela empresa Flamingo, e trata da criação das recomendações de categorias ou filtros, sendo depois acedido pelo segundo módulo (*Módulo de visualização dos resultados*) que fornece uma *User interface* ao analista de dados para a visualização das recomendações.

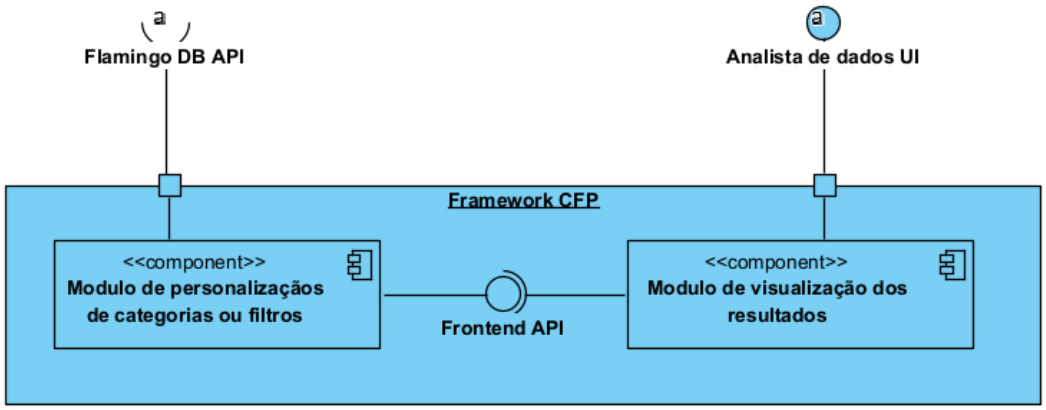


Figura 7 - Nível 2 vista lógica

Na vista de implementação de nível 2 mostra os dois módulos da *Framework CFP*, que possui um contentor para o *backend* e outro para o *frontend*, como se pode observar na Figura 8.

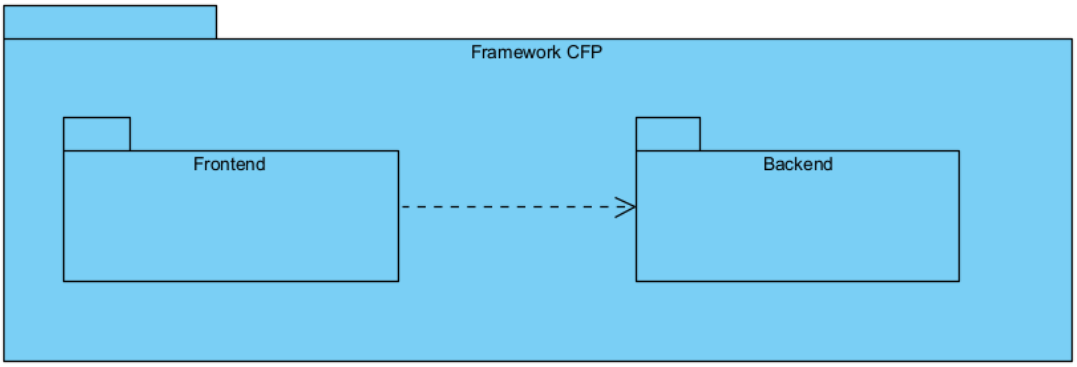


Figura 8 - Nível 2 vista de implementação

Relativamente à vista física da solução, a Figura 9 representa o servidor local que possui os dois módulos da *Framework CFP*, bem como a base de dados a que possuí acesso.

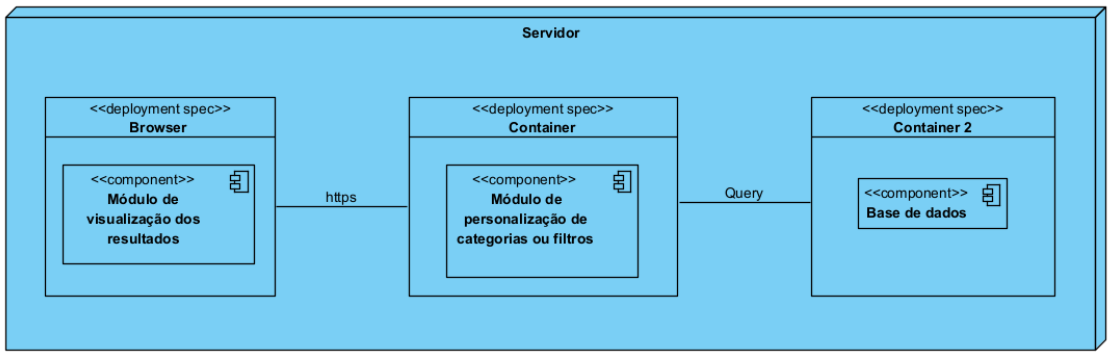


Figura 9 - Nível 2 vista física

Quanto à vista de processo, na Figura 10 pode-se observar o *Analista de dados* que pretende obter as recomendações no *Módulo de visualização*, para isso é feito um pedido HTTP GET de todos os clientes ao *Módulo de personalização* que por sua vez fará uma *query* à *Base de dados* e são retornados os clientes. O *Analista de dados*, de seguida, seleciona o cliente para o qual pretende obter as recomendações de contextos ou filtros no *Módulo de visualização*, e para isso é feito um HTTP POST com os dados do cliente selecionado para o *Módulo de personalização*, por sua vez este desempenha a sua análise e devolve as melhores recomendações através de um HTTP Response.

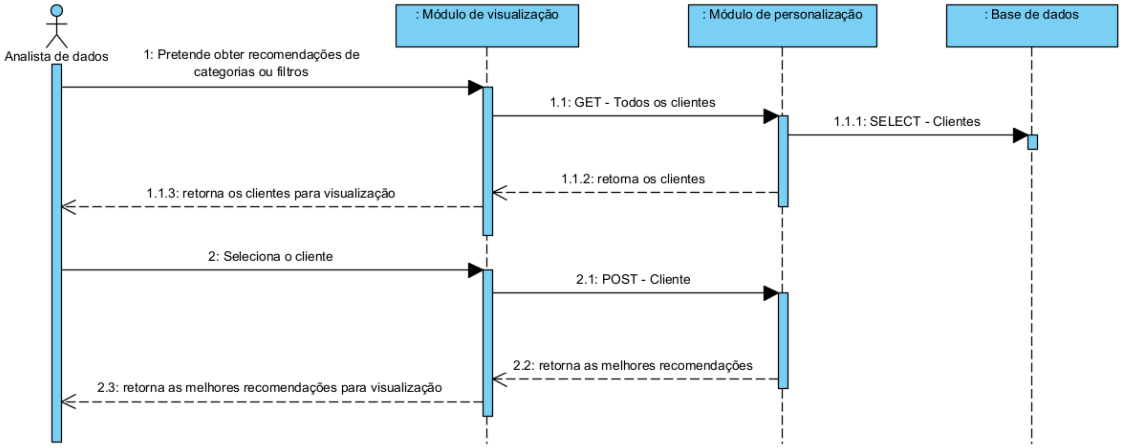


Figura 10 - Nível 2 vista processo

## 4.3 Nível 3

No nível 3 foram projetadas a vista lógica, implementação e de processo. A vista lógica (Figura 11) utiliza os padrões de desenvolvimento *Route*[29], que tem como função receber pedidos HTTP e retornar as respetivas respostas; *Controller*[30], que não apresenta a lógica de negócio, delegando essa responsabilidade aos métodos que instância; *Data Transfer Object*[31] (DTO), que também não apresentam nenhuma lógica de negócio, sendo apenas estruturas de dados utilizadas para transferir dados durante as operações; *Model*[32], que contém os objetos de domínio; *Service*[33], que implementa a lógica do negócio; e *Repository*[34], que tem como função estabelecer a conexão com a base de dados. A arquitetura efetuada segue os padrões *Onion*[35], na medida em que utiliza camadas concêntricas e em que as camadas interiores não dependem nem conhecem as camadas exteriores. Esta arquitetura reduz o acoplamento entre as várias classes tornando-a mais simples e fácil de testar. Este design com dois componentes foi escolhido, de forma a ser mais fácil de aproveitar futuramente apenas o componente *Módulo de personalização de categorias ou filtros,* visto que é apenas nesse que a Flamingo S.A. possuí interesse, sendo o outro meramente demonstrativo.

[29] Route: https://www.linkedin.com/pulse/gateway-routing-design-pattern-microservices-examples codeone-digest
[30] Controller: https://www.linkedin.com/advice/0/what-benefits-challenges-using-controller-pattern ooad-skills-ooad
[31] DTO: https://www.baeldung.com/java-dto-pattern
[32] Model: https://www.netsolutions.com/insights/software-design-pattern/
[33] Service: https://java-design-patterns.com/patterns/service-layer/
[34] Repository: https://medium.com/@pererikbergman/repository-design-pattern-e28c0f3e4a30
[35] Onion: https://www.codeguru.com/csharp/understanding-onion-architecture/

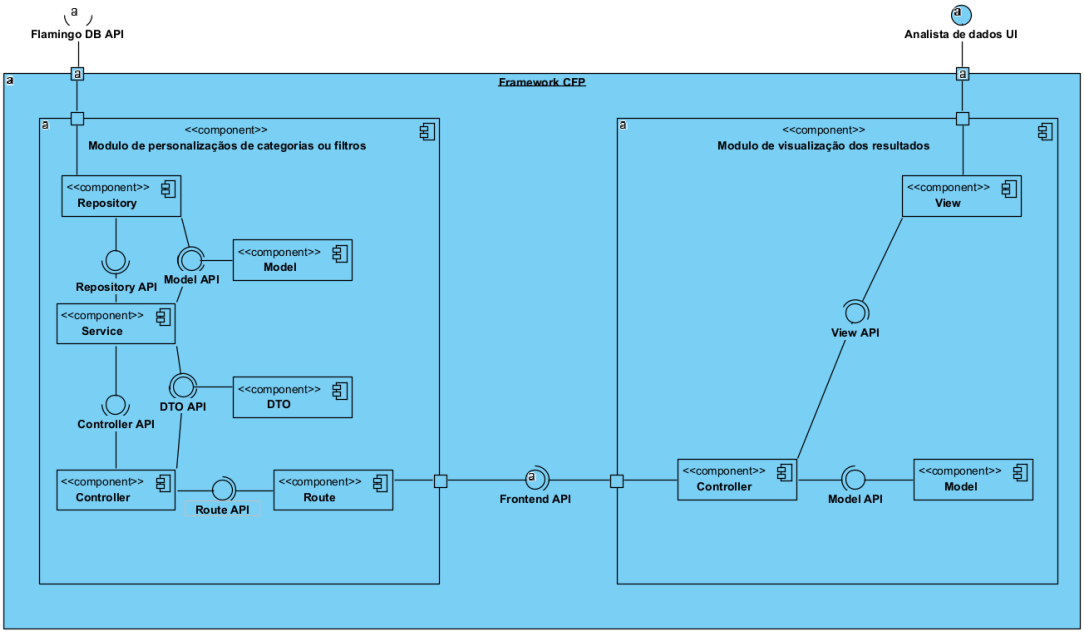


Figura 11 - Nível 3 vista lógica

Para além disso, esta arquitetura segue os princípios de *SOLID*[36]: *Single Responsibility Principle* (S), princípio que diz que cada classe dever ter apenas uma funcionalidade; *Open/Closed Principle* (O), ou seja, as entidades de software devem estar abertas para extensão, mas fechadas para modificações; *Liskov Substitution Principle* (L), relata que objetos de uma superclasse podem ser substituídos por objetos de uma subclasse, sem que isso afete o desenvolvimento do software; Interface *Segregation Principle* (I), princípio que refere que deve-se recorrer a interfaces para tornar a arquitetura desacoplada, mas evitando dependências desnecessárias nas interfaces; e *Dependency Inversion Principle* (D), que diz que os módulos de nível mais alto não devem depender dos de nível mais baixo e que as abstrações não devem depender de detalhes, mas sim, detalhes devem depender de abstrações. A aplicação destes princípios torna a aplicações mais flexíveis, modularizadas e de fácil manutenção.

A vista de implementação de nível 3, evidência as dependências entre os vários componentes da *framework*. Na Figura 12 pode-se ver no *Backend* que os *packages Repository* e *Model* dependem apenas do *Service*. O *Service* depende apenas do *Controller*, que por sua vez depende do *Router*. O *package DTO* é apenas usado para auxiliar as comunicações entre o *Service* e o *Controller*. No *Frontend* o *package Controller* interage com o *Model* e a *View*, e faz a ligação com o componente *Backend*.

[36] SOLID: https://www.bmc.com/blogs/solid-design-principles/

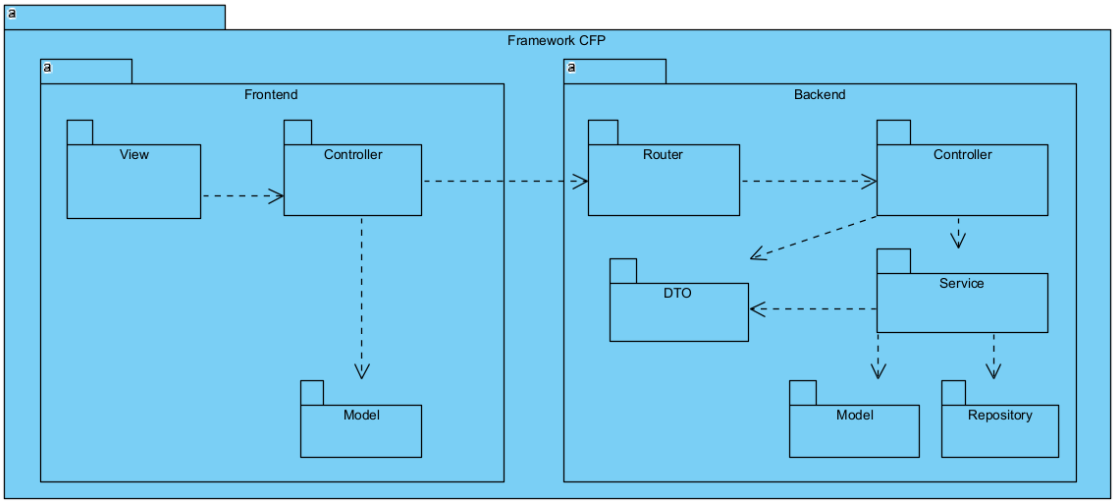


Figura 12 - Nível 3 vista de implementação

A Figura 13 demonstra o processo interno da *framework* quando o analista de dados solicita a aquisição de recomendações de categorias ou filtros. Inicialmente, o analista de dados irá receber uma lista com todos os clientes disponíveis. Para isso, a camada *RecomendationService,* irá interagir com a camada *RecomendationRepository*, de forma a proceder com a aquisição dos clientes à base de dados, de seguida estes serão passados à camada *RecomendationController*, e por fim à *View.* O analista de dados seleciona o cliente na camada *View* e este é recebido pelo *HTTP Client*, que, por sua vez, os transmite para a classe *Router*. De seguida, o *Router* transmite o pedido recebido para o *RecomendationController*, que vai numa primeira fase delegar responsabilidades ao *RecomendationService*, este aplicará técnicas de pré-processamento, executará os vários algoritmos de recomendações (caso o cliente possua *N* interações no *website*, caso contrário apenas o algoritmo de popularidade é executado), procederá a avaliação dos algoritmos através de métricas e decidirá o mais adequado àquele cliente. Após, chama a classe *Recomendation* que vai verifica as regras de negócio e, caso se cumpram, cria a recomendação. De seguida, o objeto é enviado para o *RecomendationRepository*, que vai guardar a recomendação na base de dados. Agora o objeto é retornado para o *RecomendationController*, convertido para DTO através da classe *RecomendationDTO* e enviado para a visualização do analista de dados.

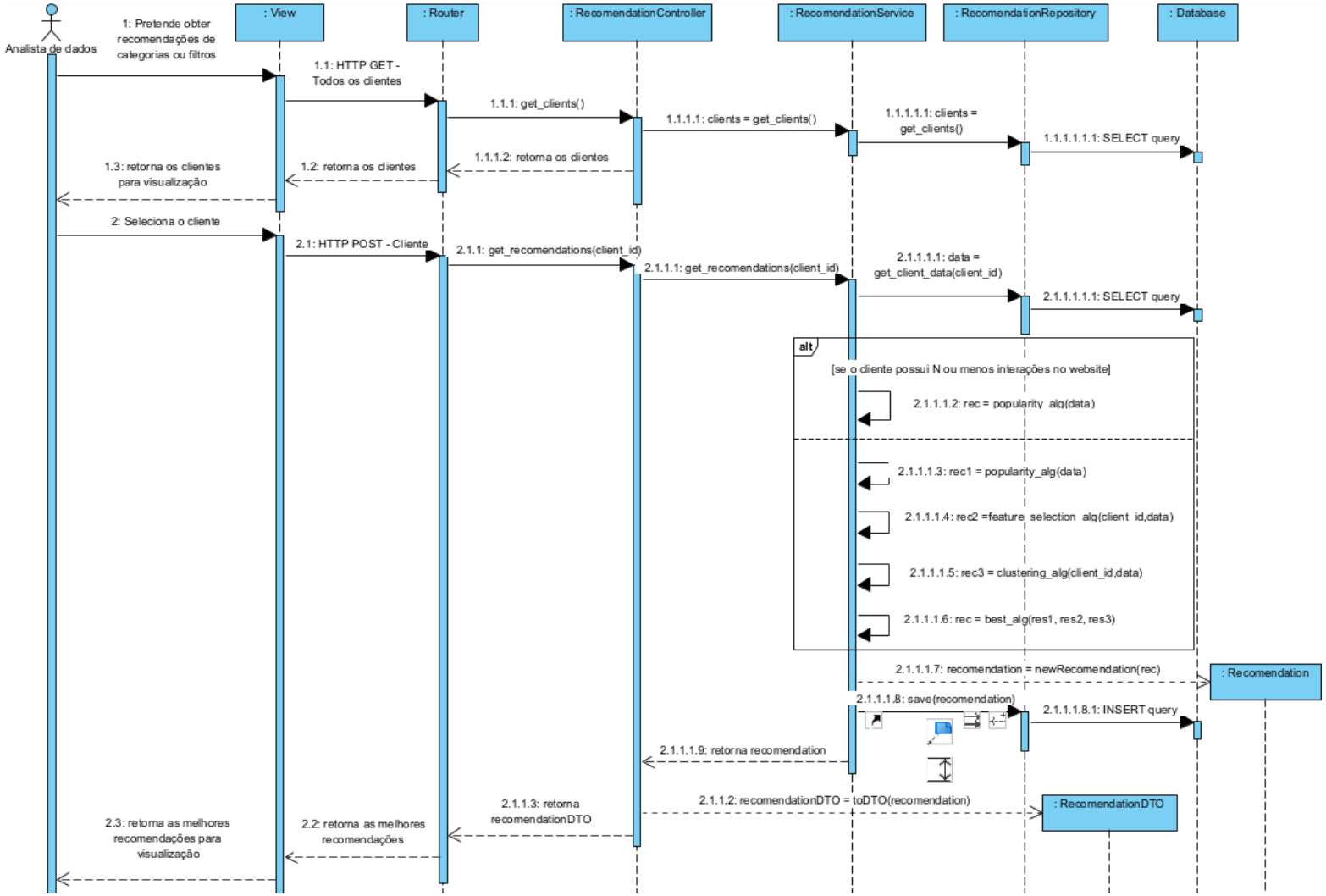


Figura 13 - Nível 3 vista de processo

# 5 Descrição da implementação

Nesta secção estão detalhados os aspetos da implementação da *framework* desenvolvida, bem como as tecnologias utilizadas.

A framework para deteção de contextos em tempo real foi desenvolvida na linguagem de programação *Python* (versão 3.10.8), em que as dependências necessárias ao desenvolvimento, foram instaladas através do gestor de *packages pip*[37]. De forma a poder disponibilizar uma *API RESTful*[38] que permita a interação entre o sistema e uma interface web para o analista de dados, recorreu-se à biblioteca *FastAPI*[39], que oferece suporte para integração com outras tecnologias modernas de forma simples. A solução centra-se na recomendação de categorias e filtros personalizados a utilizadores do *website* da Flamingo S.A., com base no seu histórico de interações. Para isso, foram desenvolvidos três algoritmos de recomendação distintos. O primeiro é o algoritmo de popularidade, que recomenda as categorias mais frequentes globalmente. O segundo baseia-se na similaridade e utiliza a métrica *cosine similarity*[40] para encontrar utilizadores com perfis semelhantes. O terceiro recorre ao *clustering*, agrupando utilizadores com base em perfis semelhantes através do algoritmo *K-Means*, sendo o número ótimo de clusters determinado com o auxílio do *silhouette score*[41].

Antes de aplicar os algoritmos de recomendação, os dados são pré-processados para melhorar a sua qualidade e relevância. Utiliza-se a transformação TF-IDF, utilizando a classe *TfidfTransformer*[42] da biblioteca *scikit-learn*, com o objetivo de ponderar a importância relativa das interações entre utilizadores e categorias ou filtros. Em seguida, procede-se à normalização e ao escalonamento dos dados com o *StandardScaler*[43], garantindo que todas as variáveis se encontram numa escala comparável. Por fim, é aplicada uma redução de dimensionalidade através do método PCA[44] (*Principal Component Analysis*), que permite extrair os componentes principais mais informativos para o processo de recomendação.

[37] Pip: https://medium.com/@habbema/python-pip-50bb24558e59
[38] API RESTful: https://aws.amazon.com/pt/what-is/restful-api/
[39] FastAPI: https://pythonacademy.com.br/blog/como-usar-o-fastapi-para-construir-apis-no-python
[40] Cosine similarity: https://scikit-learn.org/stable/modules/metrics.html#cosine-similarity
[41] Silhouette score: https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient
[42] TF-IDF: https://scikit-learn.org/stable/modules/feature_extraction.html#TfidfTransformer
[43] StandardScaler: https://scikit-learn.org/stable/modules/preprocessing.html#StandardScaler
[44] PCA: https://scikit-learn.org/stable/modules/decomposition.html#PCA

A *framework* incorpora ainda um mecanismo de avaliação dos algoritmos com base na métrica *F1-score*, permitindo testar cada abordagem individualmente. Para isso, é realizada uma divisão dos dados em conjuntos de treino e teste, de forma a simular cenários reais de recomendação.

Por fim, a API disponibiliza dois *endpoints*:

- /users, para obter a lista de utilizadores disponíveis;
- /recommend/{user_id}, para obter recomendações personalizadas para um utilizador específico, recorrendo dinamicamente ao algoritmo considerado mais eficaz com base no número de interações e nos resultados da avaliação.

## 5.1 Requisitos do sistema

O correto funcionamento do sistema de recomendação depende da qualidade dos dados de entrada armazenados na base de dados *SQLite*[45]. Esta secção descreve os requisitos formais que os dados devem cumprir para que o sistema possa gerar recomendações personalizadas de categorias e filtros.

Para que cada registo de interação possa ser utilizado pelo sistema, este deve conter, os seguintes atributos:

- um identificador único do utilizador (*cliente_id*);
- a data e hora da interação no formato "YYYY-MM-DD HH:MM:SS";
- a categoria/filtro consultado e o filtro selecionado

Estes atributos são necessários para identificar o comportamento de cada cliente e permitir a análise das suas interações.

O sistema exige, que no mínimo, cada utilizador tenha pelo menos dez interações (valor acordado com a Flamingo S.A.) registadas para que seja possível construir uma recomendação significativa. Para garantir a representatividade comportamental, pretende-se a disponibilização da maior quantidade de dados históricos possível. Neste projeto foram utilizados os dados de três meses (setembro, outubro e novembro) de 2024. Clientes com volume de dados baixo podem, ainda assim, receber recomendações baseadas na popularidade geral das categorias ou filtros.

A qualidade dos dados é igualmente fundamental para o desempenho do sistema. Os registos devem estar completos, sem valores em branco nos campos obrigatórios, e livres de duplicados. Entradas com categorias ou filtros inexistentes, inválidos ou descontinuados devem ser evitadas. Pressupõe-se ainda que os dados registados correspondem a interações reais na plataforma e não a dados simulados ou incompletos.

Por fim, o sistema deve permitir a integração de novos dados, por exemplo, através de atualizações diárias, sem comprometer a integridade ou desempenho dos modelos existentes.

[45] SQLite: https://sqlite.org/

Esses dados adicionais devem respeitar os mesmos critérios de formato e qualidade definidos para os dados históricos. Deste modo, assegura-se a atualização contínua das recomendações e a capacidade de adaptação do sistema à evolução dos comportamentos dos utilizadores.

## 5.2 Processo de obtenção de recomendações

Nesta subsecção é abordado o processo de obtenção das recomendações, mediante as interações que o utilizador fez no *website*, começando pelo pré-processamento dos dados e culminando na análise da eficácia dos algoritmos de recomendação utilizados.

### 5.2.1 Pré-processamento dos dados

Os métodos implementados nesta subsecção, necessitaram da biblioteca *pandas* para o tratamento de *dataframes*, que consistem numa estrutura de dados que se assemelha uma tabela [37]. As suas implementações estão presentes na classe *RecomendationService*, seguindo o princípio da programação orientada a objetos *Single Responsability Principle*, que promove a coesão, manutenibilidade e reutilização de código, evitando que uma classe acumule múltiplas responsabilidades, dificultando o seu entendimento e modificação [38].

Um dos passos executados na fase de pré-processamento consiste no processo de normalização dos dados. Neste caso, é aplicada a função *MinMaxScaler*, o que fez com que todas as interações fossem ajustadas para uma escala comum entre 0 e 1.

De seguida, foi aplicada a técnica TF-IDF para atribuir maior peso às categorias/filtros com maior valor informativo, ou seja, categorias que são relevantes para um utilizador, mas não comuns entre todos. Por exemplo, se uma determinada categoria/filtro for muito comum entre todos os utilizadores (como "Pulseiras"), o seu valor informativo é baixo. Em contraste, categorias/filtros que são pouco comuns, mas em que o utilizador revela mais interesse terão um peso mais elevado. Com isso, evita-se que categorias/filtros generalistas dominem as recomendações, o que promove recomendações mais personalizadas e diversificadas.

Posteriormente, foi realizada uma redução de dimensionalidade através do PCA, com o objetivo de reduzir o ruído dos dados, neste caso, remover categorias/filtros que ou não possuíam nenhuma interação ou que não possuíam um número relevante, desta forma possibilitou-se e melhorar a eficiência computacional e precisão dos algoritmos seguintes.

### 5.2.2 Algoritmos de recomendação

Foram usados três algoritmos distintos (popularidade, similaridade e *clustering*), com o objetivo de responder a diferentes desafios como o problema do *cold start*, a *sparcity* e a personalização das recomendações. As implementações destes métodos estão presentes na classe *RecomendationService*. Um quarto algoritmo (árvores de decisão) foi inicialmente pensado, mas descartado devido à escassez de dados. Com apenas 14 utilizadores com pelo menos 10 interações, modelos como *DecisionTreeClassifier* da biblioteca *sklearn,* não teriam dados suficientes para aprender padrões significativos, o que levaria a um elevado risco de *overfitting*.

#### 5.2.2.1 Algoritmo de popularidade

Este método baseia-se na frequência global das categorias mais interagidas por todos os utilizadores. É particularmente eficaz em situações de *cold start*, quando não existem interações suficientes de um novo utilizador para gerar recomendações personalizadas (menos de 10 interações). Apesar da sua simplicidade, serve como uma boa linha de base comparativa para os restantes métodos. A sua implementação foi feita com recurso às funções de agregação da biblioteca *pandas*, para calcular as categorias/filtros mais populares no conjunto de dados.

#### 5.2.2.2 Algoritmo de similaridade

Este algoritmo baseia-se na distância do cosseno, utilizando a função *cosine_similarity* da biblioteca *sklearn*. Este método mede o grau de semelhança entre vetores de clientes, com base na sua orientação [39]. Após calcular a similaridade entre os vetores dos clientes, foram identificados os perfis mais semelhantes, o que permitiu recomendar categorias/filtros que outros clientes com perfis parecidos utilizaram frequentemente. Esta abordagem revelou-se eficaz, mesmo com poucos dados por cliente, já que obtém a informação a partir de padrões partilhados entre utilizadores com interesses comuns.

#### 5.2.2.3 Algoritmo de clustering

Para identificar grupos de utilizadores com padrões de comportamento semelhantes, foi utilizado o algoritmo *K-Means* da biblioteca *sklearn*. Optou-se pela utilização do algoritmo *K-Means* em detrimento de alternativas como *Fuzzy C-Means*, DBSCAN, *Gaussian Mixture* ou *Spectral Clustering*, por se adequar melhor às características do conjunto de dados após a vectorização com TF-IDF e a redução de dimensionalidade com PCA. O *K-Means* é eficiente em contextos com *clusters* bem definidos e separáveis, sendo também computacionalmente leve e escalável, o que permitiu testar diferentes configurações com facilidade. As alternativas foram descartadas por motivos específicos: o *Fuzzy C-Means* introduz complexidade desnecessária ao permitir pertença parcial a múltiplos *clusters*; o DBSCAN tem dificuldades com muitas variáveis; o *Gaussian Mixture* requer que os dados estejam razoavelmente alinhados com distribuições gaussianas bem definidas (caso contrário os resultados podem ser imprecisos ou enganosos), o que não era garantido; e o *Spectral Clustering*, embora uma boa alternativa, apresenta maior custo computacional. O número ideal de clusters foi determinado com base na métrica de *silhouette score*, podendo também ter sido utilizado como alternativa o método *elbow*, ambos presentes na biblioteca *sklearn.* Esta abordagem permitiu segmentar os clientes, de modo a gerar as recomendações pretendidas.

### 5.2.3 Avaliação do desempenho

Para medir a eficácia de cada algoritmo, foi implementado um processo de avaliação com base no histórico real de interações dos utilizadores. Com o objetivo de medir até que ponto as recomendações geradas por cada algoritmo conseguiam antecipar corretamente as preferências futuras dos clientes. Inicialmente, os dados foram divididos em dois subconjuntos, o conjunto de treino, usado para gerar os perfis e realizar as recomendações, e o conjunto de teste, contendo interações reais posteriores, utilizadas para verificar se as recomendações eram concordantes com o comportamento efetivo. Para quantificar o desempenho, foi escolhida a métrica *F1-score*, da biblioteca *sklearn*. O F1-score combina duas métricas relevantes para este caso de estudo, a *Precision*, que avalia a proporção de recomendações

corretas entre todas as categorias/filtros recomendados e o *Recall*, que mede a proporção de categorias/filtros corretos (do conjunto de teste) que foram efetivamente recomendadas. Assim, o *F1-score* oferece uma medida mais equilibrada da eficácia do sistema, especialmente em casos em que os dados são limitados e as recomendações corretas são uma minoria entre muitas possibilidades [40]. Desta forma, foi possível identificar qual o algoritmo que apresentou melhor desempenho, permitindo uma análise fundamentada da sua adequação ao problema e aos dados disponíveis para, por fim, ser recomendado ao cliente.

### 5.2.4 Cálculo de probabilidade e margem de erro

Na fase final das recomendações, foi implementado um método de avaliação estatística com o objetivo de estimar a fiabilidade dos resultados produzidos por cada algoritmo. Para isso, recorreu-se ao cálculo de intervalos de confiança para proporções, de modo a quantificar a incerteza associada à taxa de acerto observada nas recomendações. A taxa de acerto (*p*) foi calculada como a proporção de categorias recomendadas que coincidiam com o histórico real de interações dos utilizadores. Com base nessa proporção e no número de observações (*n*), foi utilizada a fórmula da margem de erro de um intervalo de confiança para proporções:

$$ME = Z * \sqrt{\frac{p(1-p)}{n}}$$

Equação 9 – Margem de erro

Neste contexto, o valor de *Z* corresponde ao quantil da distribuição normal padrão. Para um intervalo de confiança de 95%, o valor de *Z* é aproximadamente 1.96. O método desenvolvido com auxílio das bibliotecas *numpy* e *scipy*, recebe a proporção *p*, o número de observações *n* e o nível de confiança, e retorna a margem de erro associada. Esta abordagem permite expressar os resultados de cada algoritmo não apenas em termos de desempenho médio, mas também em termos da sua robustez estatística.

## 5.3 Base de dados

Depois de avaliadas outras opções de base de dados online decidiu-se optar por utilizar um servidor *SQLite* para armazenar os dados das interações dos clientes com os filtros e categorias do *website* e a informação relativa às recomendações. Esta escolha foi tomada, devido ao facto de a base de dados ser gratuita, segura, ter compatibilidade com *Python* e ser capaz de guardar os formatos de dados pretendidos. A Figura 14 mostra o modelo relacional implementado na base de dados e as tabelas utilizadas na realização desta *framework.*

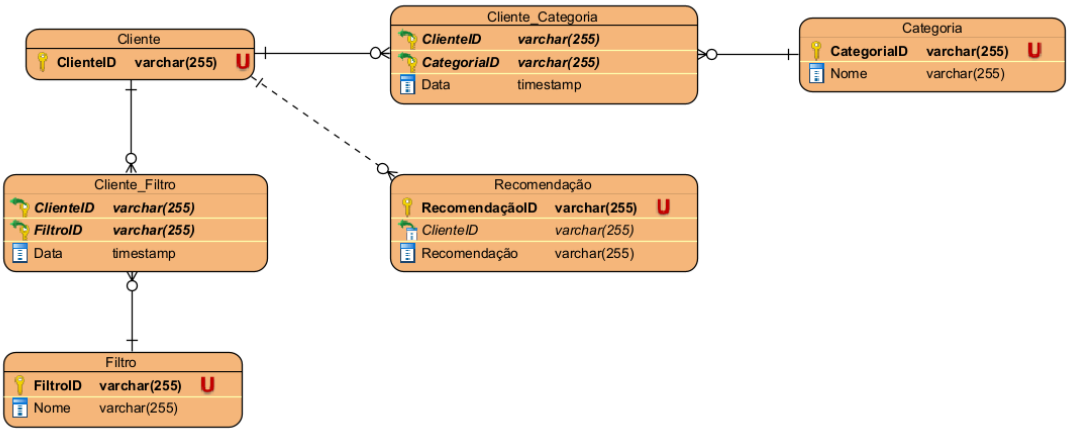


Figura 14 - Modelo relacional

A tabela *Cliente* guarda os utilizadores da plataforma, sendo o campo *ClienteID* a chave primária que identifica univocamente cada cliente. Esta tabela funciona como ponto de partida para as relações com os dados comportamentais. A tabela *Categoria* armazena as diferentes categorias disponíveis no sistema, sendo o campo *CategoriaID* a chave primária e com um *Nome* associado. De forma semelhante, a tabela *Filtro* contém os diferentes filtros que os clientes podem aplicar nas suas interações, identificados por *FiltroID* e respetivo *Nome*. Nestas três tabelas foram usados apenas os campos necessários para a análise. As interações dos clientes com categorias são registadas na tabela *Cliente_Categoria*, que liga a tabela *Cliente* à tabela *Categoria* através das chaves *ClienteID* e *CategoriaID*. Esta tabela inclui também um campo *Data*, com o momento da interação. De mesma forma, a tabela *Cliente_Filtro* representa a aplicação de filtros por parte dos clientes. Esta tabela liga-se com recurso ao *ClienteID* e *FiltroID*, com a respetiva data de cada interação. Por fim, a tabela Recomendação armazena os resultados finais do sistema de recomendação. Cada recomendação é identificada por *RecomendaçãoID* e associa um *ClienteID* ao conteúdo recomendado, armazenado no campo *Recomendação*. Esta tabela reflete diretamente o output do sistema, ou seja, as categorias ou filtros recomendados. Embora o modelo não inclua relações complexas como herança ou dependências funcionais avançadas, escolheu-se uma base de dados relacional para que a solução fosse escalável para possíveis novos requisitos.

## 5.4 Interface gráfica

De forma a disponibilizar uma simples interface para demonstração da *framework*, foi desenvolvido um *frontend* em *React*[46] com o auxílio da *framework bootstrap*[47]. A Figura 15 apresenta a página inicial onde se poderá escolher o cliente desejado para executar a análise.

[46] React: https://react.dev/
[47] Bootstrap: https://www.hostinger.com.br/tutoriais/o-que-e-bootstrap

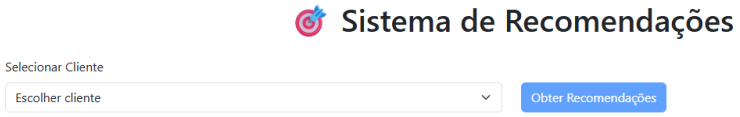


Figura 15 - Interface para obter as recomendações

Após a seleção do cliente pretendido, são apresentados os dados das categorias com as quais interagiu (Figura 16), seguidos pelas recomendações geradas por cada um dos algoritmos implementados (Figura 17). Por fim, é destacada a recomendação considerada mais adequada com base na avaliação de desempenho dos algoritmos (Figura 18).

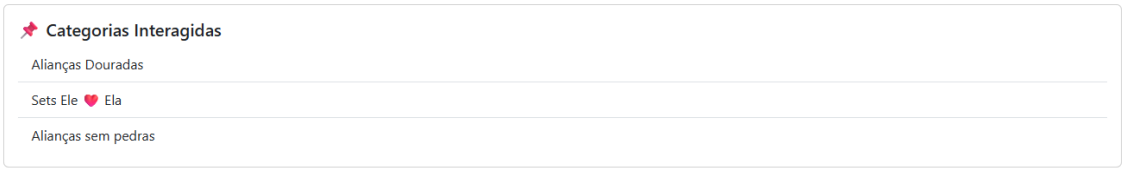


Figura 16 - Categorias interagidas

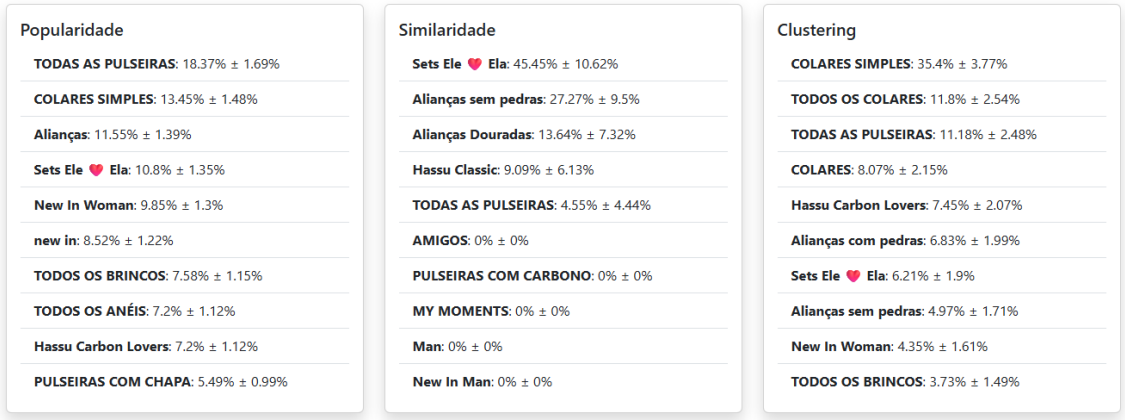


Figura 17 - Output dos algoritmos


Figura 18 - Recomendação final

## 5.5 Testes

Nesta secção descrevem-se os testes realizados na *framework*, incluindo testes unitários e de integração. Os testes unitários focaram-se em verificar o funcionamento correto de pequenas partes do código, como funções e métodos específicos, e garantem que as regras de negócio foram corretamente aplicadas. Já os testes de integração avaliaram se os diferentes componentes da *framework* interagem corretamente entre si, e asseguram que o sistema funciona como um todo. A definição dos testes foi feita por mim, que, por conhecer bem o projeto e os seus objetivos, desenhei os casos de teste mais relevantes para garantir a qualidade da solução. Os testes foram também executados e validados por mim, que analisei os resultados cuidadosamente para identificar e corrigir possíveis erros ao longo do processo de desenvolvimento. Devido à natureza do projeto, não foram incluídas evidências formais dos testes. No entanto, todos os testes foram executados manualmente e os seus resultados avaliados, para garantir a qualidade da *framework*.

### 5.5.1 Testes unitários

Ao longo do desenvolvimento da *framework*, foram desenvolvidos os testes unitários apropriados. Estes testes têm como objetivo testar pequenas unidades do código, o que possibilita focar a atenção apenas nessa pequena porção e localizar a falha rapidamente.

Foram utilizados testes unitários para testar o objeto de domínio *Recomendation*. Neste caso, testou-se a criação do objeto para verificar se as regras de negócio foram devidamente implementadas. Estas especificam que nenhum campo pode ser nulo e, no caso da lista de recomendações também não pode ser vazia. Os testes foram efetuados aos possíveis casos de insucesso, bem como aos casos de sucesso.

Quanto à classe *Service*, foram implementados teste unitários a todos os métodos da classe, visto que são estes que implementam as regras de negócio. Na classe *RecomendationService* foi testado se os métodos retornam algo, visto que não é possível prever a decisão de classificação dos métodos.

Relativamente à classe *Repository*, os testes unitários verificam a funcionalidade das *queries* efetuadas. Para tal, utilizou-se *mocks*, recorrendo à biblioteca *unittest*, disponível no *Python*, para simular a conexão com a base de dados. Desta forma, foram testados os métodos da classe *RecomendationRepository*. Depois da execução da *query* do tipo *INSERT*, o teste unitário verifica se o *commit* foi efetuado na base de dados para testes. Caso a *query* seja do tipo *SELECT*, o teste unitário verifica se o valor de retorno da *query* é o esperado.

Por fim, foram realizados testes unitários à classe *Controller*. Estes testes asseguram que todos os métodos estão a ser corretamente invocados. Para o teste foi necessário usar *mocks* na chamada, que o *RecomendationController*, faz de todos os métodos.

Não foram realizados testes unitários ao *frontend*, visto que este foi criado com apenas o objetivo de demonstrar a *framework* e possivelmente será descartado futuramente.

### 5.5.2 Testes de integração

Com a finalidade de analisar a integração da *framework* desenvolvida com a base de dados *SQLite*, foram elaborados alguns testes de integração. Estes testes, têm por objetivo, testar o funcionamento da aplicação como um todo. Com isso em mente, foi criada uma base de dados local, idêntica à referida na secção 5.3, exclusivamente para fins de teste.

Os testes realizados, foram implementados recorrendo à ferramenta *Postman*[48], que oferece a capacidade de enviar pedidos HTTP e fornece recursos para executar *scripts* antes e depois do pedido. A integração com a base de dados é realizada por intermédio dos métodos de *backend* que realizam *queries* de *SELECT* e *INSERT*. Na Tabela 3, estão representados os vários testes que foram desenvolvidos. Todos os testes aqui descritos foram realizados com sucesso.

Tabela 3 - Testes de integração

| **Objetivo** | Verificar o correto funcionamento dos *endpoints* de listagem dos clientes e geração de recomendações |
|---|---|
| **Pré-requisito** | A base de dados deve conter clientes e o histórico de interações devidamente carregados |
| **Procedimento** | |
| **Ação** | **Resultado esperado** |
| HTTP GET dos clientes | Sucesso (código 200) e retorno de uma lista ordenada de *client_id*s existentes |
| HTTP GET para obter as recomendações de um *client_id* inexistente | Erro (código 404) |
| HTTP GET para obter as recomendações de um *client_id* com menos de 10 interações | Sucesso (código 200), recomendações geradas com o algoritmo de popularidade como melhor algoritmo |
| HTTP GET para obter as recomendações de um *client_id* com menos de 10 interações | Sucesso (código 200), recomendações geradas com o melhor algoritmo identificado com base no *F1-score* |

[48] Postman: https://www.postman.com

# 6 Avaliação da solução

Nesta secção, vão ser apresentadas as funcionalidades da Framework através de um caso de estudo, que pretende demonstrar 1) *input* para a geração de recomendações, 2) a visualização dos resultados e 3) as métricas de avaliações usadas. Para esse efeito, os dados relativos ao acesso a categorias registados entre os dias 1 de setembro e 24 de novembro de 2024, referentes a 85 dias e 1000 registos, foram devidamente carregados para uma base de dados *SQLite*. O analista de dados só tem de selecionar o *cliente_id* (é usado o *cliente_id* e não o nome do cliente devido à proteção de dados pessoais), como a Figura 19 mostra, para que a análise seja feita.

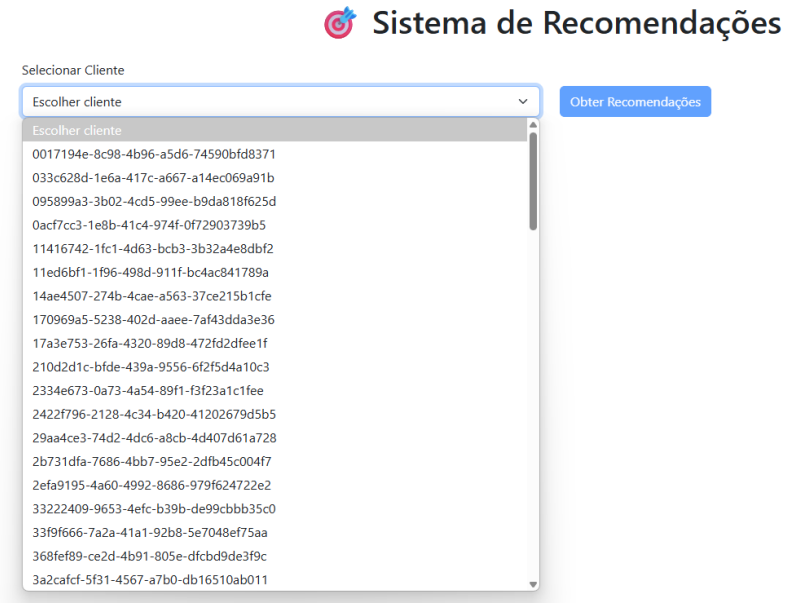


Figura 19 - Inputs para a geração de recomendações

Após a seleção são gerados os vários outputs e, caso o cliente tenha poucas interações, as recomendações guardadas na base de dados serão as correspondentes ao algoritmo de popularidade (combater o *cold start*). Caso o cliente possua dez ou mais interações o algoritmo

recomendado será baseado no melhor resultado da métrica *F1-score*. A Figura 20 e a Figura 21 mostram o output gerado quando o cliente possuí várias interações com as categorias do *website* da Flamingo S.A.


Figura 20 - Visualização dos outputs parte 1

Conforme a Figura 21 mostra, todos os algoritmos são executados e são apresentados com uma avaliação percentual, que representa a métrica de precisão e a respetiva margem de erro.

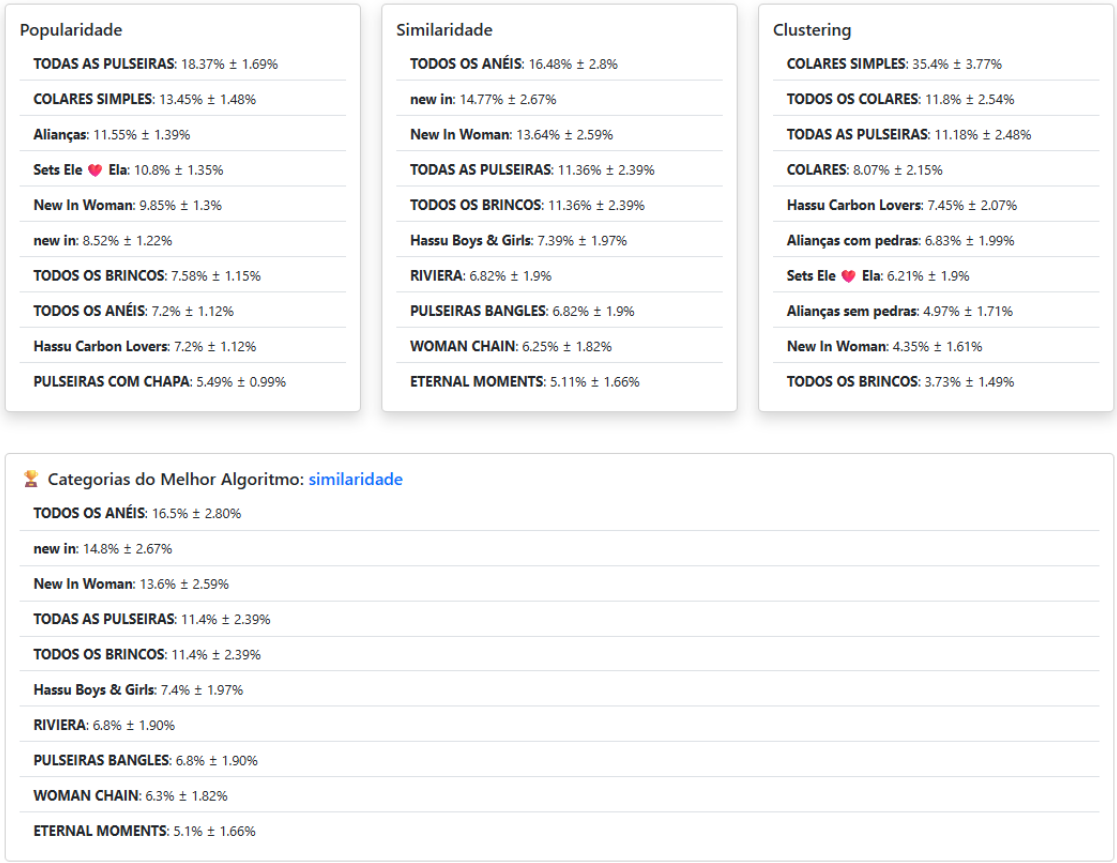


Figura 21 - Visualização dos outputs parte 2

Esta métrica representa a capacidade de cada algoritmo gerar recomendações que correspondem efetivamente às categorias com as quais o utilizador interagiu no futuro. Para isso, foi utilizada uma divisão dos dados em conjuntos de treino e teste. A percentagem indica a taxa de acerto nas recomendações, enquanto a margem de erro reflete a incerteza estatística associada ao cálculo. Esta métrica foi usada para proporcionar uma visão mais informada sobre a eficácia da previsão de cada categoria.

Para além disso, todos os algoritmos são executados e avaliados com base na métrica de *F1-score*, que combina a *precision* e *recall* numa única medida, o que permite avaliar a capacidade de cada algoritmo em gerar recomendações que correspondem efetivamente às categorias com as quais o utilizador viria a interagir no futuro. Para tal, os dados foram divididos em conjuntos de treino e teste. A Tabela 4 apresentada ilustra os *F1-scores* obtidos para diferentes quantidades de recomendações (de 1 a 10). Esta considera apenas clientes com mais de 10 interações nas categorias do *website* e oferece uma perspetiva do desempenho dos três algoritmos.

O algoritmo de popularidade apresenta valores baixos, o que já era expectável, dado que é concebido para clientes com poucas interações, servindo essencialmente como estratégia para mitigar o problema do *cold start*. O algoritmo de similaridade mostra um desempenho bastante satisfatório, especialmente nas primeiras cinco recomendações, atingindo o seu pico com quatro recomendações (*F1-score* de 0.6272). Já o algoritmo de *clustering* apresenta um desempenho inferior, possivelmente devido à limitação atual de dados disponíveis (1000 dados). Contudo, penso que, com a quantidade de dados de (pelo menos um ano) que a

empresa possuí, este método venha a apresentar melhorias substanciais, dado o seu potencial para identificar padrões em conjuntos de dados maiores.

Tabela 4 - F1-score dos algoritmos (clientes >10 interações)

| | F1-Score dos algoritmos | | |
|---|---|---|---|
| Nº recomendações | Popularidade | Similaridade | Clustering |
| 1 | 0.0923 | 0.3689 | 0.2706 |
| 2 | 0.0735 | 0.6048 | 0.4 |
| 3 | 0.0624 | 0.6188 | 0.3411 |
| 4 | 0.107 | 0.6272 | 0.3289 |
| 5 | 0.1123 | 0.6062 | 0.3154 |
| 6 | 0.1127 | 0.5652 | 0.2978 |
| 7 | 0.1164 | 0.5571 | 0.2789 |
| 8 | 0.1094 | 0.5162 | 0.2551 |
| 9 | 0.1079 | 0.4909 | 0.2565 |
| 10 | 0.1092 | 0.4701 | 0.2459 |

Adicionalmente, foi realizada uma nova avaliação, mas desta vez considerando clientes com pelo menos 5 interações, em vez de 10 como na análise anterior. Tentou-se ainda aplicar um filtro mais restritivo, excluindo clientes com menos de 15 interações, mas tal não foi possível devido à inexistência de clientes que cumprissem esse critério no conjunto de dados disponibilizado.

Na Tabela 5, foi possível observar uma redução no desempenho dos algoritmos baseados em similaridade e *clustering*. Em média, *o F1-score* do algoritmo de similaridade caiu de 0.5366 para 0.4317, correspondendo a uma redução aproximada de 19,5%. Já o algoritmo de *clustering* passou de uma média de 0.3115 para 0.2973, o que representa uma diminuição de cerca de 4,6%. Estes resultados indicam que a inclusão de clientes com menos interações torna mais difícil a personalização eficaz das recomendações, especialmente nos métodos que dependem de padrões de comportamento para funcionarem corretamente.

Tabela 5 - F1-score dos algoritmos (clientes >5 interações)

| | F1-Score dos algoritmos | | |
|---|---|---|---|
| Nº recomendações | Popularidade | Similaridade | Clustering |
| 1 | 0.0923 | 0.427 | 0.3625 |
| 2 | 0.0735 | 0.5354 | 0.3833 |

| | F1-Score dos algoritmos | | |
|---|---|---|---|
| Nº recomendações | Popularidade | Similaridade | Clustering |
| 3 | 0.0624 | 0.5009 | 0.3488 |
| 4 | 0.107 | 0.4758 | 0.3257 |
| 5 | 0.1123 | 0.4603 | 0.2991 |
| 6 | 0.1127 | 0.4387 | 0.2667 |
| 7 | 0.1164 | 0.409 | 0.2625 |
| 8 | 0.1094 | 0.3827 | 0.247 |
| 9 | 0.1079 | 0.353 | 0.2352 |
| 10 | 0.1092 | 0.3284 | 0.2321 |

Por fim, com o objetivo de avaliar a eficácia do pré-processamento aplicado aos dados (incluindo normalização, transformação TF-IDF e redução de dimensionalidade via PCA), foi realizada uma comparação entre os *F1-scores* dos algoritmos com e sem este tratamento. Os resultados comparação entre a Tabela 4 e a Tabela 6 demonstram uma melhoria clara no desempenho dos métodos de similaridade e *clustering*.

No caso do algoritmo de similaridade, o *F1-score* médio passou de 0.3207 (sem pré-processamento) para 0.5366 (com pré-processamento), representando um aumento de aproximadamente 67,3%. Já para o *clustering*, a média dos *F1-scores* subiu de 0.1311 para 0.3115, o que equivale a uma melhoria de cerca de 137,5%. Estes resultados comprovam que o pré-processamento contribuiu significativamente para a qualidade das recomendações.

Tabela 6 - F1-score dos algoritmos sem pré-processamento aplicado

| | F1-Score dos algoritmos | | |
|---|---|---|---|
| Nº recomendações | Popularidade | Similaridade | Clustering |
| 1 | 0.0923 | 0.4743 | 0.1346 |
| 2 | 0.0735 | 0.4496 | 0.1363 |
| 3 | 0.0624 | 0.3822 | 0.1523 |
| 4 | 0.107 | 0.3334 | 0.1422 |
| 5 | 0.1123 | 0.3076 | 0.1325 |
| 6 | 0.1127 | 0.292 | 0.1212 |
| 7 | 0.1164 | 0.2684 | 0.1159 |

| | F1-Score dos algoritmos | | |
|---|---|---|---|
| Nº recomendações | Popularidade | Similaridade | Clustering |
| 8 | 0.1094 | 0.2483 | 0.1212 |
| 9 | 0.1079 | 0.232 | 0.1219 |
| 10 | 0.1092 | 0.2195 | 0.1135 |

# 7 Conclusão

Esta secção apresenta as reflexões finais, abordando os objetivos concretizados, limitações, trabalho futuro e a apreciação final global do trabalho.

## 7.1 Objetivos concretizados

Durante o período do estágio, os objetivos inicialmente propostos foram alcançados, tendo sido os seguintes:

- Estudo e formulação do estado de arte, que inclui uma descrição dos pontos principais em que o projeto iria incidir, uma vista sobre as várias tecnologias existentes e trabalhos relevantes relativos a *clustering*, estratégias de recomendação e métricas de avaliação;
- Cumprimentos de todos os requisitos funcionais e não funcionais;
- Análise e desenho de uma proposta de solução e de apresentação de possíveis alternativas à mesma;
- Desenvolvimento da *framework,* seguindo o desenho da solução proposta, sendo ela capaz de:
    - Pré-processar os dados obtidos para, primeiro, obter um conjunto de dados consistente, segundo, normalizado para a melhor performance dos algoritmos, e, terceiro, com apenas o conjunto de dados relevantes para a gerações de recomendações (remoção categorias/filtros com poucas ou nenhumas interações);
    - Gerar recomendações recorrendo a vários algoritmos de recomendação;
    - Se adaptar ao *cold start* presente em novos clientes através do algoritmo de popularidade;
    - Avaliar os algoritmos com base no *F1-score* para obter a melhor recomendação;

o Guardar na base de dados, de forma automática, o *output* do melhor algoritmo;

• Teste e validação da solução implementada, recorrendo a testes unitários, testes de integração e aplicação de casos de estudo com dados reais.

Com base nos resultados obtidos, é possível concluir que as funcionalidades solicitadas foram implementadas e testadas com sucesso. A *framework* foi implementada seguindo o planeamento efetuado na fase de *design* arquitetural, produzindo uma solução capaz de utilizar várias técnicas para a obtenção de recomendações.

## 7.2 Limitações e trabalho futuro

Ao longo destes meses de desenvolvimento, foi possível cumprir os objetivos do projeto, tendo-se desenvolvido uma solução com a capacidade de integrar novas funcionalidades. Após a conclusão do estágio e entrega deste projeto, a solução poderá continuar a ser desenvolvida, estando aberta a novos requisitos. Neste momento, é possível identificar possíveis novos requisitos para um trabalho futuro, podendo estes ser:

• Inclusão de uma métrica que relacione as categorias com os filtros interagidas pelo cliente;

• Utilização de um conjunto maior de dados para avaliações mais precisas;

• Automação para ler dados em tempo real de uma base de dados e gerar as recomendações em vez de necessitar da interação do analista de dados;

• Acrescentar outras etapas de pré-processamento que possam se revelar importantes;

• Utilização de outros métodos de *clustering*.

Também foi possível identificar uma limitação no trabalho desenvolvido, que é a escalabilidade da aplicação, visto que, futuramente se lhe for fornecido um grande volume de dados, existe a possibilidade de a aplicação demorar muito tempo a executar as várias etapas.

## 7.3 Apreciação final

Na minha perspetiva, avalio a realização do estágio da Flamingo S.A. de uma forma positiva. Isto deve-se ao facto de ter ampliado significativamente os meus conhecimentos na área de *machine learning*, com especial foco na implementação de sistemas de recomendação. Para além disso, fui capaz de alargar as minhas capacidades de desenvolvimento de *software*, tendo aprofundado os meus conhecimentos na linguagem de programação *Python*. Em suma, considero que este estágio teve um impacto muito relevante no meu percurso académico, por me ter permitido aplicar conhecimentos adquiridos ao longo da licenciatura e mestrado e desenvolver novas competências técnicas e práticas alinhadas com os desafios atuais do mercado tecnológico. Importa ainda referir que a empresa Flamingo S.A. manifestou satisfação com os resultados obtidos, tendo demonstrado interesse em implementar o algoritmo desenvolvido no seu *website* num futuro próximo.

# Referências


[1] FLAMINGO, “FLAMINGO,” 2024. [Online]. Available: https://www.flamingo.pt/. [Acedido em 30 11 2024].

[2] J. D. B. C. C. M. Nabile M. Safdar, “Ethical considerations in artificial intelligence,” *https://www.sciencedirect.com/science/article/pii/S0720048X19304188?casa_token=7UvqBpEPH08AAAAA:6joYErNyQcRFoQNG3A2yv1nhtCrtML0OINR7Ww3k0RlHX7gShz-0rXkyA7ybzR_FTKjUzfj-8A,* 2020.

[3] A. G. A. G, H.-K. Su e W.-K. Kuo, “Personalized E-commerce: Enhancing Customer Experience through Machine Learning-driven Personalization,” *https://ieeexplore.ieee.org/abstract/document/10624901,* 2024.

[4] M. F. Dzulfikar, B. Purwandari, D. I. Sensuse, J. S. Lusa, I. Solichah e P. Prima, “Personalization Features on Business-to-Consumer E-Commerce: Review and,” *https://ieeexplore.ieee.org/abstract/document/8392839,* 2018.

[5] H. K. Thakur, J. Singh, A. Saxena, D. Bhaskar, A. P. Singh e P. K. Garg, “Enhancing Customer Experience through AI-Powered Personalization: A Data Science Perspective in E-Commerce,” *https://ieeexplore.ieee.org/abstract/document/10592893,* 2024.

[6] Y. Luo, “High-Performance E-Commerce Personalized Recommendation System Based on Matrix Factorization and Convolutional Neural Networks,” *https://ieeexplore.ieee.org/abstract/document/10527715,* 2023.

[7] T. Patricio, “Medium,” 16 9 2021. [Online]. Available: https://thaispatricio.medium.com/awari-bookstore-recsys-cc50dba00cf1. [Acedido em 11 2024].

[8] S. Koul e T. M. Philip, “Customer Segmentation Techniques on E-Commerce,” *https://ieeexplore.ieee.org/abstract/document/9404659,* 2021.

[9] L. Rajput e S. N. Singh, “Customer Segmentation of E-commerce data using K-means Clustering Algorithm,” *https://ieeexplore.ieee.org/abstract/document/10048834,* 2023.

[10] S. M. N. Mustafa, A. Akhtar, J. T. P. Noronha, M. Salman e M. A. Baig, “Customer Segmentation using Machine learning Techniques,” *https://ieeexplore.ieee.org/abstract/document/10075194,* 2023.

[11] N. S. S. Reddy, V. V. A. Rohith, P. S. Abhiram, M. D. S. R. Saran e S. Rebecca, “Enhancing Product Categorization in E-commerce using NLP and Machine Learning,” *https://ieeexplore.ieee.org/abstract/document/10544665,* 2024.

[12] A. G. Kanaan, F. R. Wahsheh, Y. A. B. El-Ebiary, W. M. A. F. W. Hamzah, B. Pandey e S. N. P, “An Evaluation and Annotation Methodology for Product Category Matching in E-Commerce Using GPT,” *https://ieeexplore.ieee.org/abstract/document/10346684,* 2023.

[13] T. M. Tashu, S. Fattouh, P. Kiss e T. Horváth, “Multimodal E-Commerce Product Classification Using Hierarchical Fusion,” *https://ieeexplore.ieee.org/abstract/document/9914136,* 2022.

[14] S. Lloyd, “Least squares quantization in PCM,” *https://ieeexplore.ieee.org/document/1056489,* 1982.

[15] Y. B. Léon Bottou, “Convergence Properties of the K-Means,” *https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=cee30ccc0c341dfacaac078f1560526ceae701df,* 1994.

[16] J. C. Dunn, “A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters,” *https://www.tandfonline.com/doi/abs/10.1080/01969727308546046,* 2008.

[17] J. C. Bezdek, “Pattern Recognition with Fuzzy Objective Function Algorithms,” *https://www.researchgate.net/publication/233932672_Pattern_Recognition_With_Fuzzy_Objective_Function_Algorithms,* 1981.

[18] J. Kolen e T. Hutcheson, “Reducing the time complexity of the fuzzy c-means algorithm,” *https://ieeexplore.ieee.org/document/995126,* 2002.

[19] A. P. Dempster, N. M. Laird e D. B. Rubin, “Maximum Likelihood from Incomplete Data Via the EM Algorithm,” *https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x,* 1977.

[20] C.-T. Lin, C.-S. Huang, W.-Y. Yang, A. K. Singh, C.-H. Chuang e Y.-K. Wang, “Real-Time EEG Signal Enhancement Using Canonical Correlation Analysis and Gaussian Mixture Clustering,” *https://onlinelibrary.wiley.com/doi/full/10.1155/2018/5081258,* 2018.

[21] A. K. Jain, M. N. Murty e P. J. Flynn, “Data clustering: a review,” *https://dl.acm.org/doi/10.1145/331499.331504,* 1999.

[22] D. Cai e X. Chen, “Large Scale Spectral Clustering Via Landmark-Based Sparse Representation,” *https://ieeexplore.ieee.org/abstract/document/6910247,* 2011.

[23] M. Ester, H.-P. Kriegel, J. Sander e X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” *https://dl.acm.org/doi/10.5555/3001460.3001507,* 1996.

[24] K. Khan, S. U. Rehman, K. Aziz, S. Fong e S. Sarasvady, “DBSCAN: Past, present and future,” *https://ieeexplore.ieee.org/document/6814687,* 2014.

[25] S. N. Hasan e R. Khatwal, “Cold Start Problem in Recommendation System: A Solution Model Based on Clustering and Association Rule Techniques,” *https://ieeexplore.ieee.org/document/10029293,* 2022.

[26] R. Alabdulrahman e H. Viktor, “Catering for unique tastes: Targeting grey-sheep users recommender systems through one-class machine learning,” *https://www.sciencedirect.com/science/article/abs/pii/S0957417420308241,* 2021.

[27] G. Pastor, I. Mora-Jiménez, R. Jäntti e A. J. Caamaño, “Constructing Measures of Sparsity,” *https://ieeexplore.ieee.org/document/9219175,* 2020.

[28] S. Wei, J. Zhang, Z. Yang, Q. Li, Y. Pang e Y. Xiao, “A Click Conversion Rate Model of E-Commerce Platforms Aiming at Effective Data Sparse,” *https://ieeexplore.ieee.org/abstract/document/10418583,* 2024.

[29] M. Bakhtyari e S. Mirzaei, “Click-Through Rate Prediction Using Feature Engineered Boosting Algorithms,” *https://ieeexplore.ieee.org/document/9420546,* 2021.

[30] A. R. Panda, S. Rout, M. Narsipuram, A. Pandey e J. J. Jena, “Ad Click-Through Rate Prediction: A Comparative Study of Machine Learning Models,” *https://ieeexplore.ieee.org/document/10481562,* 2024.

[31] A. Lakshmanarao, S. Ushanag e B. S. Leela, “Ad Prediction using Click Through Rate and Machine Learning with Reinforcement Learning,” *https://ieeexplore.ieee.org/document/9616653,* 2021.

[32] G. Chauhan e D. V. Mishra, “Evaluating deep learning based models for predicting click through rate,” *https://ieeexplore.ieee.org/document/9031059,* 2019.

[33] K. Xu, H. Zhou, H. Zheng, M. Zhu e Q. Xin, “Intelligent Classification and Personalized Recommendation of E-commerce Products Based on Machine,” *https://arxiv.org/abs/2403.19345,* 2024.

[34] M. Z. Haque, “E-Commerce Product Recommendation System based on ML Algorithms,” *https://arxiv.org/abs/2407.21026,* 2024.

[35] Q. Ai, Y. Zhang, K. Bi, X. Chen e W. B. Croft, “Learning a Hierarchical Embedding Model for Personalized Product Search,” *https://dl.acm.org/doi/10.1145/3077136.3080813,* 2017.

[36] Amazon, “https://www.theverge.com/2024/9/19/24249046/amazon-generative-ai-tools-personalized-shopping-recommendations,” 2024. [Online]. Available: https://www.theverge.com/2024/9/19/24249046/amazon-generative-ai-tools-personalized-shopping-recommendations.

[37] W. McKinney, “Data Structures for Statistical Computing in Python,” *https://proceedings.scipy.org/articles/Majora-92bf1922-00a,* 2010.

[38] R. C. Martin, “Agile Software Development: Principles, Patterns, and Practices,” *https://dl.acm.org/doi/10.5555/515230,* 2003.

[39] M. Alodadi e V. P. Janeja, “Similarity in Patient Support Forums Using TF-IDF and Cosine Similarity Metrics,” *https://ieeexplore.ieee.org/document/7349760,* 2015.

[40] P. Cremonesi, Y. Koren e R. Turrin, “Performance of recommender algorithms on top-n recommendation tasks,” *https://dl.acm.org/doi/10.1145/1864708.1864721,* 2010.

# Anexo A - WBS

Anexo 1 - WBS

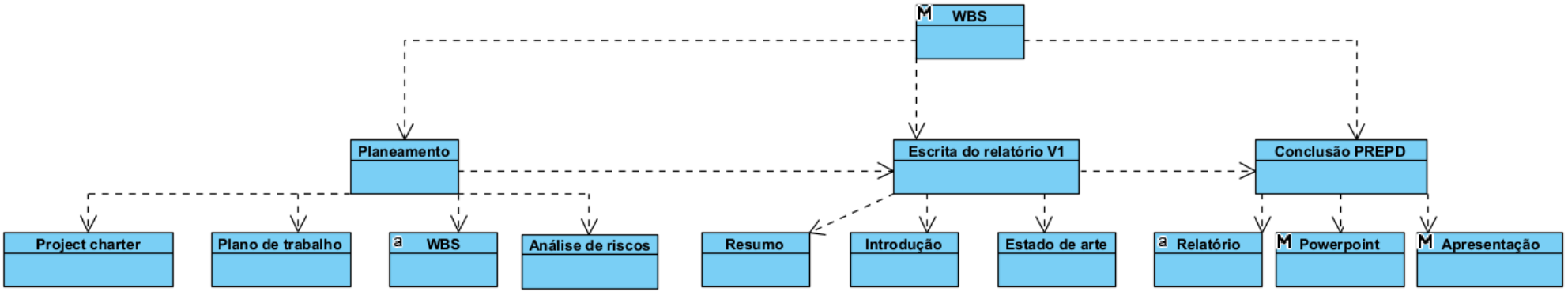


Figura 22 - WBS Preparação para a dissertação

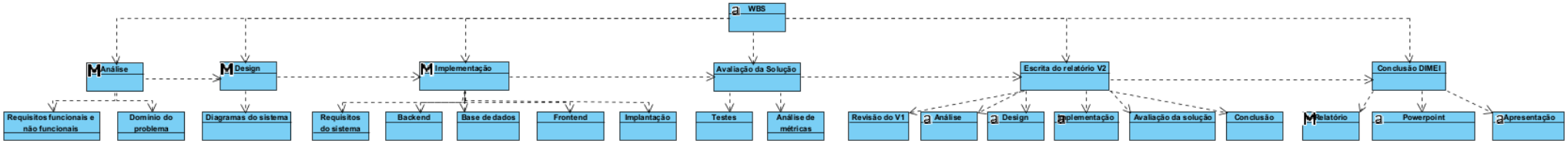


Figura 23 - WBS Dissertação

# Anexo B - Diagrama de Gantt

Anexo 2 - Diagrama de Gantt

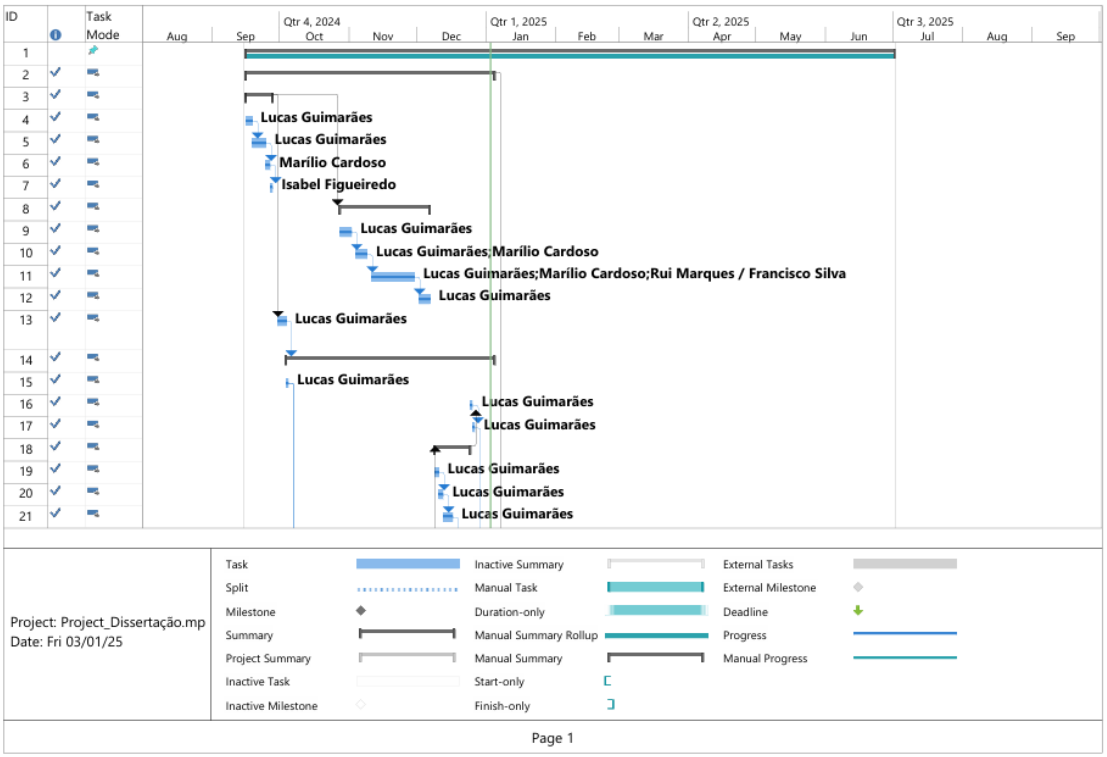


Figura 24 - Diagrama de Gantt parte 1

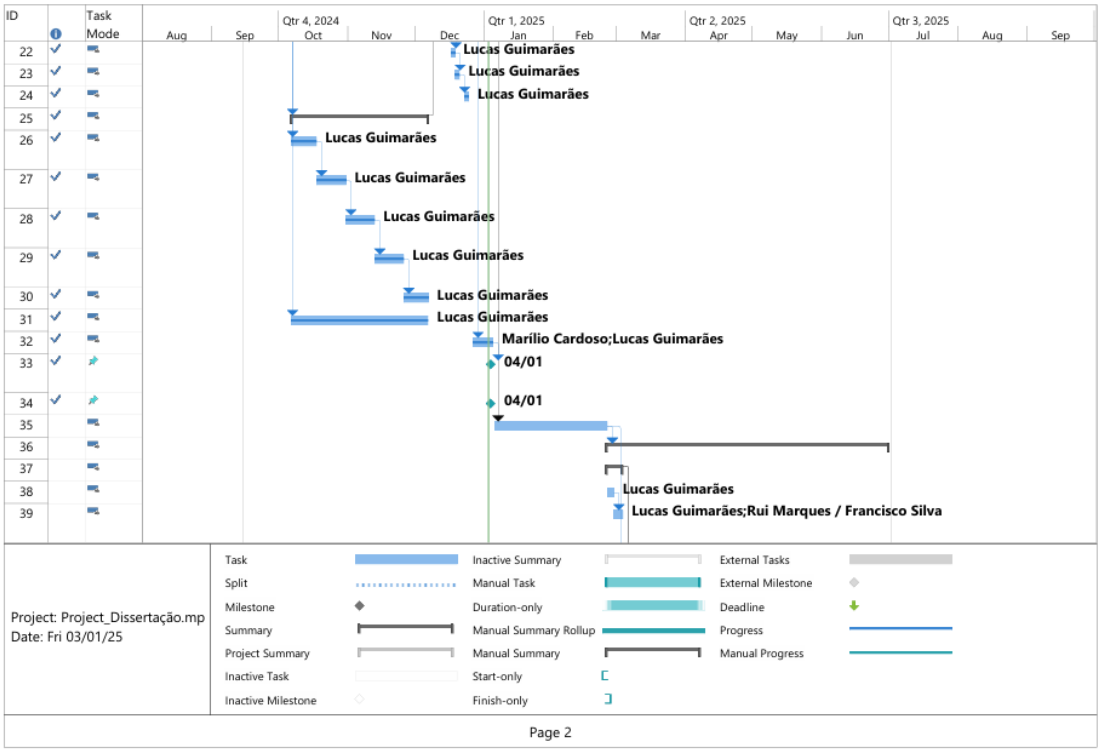


Figura 25 - Diagrama de Gantt parte 2

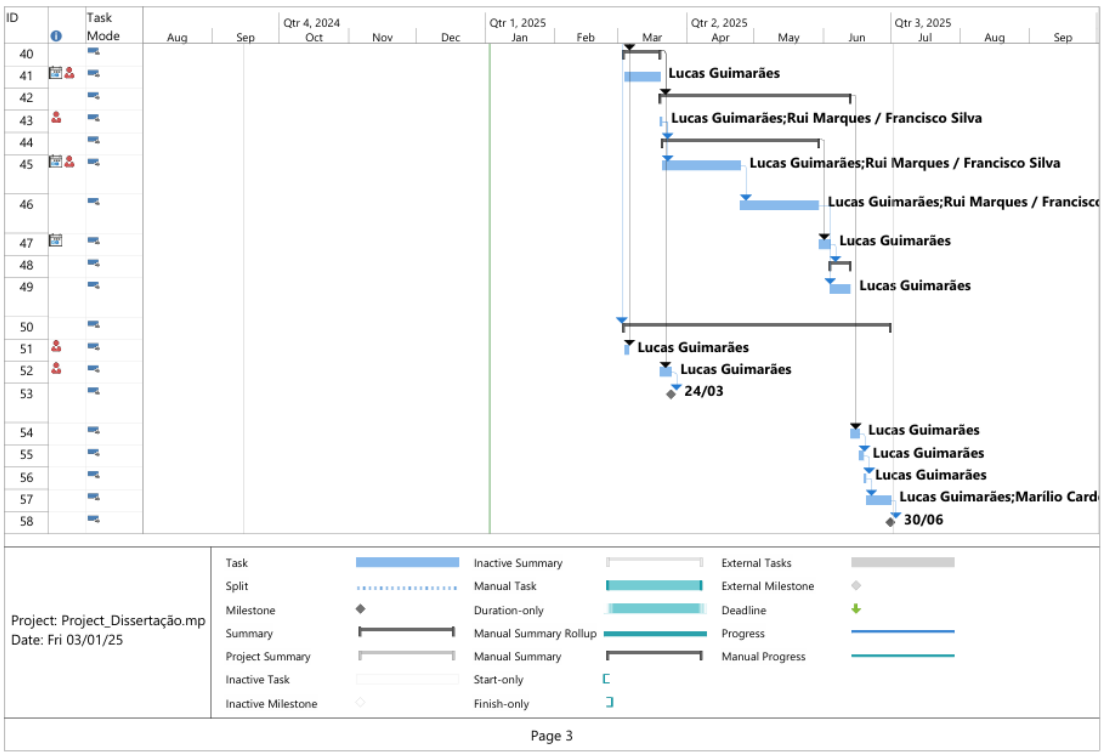


Figura 26 - Diagrama de Gantt parte 3

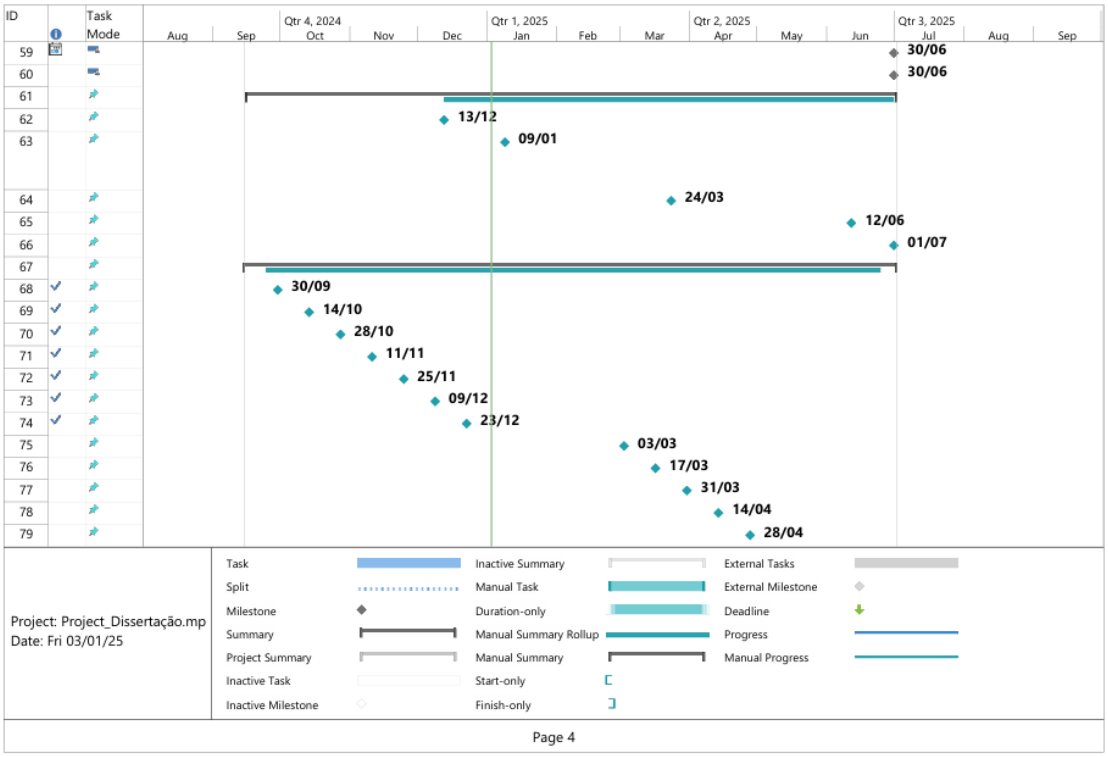


Figura 27 - Diagrama de Gantt parte 4

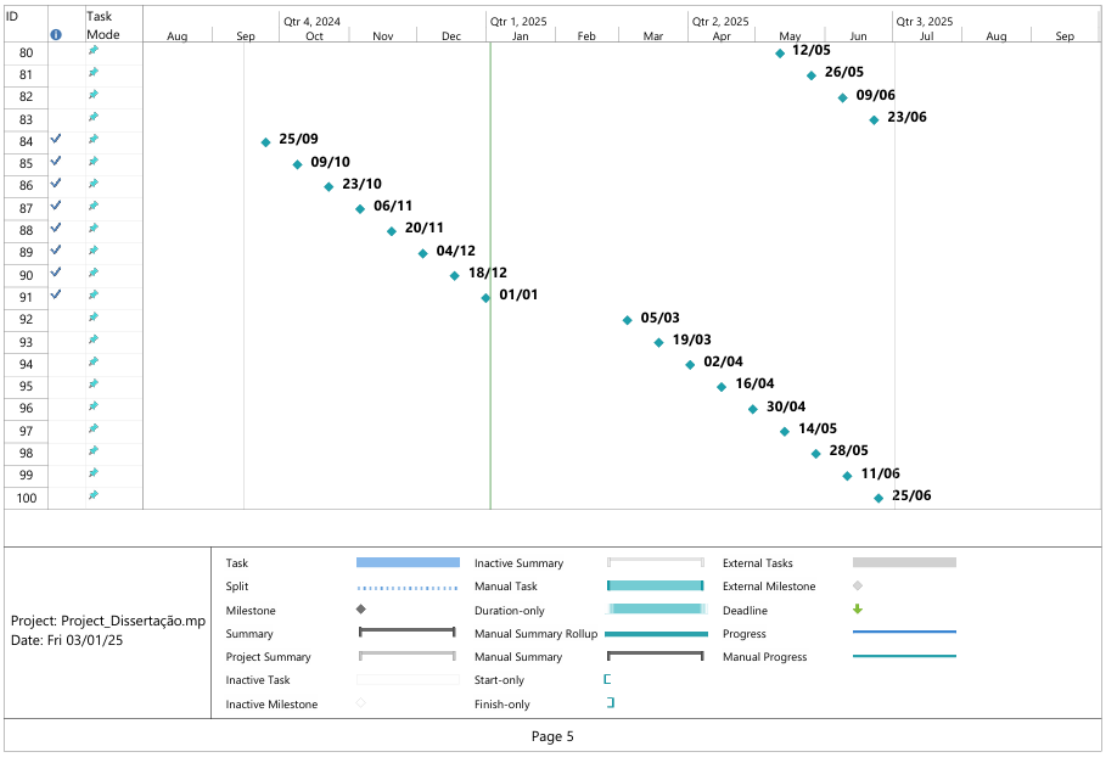


Figura 28 - Diagrama de Gantt parte 5

# Anexo C – Gráfico de Gantt Inputs

Anexo 3 - Gráfico de Gantt Inputs

Tabela 7 - Tabela de inputs (parte 1)

| Task | WBS | Nível WBS | Task Name | Duration | Start | Finish | Predecessors | Resource Names | % Complete | Add New Column |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **Projeto** | **Projeto Dissertação** | **208 days?** | **Mon 16/09/24** | **Tue 01/07/25** | | | **54%** | |
| ✓ | **1.1** | **Projeto** | **Preparação Dissertação** | **81 days** | **Mon 16/09/24** | **Sat 04/01/25** | | | **100%** | |
| ✓ | **1.1.1** | **Fase** | **Atribuição de Projeto** | **10 days** | **Mon 16/09/24** | **Fri 27/09/24** | | | **100%** | |
| ✓ | 1.1.1.1 | Entregável | Escolha da Proposta | 3 days | Mon 16/09/24 | Wed 18/09/24 | | Lucas Guimarães | 100% | |
| ✓ | 1.1.1.2 | Entregável | Formalização | 4 days | Thu 19/09/24 | Tue 24/09/24 | 4 | Lucas Guimarães | 100% | |
| ✓ | 1.1.1.3 | Entregável | Validação | 2 days | Wed 25/09/24 | Thu 26/09/24 | 5 | Marílio Cardoso | 100% | |
| ✓ | 1.1.1.4 | Entregável | Revisão e Aprovação | 1 day | Fri 27/09/24 | Fri 27/09/24 | 6 | Isabel Figueiredo | 100% | |
| ✓ | **1.1.2** | **Fase** | **Planeamento** | **30 days** | **Mon 28/10/24** | **Fri 06/12/24** | **3** | | **100%** | |
| ✓ | 1.1.2.1 | Entregável | Project Charter | 5 days | Mon 28/10/24 | Fri 01/11/24 | | Lucas Guimarães | 100% | |
| ✓ | 1.1.2.2 | Entregável | WBS | 5 days | Mon 04/11/24 | Fri 08/11/24 | 9 | Lucas Guimarães;M | 100% | |
| ✓ | 1.1.2.3 | Entregável | Plano | 15 days | Mon 11/11/24 | Fri 29/11/24 | 10 | Lucas Guimarães;M | 100% | |
| ✓ | 1.1.2.4 | Entregável | Matriz de riscos | 5 days | Mon 02/12/24 | Fri 06/12/24 | 11 | Lucas Guimarães | 100% | |
| ✓ | 1.1.3 | Tarefa | Exploração do tema da prop | 4 days | Mon 30/09/24 | Thu 03/10/24 | 3 | Lucas Guimarães | 100% | |
| ✓ | **1.1.4** | **Fase** | **Escrita do Relatório** | **67 days** | **Fri 04/10/24** | **Sat 04/01/25** | **13** | | **100%** | |
| ✓ | 1.1.4.1 | Entregável | Formatações iniciais | 1 day | Fri 04/10/24 | Fri 04/10/24 | | Lucas Guimarães | 100% | |
| ✓ | 1.1.4.2 | Entregável | Resumo | 1 day | Wed 25/12/24 | Wed 25/12/24 | 18 | Lucas Guimarães | 100% | |
| ✓ | 1.1.4.3 | Entregável | Abstract | 1 day | Thu 26/12/24 | Thu 26/12/24 | 16 | Lucas Guimarães | 100% | |
| ✓ | **1.1.4.4** | **Entregável** | **Introdução** | **12 days** | **Mon 09/12/24** | **Tue 24/12/24** | **25** | | **100%** | |
| ✓ | 1.1.4.4.1 | Entregável | Contexto | 2 days | Mon 09/12/24 | Tue 10/12/24 | | Lucas Guimarães | 100% | |
| ✓ | 1.1.4.4.2 | Entregável | Problema | 2 days | Wed 11/12/24 | Thu 12/12/24 | 19 | Lucas Guimarães | 100% | |
| ✓ | 1.1.4.4.3 | Entregável | Motivação e objetivos | 2 days | Fri 13/12/24 | Mon 16/12/24 | 20 | Lucas Guimarães | 100% | |
| ✓ | 1.1.4.4.4 | Entregável | Considerações éticas | 2 days | Tue 17/12/24 | Wed 18/12/24 | 21 | Lucas Guimarães | 100% | |
| ✓ | 1.1.4.4.5 | Entregável | Metedologia | 2 days | Thu 19/12/24 | Fri 20/12/24 | 22 | Lucas Guimarães | 100% | |
| ✓ | 1.1.4.4.6 | Entregável | Estrutura do relatório | 2 days | Mon 23/12/24 | Tue 24/12/24 | 23 | Lucas Guimarães | 100% | |

Tabela 8 - Tabela de inputs (parte 2)

| Task | WBS | Nível WBS | Task Name | Duration | Start | Finish | Predecessors | Resource Names | % Complete | Add New Column |
|---|---|---|---|---|---|---|---|---|---|---|
| **1.1.4.5** | **Entregável** | **Estado da Arte** | **45 days** | **Mon 07/10/24** | **Fri 06/12/24** | **15** | | **100%** | |
| 1.1.4.5.1 | Entregável | Personalização em e-co | 9 days | Mon 07/10/24 | Thu 17/10/24 | 15 | Lucas Guimarães | 100% | |
| 1.1.4.5.2 | Entregável | Segmentação de utilizad | 9 days | Fri 18/10/24 | Wed 30/10/24 | 26 | Lucas Guimarães | 100% | |
| 1.1.4.5.3 | Entregável | Categorização de produ | 9 days | Thu 31/10/24 | Tue 12/11/24 | 27 | Lucas Guimarães | 100% | |
| 1.1.4.5.4 | Entregável | Tecnologias existentes / | 9 days | Wed 13/11/24 | Mon 25/11/24 | 28 | Lucas Guimarães | 100% | |
| 1.1.4.5.5 | Entregável | Aplicações semelhantes | 9 days | Tue 26/11/24 | Fri 06/12/24 | 29 | Lucas Guimarães | 100% | |
| 1.1.4.6 | Entregável | Referências | 45 days | Mon 07/10/24 | Fri 06/12/24 | 15 | Lucas Guimarães | 100% | |
| 1.1.4.7 | Entregável | Revisões | 7 days | Fri 27/12/24 | Sat 04/01/25 | 17 | Marílio Cardoso;Lu | 100% | |
| **1.1.5** | **Fase** | **Conclusão Preparação para** | **0 days** | **Sat 04/01/25** | **Sat 04/01/25** | **32** | | **100%** | |
| 1.1.5.1 | Entregável | Entrega do relatório | 0 days | Sat 04/01/25 | Sat 04/01/25 | | Lucas Guimarães | 100% | |
| 1.2 | | Época de exames | 36 days | Mon 06/01/25 | Mon 24/02/25 | 2 | | 0% | |
| **1.3** | **Projeto** | **Dissertação** | **90 days** | **Tue 25/02/25** | **Mon 30/06/25** | **35** | | **0%** | |
| **1.3.1** | **Fase** | **Análise** | **5 days** | **Tue 25/02/25** | **Mon 03/03/25** | | | **0%** | |
| 1.3.1.1 | Tarefa | Domínio do problema | 3 days | Tue 25/02/25 | Thu 27/02/25 | | Lucas Guimarães | 0% | |
| 1.3.1.2 | Tarefa | Requisitos funcionais e nã | 2 days | Fri 28/02/25 | Mon 03/03/25 | 38 | Lucas Guimarães;F | 0% | |
| **1.3.2** | **Fase** | **Design** | **12 days** | **Tue 04/03/25** | **Wed 19/03/25** | **37** | | **0%** | |
| 1.3.2.1 | Tarefa | Arquitetura | 12 days | Tue 04/03/25 | Wed 19/03/25 | | Lucas Guimarães | 0% | |
| **1.3.3** | **Tarefa** | **Implementação** | **61 days** | **Thu 20/03/25** | **Thu 12/06/25** | **40** | | **0%** | |
| 1.3.3.1 | Tarefa | Requisitos do sistema | 1 day | Thu 20/03/25 | Thu 20/03/25 | | Lucas Guimarães;F | 0% | |
| **1.3.3.2** | **Tarefa** | **Backend** | **50 days** | **Fri 21/03/25** | **Thu 29/05/25** | **43** | | **0%** | |
| 1.3.3.2.1 | Tarefa | Implementação do algo | 25 days | Fri 21/03/25 | Thu 24/04/25 | 43 | Lucas Guimarães;F | 0% | |
| 1.3.3.2.2 | Tarefa | Avaliação da solução e t | 25 days | Fri 25/04/25 | Thu 29/05/25 | 45 | Lucas Guimarães;F | 0% | |
| 1.3.3.3 | Tarefa | Base de Dados | 3 days | Fri 30/05/25 | Tue 03/06/25 | 44 | Lucas Guimarães | 0% | |

Tabela 9 - Tabela de inputs (parte 3)

| Task | WBS | Nível WBS | Task Name | Duration | Start | Finish | Predecessors | Resource Names | % Complete | Add New Column |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1.3.3.4** | **Tarefa** | **Frontend** | **7 days** | **Wed 04/06/25** | **Thu 12/06/25** | **47** | | **0%** | |
| | 1.3.3.4.1 | Tarefa | Visualização de resultad | 7 days | Wed 04/06/25 | Thu 12/06/25 | 46 | Lucas Guimarães | 0% | |
| | **1.3.4** | **Fase** | **Escrita do relatório** | **85 days** | **Tue 04/03/25** | **Mon 30/06/25** | **35** | | **0%** | |
| | 1.3.4.1 | Entregável | Análise | 2 days | Tue 04/03/25 | Wed 05/03/25 | 37 | Lucas Guimarães | 0% | |
| | 1.3.4.2 | Entregável | Design | 3 days | Thu 20/03/25 | Mon 24/03/25 | 40 | Lucas Guimarães | 0% | |
| | 1.3.4.3 | Entregável | Entrega de versão parcial ( | 0 days | Mon 24/03/25 | Mon 24/03/25 | 52 | Lucas Guimarães;M | 0% | |
| | 1.3.4.4 | Entregável | Implementação | 2 days | Fri 13/06/25 | Mon 16/06/25 | 42 | Lucas Guimarães | 0% | |
| | 1.3.4.5 | Entregável | Avaliação da Solução | 2 days | Tue 17/06/25 | Wed 18/06/25 | 54 | Lucas Guimarães | 0% | |
| | 1.3.4.6 | Entregável | Conclusão | 1 day | Thu 19/06/25 | Thu 19/06/25 | 55 | Lucas Guimarães | 0% | |
| | 1.3.4.7 | Entregável | Revisão | 7 days | Fri 20/06/25 | Mon 30/06/25 | 56 | Lucas Guimarães;M | 0% | |
| | **1.3.5** | **Fase** | **Conclusão Dissertação** | **0 days** | **Mon 30/06/25** | **Mon 30/06/25** | **57** | | **0%** | |
| | 1.3.5.1 | Entregável | Entrega do relatório | 0 days | Mon 30/06/25 | Mon 30/06/25 | | Lucas Guimarães | 0% | |
| | 1.3.5.2 | Tarefa | Apresentação final | 0 days | Mon 30/06/25 | Mon 30/06/25 | | Lucas Guimarães;M | 0% | |
| | **2** | **Milestone** | **Milestones** | **208 days** | **Mon 16/09/24** | **Tue 01/07/25** | | | **0%** | |
| | 2.1 | Milestone | Planeamento aprovado | 0 days | Fri 13/12/24 | Fri 13/12/24 | | Lucas Guimarães;M | 0% | |
| | 2.2 | Milestone | Apresentação da Preparação d | 0 days | Thu 09/01/25 | Thu 09/01/25 | | Lucas Guimarães | 0% | |
| | 2.3 | Milestone | Arquitetura Concluida | 0 days | Mon 24/03/25 | Mon 24/03/25 | | Lucas Guimarães | 0% | |
| | 2.4 | Milestone | Aplicação concluída | 0 days | Thu 12/06/25 | Thu 12/06/25 | | Lucas Guimarães | 0% | |
| | 2.5 | Milestone | Apresentação Dissertação | 0 days | Tue 01/07/25 | Tue 01/07/25 | | Lucas Guimarães | 0% | |
| | **3** | **Reunião** | **Reuniões** | **209 days** | **Sun 15/09/24** | **Tue 01/07/25** | | | **0%** | |
| | 3.1 | Reunião | Reunião Orientador | 0 days | Mon 30/09/24 | Mon 30/09/24 | | Lucas Guimarães;M | 100% | |
| | 3.2 | Reunião | Reunião Orientador | 0 days | Mon 14/10/24 | Mon 14/10/24 | | Lucas Guimarães;M | 100% | |
| | 3.3 | Reunião | Reunião Orientador | 0 days | Mon 28/10/24 | Mon 28/10/24 | | Lucas Guimarães;M | 100% | |
| | 3.4 | Reunião | Reunião Orientador | 0 days | Mon 11/11/24 | Mon 11/11/24 | | Lucas Guimarães;M | 100% | |

Tabela 10 - Tabela de inputs (parte 4)

| | Task | WBS | Nível WBS | Task Name | Duration | Start | Finish | Predecessors | Resource Names | % Complete | Add New Column |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 3.10 | Reunião | Reunião Orientador | 0 days | Mon 31/03/25 | Mon 31/03/25 | | Lucas Guimarães;N | 0% | |
| | | 3.11 | Reunião | Reunião Orientador | 0 days | Mon 14/04/25 | Mon 14/04/25 | | Lucas Guimarães;N | 0% | |
| | | 3.12 | Reunião | Reunião Orientador | 0 days | Mon 28/04/25 | Mon 28/04/25 | | Lucas Guimarães;N | 0% | |
| | | 3.13 | Reunião | Reunião Orientador | 0 days | Mon 12/05/25 | Mon 12/05/25 | | Lucas Guimarães;N | 0% | |
| | | 3.14 | Reunião | Reunião Orientador | 0 days | Mon 26/05/25 | Mon 26/05/25 | | Lucas Guimarães;N | 0% | |
| | | 3.15 | Reunião | Reunião Orientador | 0 days | Mon 09/06/25 | Mon 09/06/25 | | Lucas Guimarães;N | 0% | |
| | | 3.16 | Reunião | Reunião Orientador | 0 days | Mon 23/06/25 | Mon 23/06/25 | | Lucas Guimarães;N | 0% | |
| ✓ | | 3.17 | Reunião | Reunião Supervisor | 0 days | Wed 25/09/24 | Wed 25/09/24 | | Lucas Guimarães;F | 100% | |
| ✓ | | 3.18 | Reunião | Reunião Supervisor | 0 days | Wed 09/10/24 | Wed 09/10/24 | | Lucas Guimarães;F | 100% | |
| ✓ | | 3.19 | Reunião | Reunião Supervisor | 0 days | Wed 23/10/24 | Wed 23/10/24 | | Lucas Guimarães;F | 100% | |
| ✓ | | 3.20 | Reunião | Reunião Supervisor | 0 days | Wed 06/11/24 | Wed 06/11/24 | | Lucas Guimarães;F | 100% | |
| ✓ | | 3.21 | Reunião | Reunião Supervisor | 0 days | Wed 20/11/24 | Wed 20/11/24 | | Lucas Guimarães;F | 100% | |
| ✓ | | 3.22 | Reunião | Reunião Supervisor | 0 days | Wed 04/12/24 | Wed 04/12/24 | | Lucas Guimarães;F | 100% | |
| ✓ | | 3.23 | Reunião | Reunião Supervisor | 0 days | Wed 18/12/24 | Wed 18/12/24 | | Lucas Guimarães;F | 100% | |
| ✓ | | 3.24 | Reunião | Reunião Supervisor | 0 days | Wed 01/01/25 | Wed 01/01/25 | | Lucas Guimarães;F | 100% | |
| | | 3.25 | Reunião | Reunião Supervisor | 0 days | Wed 05/03/25 | Wed 05/03/25 | | Lucas Guimarães;F | 0% | |
| | | 3.26 | Reunião | Reunião Supervisor | 0 days | Wed 19/03/25 | Wed 19/03/25 | | Lucas Guimarães;F | 0% | |
| | | 3.27 | Reunião | Reunião Supervisor | 0 days | Wed 02/04/25 | Wed 02/04/25 | | Lucas Guimarães;F | 0% | |
| | | 3.28 | Reunião | Reunião Supervisor | 0 days | Wed 16/04/25 | Wed 16/04/25 | | Lucas Guimarães;F | 0% | |
| | | 3.29 | Reunião | Reunião Supervisor | 0 days | Wed 30/04/25 | Wed 30/04/25 | | Lucas Guimarães;F | 0% | |
| | | 3.30 | Reunião | Reunião Supervisor | 0 days | Wed 14/05/25 | Wed 14/05/25 | | Lucas Guimarães;F | 0% | |
| | | 3.31 | Reunião | Reunião Supervisor | 0 days | Wed 28/05/25 | Wed 28/05/25 | | Lucas Guimarães;F | 0% | |
| | | 3.32 | Reunião | Reunião Supervisor | 0 days | Wed 11/06/25 | Wed 11/06/25 | | Lucas Guimarães;F | 0% | |
| | | 3.33 | Reunião | Reunião Supervisor | 0 days | Wed 25/06/25 | Wed 25/06/25 | | Lucas Guimarães;F | 0% | |

# Anexo D – Matriz de riscos

Anexo 4 - Matriz de riscos

Tabela 11 - Matriz de riscos

| Description | Cause | Effect | Risk Owner | Probability (1-5) | Impact (1-5) | PI Score | Expected Result, No Action | Risk Response Type | Response description |
|---|---|---|---|---|---|---|---|---|---|
| Description of the risk | Cause of the risk | Effect on the project | Name of person who monitors the risk | Group sourced rough estimate of how likely this is to occur | Rough estimate of how significant the impact of this risk | Probability multiplied by Impact | What will happen if the risk becomes an issue and no action is taken | Decision made by group on how to respond to this risk (see above in blue) | How do you know it is time to put the response into play |
| A empresa não entrega os dados atempadamente | Problemas internos | Atrasos no desenvolvimento | Lucas Guimarães | 3 | 4 | 12 | O cronograma do projeto será comprometido e os marcos podem não ser atingidos | Mitigar | Na segunda reunião com a empresa após o começo do projeto |
| Acesso limitado a recursos técnicos para o treinamento do modelo | Restrições orçamentais | Soluções com menor qualidade | Lucas Guimarães | 3 | 3 | 9 | O projeto pode entregar um modelo com desempenho inferior | Aceitar | |
| Desalinhamento entre os objetivos do projeto e as expectativas da empresa | Falta de comunicação | Atrasos e desperdício de recursos | Lucas Guimarães | 1 | 5 | 5 | Tempo e esforço significativos serão desperdiçados revisando o escopo e os entregáveis do projeto | Mitigate | Mal seja detetado |