



Sistema de recomendação para uma plataforma de comércio eletrónico

ANTÓNIO JOSÉ BRAGA MOREIRA

Julho de 2019

Sistema de recomendação para uma plataforma de comércio eletrónico

António José Braga Moreira

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

Orientador: Nuno Filipe Malheiro

Coorientador: Paulo Costa

Resumo

Atualmente, devido ao desenvolvimento das tecnologias de informação e computação e consequente melhoria do acesso à internet, verifica-se uma maior utilização das plataformas de comércio eletrónico para a compra de produtos.

Contudo, a forte adesão das pessoas a este tipo de plataformas originou dificuldades para as empresas que criam estas plataformas. Um dos principais problemas é a grande quantidade de produtos disponíveis o que dificulta os utilizadores encontrarem os produtos que pretendem. No caso da *Netflix*, por exemplo, “caso um utilizador, não encontre algo que lhe interesse após um período de 60 a 90 segundos de navegação, a probabilidade de o mesmo não aderir ao serviço aumenta consideravelmente”.

De forma a colmatar esse problema, foram criados os sistemas de recomendação, que visam ajudar os utilizadores no processo de compra, através da sugestão de produtos com base em diversos fatores.

Este documento analisa algumas das arquiteturas utilizadas pelas plataformas de comércio eletrónico, diferentes técnicas de recomendação e os problemas associados a cada uma delas, diferentes métodos e métricas de avaliação de sistemas de recomendação. Além disso, apresenta alguns exemplos de sistemas de recomendação.

Esta dissertação culminou na proposta de um sistema de recomendação para uma plataforma de comércio eletrónico. Os resultados obtidos foram positivos uma vez que o sistema criado passou nos testes de integração, os três algoritmos implementados obtiveram uma taxa de sucesso superior à da *baseline*¹, tendo os testes de carga realizados à API sido melhores do que o esperado.

Palavras-chave: sistema de recomendação, comércio eletrónico, regras associativas, filtragem colaborativa

¹ Baseline: valor mínimo utilizado para avaliar a qualidade do sistema de recomendação

Abstract

Nowadays, due to the development of technology and internet access there is a greater use of e-commerce platforms for the purchase of products.

The increased use of these platforms created several management challenges for these companies. One of the main problems is the large number of products available which makes it difficult for users to find the products they want. In the case of Netflix, for example, "if a user does not find something that appeals to him after a period of 60 to 90 seconds of browsing, the likelihood that he will not join the service increases considerably."

In order to overcome this problem, recommendation systems have been created, to help users in the purchase process, by suggesting products based on several factors.

This document analyses some of the architectures used by e-commerce platforms, the different recommendation techniques, problems associated with each of them, different methodologies and metrics for evaluating recommendation systems. In addition, it presents some examples of recommendation systems.

This dissertation led to the proposal of a recommendation system for an electronic commerce platform. The obtained results were positive given the success of the integration tests; the three implemented algorithms had a higher success rate than the baseline²; and the performed API load tests were better than expected.

Keywords: recommendation system, e-commerce, association rules, collaborative filtering

² Baseline: a minimum value used for evaluating the quality of the recommendation system.

Agradecimentos

Quero agradecer todas as pessoas que me ajudaram e apoiaram, ao longo do meu percurso académico.

Ao professor Nuno Malheiro e ao Eng.º Paulo Costa pela orientação e constante disponibilidade ao longo deste projeto.

À minha família, particularmente a minha mãe, pelo apoio ao longo do percurso que culminou na realização deste relatório.

Índice

1	Introdução	15
1.1	Âmbito	15
1.2	Contexto	15
1.2.1	GNG	16
1.3	Objetivos	17
1.4	Estrutura do relatório	17
2	Análise de valor	19
2.1	Análise de valor	19
2.1.1	Identificação e análise da oportunidade	20
2.1.2	Geração da ideia, seleção da ideia e definição do conceito	20
2.2	Identificação e análise funcional	21
2.2.1	Proposta de valor	24
2.2.2	Valor para o cliente	24
2.3	Análise e avaliação de alternativas	25
2.3.1	Análise de frameworks/bibliotecas	25
2.3.2	Seleção da framework/bibliotecas	27
3	Estado de Arte	29
3.1	Comércio eletrónico	29
3.1.1	Arquitetura monolítica	30
3.1.2	Arquitetura orientada a microsserviços	31
3.1.3	Resumo das arquiteturas	32
3.2	Sistemas de recomendação	32
3.3	Recolha de Feedback	33
3.4	Técnicas de sistemas recomendação	34
3.4.1	Baseado em conteúdos	35
3.4.2	Filtragem colaborativa	37
3.4.3	Híbrida	41
3.5	Limitações e vantagens dos sistemas de recomendação	42
3.5.1	Limitações comuns às diferentes técnicas	42
3.5.2	Limitações e Vantagens da técnica baseada em conteúdos	42
3.5.3	Limitações e Vantagens da técnica filtragem colaborativa	43
3.5.4	Limitações e Vantagens da técnica híbrida	43
3.6	Medidas de similaridade	43
3.7	Métodos de avaliação	45
3.7.1	Avaliação offline	45
3.7.2	Avaliação online	46
3.8	Avaliar a qualidade de sistema de recomendação	46

3.9	Aplicações	48
3.9.1	Amazon.....	48
3.9.2	Netflix.....	48
3.9.3	Youtube	51
3.10	Frameworks	52
4	Design.....	53
4.1	Análise de dados	53
4.2	Proposta de solução	54
4.2.1	Regras associativas	54
4.2.2	Baseada em caraterísticas.....	55
4.2.3	Histórico de compras.....	55
4.3	Requisitos funcionais e não funcionais.....	55
4.4	Propostas de Arquitetura	56
4.4.1	Arquitetura nº 1.....	56
4.4.2	Arquitetura nº 2.....	57
4.4.3	Seleção da arquitetura.....	58
4.5	Base de dados	58
4.5.1	Microsoft SQL Server	59
4.5.2	MongoDB	59
4.5.3	Seleção da base de dados.....	59
4.6	Conceitos do domínio	59
4.6.1	Modelo de dados	60
4.6.2	Recomendações	61
4.7	Processos de recomendação.....	62
4.7.1	Processo de importação de encomendas	62
4.7.2	Processo de recomendação para um produto	63
5	Implementação	65
5.1	Azure DevOps.....	65
5.2	Componentes implementados.....	66
5.2.1	Base de dados	66
5.2.2	Motor de recomendação	67
5.2.3	API de Recomendação.....	69
6	Avaliação.....	71
6.1	Definição da metodologia de avaliação.....	71
6.2	Definição das hipóteses.....	72
6.3	Avaliação dos resultados.....	73
6.3.1	Avaliação processo de integração de produtos	73
6.3.2	Avaliação processo de integração de compras.....	74
6.3.3	Avaliação processo de integração de stock.....	75
6.3.4	Avaliação performance da API Recomendação.....	76
6.3.5	Avaliação da abordagem baseada em caraterísticas	77

6.3.6	Avaliação da abordagem regras associativas.....	78
6.3.7	Avaliação da abordagem com base no histórico de compras	80
7	Conclusão	83
7.1	Apreciação final	83
7.2	Trabalho futuro e limitações.....	84
8	Anexos	90

Lista de Figuras

Figura 1- Previsão de vendas e-commerce	20
Figura 2- Funções do sistema de recomendação	22
Figura 3- Diagrama FAST	23
Figura 4- Impacto do Sistema de recomendação	24
Figura 5- Árvore de decisão	27
Figura 6- Arquitetura monolítica	30
Figura 7- Arquitetura orientada a microsserviços	32
Figura 8- Tipos de sistemas de recomendação	35
Figura 9 -Arquitetura de um sistema baseado em conteúdos	36
Figura 10- Baseado no utilizador	38
Figura 11- Baseado em produtos	39
Figura 12- Página inicial da <i>Netflix</i>	49
Figura 13 – Because you watched “Orange is the New Black”	51
Figura 14- Visão geral da arquitetura do sistema de recomendação do youtube.	52
Figura 15- Proposta de arquitetura nº 1	57
Figura 16- Proposta de arquitetura nº 2	58
Figura 17- Diagrama da base de dados	61
Figura 18- Diagrama de classes para a geração de recomendações	62
Figura 19- Diagrama de sequência da importação de encomendas	63
Figura 20- Diagrama de sequência da recomendação de um produto	64
Figura 21- Azure Boards	66
Figura 22 – Diagrama de classes do agendamento dos processos	68
Figura 23- Interface do motor de recomendação	69
Figura 24- Documentação gerada pelo <i>Swagger</i>	70
Figura 25 – Resultados da variação do suporte	79
Figura 26 - Resultados da variação da confiança	80

Lista de Tabelas

Tabela 1- Comparação das funcionalidades	23
Tabela 2- Benefícios e Sacrifícios do sistema de recomendação.....	25
Tabela 3- Matriz de comparação	28
Tabela 4- Matriz normalizada	28
Tabela 5- Escolha de alternativa	28
Tabela 6- Comparação das arquiteturas.....	32
Tabela 7- Características do <i>feedback</i> Implícito e Explícito.....	34
Tabela 8 - Classificações dos utilizadores mais semelhantes à Rita	39
Tabela 9- Requisitos funcionais e não funcionais	56
Tabela 10- Resultado do caso de teste “Introduzir uma data com produtos para importar” ...	73
Tabela 11- Resultado do caso de teste “Introduzir uma data sem dados para importar”	73
Tabela 12- Resultado do caso de teste “Falha na ligação à base de dados no processo de importação de produtos”	74
Tabela 13- Resultado do caso de teste “Introduzir uma data com compras para importar”	74
Tabela 14- Resultado do caso de teste “Introduzir uma data sem compras para importar”	74
Tabela 15- Resultado do caso de teste “Falha na ligação à base de dados no processo de importação de compras”	75
Tabela 16 - Resultado do caso de teste “Importar o stock”	75
Tabela 17 - Resultado do caso de teste “Falha na ligação à base de dados no processo de importação de stock”	76
Tabela 18- Resultados da avaliação da performance da API	76
Tabela 19 - Resultados das abordagens BC e baseline para cinco produtos.....	77
Tabela 20 - Resultados das abordagens BC e baseline para dez produtos.....	78
Tabela 21- Comparação dos resultados da abordagem RA e a <i>baseline</i>	80
Tabela 22 – Comparação dos resultados da abordagem com base no histórico de compras e a <i>baseline</i>	81
Tabela 23- Conclusão dos objetivos propostos	83

Acrónimos e Símbolos

Lista de Acrónimos

AHP	Analytic Hierarchy Process
BC	Baseado em características
FAST	Function Analysis System Technique
HTTP	Hypertext Transfer Protocol
MAE	Erro médio absoluto
NCD	New Concept Development
PVR	Personalized Video Ranker
RA	Regras associativas
RC	Rácio de consistência
RMSE	Raiz quadrada do erro quadrático médio
TF-IDF	Frequency/inverse document frequency

1 Introdução

Neste capítulo é realizada a introdução ao tema desta dissertação. Inicialmente é descrito o âmbito desta dissertação, de seguida é apresentado o contexto do problema que se pretende resolver, assim como os objetivos atingir, por fim é descrita a estrutura deste relatório.

1.1 Âmbito

Com vista à aquisição do grau de mestre em Engenharia Software foi elaborada a presente dissertação, que visa desenvolver um sistema capaz de gerar recomendações de peças de vestuário. Esta dissertação foi desenvolvida no âmbito do Mestrado de Engenharia Informática, a decorrer no Instituto Superior de Engenharia do Porto, no ano letivo 2018/2019 e em colaboração com a GNG.

1.2 Contexto

Com a disseminação da internet a produção de dados aumentou consideravelmente, dificultando o acesso à informação que realmente queremos. Como consequência, começaram a ser criados sistemas capazes de filtrar dados com base nas preferências dos utilizadores, ajudando-os desta forma, a lidar com o excesso de dados (Konstan and Riedl, 2012).

Um sistema de recomendação é capaz de filtrar, para um determinado contexto, a informação mais importante de acordo com um conjunto de preferências (Konstan and Riedl, 2012). Estes sistemas foram criados para facilitar o processo de escolha dos utilizadores, com base em experiências adquiridas no passado, fornecendo alternativas quando não existe informação suficiente para tomar uma decisão (Isinkaye, Folajimi and Ojokoh, 2015).

Como estes sistemas tem a capacidade de prever a utilidade de um determinado produto ou de produzir uma lista com os N produtos mais interessantes para o utilizador, fazem-nos

perfeitos para as plataformas de comércio eletrónico. Estas plataformas possuem uma grande variedade de produtos, tornando-se complicado para o comprador encontrar o produto pretendido. Desta forma, a introdução destes sistemas neste contexto facilita o processo de procura de produtos para o utilizador. Além disso, podem ser utilizados para promover vendas cruzadas, isto é, na compra de um produto recomenda outro produto que combine com o produto a ser adquirido. De acordo com *Giang* em (*Giang, 2019*), a personalização, através de sistemas de recomendação é bastante importante para os utilizadores, pois estes preferem plataformas com sistema de recomendações. Um exemplo disso é a *Netflix*, que de acordo com (*Gomez-Uribe and Hunt, 2015*) “caso um utilizador do *Netflix*, não encontre algo que lhe interesse após um período de 60 a 90 segundos de navegação, a probabilidade de o mesmo não aderir ao serviço aumenta consideravelmente”.

Ora isto tem um impacto direto no volume de vendas, pois ao simplificar o acesso ao produto pretendido, será possível a venda de mais produtos. A utilização destes sistemas cria *win-win scenario*, isto é, beneficia tanto o utilizador como as plataformas de comércio eletrónico (*Sarwar et al., 2000; Pu and Chen, 2010*).

Devido ao facto de estes sistemas desempenharem um papel essencial neste tipo de plataformas, levou a que empresas como *Amazon* e a *Netflix*, investissem no desenvolvimento dos mesmos. Prova disso, foi a competição criada pela *Netflix*, “*The Netflix Price*” em 2006. Esta tinha como objetivo, o desenvolvimento de um sistema de recomendação que conseguisse superar, o *Cinematch*, em 10% nos testes *offline*. Os vencedores da competição, tinham de publicar e explicar os benefícios da sua abordagem e recebiam um prémio de um milhão de dólares (*Bennett and Lanning, 2007*).

Devido ao crescimento das plataformas de comércio eletrónico e aos claros benefícios dos sistemas de recomendação, é imperativo a sua integração nestas plataformas, para que se possa criar uma experiência personalizada para o cliente (*Giang, 2019; Statista, 2019a*).

Em congruência com o que acima foi referido, o sistema de recomendação a ser desenvolvido será integrado na plataforma de comércio eletrónico da GNG, mais concretamente em *levi.pt*, com o propósito de proporcionar ao cliente um serviço personalizado na procura do produto ideal.

1.2.1 GNG

A principal atividade da GNG é o comércio de retalho de vestuário para adultos em estabelecimentos especializados. A GNG teve origem em 2010, depois da cisão de uma outra sociedade do mesmo ramo. Atualmente, a GNG gere 51 estabelecimentos das marcas *Levi Strauss® (Levi's®)*, *G-Star RAW®*, *Dockers®* e *Adidas®* em regime de franchising, em Portugal e Espanha. Em 2015, lançou a aplicação GNG para dispositivos móveis, que permite aos seus utilizadores usufruir de descontos e de campanhas promocionais nos diferentes estabelecimentos da empresa. No ano seguinte, em 2016, lançou a sua plataforma de comércio eletrónico, onde vende os produtos das marcas *Dockers®* e *G-Star RAW®*, para

Portugal e Espanha. Em 2018, começou a gerir a plataforma de comércio eletrónico da *Levi's*[®], em levi.pt.

1.3 Objetivos

De forma atingir a finalidade acima proposta foram estabelecidos alguns objetivos para a dissertação:

- Investigação do estado da arte de sistemas de recomendação
- Desenvolvimento de um sistema de importação de dados
- Desenvolvimento de uma API capaz de fornecer recomendações
- Avaliação do sistema de recomendação

1.4 Estrutura do relatório

Este documento é constituído por sete capítulos.

No primeiro capítulo é realizada uma breve apresentação desta dissertação, onde é descrito o seu âmbito, todo o contexto envolvente, incluindo a empresa onde foi realizada, objetivos e a estrutura deste documento.

No segundo capítulo é realizada a análise de valor da solução, abordando o tema de sistemas de recomendação numa perspetiva de negócio, apresentando dados importantes sobre as plataformas de comércio eletrónico e o impacto de um sistema de recomendação nas mesmas.

No terceiro capítulo é descrito o contexto teórico da presente dissertação. Inicialmente são descritas algumas arquiteturas utilizadas no desenvolvimento de plataformas de comércio eletrónico. De seguida, é realizada a introdução a sistemas de recomendação, descrevendo os vários métodos de recolha de *feedback*, técnicas de recomendação, metodologias e métricas para a sua de avaliação, bem como exemplos de sistemas de recomendação.

No quarto capítulo é descrito o *design* da solução a implementar. Serão apresentados os dados disponíveis para o desenvolvimento do sistema de recomendação. De seguida, serão descritas as técnicas que serão utilizadas para o desenvolvimento do sistema de recomendação, bem como os requisitos funcionais e não funcionais do mesmo. Além disso, serão descritas duas alternativas para a arquitetura deste sistema; ao longo do mesmo capítulo, iremos selecionar apenas uma, processo esse que será justificado. O capítulo culmina com uma breve explicação sobre a base de dados a utilizar.

No quinto capítulo é descrito o processo utilizado para o desenvolvimento do projeto, bem como as plataformas utilizadas. Além disso, é realizada uma breve descrição dos componentes desenvolvidos no âmbito deste projeto.

No sexto capítulo é descrita a metodologia de avaliação utilizada para avaliar a solução, bem como a formulação das hipóteses que se pretende comprovar; seguindo-se da interpretação dos resultados obtidos.

No sétimo capítulo são apresentadas as conclusões da presente dissertação, é realizada a apreciação global do projeto, além disso, abordam-se as limitações e o trabalho a realizar-se no futuro.

2 Análise de valor

Neste capítulo será feita uma apreciação do presente trabalho numa perspetiva de negócio. Numa primeira fase, será apresentada a análise de valor da solução a ser desenvolvida. De seguida, será feita uma identificação e análise funcional do sistema a desenvolver. Por fim, será feita uma análise das *frameworks*/bibliotecas que podem ser utilizadas para o desenvolvimento de sistemas de recomendação.

2.1 Análise de valor

Peter Koen propôs um modelo, *New Concept Development* (NCD), em (*Koen et al., 2002*), constituído por cinco elementos:

- **Identificação da oportunidade:** está normalmente relacionado com os objetivos da empresa. Pode surgir por necessidade, isto é, dar resposta a uma necessidade do mercado, desta forma obtendo uma vantagem competitiva.
- **Análise da oportunidade:** serve para analisar se a oportunidade identificada anteriormente é viável.
- **Geração da ideia:** transforma a oportunidade identificada anteriormente num produto.
- **Seleção da ideia:** serve para analisar o valor da ideia.
- **Definição do conceito:** é criado o conceito de negócio

Nas secções seguintes serão apresentados os diferentes elementos do modelo NCD, enquadrados no contexto desta dissertação.

2.1.1 Identificação e análise da oportunidade

Devido à expansão da internet, atualmente, os clientes utilizam cada vez mais as plataformas de comércio eletrônico para comprarem os seus produtos. Estatísticas recentes, comprovam o ritmo explosivo de crescimento destas plataformas. Em 2017, o volume de vendas foi de 2,3 trilhões de dólares. Com trajetória ascendente, estima-se que em 2021 as vendas neste setor cheguem aos 4.88 trilhões de dólares, ver Figura 1 (Statista, 2019a). Outros estudos demonstram que 40% da população mundial já comprou produtos através de plataformas de comércio eletrônico (Statista, 2019b).

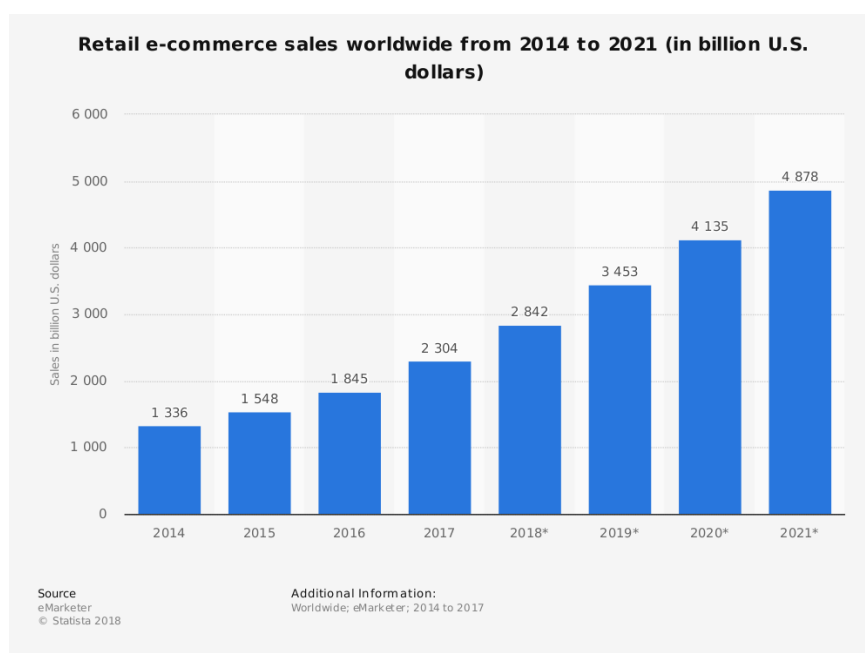


Figura 1- Previsão de vendas e-commerce

Um dos maiores desafios atualmente para as plataformas de comércio eletrônico é lidar com a enorme quantidade e variedade de produtos disponíveis. De acordo com (Gomez-Uribe and Hunt, 2015) “caso um utilizador do *Netflix*, não encontre algo que lhe interesse após um período de 60 a 90 segundos de navegação, a probabilidade de o mesmo não aderir ao serviço aumenta consideravelmente”. Por isso, é necessário encontrar soluções que lidem com o excesso de produtos.

2.1.2 Geração da ideia, seleção da ideia e definição do conceito

Um sistema de recomendação é uma solução interessante, na medida em que consegue apresentar os produtos mais pertinentes para um dado cliente. Das várias funcionalidades que um sistema de recomendação pode ter, destacam-se as seguintes: encontrar os produtos mais interessantes para um cliente; ordenar os produtos por grau de preferência; ajudar o

cliente a navegar pela plataforma; recomendar produtos em pacotes, como por exemplo: o cliente tem no seu carrinho um par de calças, e sistema de recomendação, sugere um cinto.

No caso da *Netflix*, o sistema de recomendação ajuda-os a adquirir clientes nos momentos certos, por exemplo: quando o cliente inicia pela primeira vez o serviço, o sistema de recomendação consegue encontrar produtos adequados para o cliente, prevenindo, desta forma o abandono do serviço.

A personalização deste tipo de plataformas é cada vez mais importante, uma vez que cerca de 45% dos clientes preferem comprar em plataformas que possuam um sistema de recomendação e cerca 56% desses clientes voltam a comprar na mesma (Giang, 2019). De acordo com (MacKenzie, Meyer and Noble, 2013), 35% das compras na *Amazon* e 75% do que os clientes veem na *Netflix* é fruto dos sistemas de recomendação.

Devido ao crescimento das plataformas de comércio eletrónico e aos claros benefícios dos sistemas de recomendação, é imperativo a sua integração neste tipo de plataformas, para que se possa criar uma experiência personalizada para o cliente.

2.2 Identificação e análise funcional

Um sistema de recomendação desempenha um papel importante, nas plataformas de comércio eletrónico, sendo que a principal função é a recomendação de produtos. Para o presente sistema de recomendação, esta tarefa pode ser alcançada de diferentes formas:

- **Encontrar produtos com base no histórico:** isto implica conhecer as preferências do cliente, através da análise do histórico de compras e/ou histórico de navegação.
- **Encontrar produtos em pacote:** isto implica conhecer que produtos que são vendidos em conjunto.
- **Encontrar produtos semelhantes:** isto implica conhecer todos os produtos disponíveis.

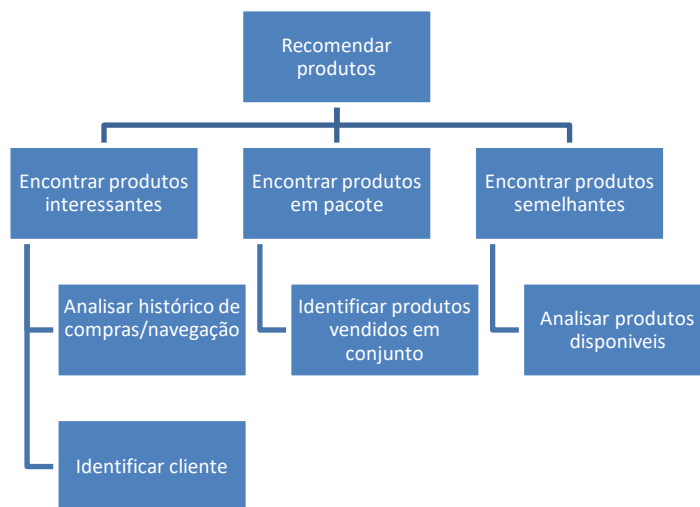


Figura 2- Funções do sistema de recomendação

Function Analysis System Technique (FAST) permite representar graficamente as diferentes funções de um sistema, permitindo ver como estas estão ligadas e como funcionam em conjunto, para a criação de valor. Esta técnica foi apresentada pelo *Charles Butcheway*, em 1965 na *SAVE Internacional Conference*, e rapidamente foi adotada como uma ferramenta na análise de valor (Borza, 2011).

Com base na Figura 3, podemos identificar três caminhos críticos:

- Apresentar produtos; Analisar histórico, Identificar cliente; Registo cliente
- Apresentar produtos; Identificar produtos complementares; Identificar contexto
- Apresentar produtos; Identificar produtos semelhantes; Identificar contexto

Apesar de não ser o comportamento mais correto, optou-se por representar os três caminhos críticos no mesmo diagrama FAST, com o objetivo de proporcionar uma visão geral das diferentes funcionalidades e do encadeamento das mesmas, em apenas num diagrama. Como requisitos de design definiu-se escalabilidade e desempenho. A nível de manutenção, os algoritmos deverão ser constantemente melhorados, quer a nível de performance quer a nível de precisão.

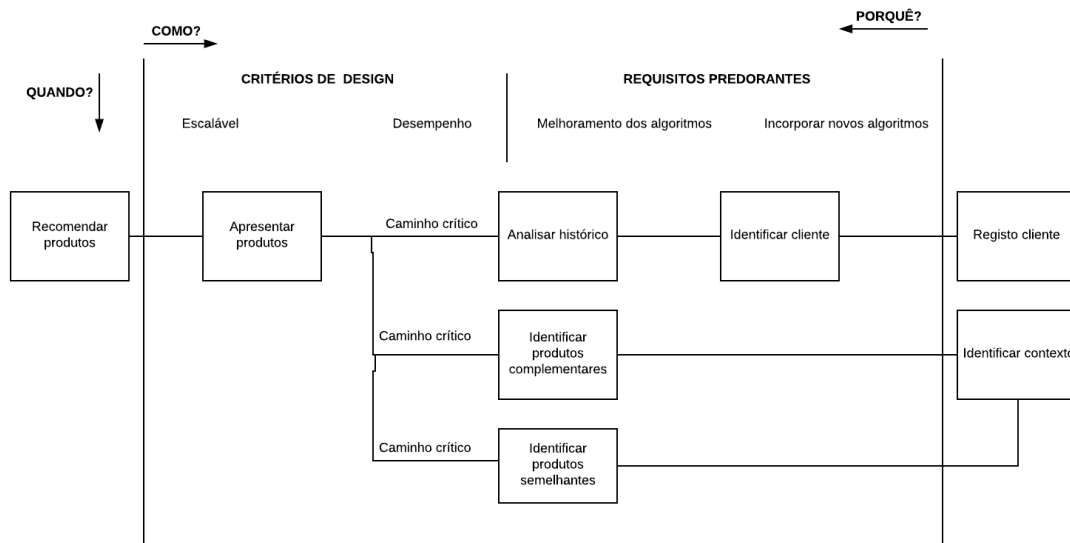


Figura 3- Diagrama FAST

Identificadas as funcionalidades do sistema de recomendação é necessário perceber a sua importância, por isso, na Tabela 1 é feita uma comparação das funcionalidades do presente sistema de recomendação, para percebermos qual a ordem de importância de cada uma delas.

Com base no resultado, podemos concluir a ordem de importância das funcionalidades é a seguinte:

- Encontrar produtos com base no histórico
- Encontrar produtos em pacote
- Encontrar produtos semelhantes

Tabela 1- Comparação das funcionalidades

	Encontrar produtos interessantes	Encontrar produtos em pacote	Encontrar produtos semelhantes	Total
Encontrar produtos com base no histórico		1	1	2
Encontrar produtos em pacote	0		1	1
Encontrar produtos semelhantes	0	0		0

As funcionalidades acima apresentadas possuem diferentes graus de importância, devido ao seu diferente impacto, quer para os clientes quer para a plataforma de comércio eletrônico. Na Figura 4, podemos ver um resumo do impacto nos diferentes intervenientes.

- **Cientes:** utilizadores da plataforma de comércio eletrônico. A personalização do sistema de recomendação fará com que o cliente sinta uma maior confiança, bem como um aumento na sua satisfação.
- **Comércio eletrônico (Empresa):** plataforma onde será integrado o sistema de recomendação. Uma consequência do aumento da satisfação do cliente é o aumento do nº de vendas, em todas as suas componentes, em volume e diversificação; bem como o aumento da fidelização.

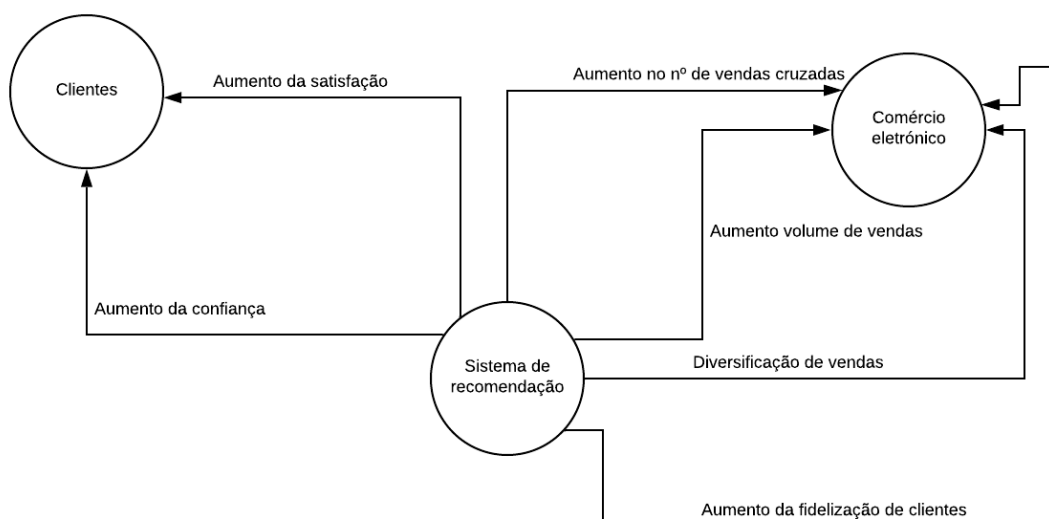


Figura 4- Impacto do Sistema de recomendação

2.2.1 Proposta de valor

O valor pode ser visto como o benefício que uma pessoa ou organização tem ao utilizar um determinado produto ou serviço.

A proposta de valor da presente tese é proporcionar ao consumidor um serviço personalizado de aconselhamento na procura do produto ideal.

2.2.2 Valor para o cliente

Para *Woodall* em (*Woodall, 2003*), o valor para o cliente é o reconhecimento por parte do cliente dos custos-benefícios de um determinado produto.

Neste caso, existem dois tipos de clientes:

- **Cliente final:** utilizador da plataforma de comércio eletrónico.
- **Empresa:** quem gere a plataforma de comércio eletrónico.

Na Tabela 2 podemos ver a previsão dos benefícios e sacrifícios para os clientes identificados.

Tabela 2- Benefícios e Sacrifícios do sistema de recomendação

Benefícios	Sacrifícios
Cliente final	
1. Apresentação de produtos interessantes para o cliente. 2. Apresentação de produtos ao cliente que de outra forma o poderia nunca vir a descobrir. 3. Recomendar produtos em pacotes	1. Como as recomendações serão apresentadas na página de detalhes do produto irá demorar mais algum tempo a ser carregada.
Empresa	
1. Aumento do volume de vendas 2. Aumento no nº de vendas cruzadas 3. Diversificação de vendas 4. Aumento da fidelização de clientes	1. Investimento no desenvolvimento do sistema de recomendação 2. Custos com os servidores

2.3 Análise e avaliação de alternativas

2.3.1 Análise de frameworks/bibliotecas

Existem diversas frameworks/bibliotecas que auxiliam o desenvolvimento de sistema de recomendação.

Foi realizada uma pesquisa, de modo a conhecer as diferentes frameworks/bibliotecas que existem, para que possa ser possível selecionar a ideal para o desenvolvimento do presente sistema de recomendação.

Dado restrições temporais, foram apenas analisadas três frameworks/bibliotecas que serão, de seguida, apresentadas.

2.3.1.1 Apache Mahout

Apache Mahout é uma *framework*, criada pela fundação *Apache Software*, que disponibiliza técnicas de aprendizagem automática, em diferentes áreas como por exemplo:

- **Recomendação:** efetua recomendações de produtos consoante o histórico de preferências do utilizador
- **Clustering:** agrupa dados consoante a sua semelhança
- **Classificação:** permite a categorização de produtos não classificados

Além disso, disponibiliza um conjunto de bibliotecas para a realização de operações matemáticas.

A evolução da *Apache mahout* para a versão 0.10.0 permitiu a introdução de diversas mudanças que foram responsáveis por modificar o foco do projeto. Assim, o foco foi orientado para a criação de uma plataforma o que permite a matemáticos, estatísticos e cientistas implementarem rapidamente os seus algoritmos de álgebra e matemática. Esta plataforma engloba ainda o *Apache Spark* que é um sistema de recomendação *out-of-the-box* que permite o processamento um grande volume de dados rapidamente, de uma forma paralela e distribuída (Foundation, no date).

2.3.1.2 Accord.net

Accord.net é uma *framework* desenvolvida em C# que combina algoritmos de aprendizagem automática com bibliotecas para o processamento de imagens e áudio e que visa auxiliar a construção de aplicações em diversas áreas, como por exemplo:

- **Visão computacional:** oferecendo diversos métodos para identificar e seguir faces/objetos.
- **Análise de sons e de sinais:** permite o carregamento, filtragem e a transformação de sinais de áudio, através da aplicação de filtros de áudio.
- **Estatística:** possui diversos métodos de classificação, regressão e *clustering* que permitem analisar estatisticamente dados.

A *framework* supracitada disponibiliza um conjunto de aplicações demo nas diferentes áreas que visam facilitar a compreensão das mesmas através de exemplos que demonstram o funcionamento dos diferentes métodos (Accord.NET, 2019).

2.3.1.3 ML.net

É uma *framework* de aprendizagem automática, de código aberto criada pela *Microsoft*. É multiplataforma, isto é, pode ser executada em diversos sistemas operativos, entre eles, *Windows, Linux, macOS*.

Pode ser utilizada para desenvolver e integrar algoritmos de aprendizagem automática, em qualquer tipo de aplicação, *Web*, móvel, *desktop*, jogos e até para a internet das coisas.

Combina o carregamento/transformação de dados, e o treino de modelos em um único *pipeline*. As transformações são aplicadas, tanto para os dados de treino, como para os dados de teste (Microsoft, no date).

2.3.2 Seleção da framework/bibliotecas

Para a seleção da framework/biblioteca foi utilizado o método *Analytic Hierarchy Process* (AHP).

AHP foi criado em 1970 por *Thomas Saaty*, e visa auxiliar as pessoas na tomada de decisão complexas. Este método é estruturado da seguinte forma: formulação do problema; definição dos critérios; formulação das alternativas (Saaty, 1990).

A Figura 5 representa a estruturação do método AHP para o problema em estudo.

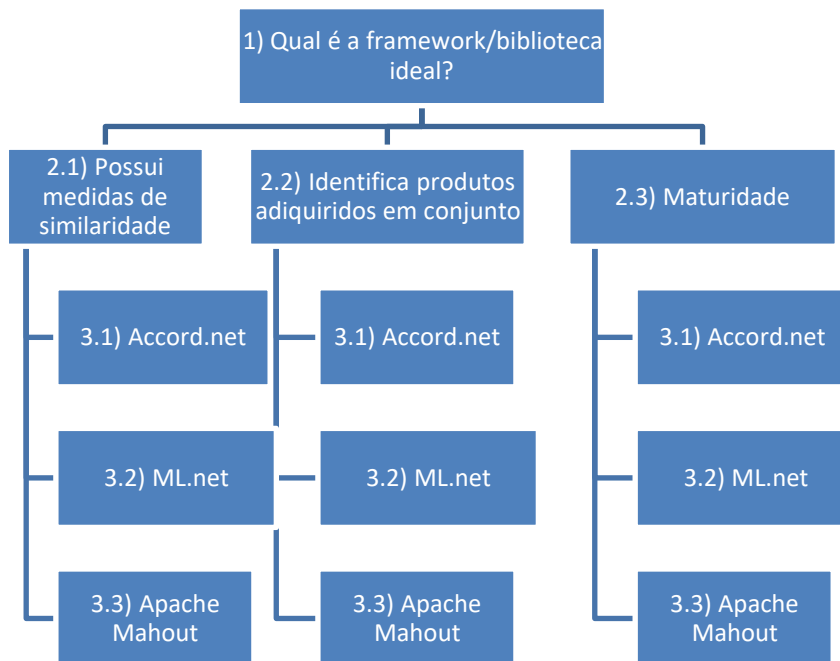


Figura 5- Árvore de decisão

Após a construção da árvore de decisão, a segunda fase consiste em definir as prioridades dos diferentes critérios, através de uma matriz de comparação, utilizando uma escala de valores de 1 a 9, ver Tabela 3 (Saaty, 1990).

Tabela 3- Matriz de comparação

	2.1	2.2	2.3
2.1	1	1/2	3
2.2	2	1	4
2.3	1/3	1/4	1

De seguida, devemos proceder à normalização da tabela e a determinação do vetor de prioridade, ver Tabela 4 **Error! Reference source not found.**.

Tabela 4- Matriz normalizada

	2.1	2.2	2.3	Prioridade relativa
2.1	3/10	2/7	3/8	32%
2.2	3/5	4/7	4	56%
2.3	1/3	1/4	1	12%

Após a normalização da matriz de comparação, devemos calcular o rácio de consistência (RC), para verificar a consistência das prioridades anteriormente definidas. Caso o RC seja superior 0.1, devemos redefinir as prioridades definidas na Tabela 3.

Neste caso, a consistência é de 0.02, por isso, podemos concluir que os valores definidos são consistentes.

O processo realizado anteriormente deve ser realizado para cada uma das alternativas, os cálculos podem ser consultados no Anexo A.

Tabela 5- Escolha de alternativa

	2.1	2.2	2.3	Total
3.1	0.54	0.52	0.33	50%
3.2	0.11	0.33	0.14	24%
3.3	0.35	0.14	0.52	26%

Com base nos resultados obtidos,(Tabela 5) através da aplicação do método AHP, podemos concluir que a *Accord.net* é a melhor opção para o desenvolvimento do presente sistema de recomendação.

3 Estado de Arte

Este capítulo descreve o conhecimento técnico necessário para a compreensão da solução implementada. Numa primeira fase, será feita uma introdução ao comércio eletrónico. De seguida, serão introduzidos os sistemas de recomendação, serão descritas as diferentes formas de obter *feedback* dos utilizadores, bem como as diferentes técnicas utilizadas para o desenvolvimento de sistemas de recomendação e algumas das diferentes técnicas para o cálculo da similaridade entre utilizadores e produtos. Depois, serão descritos os métodos de avaliação dos sistemas de recomendação, bem como, algumas das métricas para avaliar a qualidade dos sistemas de recomendação. No final, serão apresentados alguns exemplos de sistemas de recomendação.

3.1 Comércio eletrónico

Atualmente, existem diversas plataformas, que permitem aquisição de produtos através da internet, como por exemplo: *Amazon*, *EBay*, entre outras. A proliferação deste tipo de plataformas só foi possível devido à evolução da internet. Esta evolução originou uma revolução na forma como as empresas efetuam os seus negócios, criando novos canais de comunicação e distribuição (Gunasekaran *et al.*, 2002).

Quando desenvolvemos qualquer tipo de plataforma devemos ter sempre em consideração quais são os requisitos da mesma. Os requisitos direcionam a forma como desenvolvemos software. Inicialmente, as plataformas de comércio eletrónico eram construídas sobre uma

arquitetura monolítica, no entanto, com a evolução dos requisitos/necessidades foram adotando uma arquitetura orientada a microsserviços. De seguida, será feita uma breve introdução aos dois estilos arquiteturais.

3.1.1 Arquitetura monolítica

Normalmente, as aplicações empresariais são constituídas por três componentes, ver Figura 6 a interface com utilizador (*views*), que pode ser constituída por páginas HTML e *javascript* que é executado no browser do computador do utilizador; pela base de dados e pela componente aplicacional.

A componente aplicacional é executada num único processo; responsável por processar todos os pedidos HTTP; executar a lógica de negócio; atualizar os dados da base de dados e por atualizar a informação das *views*. Qualquer alteração na componente aplicacional implica efetuar o *deploy* das restantes componentes.

O desenvolvimento de aplicações monolíticas é simples, porque a lógica de negócio reside apenas num local, o que permite um maior controlo na ordem como as diferentes regras de negócio são executadas, é simples de testar, podem ser escaladas verticalmente, através do aumento dos recursos computacionais do servidor onde esta é executada, ou horizontalmente, executando várias instâncias da aplicação. No entanto, existem algumas desvantagens, como por exemplo: a manutenção torna-se mais complicada à medida que aplicação aumenta de tamanho; um erro num dos módulos pode impossibilitar o correto funcionamento de toda aplicação (Fowler and Lewis, 2014).

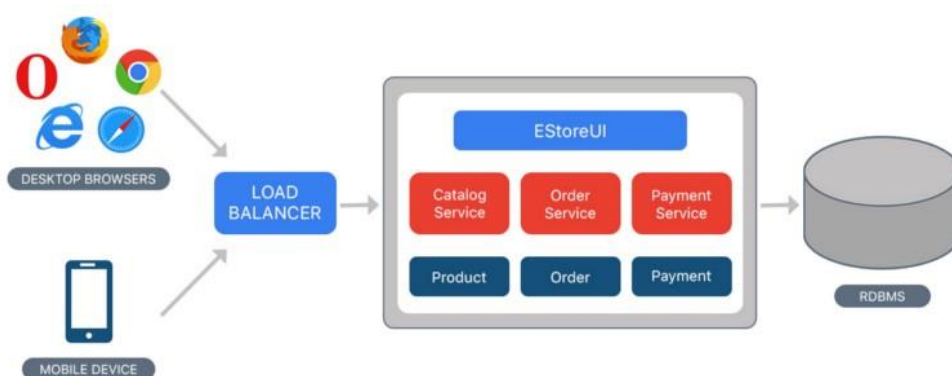


Figura 6- Arquitetura monolítica³

³ Disponível em: <https://medium.com/koderlabs/introduction-to-monolithic-architecture-and-microservices-architecture-b211a5955c63>. Acedido em 2019-01-31

3.1.2 Arquitetura orientada a microsserviços

De acordo com *Martin e Lewis* em (Fowler and Lewis, 2014), não existe apenas uma definição para arquitetura orientada a microsserviços, no entanto, podemos descrevê-la como um estilo arquitetural que estrutura uma aplicação através de um conjunto de serviços, em que cada um é executado de forma independente, comunicando entre si através de interfaces bem definidas.

Tal como não existe uma definição formal de microsserviços, não existe uma estrutura *standard* para este tipo de arquitetura, ver Figura 7, no entanto, normalmente, possuem as seguintes características:

- **Composição via serviços:** o facto de dividirmos *software* em serviços permite-nos efetuar o *deploy* individual dos mesmos sem comprometer a integridade da aplicação. Isto é, caso seja necessário modificar um serviço, apenas será necessário, efetuar o *deploy* do mesmo. No entanto, a composição via serviços possui algumas desvantagens: a comunicação entre serviços é mais cara que a invocação de métodos do mesmo processo; maior complexidade na atribuição de responsabilidades.
- **Organizado em torno de requisitos de negócio:** cada serviço é responsável por uma área específica de negócio. Esta organização origina equipas multidisciplinares, que são responsáveis por todo ciclo de vida do serviço.
- **Smart endpoints and dump pipes:** os serviços são responsáveis por processar a lógica de negócio e utilizam protocolos, como por exemplo HTTP ou *messaging*, para transferir a informação de serviço em serviço.
- **Descentralização:** permite que cada equipa selecione o conjunto de tecnologias que melhor se adapte ao problema que estão a resolver. Cada serviço é responsável por gerir a sua própria base de dados.
- **Tolerante à falha:** como os serviços podem falhar a qualquer momento, é importante detetar falhas, e, se possível restaurar o sistema automaticamente. Esta tarefa pode ser realizada através da implementação de mecanismos de resiliência e da monitorização em tempo real do estado do sistema.
- **Design evolutivo:** permite evoluir gradualmente de qualquer arquitetura para uma arquitetura orientada a microsserviços.

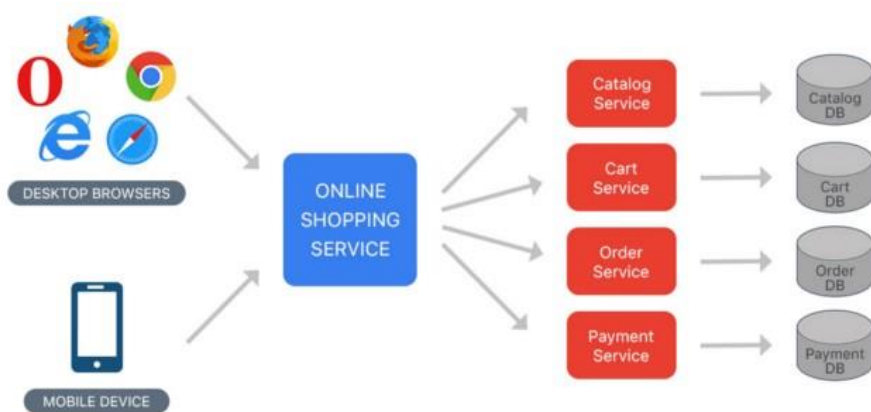


Figura 7- Arquitetura orientada a microsserviços⁴

3.1.3 Resumo das arquiteturas

Nas secções anteriores foram descritas duas das arquiteturas utilizadas no desenvolvimento de plataformas de comércio eletrónico. Na Tabela 6 é efetuada uma comparação das mesmas.

Tabela 6- Comparação das arquiteturas

Caraterísticas	Arquitetura monolítica	Arquitetura orientada a microsserviços
Manutenção	Com o passar do tempo e devido ao acréscimo de novas funcionalidades aplicação torna-se cada vez maior.	O fato de ser constituído por múltiplos serviços, estes permanecem relativamente pequenos
Deployment	Qualquer alteração à plataforma, obriga o <i>deploy</i> de toda a plataforma	Permite o <i>deploy</i> de cada serviço de forma independente
Escalabilidade	Não é possível escalar apenas uma parte do sistema	O fato de ser constituído por múltiplos serviços possibilita escalar individualmente cada um deles

3.2 Sistemas de recomendação

No nosso dia-a-dia temos de tomar decisões, quer no âmbito profissional como no pessoal, sem, muitas das vezes termos a experiência pessoal e/ou conhecimento suficiente para conhecer as diferentes opções. Por isso, baseamos as nossas decisões nas recomendações de

⁴ Disponível em: <https://medium.com/koderlabs/introduction-to-monolithic-architecture-and-microservices-architecture-b211a5955c63>. Acedido em 2019-01-31

outros. Um sistema de recomendação auxilia o processo de tomada decisão baseado na experiência de outros (Resnick and Varian, 1997).

Podemos definir um sistema de recomendação, como um sistema que é capaz de efetuar sugestões pertinentes ajustadas às necessidades de cada um.

Existem dois tipos de sistemas de recomendação: os sistemas de recomendação genéricos, apresentam as mesmas recomendações para todos os utilizadores, como por exemplo: os produtos mais populares; e os sistemas de recomendação personalizáveis, apresentam recomendações consoante as preferências dos utilizadores (Silva, 2011).

Em (Ricci, Rokach and Shapira, 2011) foi feita uma análise aos motivos que levam à adoção de um sistema de recomendação da perspetiva do prestador de serviço.

- **Aumento o nº de produtos vendidos:** este é o papel mais importante de um sistema de recomendação comercial. Estimular a venda de produtos, ao apresentar produtos interessantes para o utilizador
- **Diversificar as vendas:** estimular a venda de produtos, menos conhecidos pelo utilizador.
- **Aumentar a satisfação do utilizador:** a satisfação do cliente está diretamente relacionada com a precisão do sistema. Quanto maior for a capacidade do sistema de gerar recomendações fiáveis e assertivas, maior é a satisfação do utilizador.
- **Aumentar a fidelização do utilizador:** para os utilizadores mais antigos o sistema de recomendação deve ter a capacidade de promover uma experiência mais enriquecedora e pessoal, oferecendo-lhes, de certa forma, um tratamento especial.
- **Identificar as preferências do utilizador:** identificar o que os utilizadores realmente querem, ajudar a estabelecer novas necessidades de stock e ou à criação de novos serviços.

Como podemos verificar a maior parte dos pontos acima referidos, está dependente da capacidade do sistema de recomendação compreender as preferências dos utilizadores. Por isso, na seção 3.3, será feita uma análise ao tipo de técnicas existentes para a recolha das preferências dos utilizadores.

3.3 Recolha de Feedback

Os sistemas de recomendação dependem principalmente da qualidade da informação que direta ou indiretamente conseguem obter do utilizador. Um sistema de recomendação precisa de ter acesso às preferências do utilizador, para conseguir efetuar recomendações precisas (Isinkaye, Folajimi and Ojokoh, 2015). De acordo com *Brafman* e *Domshlak* em (Brafman and Domshlak, 2009) a preferência pode ser vista como o instrumento utilizado para a tomada de decisão, permitindo, desta forma selecionar o que realmente queremos. Desta maneira, o

sucesso de um sistema recomendação está diretamente ligado à capacidade de representar as preferências dos utilizadores.

A recolha das preferências do utilizador pode ser efetuada de três formas distintas: explícita, implícita ou híbrida (Isinkaye, Folajimi and Ojokoh, 2015).

- **O *feedback* explícito:** permite aos utilizadores expressarem de forma inequívoca a sua admiração ou descontentamento sobre um determinado produto. Esta decorre normalmente através do preenchimento de algum tipo de formulário, utilizando uma escala previamente definida. As escalas podem ser numéricas, as mais comuns de 1-5, ou binárias Gosto/Não Gosto. As redes sociais, como por exemplo *Facebook* e *LinkedIn* utilizam o sistema de classificação binário, enquanto que a *Amazon* utiliza a classificação numérica. Apesar deste tipo de *feedback* ser mais preciso, é mais difícil de obter, pois obriga a uma interação direta do utilizador (Jawaheer, Szomszor and Kostkova, 2010; Núñez-Valdez *et al.*, 2018).
- **O *feedback* implícito:** consiste em inferir as preferências do utilizador através da análise das suas ações/comportamentos, como por exemplo: *clicks*, histórico de compras e/ou de navegação. Este método é menos intrusivo para o utilizador, no entanto, é menos preciso e apenas permite capturar *feedback* positivo (Jawaheer, Szomszor and Kostkova, 2010; Isinkaye, Folajimi and Ojokoh, 2015; Núñez-Valdez *et al.*, 2018).
- **O *feedback* híbrido:** permite combinar os dois tipos de *feedback* anteriormente referidos, com o objetivo de minimizar as suas fraquezas e aproveitando os seus pontos fortes (Isinkaye, Folajimi and Ojokoh, 2015).

A Tabela 7 apresenta um resumo das características dos tipos de *feedback* implícito e explícito apresentados acima.

Tabela 7- Características do *feedback* Implícito e Explícito

	Implícito	Explícito
Precisão	Baixa	Alta
Popularidade	Alta	Baixa
Níveis de expressividade	Positivo	Positivo e negativo
Sensível ao contexto	Sim	Sim

3.4 Técnicas de sistemas recomendação

Nesta secção serão apresentadas as técnicas mais comuns para o desenvolvimento de sistemas de recomendação, que podem ser divididas em três categorias: baseado em conteúdos, filtragem colaborativa, ou híbrido (Isinkaye, Folajimi and Ojokoh, 2015).

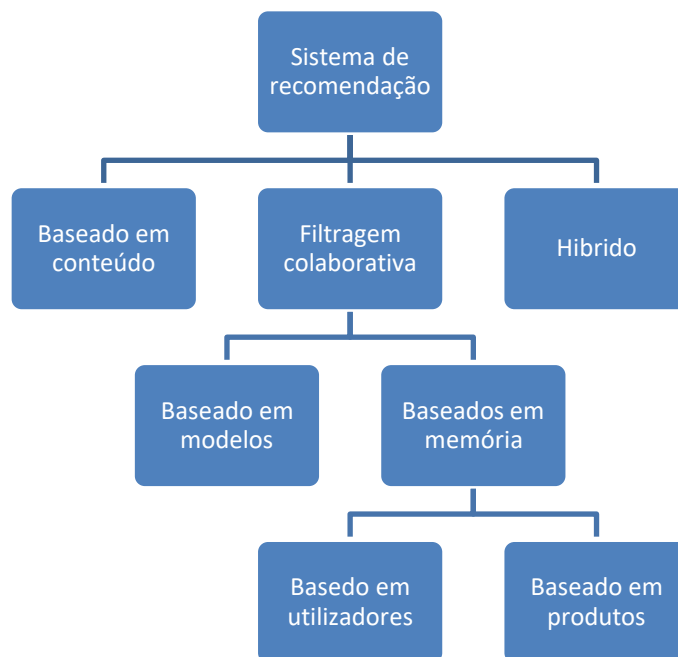


Figura 8- Tipos de sistemas de recomendação⁵

3.4.1 Baseado em conteúdos

Estes sistemas geram recomendações com base nos interesses do utilizador, comparando os produtos considerados por este de interessante, com os produtos disponíveis. Os produtos que mais se assemelham aos interesses do utilizador são os recomendados (Isinkaye, Folajimi and Ojokoh, 2015).

Estes sistemas são usualmente constituídos por três componentes:

- **Content Analyser:** responsável por analisar/normalizar os dados
- **Profile learner:** responsável por criar o perfil do utilizador
- **Filtering component:** responsável por filtrar os produtos mais interessantes para o utilizador

⁵ Disponível em <https://www.sciencedirect.com/science/article/pii/S1110866515000341>. Acedido em 2018-12-26

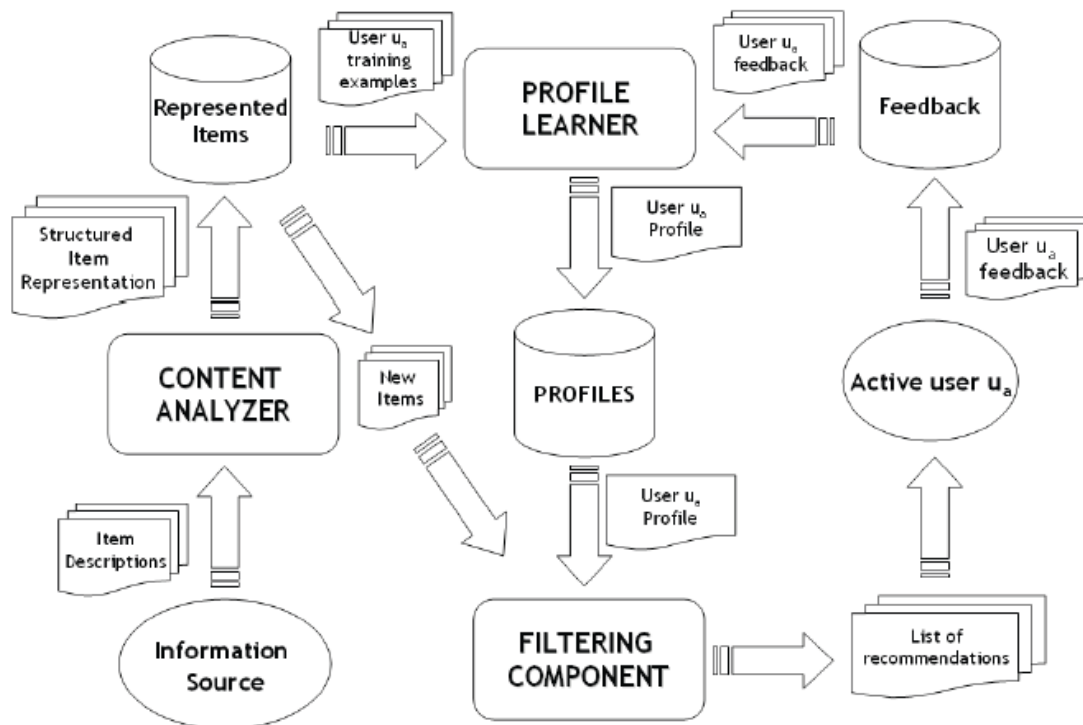


Figura 9 -Arquitetura de um sistema baseado em conteúdos⁶

Esta técnica necessita que as características dos produtos estejam bem definidas. No entanto, nem sempre isto é possível, como por exemplo, os documentos de texto.

Nestes casos, o *Content Analyser* efetua um pré-processamento dos documentos, normalmente com recurso à técnica *frequency/inverse document frequency* (TF-IDF).

A TF-IDF permite medir a importância (peso) de cada palavra num documento de texto. O peso está diretamente relacionado com o nº de vezes que uma determinada palavra é utilizada no documento, no entanto, esse valor é equilibrado pela frequência da palavra no resto dos documentos. Dessa forma, é possível fazer a diferenciação das palavras que, embora figurem várias vezes num documento, não apresentam grande relevância para o mesmo, como é o caso das preposições. Contudo, por esta técnica se basear apenas em palavras, apresenta algumas limitações:

- **Sinónimos:** palavras que embora sejam diferentes significam a mesma coisa.
- **Polissemia:** palavras que utilizadas em contextos diferentes possuem significados distintos.

O resultado deste pré-processamento é guardado numa base de dados para, mais tarde, ser utilizado pelo *Profile learner* e pelo *Filtering Component*.

⁶ Disponível em https://www.researchgate.net/figure/High-level-architecture-of-a-Content-based-recommender-system_fig1_300884439. Acedido a 2019-05-12

Antes de sugerir produtos ao utilizador, é necessário conhecer as suas preferências. Para tal, o *Profile Learner*, utiliza geralmente algoritmos de aprendizagem automática, como por exemplo *Naive Bayes*, que permitem inferir os interesses do utilizador (modelo) através do *feedback*, fornecido pelo mesmo. O *feedback* pode ser fornecido de forma explícita ou implícita. Como os interesses dos utilizadores variam ao longo do tempo é importante que o modelo do utilizador seja atualizado. A frequência da atualização depende do tipo de ambiente.

Após a construção do perfil dos produtos e do utilizador podemos começar a gerar recomendações. O *Filtering Component* é o responsável por determinar se um utilizador tem interesse num determinado produto, comparando os atributos do produto com as características do perfil do utilizador, através de algoritmos de correlação, por exemplo: a similaridade do cosseno. Este componente dispõe de ferramentas que o permitem priorizar os produtos semelhantes, apresentando, desta forma, apenas uma sublista dos produtos semelhantes (Lops, de Gemmis and Semeraro, 2011).

3.4.2 Filtragem colaborativa

Esta técnica baseia as suas recomendações nos comportamentos/interesses dos utilizadores do sistema. O interesse do utilizador sobre um produto, pode ser obtido de forma explícita ou implícita. Esta técnica baseia-se na premissa de que os utilizadores podem ser agrupados com base nas suas preferências. Assume, que se um conjunto de utilizadores, partilha da mesma opinião sobre conjunto de produtos é provável que partilhem da mesma opinião sobre outros produtos (Sarwar *et al.*, 2001; Herlocker *et al.*, 2004).

Esta técnica pode ser dividida em duas categorias: baseados em memória e em modelos (Isinkaye, Folajimi and Ojokoh, 2015).

3.4.2.1 Baseados em memória

Esta utiliza os dados disponíveis no sistema para estabelecer correlações entre produtos ou utilizadores, através da sua semelhança. Devido à sua eficácia, atualmente, são utilizadas em diversos sistemas de recomendação. Além de conseguirem prever a utilidade de um produto, conseguem gerar uma lista com os N produtos mais interessantes para um utilizador (Herlocker *et al.*, 2004; Su and Khoshgoftaar, 2009).

Esta categoria pode ser dividida em duas abordagens: baseada em utilizadores ou baseada em produtos (Isinkaye, Folajimi and Ojokoh, 2015).

A baseada em utilizadores, de acordo com (Herlocker *et al.*, 2004), foi apresentada, pela primeira vez por Resnick em (Resnick *et al.*, 1994). Procura identificar, para um determinado utilizador, outros utilizadores com interesses semelhantes, denominados vizinhos, e visa recomendar produtos que estes achem de interessante e que o utilizador em causa ainda não tenha comprado/classificado.

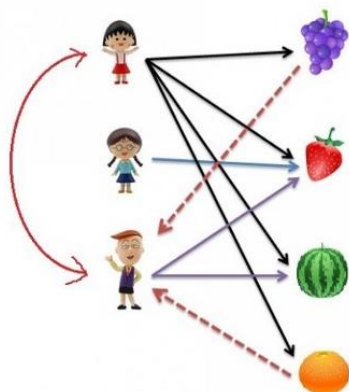


Figura 10- Baseado no utilizador⁷

O processo para calcular a utilidade de um produto ou de produzir a lista de N produtos é semelhante, e pode ser dividido em duas tarefas: encontrar os vizinhos para um utilizador; produzir a lista com os N produtos mais relevantes ou prever a utilidade de um produto.

Antes de se iniciar este processo, é necessário a construção da matriz Utilizador-Produto. Esta é contruída com as com as classificações ou compras de todos os utilizadores do sistema. Assim que esta matriz esteja contruída, podemos dar início ao processo de recomendação (Isinkaye, Folajimi and Ojokoh, 2015).

A primeira tarefa do processo é encontrar os vizinhos para um determinado utilizador. Para isso, precisamos de calcular a semelhança entre os mesmos. Existem vários algoritmos, que podem ser utilizados para esse efeito, como por exemplo: a correlação de *pearson*, a similaridade do cosseno, entre outros. De seguida, são seleccionados k utilizadores, para serem os vizinhos. O número de vizinhos tem um impacto na qualidade das recomendações, devendo este ser ajustado caso a caso, como comprovado em (Sarwar *et al.*, 2000).

A segunda tarefa, difere consoante o objetivo do sistema de recomendação. Caso seja para produzir a lista com os N produtos, consiste em agrupar os produtos classificados/comprados pelos vizinhos, juntamente com a sua frequência. Por fim, são seleccionados os N produtos com maior frequência que ainda não tenham sido classificados/comprados pelo utilizador em causa (Karypis, 2001). Caso o objetivo seja prever a utilidade de um produto para um utilizador, a segunda tarefa consiste em calcular a classificação do produto, esta é calculada

⁷ Disponível em <https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f> Acedido em 2018-12-26

através da média das classificações dos vizinhos para o produto em causa (Zhao and Shang, 2010).

Na Tabela 8, podemos ver a Rita e os seus vizinhos (João e Rui). Ambos os vizinhos da Rita viram o Homem-Formiga e *Thor*, no entanto, a Rita ainda não viu nenhum desses filmes. Caso o objetivo fosse produzir uma lista os *N* produtos mais relevantes, os filmes que seriam recomendados à Rita seriam o Homem-Formiga e o *Thor*.

Tabela 8 - Classificações dos utilizadores mais semelhantes à Rita

Utilizador/Produto	Deadpool	Homem-Formiga	Pantera Negra	Thor
Rita	5	?	4	?
João	5	5	5	5
Rui	3	3	3	3

Como os interesses dos utilizadores mudam ao longo do tempo, a maior parte destes sistemas encontram os vizinhos para um determinado utilizador *onfly*, isto é, quando é necessário efetuar recomendações. À medida que o número de utilizadores aumenta, este processo torna-se cada vez mais dispendioso (Herlocker et al., 2004). Para lidar com esta limitação foi criada a abordagem baseada em produtos (Sarwar et al., 2001; Linden, Smith and York, 2003).

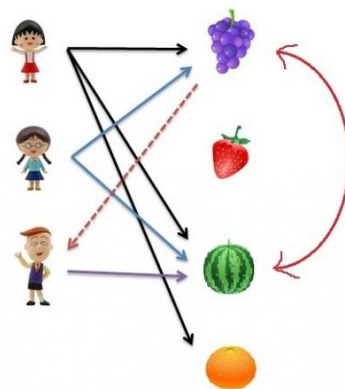


Figura 11- Baseado em produtos⁸

A abordagem baseada em produtos, foca-se em encontrar produtos semelhantes aos produtos comprados/classificados pelo utilizador e a recomendar os mais semelhantes (Herlocker et al., 2004).

Esta abordagem é constituída por duas componentes:

⁸ Disponível em <https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f> Acedido em 2018-12-26

- **Componente *offline*:** consiste na construção da matriz Produto-Produto.

A construção desta matriz só é possível devido à natureza estática dos produtos. De acordo com (Linden, Smith and York, 2003), a matriz pode ser contruída iterando sobre os produtos disponíveis e verificando para cada produto (P) os utilizadores (U), que compraram P, calculando de seguida a semelhança entre P e os produtos adquiridos pelo utilizador (P2). A semelhança entre os produtos pode ser calculada através de vários algoritmos, como por exemplo: a similaridade do cosseno, a correlação de *pearson* (Sarwar et al., 2001; Herlocker et al., 2004).

- **Componente *online*:** consiste em procurar os *N* produtos mais relevantes ou calcular a utilidade de um determinado produto para um dado utilizador.

Com o passar do tempo, a matriz irá ficar desatualizada, devido a novas classificações/compras, no entanto, esta pode ser recalculada sempre que necessário (Linden, Smith and York, 2003; Herlocker *et al.*, 2004). A chave do sucesso desta abordagem reside exatamente nisso, apesar de se utilizar apenas parte dos produtos disponíveis, esta abordagem consegue obter melhores resultados do que abordagem baseada em utilizadores (Sarwar *et al.*, 2001).

3.4.2.2 Baseada em modelos

Esta abordagem utiliza técnicas de aprendizagem automática, para criar um modelo capaz de representar os interesses dos utilizadores (Aditya, Budi and Munajat, 2017). De acordo com (Isinkaye, Folajimi and Ojokoh, 2015), esta abordagem permite gerar recomendações rapidamente, com resultados semelhantes à abordagem baseada em memória e resolve os problemas associados com a esparsidade de dados.

De acordo com (Sarwar *et al.*, 2001), estas são algumas das técnicas de aprendizagem automática utilizadas para a construção do modelo:

- **Regras associativas:** analisa um conjunto de transações, com o objetivo de identificar padrões, isto é, objetos que ocorrem juntos com algum grau de frequência. Um dos algoritmos mais utilizados é o *Apriori* (Sarwar *et al.*, 2001; Aditya, Budi and Munajat, 2017).
- **Clustering:** procura identificar um conjunto de características semelhantes entre utilizadores e agrupá-los consoante as mesmas. Após a devida categorização dos utilizadores, cada um deles recebe recomendações com base nos interesses do seu grupo.
- **Bayesian network:** utiliza um modelo probabilístico para resolver o problema das recomendações. Deve ser utilizado em ambientes, onde as preferências do utilizador são constantes ou mudam com pouca frequência (Sarwar *et al.*, 2001; Isinkaye, Folajimi and Ojokoh, 2015).

Singular Value Decomposition é uma técnica de fatorização de matrizes, que permite minimizar uma matriz sem perder os seus valores originais. Cada utilizador e produto são representados por um vetor, que juntos reproduzem a matriz original. A redução do tamanho de uma matriz é bastante útil, principalmente quando lidamos com matrizes esparsas (Sarwar *et al.*, 2002).

Este tipo de técnicas tem vindo a ganhar destaque, devido ao facto de o vencedor do concurso “The Netflix Prize” ter incorporado no seu algoritmo uma técnica de factorização de matrizes (Toscher and Jahrer, 2009).

3.4.3 Híbrida

Esta técnica propõe a combinação de várias técnicas de recomendação. Com o objetivo de criar sistemas de recomendação mais fiáveis e para ultrapassar as limitações das técnicas anteriormente apresentadas (Burke, 2002; Thorat, Goudar and Barve, 2015).

De acordo com (Burke, 2002) e (Çano and Morisio, 2017), esta técnica pode ser dividida em sete categorias:

- **Weighed:** combina os resultados das diferentes técnicas de recomendação, atribuindo diferentes pesos a cada uma delas. Os pesos atribuídos a cada uma das técnicas podem ser ajustadas conforme o *feedback* do utilizador.
- **Switching:** escolhe a técnica de recomendação a aplicar consoante o contexto.
- **Mixed:** agrega as recomendações geradas por diferentes técnicas.
- **Feature combination:** utiliza dados de outras técnicas como informação adicional para a segunda técnica. Por exemplo, é possível utilizar as classificações dos utilizadores sobre um produto (oriundas da técnica filtragem colaborativa), e utilizá-las como uma das características para calcular a semelhança dos produtos.
- **Cascade:** utiliza um processo iterativo para construir a lista de recomendações. As recomendações geradas pela primeira técnica são refinadas pela aplicação da segunda técnica. A segunda técnica só é aplicada nos produtos que precisam de ser diferenciados.
- **Feature augmentation:** o *output* de uma técnica é utilizado como *input* da segunda. Diferencia-se da *Feature combination*, pois nesta, são utilizados dados brutos provenientes de diferentes fontes.
- **Meta-level:** utiliza um modelo gerado pela primeira técnica e utiliza-o como *input* da segunda técnica. Normalmente, é utilizada a técnica baseada em conteúdos para produzir uma representação do produto, de seguida, é aplicada a técnica filtragem colaborativa para combinar os produtos com os utilizadores.

3.5 Limitações e vantagens dos sistemas de recomendação

Nesta secção serão descritas as vantagens e limitações das diferentes técnicas de recomendação apresentadas anteriormente. Primeiramente, serão descritas as limitações comuns às diferentes técnicas e, de seguida, serão apresentadas as limitações e vantagens individuais de cada uma delas.

3.5.1 Limitações comuns às diferentes técnicas

Um dos problemas mais comuns nos sistemas de recomendação é o chamado “arranque a frio” (*Cold Start*). O problema está associado à incapacidade do sistema de recomendação de gerar recomendações para utilizadores ou produtos novos, pois ainda não possuem informação suficiente para tal. Existem autores que subdividem este problema em: o problema do utilizador novo e do produto novo (Su and Khoshgoftaar, 2009).

A privacidade é outro problema, pois, todos eles fornecem melhores recomendações se souberem mais informações sobre os seus utilizadores, no entanto, nem todos gostam de expor a sua informação a terceiros (Su and Khoshgoftaar, 2009).

3.5.2 Limitações e Vantagens da técnica baseada em conteúdos

As limitações são (Lops, de Gemmis and Semeraro, 2011):

- **Análise limitada de conteúdo:** as características dos produtos são utilizadas para determinar se um produto é do interesse para o utilizador, no entanto, nem todos os produtos possuem características bem definidas, o que dificulta o processo de recomendação.
- **Utilizador novo:** o sistema precisa de conhecer as preferências do utilizador, por isso, enquanto este não fornecer as informações necessárias, o sistema não é capaz de gerar recomendações.
- **Superespecialização:** consiste em recomendar sempre o mesmo tipo de produtos, isto é, não é capaz de recomendar algo inesperado.

As vantagens são (Lops, de Gemmis and Semeraro, 2011):

- **Produto novo:** é capaz de recomendar produtos que ainda não possuam classificações
- **Transparência:** permite descobrir facilmente o porquê de um produto ter sido recomendado a um utilizador.
- **Independência do utilizador:** não necessita de informação de terceiros para efetuar recomendações.

3.5.3 Limitações e Vantagens da técnica filtragem colaborativa

As limitações são: (Su and Khoshgoftaar, 2009; Isinkaye, Folajimi and Ojokoh, 2015)

- **Produto novo:** não é possível efetuar recomendações de produtos para os quais não existam classificações ou compras
- **Utilizador novo:** o sistema precisa de conhecer as preferências do utilizador, por isso, enquanto este não fornecer as informações necessárias, o sistema não é capaz de gerar recomendações.
- **Utilizador Incomum:** utilizadores com preferências pouco comuns não recebem recomendações muito precisas, porque não é possível encontrar utilizadores semelhantes.
- **Esparsidade (*Sparsity*):** o número de classificações necessárias é inferior ao número de classificações realizadas pelos utilizadores.
- **Escalabilidade:** o aumento do número de utilizadores e/ou produtos pode originar um aumento nos recursos computacionais necessários para gerar recomendações.

As vantagens são (Su and Khoshgoftaar, 2009; Isinkaye, Folajimi and Ojokoh, 2015):

- **Agnóstica ao negócio:** não precisam de conhecer os produtos que estão a recomendar.
- **Serendipitous:** esta pode ser vista como a capacidade de recomendar produtos que sejam surpreendentes e ao mesmo tempo do agrado do utilizador

3.5.4 Limitações e Vantagens da técnica híbrida

As vantagens e limitações desta técnica estão diretamente relacionadas com a forma como as diferentes técnicas são combinadas. Por exemplo, a abordagem *Mixed*, permite ultrapassar uma das limitações associadas à técnica colaborativa, “Produto novo”, no entanto, sofre do problema “Utilizador novo”, isto é, não é capaz de gerar recomendações para utilizadores recentes (Burke, 2002; Thorat, Goudar and Barve, 2015).

3.6 Medidas de similaridade

A medida de similaridade é uma função que permite quantificar a semelhança entre dois objetos. A escolha da medida de similaridade é bastante importante, pois tem um impacto direto na qualidade do sistema de recomendação, como podemos verificar em (Sarwar *et al.*, 2001). Por isso, antes de escolher uma medida, devem ser efetuados testes com mais do que uma para verificar qual das medidas se aplica ao caso em estudo (Agarwal and Chauhan, 2017).

De seguida, serão descritas algumas das medidas de similaridade mais utilizadas, no desenvolvimento de sistemas de recomendação.

- **Correlação de pearson:** calcula o grau de correlação entre dois vetores. O resultado desta medida varia entre -1 e +1. -1 indica uma correlação negativa; 0 indica que não existe correlação; +1 indica uma correlação positiva. Esta medida é dada pela seguinte equação 1

$$pearson(u, v) = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2 + \sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \quad (1)$$

Onde \bar{r}_u, I e \bar{r}_v, I representam a média das classificações para os utilizadores u e v , respetivamente, para o produto I

- **Similaridade do cosseno:** é medida através do ângulo entre dois vetores. É a medida mais utilizada em sistemas de recomendação do tipo filtragem colaborativa. Uma das desvantagens desta medida é que trata as preferências nulas, como negativas. Esta é dada pela equação 2

$$cos(u, v) = \frac{\vec{R}_u \cdot \vec{R}_v}{|\vec{R}_u| \cdot |\vec{R}_v|} \quad (2)$$

Onde “.” representa o produto entre dois vetores. \vec{R}_u e \vec{R}_v são os vetores com as classificações dos utilizadores u e v , respetivamente.

- **Semelhança do cosseno ajustada:** esta medida tem em consideração os hábitos de classificação dos utilizadores. Esta medida é dada pela seguinte equação 3

$$Acos = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2 + \sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \quad (3)$$

Onde \bar{r}_u representa a média das classificações do utilizador u .

- **Semelhança de Jaccard:** calcula a semelhança entre utilizadores através do número de produtos em comum. Quanto mais produtos dois utilizadores tiverem em comum, mais semelhantes são. Esta é dada pela seguinte equação 4

$$J(u, v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \quad (4)$$

Onde I_u representa os produtos classificados pelo utilizador u , e I_v , representa os produtos classificados pelo utilizador v .

3.7 Métodos de avaliação

Avaliar a performance de algoritmos de recomendação consiste fundamentalmente em avaliar o grau de aceitação do utilizador. Esta tarefa pode ser elaborada utilizando três métodos: *offline*, *online* ou a combinação dos dois métodos anteriormente referidos. De acordo com (Gunawardana and Shani, 2009), a avaliação *offline* deve ser utilizada para identificar os sistemas de recomendação mais promissores, para que desta forma apenas os mais promissões sejam submetidos a avaliação *online*.

De seguida, será feita uma breve descrição dos métodos de avaliação *offline* e *online*.

3.7.1 Avaliação offline

Os dados utilizados na avaliação *offline* devem-se assemelhar sempre que possível aos dados com que o sistema irá lidar. Por isso, a seleção/tratamento dos dados é uma tarefa importante que deve ser efetuada de forma cuidada (Gunawardana and Shani, 2009).

Esta avaliação requer a simulação do processo *online*, isto é, precisa-se de simular o processo de aceitação/rejeição das recomendações geradas pelo sistema de recomendação. Para tal, utiliza-se um *dataset*, com classificações e/ou compras de utilizadores, e divide-se em duas partes. Basicamente, a primeira parte do *dataset* serve para o sistema “aprender” o comportamento dos utilizadores, e a segunda parte serve para avaliar a sua aprendizagem (Herlocker *et al.*, 2004; Gunawardana and Shani, 2009).

De seguida, serão apresentadas algumas das técnicas utilizadas na avaliação *offline*.

A técnica *hold-out*, consiste em dividir um *dataset*, em dois subconjuntos, um de treino e outro de teste. O subconjunto de dados de treino é utilizado para “treinar” o modelo, e o de testes é utilizado para avaliar o “treino” do modelo. Normalmente, o subconjunto de dados de treino é maior do que o subconjunto de dados de teste (Allibhai, 2018).

A técnica *k-fold cross validation*, permite maximizar os dados disponíveis. Este método consiste em dividir o *dataset* em k partes iguais. De seguida, um subconjunto é utilizado para teste e os restantes $k-1$ são utilizados para treinar o sistema. O processo é repetido k vezes alternando o subconjunto de testes a cada iteração. Desta forma, conseguimos obter resultados mais confiáveis (Herlocker *et al.*, 2004; Allibhai, 2018).

A técnica *k-fold cross validation* é normalmente a escolhida, pois permite avaliar o modelo diversas vezes apenas com um *dataset*. Por causa disso, permite-nos ficar com uma melhor perceção do comportamento futuro do modelo, no entanto, requer maior poder

computacional do que o *hold-out*. Contudo, a *hold-out* pode ser utilizada quando o temos um *dataset* de grandes dimensões e não temos muito tempo para avaliar o modelo.

Esta abordagem é de fácil implementação; não acarreta muitos custos para ser aplicada; é utilizada por diversos autores, (Sarwar *et al.*, 2001; Burke, 2002; Herlocker *et al.*, 2004) para comparar objetivamente diferentes algoritmos; no entanto, apenas permite avaliar produtos que existam no *dataset*; não permite obter *feedback* do utilizador (Herlocker *et al.*, 2004; Gunawardana and Shani, 2009).

3.7.2 Avaliação online

Como referido anteriormente, esta avaliação é normalmente utilizada para selecionar o sistema de recomendação mais promissor. Esta analisa o comportamento do utilizador perante as recomendações do sistema de recomendação. Para tal, os utilizadores são divididos em grupos, e são redirecionados, de forma aleatória para os diferentes sistemas de recomendação. O *feedback* dos utilizadores é recolhido, para, posteriormente ser analisado. Uma das formas de recolher o *feedback* do utilizador é através de *clicks*, por exemplo, apresentamos uma lista com dez de produtos ao utilizador. O utilizador clica em dois produtos, neste caso, podemos considerar que, 20% dos produtos são interessantes para o utilizador.

Apesar de esta abordagem ser a única forma de medir verdadeiramente a satisfação do utilizador, é complicada e implica custos para ser implementada. Por isso, devemos avaliar o sistema de recomendação numa primeira fase *offline*, e caso os resultados sejam satisfatórios, utilizar a abordagem *online*. Desta forma, conseguimos mitigar a apresentação de más recomendações, que de outra forma poderiam desencorajar os utilizadores de usar o sistema (Gunawardana and Shani, 2009).

3.8 Avaliar a qualidade de sistema de recomendação

De acordo com (Sarwar *et al.*, 2001; Ge, Delgado-Battenfeld and Jannach, 2010), existem diversas métricas para avaliar a qualidade de um sistema de recomendação, no entanto, a maior parte dos sistemas de recomendação são avaliados apenas quanto à sua precisão.

Podendo ser divididas em duas categorias:

- **Statistical accuracy metrics:** avaliam a precisão do sistema, comparando o resultado da previsão com as classificações reais do utilizador. Erro médio absoluto (MAE) é a medida mais utilizada nesta categoria. Esta mede a variação entre as recomendações feitas pelo sistema e o valor real da classificação. A precisão do sistema é inversamente proporcional à MAE, isto é, quanto mais baixa for a MAE, mais preciso é o sistema. Esta é dada pela seguinte equação

$$MAE = \frac{\sum_{i=1}^N |p_{u,i} - r_{u,i}|}{N} \quad (5)$$

Raiz quadrada do erro quadrático médio (RMSE) é uma métrica semelhante à MAE, no entanto, dá mais importância aos erros absolutos. Esta é dada pela seguinte equação

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (p_{u,i} - r_{u,i})^2}{N}} \quad (6)$$

Onde $p_{u,i}$ é a previsão da classificação para o produto, i para o utilizador u , e $r_{u,i}$ é a classificação real do utilizador u para o produto i . N é o número total de classificações do caso de teste.

- **Decision support accuracy metrics:** avaliam a capacidade do sistema em seleccionar os produtos mais relevantes para o utilizador. Estas assumem que o processo de recomendação é uma operação binária, ou seja, distinguem bons produtos de maus produtos.

A Exatidão é a relação entre as previsões corretas e todas as previsões feitas.

$$Exatidão = \frac{VP + VN}{VP + VN + FP + FN} \quad (7)$$

A Precisão mede a probabilidade um produto recomendado ser relevante.

$$Precisão = \frac{VP}{VP + FP} \quad (8)$$

Recall mede a probabilidade de um produto relevante ser recomendado.

$$Recall = \frac{VP}{VP + FN} \quad (9)$$

A *Fmeasure* combina as duas métricas anteriores, numa só (Sarwar et al., 2001; Isinkaye, Folajimi and Ojokoh, 2015).

$$Fmeasure = \frac{2 * Precisão * Recall}{Precisão + Recall} \quad (10)$$

A nomenclatura utilizada para a definição das equações 6,7,8 e 9 foram baseadas em (Google, 2018).

As duas categorias acima apresentadas estão relacionadas apenas com a precisão do sistema, no entanto, a qualidade do sistema pode ser medida além da sua precisão. Outra medida, que pode ser utilizada para avaliar a qualidade destes sistemas é a sua cobertura. A cobertura pode ser vista como a percentagem de produtos/utilizadores para o qual o sistema de recomendação consegue efetuar recomendações. Por fim, a *Serendipitous*, esta pode ser vista como a capacidade de recomendar produtos que sejam surpreendentes e ao mesmo tempo do agrado do utilizador (Ge, Delgado-Battenfeld and Jannach, 2010).

Existem diversas métricas para avaliar a qualidade de um sistema de recomendação, no entanto, a escolha das mesmas deve ser efetuada de forma cuidada e alinhada com os objetivos do sistema de recomendação (Ge, Delgado-Battenfeld and Jannach, 2010).

3.9 Aplicações

Nesta secção serão apresentados alguns sistemas de recomendação.

3.9.1 Amazon

Amazon é uma empresa de comércio eletrónico, fundada por *Jeff Bezos* em 1994 (Amazon, 2019).

A *Amazon* utiliza uma abordagem baseada em produtos e o seu algoritmo é constituído por duas componentes: a *offline* e a *online*. A componente *offline* é responsável por elaborar uma tabela com os produtos semelhantes. Para calcular a semelhança entre os produtos, utilizam a semelhança do cosseno. Enquanto, que a componente *online* é responsável por procurar produtos nessa tabela. A chave do sucesso deste algoritmo está no facto da componente *offline* produzir uma tabela com os produtos semelhantes. Além de produzir recomendações fiáveis, a componente *online* escala independentemente do número de produtos e utilizadores (Linden, Smith and York, 2003)

3.9.2 Netflix

A *Netflix* foi fundada em 1997 nos Estados Unidos por *Reed Hastings*, *Marc Randolph* e *Scotts Valley*. Inicialmente, surgiu como um serviço de entrega de DVDs pelo correio e, mais tarde, evoluiu para o *streaming* de filmes e series de televisão. Á data desta dissertação, a plataforma *Netflix* está disponível em mais de 190 países (Netflix, 2018).

Quando a principal fonte de rendimento da *Netflix* era o envio de DVDs por correio, a recolha de *feedback* era feita através do sistema de classificação que ia de um a cinco, sendo que um é a classificação mais baixa e cinco a mais alta. Como esta classificação era praticamente o

único *feedback* que existia as recomendações eram feitas com base nas mesmas. Os DVDs com mais estrelas eram os que eram recomendados a outros clientes. Atualmente, a *Netflix* tem acesso a diversos dados sobre cada cliente, como por exemplo: o dispositivo onde o conteúdo foi acessado, a hora do dia, o dia da semana, como é que o conteúdo foi descoberto entre outros. A experiência adquirida juntamente com o acesso a mais dados permitiu evoluir de um sistema que se baseava apenas na classificação dos filmes para um sistema de recomendações constituído por vários algoritmos (Gomez-Uribe and Hunt, 2015)

Os mesmos autores afirmam que “se um utilizador não encontrar algo que lhe interesse após um período de 60 a 90 segundos de navegação, a probabilidade de o mesmo não aderir ao serviço aumenta consideravelmente”. Por isso, o sistema de recomendações tem de ser capaz de produzir recomendações fiáveis, de modo a captar atenção do cliente o mais rapidamente possível. A página inicial da *Netflix* foi construída a pensar exatamente no problema acima apresentado. Esta organiza os seus conteúdos por tema num layout bidimensional. Ao apresentar os conteúdos em linhas/categorias permite ao cliente rapidamente escolher aquele que mais o interessa. Normalmente, os vídeos apresentados em cada linha resultam da aplicação de apenas um algoritmo (Alvino and Basilico Justin, 2015; Gomez-Uribe and Hunt, 2015)

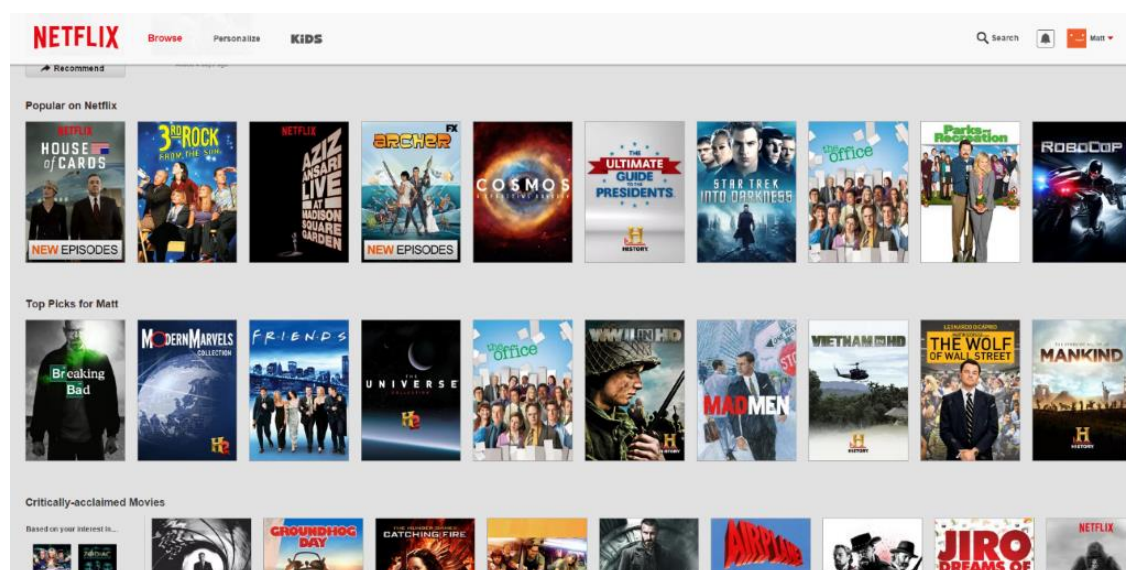


Figura 12- Página inicial da *Netflix*⁹

De seguida, serão apresentados alguns dos algoritmos utilizados na construção da página inicial.

3.9.2.1 Personalized Video Ranker

⁹ Disponível em <https://techcrunch.com/2015/03/19/fix-netflixs-user-interface-with-god-mode/?guccounter=1>. Acedido em 2019-05-12

Como foi referido, a página inicial é organizada por linhas, sendo que cada linha é uma categoria diferente. O PVR é responsável por ordenar a lista de vídeos dentro de uma categoria. Para tal, utiliza a informação do perfil do cliente para que os vídeos com mais interesse para o utilizador sejam apresentados primeiro. Desta forma, os vídeos apresentados numa categoria podem variar de utilizador para utilizador.

3.9.2.2 Top-N Video Ranker

O objetivo deste algoritmo é de encontrar os melhores vídeos para um determinado utilizador, independentemente da categoria. Combina a informação do utilizador, com popularidade, e identifica e incorpora as últimas tendências. O resultado deste algoritmo é apresentado na linha “Top Picks”

3.9.2.3 Trending Now

A *Netflix* identificou que certas categorias de vídeos têm picos de visualizações em determinados períodos com diferentes periodicidades. Para suprimir esta necessidade desenvolveu um algoritmo que utiliza esta informação e juntamente com os dados do perfil do utilizador disponibiliza conteúdo personalizado.

Este algoritmo produz recomendações para dois tipos de eventos:

- **Eventos periódicos:** eventos que se repetem de forma cíclica com intervalos de tempo diferentes, como por exemplo: O Dia dos Namorados. Este dia produz um pico de visualizações em filmes românticos.
- **Eventos esporádicos:** eventos que ocorrem de forma aleatória, como por exemplo: Furacões. Este evento produz um pico de visualizações em documentários e filmes sobre a temática.

3.9.2.4 Continue Watching

Os utilizadores da *Netflix* têm a possibilidade de visualizar os vídeos de forma faseada, isto é, ver o mesmo vídeo ao longo de várias sessões. Existem diversos motivos que levam a que um utilizador abandone a visualização de vídeo, como por exemplo: desinteresse pelo conteúdo, o conteúdo é extenso e não pode ser visualizado apenas numa sessão, entre outros. O algoritmo responsável por selecionar os vídeos desta categoria utiliza um conjunto de dados (o tempo que passou desde a última visualização, o momento em que o utilizador abandonou o vídeo, entre outros) que permite perceber quais são os filmes e séries que o utilizador pretende continuar a ver ou rever.

3.9.2.5 Video-Video Similarity

Como podemos ver na Figura 13, é apresentada uma lista de séries baseada no facto de o utilizador ter visto a série “Orange is the New Black”. O conteúdo gerado por este algoritmo não é personalizado, uma vez que gera para cada filme/série uma lista de filmes/séries similares. No entanto, o resultado apresentado na página inicial de cada utilizador é filtrado de modo a que sejam apresentados apenas as séries com interesse para o utilizador.

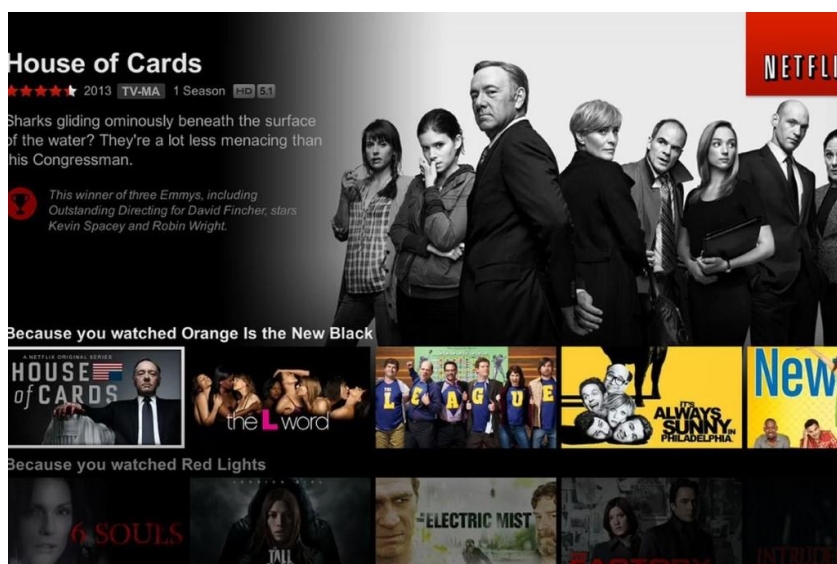


Figura 13 – Because you watched “Orange is the New Black”¹⁰

3.9.3 Youtube

O *Youtube* é a maior plataforma do mundo para criar e partilhar vídeos.

O seu sistema de recomendação é responsável por encontrar conteúdo personalizado para milhares de utilizadores.

Recomendar vídeos nesta plataforma é um grande desafio principalmente devido a três fatores:

- **Escala:** nem todos os algoritmos estão preparados para lidar a quantidade de dados disponível
- **Novidade:** como o catálogo de vídeos está constantemente a crescer, é necessário que o sistema seja flexível para enquadrar os novos vídeos nas suas recomendações.
- **Noise:** A falta de feedback e a esparsidade dos dados tornam complicado prever o comportamento do utilizador. Além disso, a meta-informação associada aos vídeos não está estruturada.

¹⁰ Disponível em <https://www.theverge.com/2013/11/13/5098224/netflix-introduces-one-unified-tv-interface-to-rule-them-all>. Acedido a 2019-05-12

Para colmatar as dificuldades referidas anteriormente, o sistema de recomendação do *Youtube* é constituído por duas redes neuronais, ver

Figura 14:

- **Geração de candidatos:** utiliza o histórico do utilizador e um subconjunto de vídeos para produzir uma lista de candidatos
- **Priorização:** classifica a lista de candidatos produzida anteriormente, apresentado ao utilizador os mais bem classificados

O facto de o sistema de recomendação ser dividido em duas etapas, permite efetuar recomendações a partir de um enorme catálogo de vídeos, sabendo que uma percentagem dos vídeos apresentados aos utilizadores são personalizados (Covington, Adams and Sargin, 2016)

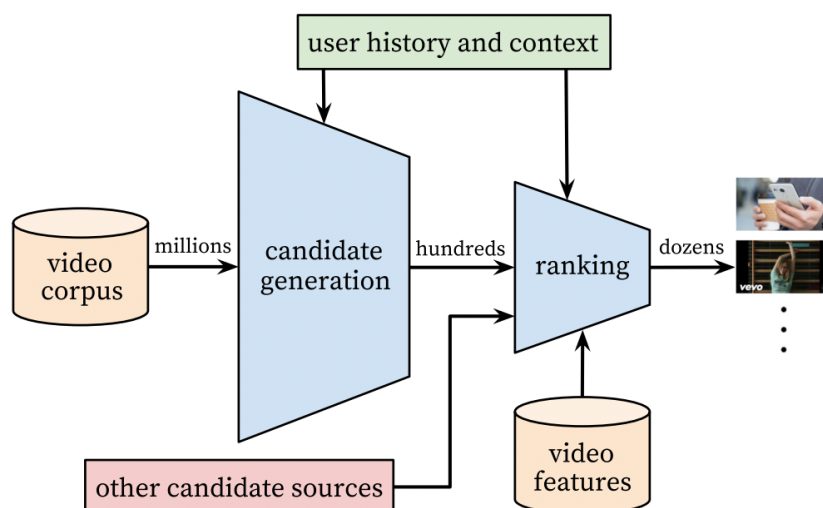


Figura 14- Visão geral da arquitetura do sistema de recomendação do youtube.¹¹

3.10 Frameworks

Na seção 2.3, foram analisadas três *frameworks* e recorrendo ao método AHP foi selecionada a *Accord.net*, para o desenvolvimento do sistema de recomendação.

¹¹ Disponível em <https://dl.acm.org/citation.cfm?doid=2959100.2959190>. Acedido em 2019-05-12

4 Design

Este capítulo irá abordar o design da solução. Iremos iniciar esta parte com uma análise dos dados disponíveis para o desenvolvimento do sistema de recomendação. Seguidamente, iremos descrever as técnicas que foram utilizadas para desenvolver o sistema supracitado e os requisitos funcionais e não funcionais dos diferentes componentes do sistema. Depois, serão apresentadas duas propostas para a arquitetura deste sistema, seguindo-se da descrição do processo de seleção utilizado para selecionar a proposta de arquitetura. De seguida, serão apresentadas duas bases de dados de tipologias diferentes, seguindo-se da descrição do processo de seleção utilizado para selecionar a base de dados utilizada para o desenvolvimento do sistema de recomendação. Seguindo-se de uma breve descrição dos conceitos de domínio. O capítulo culmina com a apresentação de alto nível de dois processos do sistema de recomendação.

4.1 Análise de dados

Os dados que serão agora analisados foram recolhidos entre 08/05/2018 a 26/04/2019 na empresa GNG.

Os dados recolhidos durante o período referido levam-nos a inferir que:

- 54% dos utilizadores comprou apenas um produto,
- 23% dos utilizadores comprou dois produtos
- 9% comprou três produtos.
- 14% dos utilizadores compraram mais de três produtos.

O site não inclui a gravação do histórico de navegação do utilizador, além disso, existem apenas dez classificações explícitas de produtos. Dado a não existência de histórico de navegação e a escassez de classificações, estes não serão utilizados para o desenvolvimento do sistema de recomendação. Apesar de não existir histórico de navegação, o site grava no *browser* uma *cookie* com os últimos dez produtos vistos por cada utilizador durante um curto período.

Também é importante realçar que a plataforma permite a compra de produtos a utilizadores não registados. Este facto impossibilita a recolha de dados demográficos, como o género, a idade e a localização do comprador.

Os autores *Kaminskas* em (*Kaminskas et al.*, 2015) e os *Chen, Miller e Dagher* em (*Chen, Miller and Dagher*, 2014), deparam-se com limitações semelhantes às apresentadas anteriormente. Face a estas limitações, ambos propuseram um sistema de recomendação orientado a produtos.

Assim, tendo em conta as limitações referidas e em congruência com os autores supracitados, o sistema de recomendação proposto irá ser orientado a produtos.

4.2 Proposta de solução

Este sistema de recomendação tem como missão produzir os N produtos mais indicados, consoante o contexto.

Para isso irá utilizar três abordagens distintas:

- **Regras associativas:** para extrair os produtos que são vendidos em conjunto.
- **Baseado em características:** para produzir uma lista de produtos semelhantes baseando se nas características dos produtos.
- **Histórico de compras:** para produzir recomendações pessoais com base no histórico de compras do utilizador

De seguida, será efetuada uma descrição mais detalhada do comportamento de cada uma das abordagens.

4.2.1 Regras associativas

Será aplicado o algoritmo *Apriori* para analisar as vendas, com o objetivo de identificar padrões, isto é, produtos que são vendidos juntos com algum grau de frequência. Este algoritmo é configurável, sendo possível definir níveis:

- **Suporte:** indica com que frequência de uma determinada regra ocorre.
- **Confiança:** calcula o número de vezes que uma determinada regra é verdadeira.

O resultado deste algoritmo será guardado numa base de dados, para que, mais tarde, possa ser utilizado para gerar recomendações.

4.2.2 Baseada em características

O cálculo da similaridade baseado em texto é uma técnica bastante utilizada em *web mining*, recuperação de informação, e em linguagem natural, pois permite calcular a semelhança entre um conjunto de documentos de texto (Kaminskas *et al.*, 2015).

Os produtos são constituídos por diversas características, com valores pré-definidos, como por exemplo, género, categoria, tipo de modelo; ou campos livres, como por exemplo descrição, composição, tamanho e corte.

Para as características com valores pré-definidos, será construída uma matriz Produto-Características, em que cada linha corresponde a um produto e a coluna a uma característica. A ausência de uma característica de um produto é indicada com 0 e existência da mesma com 1.

Para analisar as características livres, será utilizada a técnica TF-IDF.

Após a construção da matriz, e da utilização da técnica TF-IDF, o cálculo da semelhança de produtos será efetuado através da semelhança do cosseno. Para cada produto serão guardados os N produtos mais bem classificados, para que, mais tarde, possam ser utilizados para gerar recomendações.

4.2.3 Histórico de compras

Esta abordagem utiliza uma técnica colaborativa para recomendar produtos, mais concretamente, a técnica baseada em produtos, apresentada na seção 3.4.2.1.

Para implementar esta abordagem, utilizou-se outra *framework*, além da *Accord.net*. Esta nova *framework*, *NReco.Recommender*, disponibiliza vários métodos de recomendação colaborativos. Além disso, possuiu diversas implementações de medidas de similaridade. Neste caso, a medida de similaridade selecionada foi a *LogLikelihoodSimilarity*.

4.3 Requisitos funcionais e não funcionais

O presente sistema de recomendação tem de ser autossuficiente, isto é, tem de possuir toda a informação necessária para gerar recomendações. Para atingir este objetivo, os produtos, compras e stock serão importados de forma periódica para o sistema de recomendação.

A Tabela 9 apresenta os requisitos funcionais e não funcionais do sistema.

Tabela 9- Requisitos funcionais e não funcionais

Requisitos funcionais	
Identificação	Requisito
RF001	Importar produtos da plataforma de comércio eletrónico
RF002	Importar encomendas da plataforma de comércio eletrónico
RF003	Importar stock do sistema de faturação
RF004	Verificar se existe inventário
RF005	Extrair regras associativas
RF006	Recomendar produtos adquiridos em conjunto
RF007	Recomendar produtos com base no histórico de compras
RF008	Recomendar produtos semelhantes a um produto
Requisitos não funcionais	
RNF001	A API Recomendação deve gerar recomendações para cem utilizadores em paralelo num tempo inferior ou igual a dois segundos.

4.4 Propostas de Arquitetura

Nesta secção serão apresentadas duas propostas para arquitetura do sistema de recomendação.

4.4.1 Arquitetura nº 1

A Figura 15 apresenta o diagrama de componentes simplificado, demonstrando como os diferentes componentes interagem entre si. Os componentes site, base de dados comércio eletrónico e base de dados sistema de faturação fazem parte da estrutura da plataforma já existente e onde o sistema de recomendação será integrado. É importante realçar que este esquema é uma visão simplificada da estrutura da plataforma de comércio eletrónico.

O componente API Integradora será responsável por fornecer os dados necessários à geração de recomendações, tais como: informação dos produtos, encomendas e stock.

O componente API Recomendação será responsável por obter as recomendações para um determinado contexto. Além disso, deverá cruzar os produtos obtidos, com o stock disponível, apresentando, desta forma, apenas os produtos para os quais exista stock.

O componente Motor de recomendação será responsável por obter as informações dos produtos, das compras, stock, por calcular a semelhança entre os produtos e extrair as regras associativas.

O componente Base de dados será responsável por guardar toda a informação necessária para a geração de recomendações.

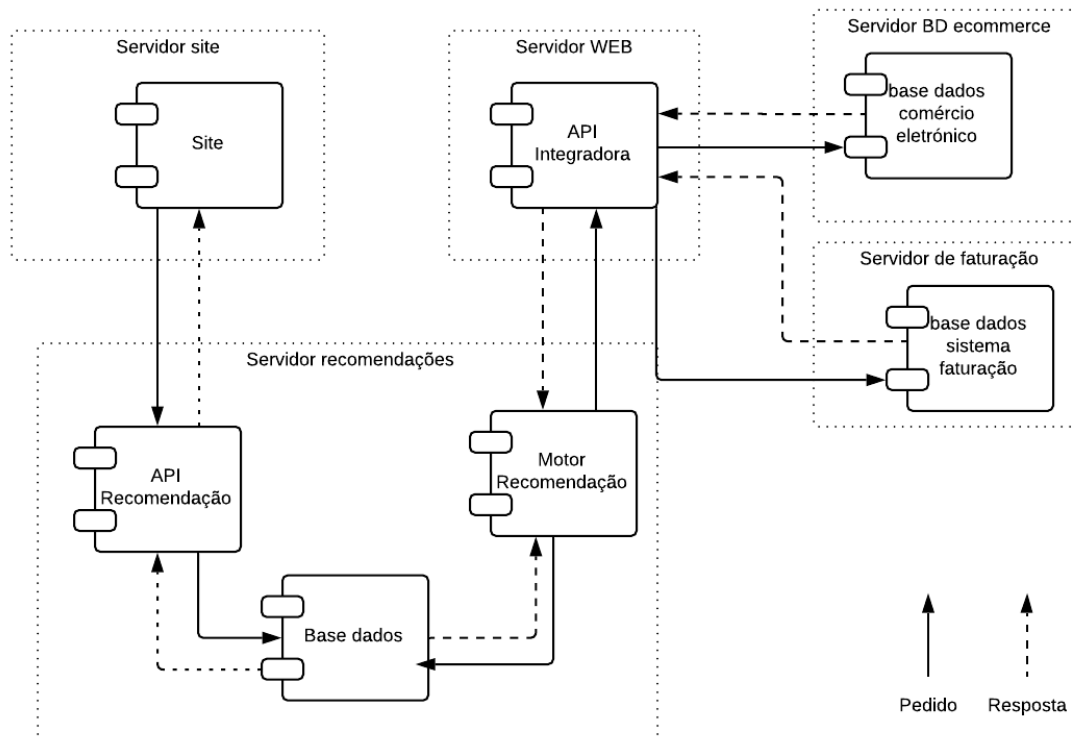


Figura 15- Proposta de arquitetura nº 1

4.4.2 Arquitetura nº 2

A Figura 16 apresenta o diagrama de componentes simplificado, demonstrando como os diferentes componentes interagem entre si. Os componentes site, base de dados comércio eletrônico e base de dados sistema de faturação fazem parte da estrutura da plataforma já existente e onde o sistema de recomendação será integrado. É importante realçar que se trata de uma visão simplificada da estrutura da plataforma de comércio eletrônico.

O componente API Recomendação será responsável por obter as recomendações para um determinado contexto. Além disso, deverá cruzar os produtos obtidos, com o stock disponível, apresentando, desta maneira, apenas os produtos para os quais existam stock.

O componente Motor de recomendação será responsável por obter as informações dos produtos, das compras e por calcular as diferentes métricas.

O componente Base de dados será responsável por guardar toda a informação necessária para a geração de recomendações.

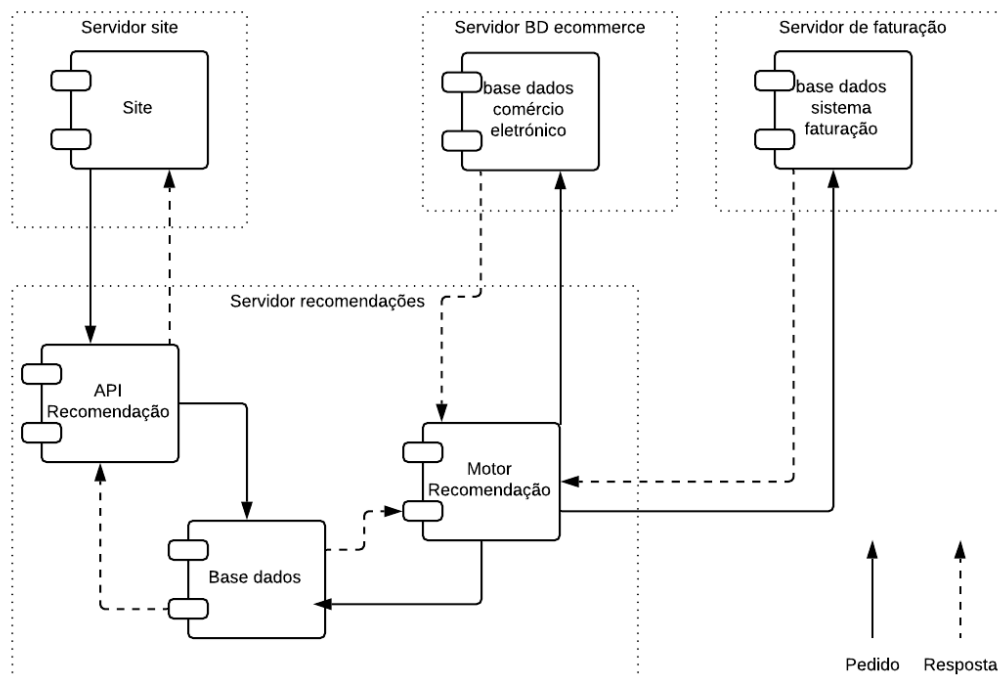


Figura 16- Proposta de arquitetura nº 2

4.4.3 Seleção da arquitetura

A única diferença entre as duas arquiteturas apresentadas é a forma como estas obtêm a informação da plataforma de comércio eletrónico

Na primeira proposta, a informação é obtida através de uma API. Desta forma, a importação dos dados é transparente. Porém, é mais lenta e existem mais componentes que exigem manutenção.

Na segunda proposta, a informação é obtida diretamente através das bases dados, pelo que, a obtenção de dados é mais rápida. Contudo, o sistema de recomendação fica a conhecer detalhes de implementação das bases dados do comércio eletrónico e de faturação.

Optou-se por implementar a segunda proposta, porque todas as bases dados mencionadas são controladas pelo autor, além disso, esta solução exige menos desenvolvimentos e é mais rápida na obtenção de dados.

4.5 Base de dados

Nesta secção será realizada uma breve descrição de dois motores de base de dados e do processo de seleção da base de dados que será utilizada.

4.5.1 Microsoft SQL Server

A *Microsoft SQL Server* é uma base de dados relacional criada pela *Microsoft* em 1989. Os dados são guardados de forma estruturada, em tabelas e cada linha possui o mesmo nº de colunas. Pode ser utilizada através de várias linguagens de programação, é escalável e permite a análise de informações em tempo real. Foi considerada pela *National Institute of Standards and Technology*, como a base de dados mais segura nos últimos sete anos (Microsoft, 2019).

4.5.2 MongoDB

O *MongoDB* é um projeto de código aberto, do tipo *NoSQL* criado em 2009, de alta performance, sem estruturas de dados pré-definidas, utilizando documentos com sintaxe JSON para a serialização dos dados. Foi concebido especialmente para o armazenamento de grandes quantidades de dados (MongoDB, 2019).

4.5.3 Seleção da base de dados

Devido a restrições temporais, não foi possível analisar todas as bases de dados existentes. Por isso, a seleção da base de dados recaiu sobre:

- **Microsoft SQL Server:** devido à experiência profissional com a mesma.
- **MongoDB:** apesar da in experiência profissional com a mesma, é uma base de dados *NoSQL* bastante conhecida e utilizada.

A base de dados tem de possuir um *driver* para C#, uma vez que será acedida através de uma aplicação e *WebAPI* desenvolvida em C#.

Uma vez que as ambas as bases de dados possuem o *driver* anteriormente referido, e são escaláveis, pode-se inferir que estas são boas alternativas para o desenvolvimento do sistema de recomendação. No entanto, dado a experiência profissional e a cultura da empresa optou-se por seleccionar o *Microsoft SQL Server*.

4.6 Conceitos do domínio

Nesta secção será realizada uma breve apresentação dos conceitos de negócio subjacentes ao problema. Inicialmente, será efetuada uma breve descrição do modelo de dados, seguido do diagrama de classes responsável pela geração das recomendações.

4.6.1 Modelo de dados

A proposta de solução apresentada na seção 4.2 é constituída por três abordagens diferentes, cada uma com necessidades diferentes. De seguida, serão descritas as entidades e os respetivos relacionamentos criados para suportar as necessidades da proposta de solução.

O sistema de recomendação terá diversas configurações, para tal, criou-se a entidade *Configurations*. Esta é responsável por guardar as diferentes configurações do sistema, como por exemplo os parâmetros utilizados pelo algoritmo *Apriori*.

Uma das abordagens apresentadas na secção 4.2 utiliza as características dos produtos para encontrar outros produtos semelhantes. Para suportar este algoritmo foi necessário criar as seguintes entidades:

- *Attributes*: representa os atributos que existem para os produtos
- *Models*: representa os produtos do sistema
- *ModelAttributes*: representa os atributos dos produtos do sistema
- *SimilarModels*: representa os N produtos mais semelhantes

Além de guardar os produtos, é necessário que o sistema armazene as diferentes compras dos clientes. Para isso, foram criadas duas entidades, *Purchases* que é responsável por armazenar os cabeçalhos compras dos clientes e a entidade *PurchaseLines* responsável por armazenar as linhas das compras.

Para armazenar as regras extraídas pelo algoritmo *Apriori*, foram criadas duas entidades. A entidade *Rules* representa o cabeçalho das regras extraídas e a entidade *RuleItems* responsável por armazenar os produtos que fazem parte das regras.

Na Figura 17 podemos ver a representação do modelo de dados utilizado neste projeto

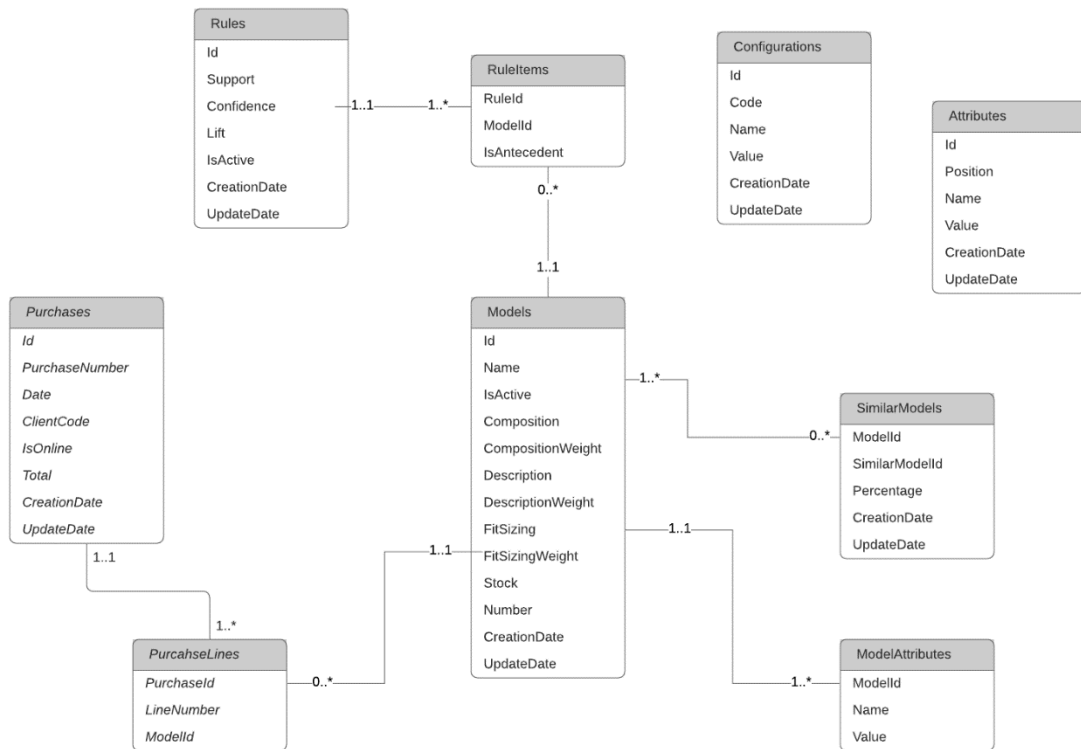


Figura 17- Diagrama da base de dados

4.6.2 Recomendações

Na Figura 18, podemos ver o diagrama de classes que suporta o processo de geração de recomendações.

A interface *IRecommendationsService* é responsável por fornecer uma camada de abstração, entre as diferentes implementações dos algoritmos de recomendação. Assim sendo, todas as abordagens do sistema devem implementar a interface mencionada, definindo, em cada uma delas, a lógica inerente à sua abordagem.

As opções relacionadas com a base de dados são efetuadas através da interface *IDatabaseContext*. Esta é responsável por efetuar a ponte entre as entidades e a base de dados, permitindo a pesquisa, inserção e atualização das mesmas na base de dados.

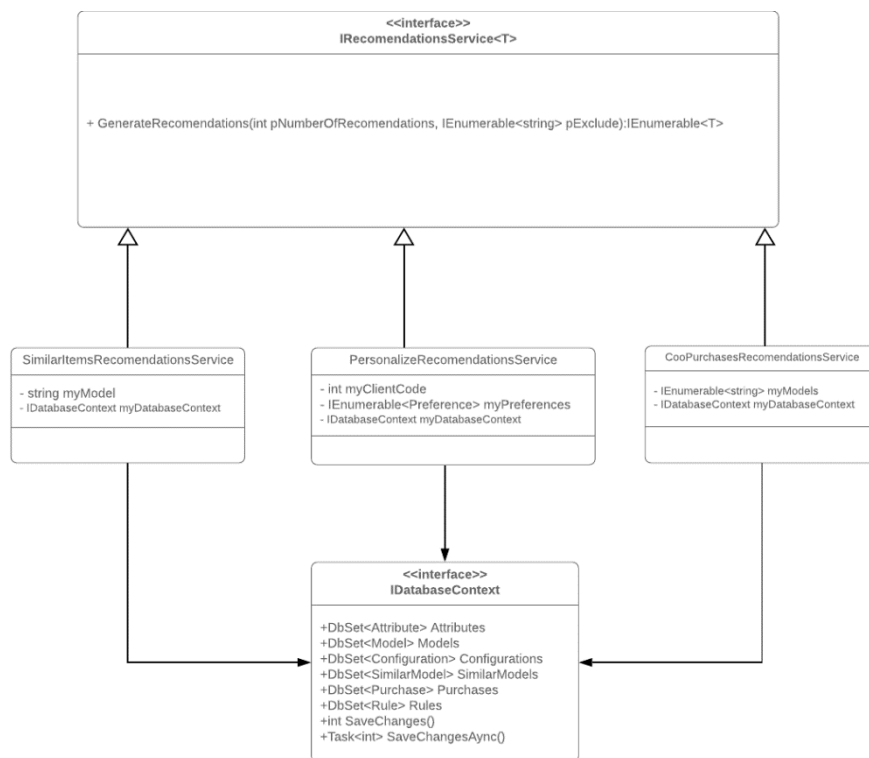


Figura 18- Diagrama de classes para a geração de recomendações

4.7 Processos de recomendação

Esta secção irá apresentar uma visão de alto nível, de dois dos vários processos do sistema de recomendação.

4.7.1 Processo de importação de encomendas

O processo de importação de encomendas será realizado de forma periódica.

A Figura 19 apresenta a proposta para a sequência de pedidos necessários para a importação de produtos.

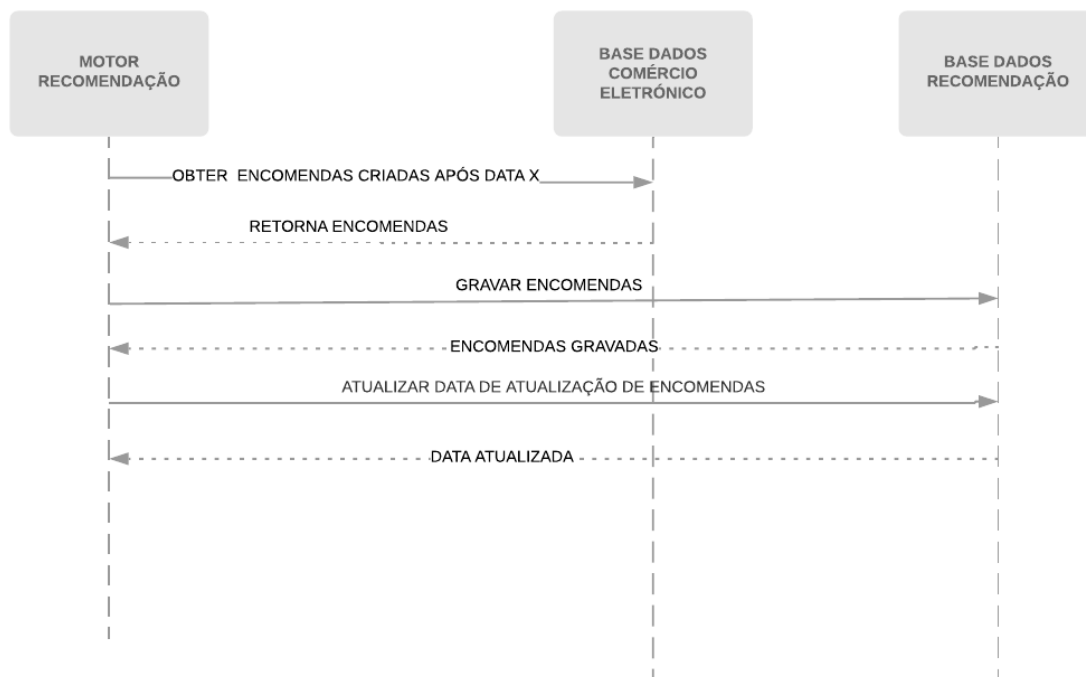


Figura 19- Diagrama de sequência da importação de encomendas

O Motor de recomendação pede à base de dados da plataforma de comércio eletrônico todas as encomendas efetuadas depois de uma determinada data e a base de dados retorna as encomendas encontradas. O Motor de recomendação grava as novas encomendas, atualizando, de seguida, a data da última importação.

4.7.2 Processo de recomendação para um produto

A Figura 20 apresenta a proposta para a sequência de pedidos necessários para a geração de recomendações.

O pedido é iniciado pelo site que pede à API de Recomendação para recomendações para o produto X. Esta efetua uma pesquisa na base de dados e retorna os produtos encontrados. De seguida, verifica se existe stock dos produtos, retornando apenas aqueles que possuem stock

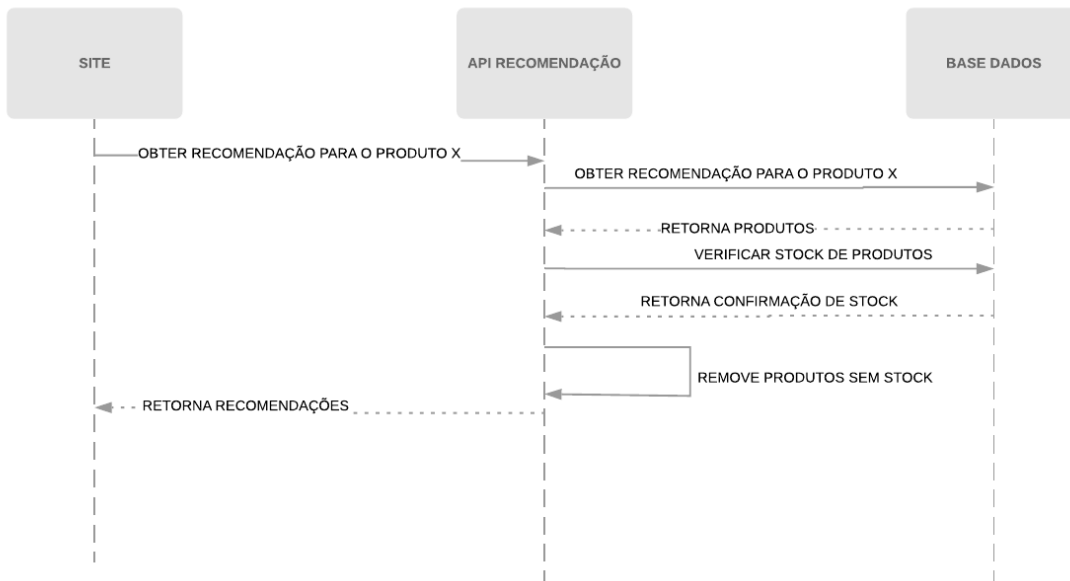


Figura 20- Diagrama de sequência da recomendação de um produto

5 Implementação

Neste capítulo, são apresentados os detalhes de desenvolvimento da solução descrita anteriormente. Inicialmente, é apresentada a plataforma utilizada para o desenvolvimento em si. Seguindo-se da descrição do desenvolvimento de cada componente.

5.1 Azure DevOps

Azure DevOps é uma plataforma que possui diversos serviços que facilitam o desenvolvimento de software:

- **Azure Boards:** permite o planeamento das tarefas.
- **Azure repo:** permite o versionamento do código fonte.
- Entre outros.

Esta plataforma foi escolhida para o desenvolvimento do sistema de recomendação, devido à experiência prévia com a mesma.

O serviço *Azure Boards* foi utilizado para planear o desenvolvimento das tarefas. A Figura 21 apresenta algumas das tarefas realizadas durante o desenvolvimento do sistema de recomendação

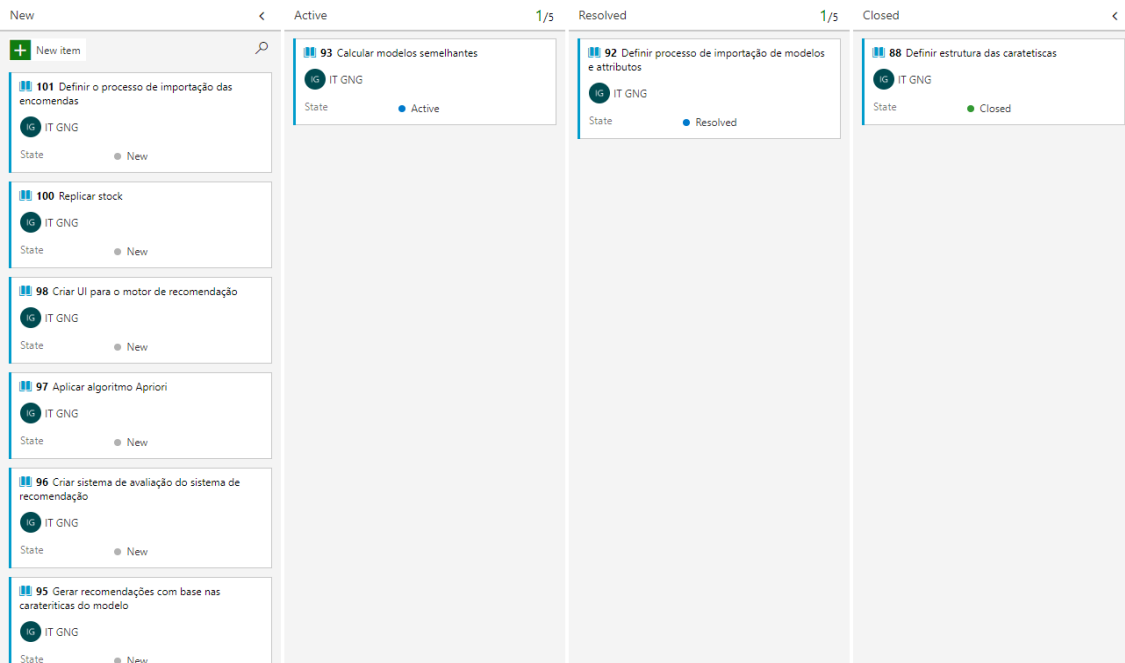


Figura 21- Azure Boards

O serviço *Azure repo* foi utilizado para efetuar o versionamento do código fonte. Foram criados dois branch o *master* e o *dev*.

Os tickets eram desenvolvidos no *branch dev*, e após o desenvolvimento eram submetidos a um conjunto de testes, caso passassem era feito o *merge* para o *master*, caso contrário, eram corrigidos os erros.

5.2 Componentes implementados

Foi implementada a proposta de solução definida na secção 4.2, com a arquitetura descrita na secção 4.4, mais concretamente, a Arquitetura nº 2. Assim sendo, o sistema foi dividido em dois componentes, cada um com uma responsabilidade específica.

De seguida, será feita uma breve descrição sobre a implementação do modelo de dados, e do processo de desenvolvimento de cada um dos componentes.

5.2.1 Base de dados

Como referido na secção 4.5.3 a base de dados selecionada para o desenvolvimento do sistema de recomendação foi a *Microsoft SQL SERVER*.

O modelo de dados implementado foi o descrito na secção 4.6.1 e decorreu sem problemas.

Como o sistema tem de armazenar dados sensíveis, foi necessário recorrer à encriptação dos mesmos. Para tal, usou-se o mecanismo de encriptação, *Always Encrypted*, oferecido pela base de dados.

Este mecanismo foi desenhado para proteger dados sensíveis, sem nunca relevar as chaves de encriptação para a base de dados.

Existem dois tipos de chaves:

- **Chaves encriptação de coluna:** são utilizadas para encriptar os dados da base de dados
- **Chave mestre:** são utilizadas para encriptar as chaves de encriptação de coluna. São armazenadas em cofres externos à base de dados, como por exemplo: o gestor de certificados do *Windows*.

Este mecanismo de encriptação é totalmente transparente para as aplicações, além disso, mesmo em memória, os dados estão encriptados, protegendo-os, desta forma, de ataques que envolvam análise dos dados em memória da base de dados (Kupcik, 2018).

5.2.2 Motor de recomendação

O Motor de recomendação foi desenvolvido utilizando a linguagem de programação C#.

Este é responsável por:

- Importar os dados da plataforma de comércio eletrónico e faturação (automaticamente e manual)
- Construir a matriz Produto-Caraterísticas
- Calcular o grau de semelhança entre os produtos
- Extrair as regras associativas

O processo de importação de dados ocorre de forma periódica. O primeiro processo a ser executado é o de produtos, seguindo-se do stock e de encomendas. Estes dois últimos processos são executados, depois da importação de produtos, porque dependem do resultado deste. Para reduzir a duração do processo de importação de dados, a importação do stock e das encomendas é realizado em paralelo.

O agendamento/montagem dos processos de importação é realizado no arranque do programa. Implementou-se o padrão observador, para permitir a atualização da interface gráfica à medida que os diferentes processos de importação são executados. No final de cada importação, é enviado um email com o relatório da importação.

Na Figura 22, podemos ver um, diagrama de classes, que apresenta os diferentes intervenientes no processo de importação de dados.

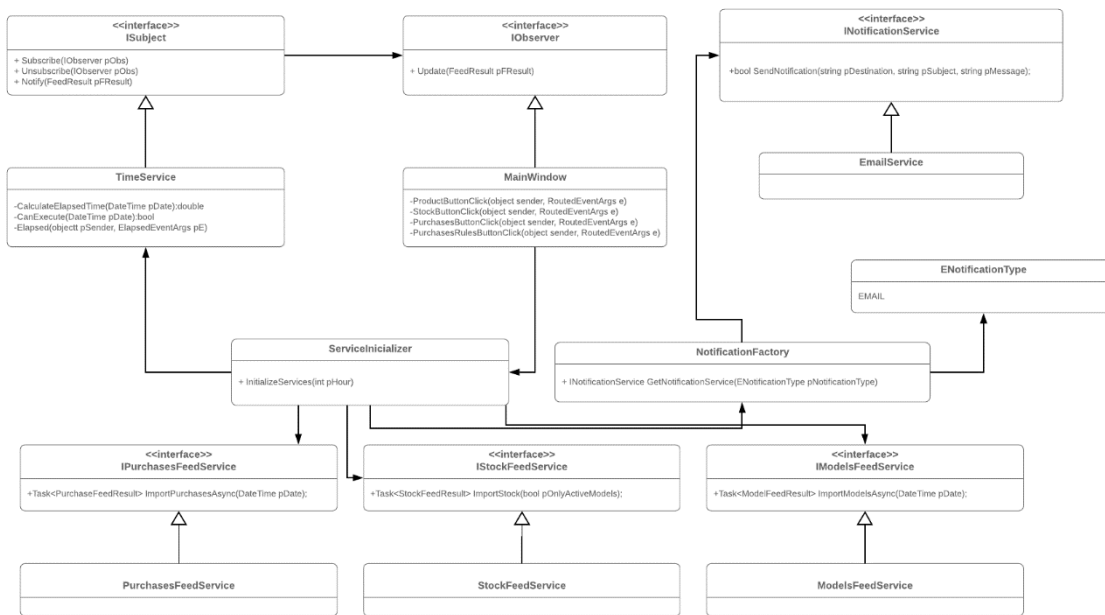


Figura 22 – Diagrama de classes do agendamento dos processos

Nesta fase, a extração das regras é feita de forma manual, no entanto, tal como os outros processos, é possível a sua automatização.

A componente gráfica foi desenvolvida em *Windows Presentation Foundation*, dado a experiência prévia com esta tecnologia.

A Figura 23 apresenta a interface que permite gerir a importação de dados entre os dois sistemas. Nesta interface, conseguimos verificar a última vez em que os processos de importação foram executados, bem como iniciar uma nova importação e executar a extração das regras associativas.

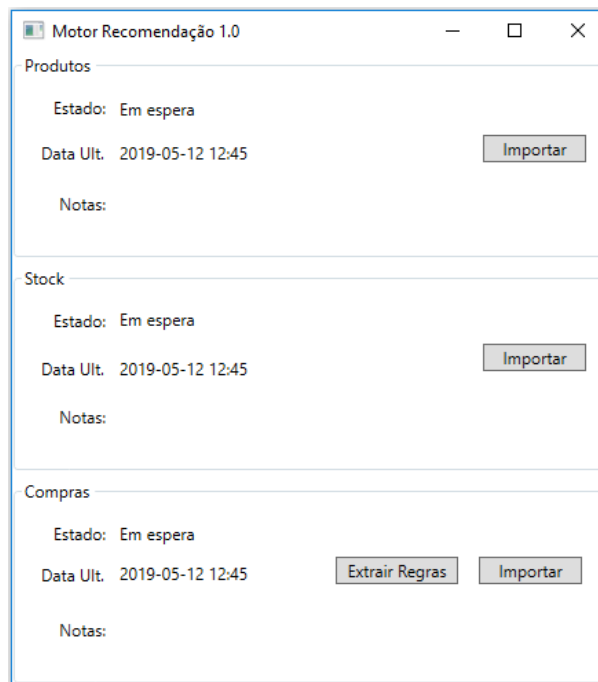


Figura 23- Interface do motor de recomendação

5.2.3 API de Recomendação

A API de Recomendação foi desenvolvida utilizando a linguagem de programação C#

A API disponibiliza três *endpoints*, permitindo, desta forma, a obtenção de recomendações para três cenários.

Os *endpoints* criados foram os seguintes:

- **SimilarModels:** responsável por obter os produtos semelhantes a um determinado produto.
- **CooPurchases:** responsável por analisar os produtos de um carrinho, e por verificar se existe alguma combinação de produtos que esteja presente nas regras extraídas pelo algoritmo *Apriori*.
- **Itembased:** responsável por obter recomendações com base no histórico de compras do cliente

Para facilitar a utilização da API, foi documentada utilizando *Swagger*. *Swagger* permite descrever a estrutura de uma API. Este processo pode ser efetuado manualmente ou automaticamente. Além disso, permite simular pedidos à API de forma simples, rápida e sem necessidade de programar (SMARTBEAR, no date).

Na Figura 24 podemos ver um *printscreen* da documentação gerada pelo *Swagger* para a API desenvolvida.

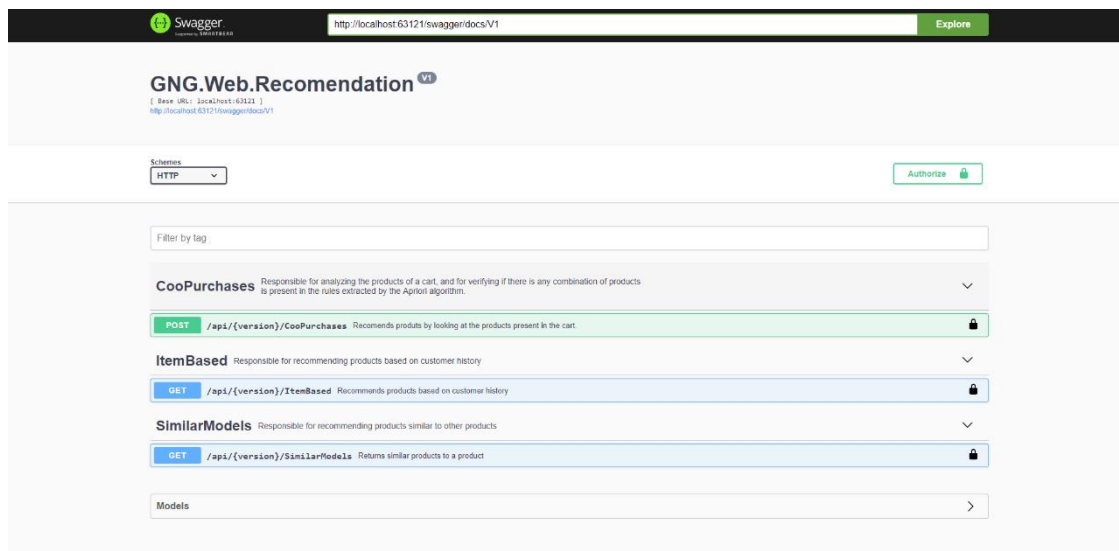


Figura 24- Documentação gerada pelo *Swagger*

Por razões de segurança sempre que é feito um pedido à API, é necessário enviar no cabeçalho do pedido uma chave. Esta é validada pela API, e se for válida, é executado o código referente ao *endpoint*, caso contrário, é retornado um erro a indicar a invalidade da chave.

6 Avaliação

Este capítulo tem como objetivo avaliar a qualidade do sistema de recomendação desenvolvido. Primeiramente, será definida a metodologia utilizada para sua avaliação. De seguida, serão definidas as hipóteses que se pretende comprovar. Seguindo-se da apresentação dos resultados e das respetivas conclusões. No final, é realizada a apreciação global do trabalho realizado, bem como a apresentação das limitações da solução e do trabalho que se pretende realizar no futuro.

6.1 Definição da metodologia de avaliação

Para analisar a performance da API de Recomendação será utilizada a aplicação *JMeter*, versão 5.1.1. Esta análise pretende provar que o requisito não funcional definido na secção 4.3 é cumprido. Além disso, pretende descobrir quando é que o sistema deixa de conseguir processar pedidos, no tempo definido na secção 4.3.

O plano de testes será constituído por quatro grupos. Cada grupo (*ThreadGroup*) será constituído por três pedidos, um para cada um dos *endpoints* criados. Os *curl* dos pedidos podem ser consultados no anexo B.

Cada grupo terá a seguinte configuração:

- **Nº utilizadores:** o primeiro grupo terá 100 utilizadores, os seguintes, terão um incremento de 100 utilizadores até ao máximo de 400 utilizadores
- **Ramp-Up:** 10 segundos
- **Duração:** 60 segundos

Cada grupo será executado sequencialmente, com um intervalo de 60 segundos, entre cada execução. Para cada um dos grupos serão registadas as seguintes métricas: média, tempo mínimo e o máximo de resposta e a taxa de transferência¹². No final, as métricas referidas serão analisadas para verificar se o requisito não funcional definido na secção 4.3 foi cumprido e para saber qual é o ponto em que o sistema não consegue responder aos pedidos, num tempo igual ou inferior a dois segundos.

O sistema de recomendação proposto apresenta três abordagens distintas. Dada a particularidade de cada uma das abordagens serão definidas metodologias de avaliação distintas.

À data desta dissertação, a plataforma de comércio eletrónico não possui um sistema de recomendação, por isso, foram selecionados os dez produtos mais vendidos. Estes produtos constituíram a *baseline*.

A avaliação abordagem BC será efetuada da seguinte forma: serão selecionados aleatoriamente dez produtos do catálogo de produtos. Para cada um dos produtos selecionados, o algoritmo em análise terá de produzir N produtos semelhantes. O resultado do algoritmo será comparado com a lista real de produtos semelhantes. De seguida, serão calculadas as métricas: precisão, *recall* e a *fmeasure* (para mais informações sobre as métricas, consultar secção 3.8). Os resultados serão comparados com os da *baseline*.

O processo de validação da abordagem com regras associativas será efetuado em três passos. O primeiro passo será a extração das regras, através da execução do algoritmo *Apriori*, para isso, será utilizado 80% do *dataset*. O segundo passo consistirá na procura da exatidão ótima; esta será alcançada através da experimentação de diferentes valores para as variáveis de confiança e suporte, nos restantes 20% do *dataset*. Por último, a exatidão desta abordagem será comparada com a da *baseline*.

A abordagem, com base no histórico de compras, será avaliada utilizando o método *hold-out*, 80% do *dataset* será utilizado para treino, sendo que os restantes 20% serão utilizados para teste. O *dataset* utilizado será constituído apenas pelos utilizadores que comparam dez ou mais produtos. No final, os resultados obtidos, precisão, *recall* e *fmeasure*, serão comparados com os resultados da *baseline*.

6.2 Definição das hipóteses

Após identificada a metodologia de avaliação do sistema, e tendo em conta os requisitos definidos na secção 4.3, foi possível definir as seguintes hipóteses:

- A API Recomendação deve gerar recomendações para cem utilizadores em paralelo num tempo inferior ou igual a dois segundos.
- A *fmeasure* da abordagem BC deve ser superior à da *baseline*.

¹² Taxa transferência é a quantidade de dados que é processado num determinado período

- A exatidão da abordagem RA deve ser superior à da *baseline*
- A fmeasure da abordagem com base no histórico deve ser superior à da *baseline*

6.3 Avaliação dos resultados

Esta secção apresenta os resultados obtidos nesta dissertação, demonstrando, desta forma, que as hipóteses definidas anteriormente foram comprovadas.

6.3.1 Avaliação processo de integração de produtos

Para garantir a integridade do processo de importação de produtos foram realizados testes de integração, que serão apresentados de seguida.

O caso de teste “Introduzir uma data com produtos para importar” pretende garantir que caso existam produtos a importar, estes são importados corretamente e que a data da última importação é atualizada.

Com podemos ver na Tabela 10, o teste de integração passou com sucesso.

Tabela 10- Resultado do caso de teste “Introduzir uma data com produtos para importar”

Caso de teste: Introduzir uma data com produtos para importar	
Resultado esperado	Resultado obtido
Importação dos dados e atualização da data da última importação	Importação dos dados e atualização da data da última importação

O caso de teste “Introduzir uma data sem produtos para importar” pretende garantir, que caso não existam produtos a importar, que a data da última importação é atualizada.

Com podemos ver na Tabela 11, o teste de integração passou com sucesso.

Tabela 11- Resultado do caso de teste “Introduzir uma data sem dados para importar”

Caso de teste: Introduzir uma data sem produtos para importar	
Resultado esperado	Resultado obtido
Importação dos dados e atualização da data da última importação	Importação dos dados e atualização da data da última importação

O caso de teste “Falha na ligação à base de dados no processo de importação de produtos” pretende garantir que, caso ocorra um problema com a ligação SQL, o sistema está preparado

para lidar com a mesma. Para simular a falha à base de dados antes da execução do teste a base de dados foi colocada em modo *offline*.

Com podemos ver na Tabela 12, o teste de integração passou com sucesso.

Tabela 12- Resultado do caso de teste “Falha na ligação à base de dados no processo de importação de produtos”

Caso de teste: Falha na ligação à base de dados no processo de importação de produtos	
Resultado esperado	Resultado obtido
Erro na ligação à base de dados	Erro na ligação à base de dado

6.3.2 Avaliação processo de integração de compras

Para garantir a integridade do processo de importação de compras foram realizados testes de integração, que serão apresentados em seguida.

O caso de teste “Introduzir uma data com compras para importar” pretende garantir, que caso existam compras a importar, estas são importadas corretamente e que a data da última importação é atualizada.

Com podemos ver na Tabela 13, o teste de integração passou com sucesso.

Tabela 13- Resultado do caso de teste “Introduzir uma data com compras para importar”

Caso de teste: Introduzir uma data com compras para importar	
Resultado esperado	Resultado obtido
Importação dos dados e atualização da data da última importação	Importação dos dados e atualização da data da última importação

O caso de teste “Introduzir uma data sem compras para importar” pretende garantir, que caso não existam compras a importar, que a data da última importação é atualizada.

Com podemos ver na Tabela 14, o teste de integração passou com sucesso.

Tabela 14- Resultado do caso de teste “Introduzir uma data sem compras para importar”

Caso de teste: Introduzir uma data sem compras para importar	
Resultado esperado	Resultado obtido
Importação dos dados e atualização da data da	Importação dos dados e atualização da data da

última importação	última importação
-------------------	-------------------

O caso de teste “Falha na ligação à base de dados no processo de importação de compras” pretende garantir que, caso ocorra um problema com a ligação SQL, o sistema está preparado para lidar com a mesma. Para simular a falha à base de dados antes da execução do teste a base de dados foi colocada em modo *offline*.

Como podemos ver na Tabela 15, o teste de integração passou com sucesso.

Tabela 15- Resultado do caso de teste “Falha na ligação à base de dados no processo de importação de compras”

Caso de teste: Falha na ligação à base de dados no processo de importação de compras	
Resultado esperado	Resultado obtido
Erro na ligação à base de dados	Erro na ligação à base de dado

6.3.3 Avaliação processo de integração de stock

Para garantir a integridade do processo de importação de stock foram realizados testes de integração, que serão apresentados em seguida.

O caso de teste “Importar o stock” pretende garantir que o processo de importação de stock ocorre sem problemas e que a data da última importação é atualizada

Com podemos ver na Tabela 16, o teste de integração passou com sucesso.

Tabela 16 - Resultado do caso de teste “Importar o stock”

Caso de teste: Importar o stock	
Resultado esperado	Resultado obtido
Atualização do stock dos produtos e atualização da última data de importação de stock	Atualização do stock dos produtos e atualização da última data de importação de stock

O caso de teste “Falha na ligação à base de dados no processo de importação de stock” pretende garantir que, caso ocorra um problema com a ligação SQL, o sistema está preparado para lidar com a mesma. Para simular a falha à base de dados antes da execução do teste a base de dados foi colocada em modo *offline*.

Como podemos ver na Tabela 17, o teste de integração passou com sucesso.

Tabela 17 - Resultado do caso de teste “Falha na ligação à base de dados no processo de importação de stock”

Caso de teste: Falha na ligação à base de dados no processo de importação de stock	
Resultado esperado	Resultado obtido
Erro na ligação à base de dados	Erro na ligação à base de dado

6.3.4 Avaliação performance da API Recomendação

A avaliação da performance da API seguiu o plano definida na secção 6.1.

Os testes de performance foram realizados, numa máquina virtual, alojada na PT, com as seguintes características:

- Dois processadores *Intel® Xeon® Processor E5-2630 v4*
- Oito gigas de memória *ram*
- Sistema operativo Windows Server 2012 R2
- IIS na versão 8.5

Tabela 18- Resultados da avaliação da performance da API

Nº Utilizadores	Nº pedidos	Média (ms)	Mínimo (ms)	Máximo (ms)	Taxa de transferência
100	23871	231	22	405	395,96
200	24231	455	23	729	401,03
300	24291	684	22	961	399,58
400	24389	912	23	3124	396,46

Para o primeiro grupo, foram executados 23.871 pedidos, com um tempo médio de resposta de 231 milissegundos. O tempo de resposta mais curto foi de 22 milissegundos e o máximo de 405 milissegundos. A taxa de transferência foi de 395,96 pedidos por segundo.

Para o segundo grupo, foram executados 24.231 pedidos, com um tempo médio de resposta de 462 milissegundos. O tempo de resposta mais curto foi de 23 milissegundos e o máximo de 729 milissegundos. A taxa de transferência foi de 401,03 pedidos por segundo.

Para o terceiro grupo, foram executados 24.291 pedidos, com um tempo médio de resposta de 684 milissegundos. O tempo de resposta mais curto foi de 22 milissegundos e o máximo de 926 milissegundos. A taxa de transferência foi de 399,58 pedidos por segundo.

Para o quarto grupo, foram executados 24.389 pedidos, com um tempo médio de resposta de 912 milissegundos. O tempo de resposta mais curto foi de 23 milissegundos e o máximo de 3124 milissegundos. A taxa de transferência foi de 396,46 pedidos por segundo.

Analisando os dados, do primeiro grupo, conclui-se que a hipótese “A API Recomendação deve gerar recomendações para cem utilizadores em paralelo num tempo inferior ou igual a dois segundos.” foi comprovada.

Como esperado, ao aumentar o nº de utilizadores, o tempo médio de reposta aumentou. O mínimo tempo de reposta não variou muito nos quatros grupos, manteve-se entre os 22 e 23 milissegundos, no entanto, o máximo tempo de reposta, obteve uma variação superior, variou entre 405 e 3124 milissegundos. Sendo que o primeiro grupo obteve o menor máximo tempo de resposta, e o quarto grupo que obteve o maior máximo tempo de resposta. Apesar do máximo tempo de resposta do quarto grupo ser superior a 2 segundos, o tempo médio de resposta é inferior a 1 segundo. Na Tabela 18, podemos ver o resumo das métricas obtidas, durante a avaliação da performance da API.

6.3.5 Avaliação da abordagem baseada em características

Para descobrir a melhor combinação de pesos, para as componentes: Atributos, Composição, Descrição e Tamanho e Corte, variou-se cada componente 0,1. Na procura da melhor combinação, foram impostas duas restrições, a soma das quatro componentes não podia ultrapassar o valor 1 e cada componente teria de ser maior ou igual a 0,1.

Na sequência do processo anteriormente referido chegou-se à seguinte melhor combinação de pesos:

- Atributos: 0,6
- Composição: 0,2
- Descrição: 0,1
- Tamanho e Corte: 0,1

Na Tabela 19, estão representados os resultados obtidos para cinco produtos semelhantes da abordagem BC e da *baseline*

Tabela 19 - Resultados das abordagens BC e baseline para cinco produtos

Abordagem	Precisão	Recall	Fmeasure
BC	0,78	0,65	0,71
<i>Baseline</i>	0,02	0,01	0,01

Na Tabela 20, estão representados os resultados obtidos para dez produtos semelhantes da abordagem BC e da *baseline*:

Tabela 20 - Resultados das abordagens BC e baseline para dez produtos

Abordagem	Precisão	Recall	Fmeasure
BC	0,51	0,82	0,63
Baseline	0,01	0,01	0,01

Como esperado, a abordagem BC apresentou melhores resultados que a *baseline*, isto é, o valor da *fmeasure* é superior à da *baseline* em ambas as experiências, por isso, podemos concluir que a BC é melhor do que a *baseline* para identificar produtos semelhantes.

Desta forma, conclui-se que a hipótese “A *fmeasure* da abordagem BC deve ser superior à da *baseline*.” foi comprovada.

6.3.6 Avaliação da abordagem regras associativas

O algoritmo *Apriori* foi executado quatro vezes com diferentes configurações, para descobrir qual a configuração ótima. Para descobrir a configuração ótima, variou-se o suporte dois a cinco, mantendo o nível de confiança estável nos 0,5.

Para suporte igual a 2 foram geradas 601 regras em 95 minutos. Para um suporte igual a 3 foram geradas 51 regras em 70 minutos. Para um suporte igual a 4 foram geradas 9 regras em 62 minutos. Finalmente, para um suporte de 5 foram geradas 3 regras em 55 minutos.

Como podemos ver pela Figura 25, existe uma descida de seis pontos percentuais na exatidão quando, passamos de suporte dois para suporte três. Quando aumentamos o suporte para quatro, existe uma subida de vinte pontos percentuais, mantendo-se estável quando aumentamos o suporte para cinco. Apesar da exatidão ser igual, o nº de regras geradas para o nível de suporte quatro é o triplo do nº regras geradas para o nível de suporte cinco.

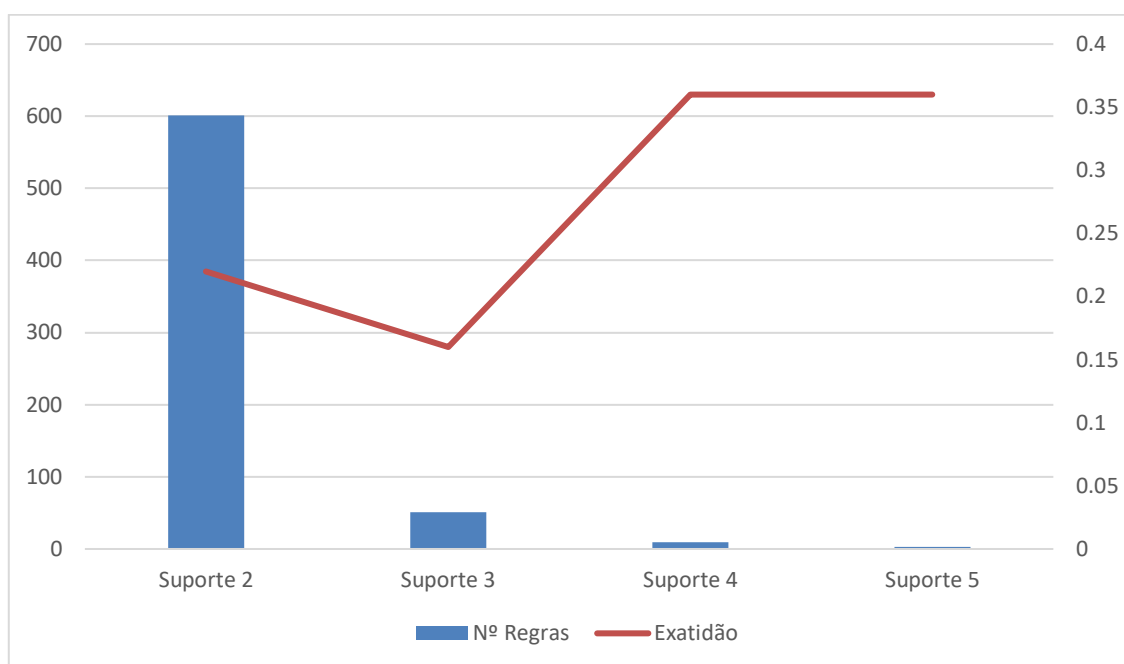


Figura 25 – Resultados da variação do suporte

De seguida, foi feita uma análise à exatidão das regras associativas na perspetiva da componente de confiança. A confiança mínima foi definida em 0,5 e para cada iteração aumentou-se 0,1, mantendo o suporte igual 2.

Como podemos ver pela Figura 26, para o nível de confiança igual 0,5 existem, seiscentas e uma regras com uma exatidão de 0,22. Ao aumentar o nível de confiança para 0,6, a exatidão aumenta para 0,25 e o nº de regras desceu para duzentas e oitenta. Ao aumentar para 0,7, a exatidão aumenta para 0,32 e o nº de regras desce para 156. Ao aumentar para 0,8, a exatidão aumenta para 0,33 e o nº de regras desce para 147. Os níveis de confiança 0,9 e 1 possuem a mesma exatidão e nº de regras.

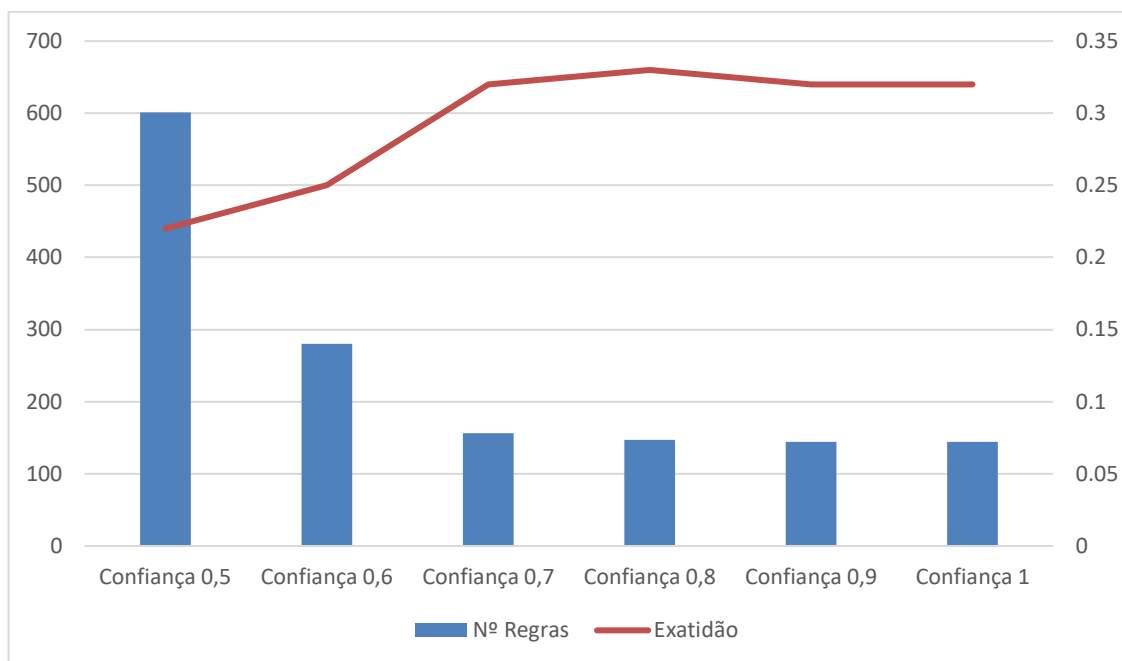


Figura 26 - Resultados da variação da confiança

Comparando os resultados das duas experiências, podemos concluir que o melhor resultado, mantendo um equilíbrio entre o nº de regras e a exatidão, é quando o nível de confiança é 0,7 e o de suporte de 2.

Como podemos ver na Tabela 21, a abordagem RA conseguiu obter melhores resultados que a *baseline*. A diferença entre a abordagem RA e a *baseline* é de onze pontos percentuais. Desta forma, conclui-se que a hipótese “A exatidão da abordagem RA deve ser superior à da *baseline*” foi comprovada.

Tabela 21- Comparação dos resultados da abordagem RA e a *baseline*

Abordagem	Exatidão
RA	0,32
<i>Baseline</i>	0,21

6.3.7 Avaliação da abordagem com base no histórico de compras

Como podemos ver na Tabela 22 a *fmeasure* da abordagem histórico de compras é de dezassete por cento, sendo desta forma superior à *baseline* em quatorze pontos percentuais. Desta forma, conclui-se que a hipótese “A *fmeasure* da abordagem com base no histórico deve ser superior à da *baseline*” foi comprovada.

Tabela 22 – Comparação dos resultados da abordagem com base no histórico de compras e a *baseline*

Abordagem	Precisão	Recall	Fmeasure
Histórico de compras	0,17	0,17	0,17
<i>Baseline</i>	0,02	0,06	0,03

7 Conclusão

Neste capítulo serão apresentadas as conclusões da dissertação. Inicialmente, é realizada a apreciação de todo o trabalho realizado. De seguida, são apresentadas as limitações do trabalho realizado, bem como o trabalho a realizar-se no futuro.

7.1 Apreciação final

No início da presente dissertação, foram definidos os objetivos que se pretendiam alcançar. Na Tabela 23, podemos ver os objetivos definidos, bem como a indicação se o mesmo foi cumprido.

Tabela 23- Conclusão dos objetivos propostos

Objetivo	Cumprido
Investigação do estado da arte de sistemas de recomendação	Sim
Desenvolvimento de um sistema de importação de dados	Sim
Desenvolvimento de uma API capaz de fornecer recomendações	Sim
Avaliação do sistema de recomendação	Sim

Nesta dissertação foi descrito o papel dos sistemas de recomendação como mecanismos de recomendação de produtos a utilizadores, partindo do seu histórico de ações, como por exemplo compras e ou classificações. Além disso, reforçou-se a importância destes sistemas no contexto do comércio eletrónico, através de uma análise de valor.

Depois, fez-se uma breve descrição do tipo de arquiteturas utilizadas no desenvolvimento de plataformas de comércio eletrónico, seguindo-se das diferentes formas de obter o *feedback* dos utilizadores, crucial para o bom funcionamento dos sistemas recomendação. Foram apresentadas as diferentes técnicas utilizadas para recomendar produtos, bem como as

respetivas vantagens e desvantagens. Além disso, apresentou-se os diferentes métodos e métricas utilizados na avaliação de sistemas de recomendação. No final, foram apresentados três sistemas de recomendação de plataformas de referência nas respetivas áreas.

Após a análise das diferentes técnicas de recomendação, foi feita uma análise aos dados disponíveis para a construção do sistema de recomendação. Dado que a plataforma de comércio eletrónico era recente, os dados sobre as compras eram escassos, por isso, houve a necessidade de combinar os dados desta plataforma com os dados das lojas físicas. Após esta combinação, o volume de dados aumentou, permitindo, desta forma, desenhar a solução do sistema de recomendação.

Após a análise dos dados, foram apresentadas duas propostas de arquitetura. As duas arquiteturas eram semelhantes, diferindo apenas na forma como os dados eram importados. Uma propunha a importação dos dados através de uma API e a outra através do SQL. Optou-se por selecionar a segunda opção, porque a sua implementação era mais rápida e, além disso, o autor tem acesso e controlo sobre os todos os sistemas apresentados.

Além disso, foram analisadas duas bases de dados, *Microsoft SQL Server* e *MongoDB*, seguindo-se do processo de seleção da base de dados a utilizar no desenvolvimento do sistema de recomendação. Optou-se por selecionar a *Microsoft SQL Server*, dado a familiaridade do autor com a mesma, além disso, é a base de dados de eleição da empresa.

O sistema de recomendação é composto por um componente que é responsável por importar os dados, das compras, produtos e o inventário, da plataforma de comércio eletrónico e do sistema de faturação; e por uma API responsável por fornecer as recomendações aos utilizadores.

Os resultados obtidos com o sistema de recomendação foram positivos, todos os testes de integração passaram, os três algoritmos implementados obtiveram uma taxa de sucesso superior à da *baseline*, além disso, os testes de performance realizados à API foram melhores do que o esperado. Verificando-se, desta forma, todas as hipóteses definidas na seção 6.2. Constatou-se que, quando o sistema tem 400 utilizadores em paralelo, o tempo máximo de resposta foi superior a 2 segundos. É importante realçar que, apesar disso, o tempo médio de resposta foi inferior a 1 segundo

O trabalho realizado ao longo da dissertação foi relevante para o autor, assim como para a empresa. Para o autor, contribui para o seu desenvolvimento profissional, permitindo a exploração de uma área interessante e desconhecida para o mesmo. Para a empresa, permitiu a aquisição de um sistema de recomendação para a sua plataforma de comércio eletrónico.

7.2 Trabalho futuro e limitações

Seria interessante adicionar outro mecanismo de importação de dados, como por exemplo, a importação de dados através de eventos. Opcionalmente, poder-se-ia construir uma API para

a importação dos dados. A inclusão de qualquer um deste mecanismo iria permitir uma importação mais dinâmica, tornando o processo mais robusto.

Devido a restrições temporais, não foi possível implementar mais do que três algoritmos de recomendação. Por isso, seria interessante implementar outros algoritmos de recomendação, nomeadamente algoritmos baseados na factorização de matrizes. Este tipo de algoritmos, além de produzirem bons resultados (Toscher and Jahrer, 2009), tem vindo a ganhar notoriedade, devido ao concurso realizado pela *Netflix*, “The Netflix Prize”. Além disso, e à semelhança do que já faz a *Netflix*, seria interessante desenvolver um algoritmo capaz de identificar tendências, permitindo, desta forma, ajustar as recomendações à medida que as tendências mudam.

Como referido na seção 3.5, as diferentes técnicas de recomendação possuem diferentes limitações. Um dos motivos que levou à implementação de mais do que um algoritmo de recomendação, foi para conseguir ultrapassar as limitações que as técnicas possuem quando usadas sozinhas. Como trabalho futuro, seria interessante realizar testes A/B, para verificar qual será melhor combinação para as técnicas implementadas.

Como referido na anteriormente, o sistema quando confrontado com 400 utilizadores em paralelo, teve o tempo máximo de resposta superior a 2 segundos, contudo, é possível reduzir o tempo de resposta. Uma das abordagens é o aumento dos recursos computacionais da máquina onde o sistema de recomendação será implantado, ou seja, escalar o sistema verticalmente. Como alternativa, poder-se-ia adicionar mais uma instância da API de Recomendação, para tal, seria necessário adicionar um componente, entre o Site e a APIs, para balancear o tráfego, entre as instâncias.

Referências

Accord.NET (2019) *Accord.NET Framework*. Available at: <http://accord-framework.net/> (Accessed: 4 February 2019).

Aditya, P. H., Budi, I. and Munajat, Q. (2017) 'A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT X', in *2016 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2016*. doi: 10.1109/ICACSIS.2016.7872755.

Agarwal, A. and Chauhan, M. (2017) 'Similarity Measures used in Recommender Systems: A Study', *Minakshi Chauhan International Journal of Engineering Technology Science and Research IJETSr* www.ijetsr.com ISSN.

Allibhai, E. (2018) *Hold-out vs. Cross-validation in Machine Learning*. Available at: <https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f> (Accessed: 23 January 2019).

Alvino, C. and Basilico Justin (2015) *Learning a Personalized Homepage*. Available at: <https://medium.com/netflix-techblog/learning-a-personalized-homepage-aa8ec670359a>.

Amazon (2019) *Amazon*. Available at: <https://www.amazon.com/> (Accessed: 12 May 2019).

Bennett, J. and Lanning, S. (2007) 'The Netflix Prize', in *Proceedings of KDD Cup and Workshop*. doi: 10.1145/1562764.1562769.

Borza, J. (2011) 'FAST Diagrames: The Foundation for Creating Effective Function Models', *Trizcon 2011*.

Brafman, R. and Domshlak, C. (2009) 'Preference Handling - An Introductory Tutorial', *AI Magazine*. doi: 10.1609/aimag.v30i1.2114.

Burke, R. (2002) 'Hybrid recommender systems: Survey and experiments', *User Modelling and User-Adapted Interaction*. doi: 10.1023/A:1021240730564.

Çano, E. and Morisio, M. (2017) 'Hybrid recommender systems: A systematic literature review', *Intelligent Data Analysis*. doi: 10.3233/IDA-163209.

Chen, J., Miller, C. and Dagher, G. G. (2014) 'Product recommendation system for small online retailers using association rules mining', in *Proceedings of the 2014 International Conference on Innovative Design and Manufacturing, ICIDM 2014*. doi: 10.1109/IDAM.2014.6912673.

Covington, P., Adams, J. and Sargin, E. (2016) 'Deep Neural Networks for YouTube Recommendations', in. doi: 10.1145/2959100.2959190.

Foundation, A. S. (no date) *MAHOUT*. Available at: <https://mahout.apache.org/> (Accessed: 4 February 2019).

Fowler, M. and Lewis, J. (2014) *Microservices*. Available at: <https://martinfowler.com/articles/microservices.html> (Accessed: 30 January 2019).

Ge, M., Delgado-Battenfeld, C. and Jannach, D. (2010) 'Beyond Accuracy: Evaluating

- Recommender Systems by Coverage and Serendipity', *RecSys*. doi: 10.1145/1864708.1864761.
- Giang (2019) *Online Retail trends: 22 ecommerce predictions for 2019*. Available at: <https://www.connectpos.com/online-retail-trends-2019/#.XEWnf1z7SUK> (Accessed: 21 January 2019).
- Gomez-Uribe, C. A. and Hunt, N. (2015) 'The Netflix Recommender System: Algorithms, Business Value, and Innovation', *ACM Transactions on Management Information Systems (TMIS)*. doi: 10.1145/2843948.
- Google (2018) *Classification: True vs. False and Positive vs. Negative*. Available at: <https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative> (Accessed: 25 January 2019).
- Gunasekaran, A. et al. (2002) 'E-commerce and its impact on operations management', *International Journal of Production Economics*. doi: 10.1016/S0925-5273(01)00191-8.
- Gunawardana, A. and Shani, G. (2009) 'A survey of accuracy evaluation metrics of recommendation tasks', *The Journal of Machine Learning Research*. doi: 10.1145/1577069.1755883.
- Herlocker, J. L. et al. (2004) 'Evaluating collaborative filtering recommender systems', *ACM Transactions on Information Systems*. doi: 10.1145/963770.963772.
- Isinkaye, F. O., Folajimi, Y. O. and Ojokoh, B. A. (2015) 'Recommendation systems: Principles, methods and evaluation', *Egyptian Informatics Journal*. doi: 10.1016/j.eij.2015.06.005.
- Jawaheer, G., Szomszor, M. and Kostkova, P. (2010) 'Comparison of implicit and explicit feedback from an online music recommendation service', in *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems - HetRec '10*. doi: 10.1145/1869446.1869453.
- Kaminskas, M. et al. (2015) 'Product recommendation for small-scale retailers', in *Lecture Notes in Business Information Processing*. doi: 10.1007/978-3-319-27729-5_2.
- Karypis, G. (2001) 'Evaluation of Item-Based Top- N Recommendation Algorithms', in *Proceedings of the tenth international conference on Information and knowledge management - CIKM'01*. doi: 10.1145/502585.502627.
- Koen, P. a et al. (2002) 'Fuzzy Front End: Effective Methods, Tools, and Techniques', in Bellviav, P., Griffin, A. and Somemeyer, S. (eds.) *The PDMA Tool-Book for New Product Development*. doi: 10.1021/ed072p378.
- Konstan, J. A. and Riedl, J. (2012) 'Recommender systems: From algorithms to user experience', *User Modeling and User-Adapted Interaction*. doi: 10.1007/s11257-011-9112-x.
- Kupcik, A. (2018) *Always Encrypted (Database Engine)*. Available at: <https://azure.microsoft.com/pt-pt/blog/transparent-data-encryption-or-always-encrypted/> (Accessed: 4 June 2019).
- Linden, G., Smith, B. and York, J. (2003) 'Amazon.com recommendations: Item-to-item collaborative filtering', *IEEE Internet Computing*. doi: 10.1109/MIC.2003.1167344.

Lops, P., de Gemmis, M. and Semeraro, G. (2011) 'Content-based Recommender Systems: State of the Art and Trends', in *Recommender Systems Handbook*. doi: 10.1007/978-0-387-85820-3_3.

MacKenzie, I., Meyer, C. and Noble, S. (2013) *How retailers can keep up with consumers*. Available at: <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers> (Accessed: 22 January 2019).

Microsoft (2019) *Execute o SQL Server na sua plataforma favorita*. Available at: <https://www.microsoft.com/pt-pt/sql-server/sql-server-2017> (Accessed: 13 February 2019).

Microsoft (no date) *What is ML.NET?* Available at: <https://dotnet.microsoft.com/learn/machinelearning-ai/what-is-mldotnet> (Accessed: 4 February 2019).

MongoDB, I. (2019) *What is MongoDB?* Available at: <https://www.mongodb.com/what-is-mongodb> (Accessed: 9 February 2019).

Netflix (2018) *Netflix*. Available at: <https://www.netflix.com/pt/>.

Núñez-Valdez, E. R. *et al.* (2018) 'A recommender system based on implicit feedback for selective dissemination of ebooks', *Information Sciences*. doi: 10.1016/j.ins.2018.07.068.

Pu, P. and Chen, L. (2010) 'A user-centric evaluation framework of recommender systems', in *CEUR Workshop Proceedings*. doi: 10.1145/2043932.2043962.

Resnick, P. *et al.* (1994) 'GroupLens: An Open Architecture for Collaborative Filtering of Netnews', in *Proceedings of the 1994 ACM conference on Computer supported cooperative work - CSCW '94*. doi: 10.1145/192844.192905.

Resnick, P. and Varian, H. R. (1997) 'Recommender systems.(Special Section: Recommender Systems)(Cover Story)', *Communications of the ACM*.

Ricci, F., Rokach, L. and Shapira, B. (2011) *Introduction to Recommender Systems Handbook*.

Saaty, T. L. (1990) 'How to make a decision: The analytic hierarchy process', *European Journal of Operational Research*. doi: 10.1016/0377-2217(90)90057-I.

Sarwar, B. *et al.* (2000) 'Analysis of recommendation algorithms for e-commerce', in *Proceedings of the 2nd ACM conference on Electronic commerce - EC '00*. doi: 10.1145/352871.352887.

Sarwar, B. *et al.* (2001) 'Item-based collaborative filtering recommendation algorithms', *Proceedings of the 10th* doi: 10.1145/371920.372071.

Sarwar, B. *et al.* (2002) 'Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems', *Proceedings of the Fifth International Conference on Computer and Information Technology (ICCIT)*. doi: 10.1.1.3.7894.

Silva, E. Q. (2011) *Revisão Bibliográfica - Sistemas de recomendação*. Available at: <https://pt.slideshare.net/EdjalmaQueirozdaSilva/sistemas-recomendacao> (Accessed: 27 January 2019).

SMARTBEAR (no date) *Swagger*. Available at: <https://swagger.io/> (Accessed: 9 June 2019).

Statista (2019a) *Retail e-commerce sales worldwide from 2014 to 2021 (in billion U.S. dollars)*. Available at: <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/> (Accessed: 21 January 2019).

Statista (2019b) *Statistics and Market Data about E-commerce*. Available at: <https://www.statista.com/markets/413/e-commerce/>.

Su, X. and Khoshgoftaar, T. M. (2009) 'A Survey of Collaborative Filtering Techniques', *Advances in Artificial Intelligence*. doi: 10.1155/2009/421425.

Thorat, P. B., Goudar, R. M. and Barve, S. (2015) 'Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System', *International Journal of Computer Applications*.

Toscher, A. and Jahrer, M. (2009) 'The BigChaos Solution to the Netflix Grand Prize', *Netflix prize documentation*. doi: 10.1093/icc/dtp008.

Woodall, T. (2003) 'Conceptualising "Value for the Customer": An Attributional, Structural and Dispositional Analysis', *Academy of Marketing Science Review*. doi: 10.1007/s11661-004-0356-5.

Zhao, Z. D. and Shang, M. S. (2010) 'User-based collaborative-filtering recommendation algorithms on hadoop', in *3rd International Conference on Knowledge Discovery and Data Mining, WKDD 2010*. doi: 10.1109/WKDD.2010.54.

8 Anexos

Anexo A- Análise AHP

	Nº de critérios	3		Nº alternativas	3
	A	Possui medidas de similaridade		1	Accord.net
	B	Identifica produtos adquiridos em conjunto		2	ML.net
	C	Maturidade		3	Apache Mahout
	Matriz de comparação de critérios				
	A	B	C		
A	1	1/2	3		
B	2	1	4		
C	1/3	1/4	1		
Total	3 1/3	1 3/4	8		
	Matriz normalizada				
	A	B	C	Prioridade relativa	
A	3/10	2/7	3/8	0,32	
B	3/5	4/7	1/2	0,56	
C	1/10	1/7	1/8	0,12	
Total	1	1	1	1	
	Análise de consistencia				
	A	B	C	Total da soma dos critérios	Racio
A	0,320238095	2/7	3/8	0,966666667	3,018587361
B	2/3	5/9	1/2	1,688095238	3,02991453
C	1/9	1/7	1/8	0,368650794	3,006472492
	Média	3,02			
	CI	0,01			
	RC	0,02	Consistencia verificada		

A	1	2	3	
1	1,00	4,00	2,00	
2	0,25	1,00	0,25	
3	0,50	4,00	1,00	
Total	1,75	9,00	3,25	
A	1	2	3	Prioridade relativa
1	0,57	0,44	0,62	0,54
2	0,14	0,11	0,08	0,11
3	0,29	0,44	0,31	0,35
Total	1,00	1,00	1,00	1,00
B	1	2	3	
1	1,00	2,00	3,00	
2	0,50	1,00	3,00	
3	0,33	0,33	1,00	
Total	1,83	3,33	7,00	
B	1	2	3	Prioridade relativa
1	0,55	0,60	0,43	0,52
2	0,27	0,30	0,43	0,33
3	0,18	0,10	0,14	0,14
Total	1,00	1,00	1,00	1,00
C	1	2	3	
1	1,00	3,00	0,50	
2	0,33	1,00	0,33	
3	2,00	3,00	1,00	
Total	3,33	7,00	1,83	
C	1	2	3	Prioridade relativa
1	0,30	0,43	0,27	0,33
2	0,10	0,14	0,18	0,14
3	0,60	0,43	0,55	0,52
Total	1,00	1,00	1,00	1,00

Anexo B- Curls

```
curl -X GET \  
  
'http://recomendations/api/v1/ItemBased?clientCode=10000030&numberOfRecomen-  
dations=5' \  
-H 'Accept: */*' \  
-H 'Authorization: 1234567890' \  
-H 'Cache-Control: no-cache' \  
-H 'Connection: keep-alive' \  
-H 'Host: recomendations.gng.pt' \  
-H 'User-Agent: PostmanRuntime/7.13.0' \  
-H 'accept-encoding: gzip, deflate' \  
-H 'cache-control: no-cache'  
  
curl -X GET \  
'http://recomendations/api/v1/SimilarModels?modelId=00501-  
0101&numberOfRecomendations=5' \  
-H 'Accept: */*' \  
-H 'Authorization: 1234567890' \  
-H 'Cache-Control: no-cache' \  
-H 'Connection: keep-alive' \  
-H 'Host: recomendations.gng.pt' \  
-H 'User-Agent: PostmanRuntime/7.13.0' \  
-H 'accept-encoding: gzip, deflate' \  
-H 'cache-control: no-cache'  
  
curl -X POST \  
http://recomendations/api/v1/CooPurchases \  
-H 'Authorization: 1234567890' \  
-H 'Content-Type: application/json' \  
-H 'Postman-Token: 4c86e725-d5d5-47af-a3cb-1a39375deac' \  
-H 'cache-control: no-cache' \  
-d '{  
  "NumberOfRecomendations":5,  
  "Cart":{  
  
    "ClientCode":"10000030",  
    "Lines":[  
      {  
        "LineNumber":1,  
        "ModelId":"22791-0052"  
      },  
      {  
        "LineNumber":2,  
        "ModelId":"77319-0397"  
      }  
    ]  
  }  
}'
```