NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

A Criação de Valor através da Reutilização de Dados Abertos

Arquitetura e desenvolvimento de uma plataforma Serverless em Cloud AWS, direcionada ao retalho alimentar online

Cláudia Maria Silva Pimenta

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Gestão de Informação

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

LOMBADA MGI

2022

Título: A Criação de Valor através da Reutilização de Dados Abertos

Subtítulo: Arquitetura e desenvolvimento de uma plataforma Serverless em Cloud AWS, direcionada ao retalho alimentar online

Cláudia Maria Silva Pimenta

MGI

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

A CRIAÇÃO DE VALOR ATRAVÉS DA REUTILIZAÇÃO DE DADOS ABERTOS

ARQUITETURA E DESENVOLVIMENTO DE UMA PLATAFORMA SERVERLESS EM CLOUD AWS, DIRECIONADA AO RETALHO ALIMENTAR ONLINE

por

Cláudia Maria Silva Pimenta

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Gestão de Informação, Especialização em Gestão do Conhecimento e Business Intelligence

Orientador: Pedro Manuel Carqueijeiro Espiga da Maia Malta

novembro 2022

DEDICATÓRIA

Não desvalorizando o processo de crescimento individual que cada ser humano inevitavelmente vive num determinado período da sua vida, acredito verdadeiramente que o processo de crescimento é substancialmente mais rico quando ultrapassado em grupo, seja de uma forma direta ou indiretamente. Num grupo coeso, forte e verdadeiro. Considero que eu e o meu irmão César formamos um desses grupos. Um grupo de vida. De amor. Existimos para evoluir os dois juntos. É ele quem me suporta. É a ele a quem recorro quando preciso de orientação. E ele, com amor, disponibiliza e partilha o sentido de clareza mental que lhe pertence. Que o conhecimento que adquiri se transforme numa força inabalável de confiança e retome novamente até ele. Porque transmitiu-me um conhecimento capaz de ser reutilizado. Capitalizei-o para a pesquisa, metodologia e análises relevantes para a àrea cientifica de dados, e isto acontece porque o ser humano é capaz de fazer coisas extraordinárias. Em grupo.

Para o meu irmão, para minha avó e para o meu pai - César Pimenta, Maria de La Salete Batista Carvalho e Armando Pimenta. Sem nunca esquecer em memória da minha mãe, Maria da Conceição Batista Silva, do meu avô Isidro Silva e de um anjo no céu, Bruno Filipe.

AGRADECIMENTOS

Para conseguir concluir a presente dissertação tenho necessariamente que agradecer a muitas pessoas.

Ao meu orientador professor Pedro Manuel Carqueijeiro Espiga da Maia Malta, pela confiança depositada em mim. Ao Bruno Miguel, pela força inabalável que me transmite e pelo que construímos/queremos construir juntos. Ao Miguel por tudo o que me ensinou. À empresa para a qual trabalho que proporcinou disponibilidade total em ajudar-me em todos os temas técnicos. À Francisca, pela partilha e amizade. À SONAE pelo contributo e disponibilidade. À minha família pelo amor e apoio incondicional que me trasmitem. A todos os meus amigos. A todas as pessoas que me ajudaram nesta caminhada.

RESUMO

O presente estudo teve como objetivo geral comprovar o valor que a reutilização de dados abertos pode representar no impacto económico de uma empresa com loja online. A recolha de informação (os dados relativos a produtos) em fontes abertas (em lojas online) constitui uma das potencialidades para a exploração do mercado online.

Para isso foram abordadas questões inerentes ao desenvolvimento de criação de valor dos dados abertos com a arquitetura e a implementação de uma plataforma de reutilização de dados, utilizando exclusivamente informação pública das principais lojas online de retalho em Portugal. O resultado traduzir-se-á numa plataforma que combina dados de diferentes retalhistas para uma exploração de dados ampla, rica e precisa acerca do mercado retalhista online, em tempo real.

No que diz respeito à implementação da plataforma serão interpelados todos os passos necessários da construção de uma ferramenta escalável e automatizada para um acesso e recolha mais fácil às informações, conteúdos e produtos das lojas online (*Websites*), resultando em ganhos de eficiência no que diz respeito a Data Analytics devido à utilização de dados em tempo real que permitem elaborar análises avançadas e assim contribuir para um conhecimento mais profundo do mercado. Com foco nos novos paradigmas da Data Science e na importância da inclusão de tecnologias que revelem uma mais-valia quando aplicadas a desenvolvimentos de projetos nesta temática, a aplicação será construída com base numa arquitetura Serverless na Cloud Amazon Web Services (AWS) utilizando as técnicas de Web Scraping e Web Crawling para a extração dos dados, encontrando soluções de resposta às diversas proteções dos Websites (lojas online).

Com foco no objetivo principal, depois da informação recolhida, transformada e armazenada, será desenvolvida uma camada de análise aos dados, a fim de observar e medir a importância dos dados no mercado de retalho online em Portugal.

Sintetizado, questões relacionadas com matéria de reutilização de dados abertos, técnicas de Web Scraping e Web Crawling, soluções contra defesas que os portais online implementam, vantagens e desafios na utilização de arquiteturas Serverless e construção de análises visando a criação de valor na compreensão do negócio, serão conceitos abordados e discutidos com detalhe durante a presente dissertação.

PALAVRAS-CHAVE

Dados Abertos, Análise de Dados, Arquitetura Serverless, Web Scraping, Web Crawling, E-commerce, ETL.

RESUMO

This study had the general objective of proving the value that the reuse of open data can represent in the economic impact of a company with an online store. The collection of information (product data) in open sources (in online stores) is one of the potentialities for exploring the online market.

To this end, issues inherent to the development of open data value creation were addressed with the architecture and implementation of a platform for data reuse, using exclusively public information from the main online retail stores in Portugal. The result will be a platform that combines data from different retailers for a broad, rich, and accurate data exploration of the online retail market, in real time.

Regarding the implementation of the platform, all necessary steps will be addressed to build a scalable and automated tool for easier access and collection of information, content, and products from online stores (Websites), resulting in efficiency gains regarding Data Analytics due to the use of real-time data that allows advanced analysis and thus contributes to a deeper understanding of the market. Focusing on the new paradigms of Data Science and the importance of including technologies that show added value when applied to project developments in this area, the application will be built based on a Serverless architecture in the Amazon Web Services (AWS) cloud using Web Scraping and Web Crawler techniques for data extraction, finding solutions to respond to the various protections of Websites (online stores).

Focusing on the main goal, after the information collected, transformed, and stored, a layer of data analysis will be developed to observe and measure the importance of data in the online retail market in Portugal.

In summary, issues related to open data reuse, Web Scraping and Web Crawler techniques, solutions against defenses that online portals implement, advantages and challenges in the use of Serverless architectures and the construction of analytics aiming to create value in business understanding, will be concepts addressed and discussed in detail during this dissertation.

PALAVRAS-CHAVE

Open Data, Data Analysis, Serverless Architecture, Web Scraping, Web Crawling, E-commerce, ETL.

ÍNDICE

1. Introdução 1
2. Revisão da Literatura 4
   2.1 Dados abertos 5
      2.1.1 Classificação das fontes de informação e Dados Abertos 5
      2.1.2 Ciclo da recolha de informação 8
      2.1.3 Reutilização de dados abertos 10
   2.2 Técnicas utilizadas para a recolha e extração de dados 12
      2.2.1 Extração de dados e os processos ETL e ELT 12
   2.3 Factualidades sobre o panorama das lojas online – Oportunidades de explorar os dados dos concorrentes 15
      2.3.1 Evolução, representatividade e caracterização das empresas no e-commerce em Portugal 15
      2.3.2 A monitorização do mercado retalhista no canal online 19
   2.4 Arquiteturas serverless e serviços Amazon Web Services (AWS) 20
3. Metodologia 23
   3.1 Contexto e objetivos do processo metodológico 23
   3.2 Arquitetura da ferramenta 29
   3.3 Desenvolvimento da ferramenta 37
      3.3.1 Pré-configurações necessárias no desenvolvimento da plataforma 37
      3.3.2 Processo de Recolha dos Dados e Processo Tratamento de Dados 42
      3.3.2.1 Processo de Recolha dos Dados 42
      3.3.2.2 Processo de Tratamento dos dados 56
      3.3.3 Fluxo de trabalho entre os dois processos: Recolha dos dados e Tratamento dos dados & Monitorização e Notificações 58
4. Resultados e Discussão 60
   4.1 Criação de valor através da reutilização de dados 60
   4.2 Modelo da economia circular de dados 69
5. Conclusões, Limitações e recomendações para trabalhos futuros 71
6. Bibliografia 73
7. Anexos 76

ÍNDICE DE FIGURAS

Figura 1 - Ciclo da recolha de informação 8
Figura 2 - Processo ETL 13
Figura 3 - Processo ELT 13
Figura 4 - Percentagem de empresas que receberam encomendas por redes eletrónicas e peso no volume de negócios 16
Figura 5 - Percentagem de empresas que receberam encomendas por redes eletrónicas e tipo de receção e peso no volume de negócios 16
Figura 6 - Percentagem de empresas que receberam encomendas via website/app por segmento e peso no volume de negócios....17
Figura 7 - Percentagem de empresas que receberam encomendas via website/app por tipologia 17
Figura 8 - Grupos de empresas que mais receberam encomendas por redes eletrónicas....18
Figura 9 - Transformar dados de uma loja online em novas análises do mercado 23
Figura 10 – Processo metodológico na transformação de dados abertos para a criação de valor 27
Figura 11 - Arquitetura Alto Nível Recolha dos Dados e Tratamentos dos Dados 29
Figura 12 - Arquitetura Alto Nível Atribuir Conhecimento aos Dados....30
Figura 13 - Relação entre a Framework Serverless e a plataforma de desenvolvimento....33
Figura 14 - Arquitetura Baixo Nível 35
Figura 15 - Processo de desenvolvimento Recolha de Dados e desenvolvimento Tratamento de Dados....42
Figura 16 - Processo de desenvolvimento Recolha de Dados 43
Figura 18 - Exemplo de um bloco de HTML de um produto disponível numa loja online de retalho....45
Figura 19 - Processo de desenvolvimento à arte de Web Scraping....46
Figura 20 - Servidor Proxy....52
Figura 21 - Armazenamento dos dados raw no Bucket Amazon S3 53
Figura 22 - Aspetos importantes a reter do desenvolvimento da Recolha de Dados 54
Figura 23 - Processo de desenvolvimento Tratamento de Dados 56
Figura 24 - Resultados: Nº de produtos por dia de extração 62
Figura 25 - Resultados: Distribuição de Preços até 40€ com intervalos de 1€....63
Figura 26 - Resultados: Distribuição de Preços por Loja: Loja A e Loja B....64
Figura 27 - Resultados: Distribuição de Preços até 1€ com intervalos de 0,01€....65
Figura 28 - Resultados: Distribuição de Preços até 10€ com intervalos de 0,1€....66

Figura 29 - Resultados: Distribuição de Promoções por Preço Original.....67
Figura 30 - Resultados: Distribuição de Promoções Percentuais 68
Figura 31 - Resultados: Modelo da economia circular dos dados 69

ÍNDICE DE TABELAS

Tabela 1 - Características de interesse do produto a extrair.....26
Tabela 2 - Stack tecnológico da ferramenta 31
Tabela 3 - Especificação dos recursos numa perspetiva de Arquitetura Alto Nível 34
Tabela 4 – Esquematização da informação para a Recolha dos Dados 43
Tabela 5 - Diferenças encontradas em Web Scraping e Web Crawling no desenvolvimento .55

LISTA DE SIGLAS E ABREVIATURAS

AWS	Amazon Web Services
IAM	Identity & Access Management
IP	Internet Protocol
OSINT	Open Source Intelligence
ETL	Extract, Transform and Load
ELT	Extract, Load and Transform
RGPD	Regulamento Geral sobre a Proteção de Dados
NATO	North Atlantic Treaty Organization
OLAP	Online Analytical Processing
YAML	YAML ain't markup language
SaaS	Software as a Service
FaaS	Function as a Service

1. INTRODUÇÃO

O presente trabalho de investigação teve por base estudar a maturidade dos conceitos: fontes abertas e dados abertos associados às lojas online (Websites) em Portugal. Se por um lado os dados abertos representam uma grande importância em diferentes áreas "da ciência— pelo seu potencial de criação de conhecimento— à administração pública—com a promoção da transparência, da economia circular— no uso eficiente dos recursos—, ao envolvimento dos cidadãos em todos os temas da sociedade." (Estratégia de Dados Abertos INCoDE.(2030)), por outro lado o mercado retalhista é um meio que beneficia do valor criado através de dados abertos, ou seja, tem todo o interesse em acompanhar e monitorizar o mercado online, as políticas praticadas. Caracteriza-se como um mercado que precisa de dados reais para monitorizar a concorrência: últimos descontos, entrada de novos produtos, principais campanhas do dia, da semana, do mês, e por isso é percetível a importância de aceder em tempo real a esta informação uma vez que a concorrência atualiza a sua página (o nosso target) de forma constante, assídua e em momentos estratégicos. Além desta monitorização, através dos dados abertos mencionados, o retalhista detém a possibilidade de expandir a reutilização dos mesmos (por exemplo, para análises preditivas).

É com esta base de estudo, que a presente dissertação pretende trabalhar a maturidade dos dados abertos através da implementação de um caso prático levado a cabo de forma sistemática para aumentar o campo dos conhecimentos, da cultura e da sociedade, e a utilização desses conhecimentos para criar aplicações traduzindo-se em investigação aplicada e desenvolvimento experimental.

Assim, a questão predominante da investigação prende-se em perceber qual o possível valor criado através da reutilização de dados abertos das lojas online (Websites). No entanto, quando levantamos esta questão, outras questões são levantadas: o acesso a lojas online (que representam as fontes abertas target) é fazível? Ou existem bloqueios de acesso por parte dos retalhistas? As páginas tendem a esconder ou a dificultar esse acesso? É possível reutilizar essa informação? Existe maturidade do assunto nas empresas privadas? Ou seja, é verificada uma rede de conhecimento sobre o assunto?

Estas são algumas das questões que serão respondidas ao longo da presente dissertação. Com foco nestas questões, será desenhada e desenvolvida uma ferramenta com o objetivo de contribuir para a mudança e expandir o conhecimento (espelhar através da experimentação que necessidades existem para uma maior maturidade da usabilidade e reutilização dos dados abertos). É neste sentido, que através de um questionário, realizado junto da SONAE, uma das maiores empresas de retalho em Portugal, com presença no mercado online, recolhemos informação para a presente produção científica acerca do retalho online e qual o posicionamento da empresa em relação a estas temáticas.

Espelhando a motivação para este trabalho, e de acordo com as respostas facultadas pela SONAE, as estratégias de monitorização implementadas no mercado físico diferem daquelas que são implementadas no mercado online "Os processos de shopping online são obtidos através de *service providers* especializados neste tipo de operações, enquanto o processo físico é suportado por uma equipa de profissionais que recolhem *in loco* os preços da concorrência". Percebemos desta forma que a monitorização do mercado online (recolha e tratamento dos dados abertos disponibilizados pela concorrência), é garantido por *service providers* especializados, o que significa que a empresa não detém uma equipa interna com este conhecimento. Além disto, é partilhado pela empresa que esta monitorização é feita de forma "Diária", e é realizada apenas nas seguintes vertentes: "Preço",

"Descontos/Promoções nos produtos" e "Grandes campanhas". Quer isto dizer que diariamente são executados programas que recolhem, tratam e armazenam dados disponibilizados pela concorrência relacionados com os preços, descontos e campanhas desses mesmos concorrentes. Ainda assim, não são monitorizadas vertentes mais complexas como por exemplos a saída e entrada de novos produtos (o que exige que seja feita uma pesquisa diária à base de dados histórica, que é armazenada, da concorrência).

Com base neste entendimento, a presente dissertação procura desenvolver não só um projeto técnico – implementação de uma ferramenta que produz avanços nas abordagens genéricas para a recolha, armazenamento, processamento, tratamento e visualização de informação reutilizada, com a identificação de lacunas de conhecimentos tecnológicos necessários nesta temática – bem como um projeto analítico – comprovar a criação de valor com a reutilização de dados abertos no mercado retalhista online, com a demonstração de análises relevantes acerca do mercado. Este valor consiste na possibilidade de validação da gama de produtos e preços que os concorrentes oferecem ao mercado, com o objetivo de validar se a oferta do retalhista é consistente e/ou diferenciadora da competição e se a sua política de pricing é adequada e/ou competitiva. Para isto é então necessária uma base de dados reutilizada com base nos dados abertos disponibilizados pela concorrência (os produtos e as usas características disponíveis nas lojas online). De referir que se trata de um âmbito passível de capitalizar em diferentes indústrias (todas aquelas com produtos de venda online).

Em relação à arquitetura e implementação da plataforma, esta será baseada em tecnologias emergentes na área de Data Science. Para a recolha diária dos dados serão utilizadas as técnicas de Web Scraping e Web Crawling, que permitem a extração de dados em Websites convertendo-os em informação estruturada para posterior análise. Todo o processo de *Extract*, *Load*, and *Transform* – ELT, será construído e armazenado através de serviços cloud, recorrendo aos serviços da Amazon Web Services - AWS. A AWS oferece tecnologias para executar código, gerir dados e integrar aplicações, tudo isso sem a necessidade de gerir servidores (*Serverless Computing*). Quer isto dizer que a plataforma será construída através de *Serverless Computing*, e assim garantir uma escalabilidade automática, alta disponibilidade integrada e um modelo de pagamento pago por utilização para aumentar a agilidade e otimizar os custos. Depois dos dados trabalhados em serviços cloud AWS (extraídos, armazenados e transformados), serão construídas análises através de bibliotecas *Python*.

Neste seguimento, é importante notar que a presente iniciativa é inserida num contexto do setor de distribuição alimentar online, com o propósito de extrair informação de duas das principais lojas de retalho alimentar online em Portugal, denominadas como a Loja A e a Loja B. Uma vez que o desenvolvimento se prende à prova de conceito da reutilização de dados, não existe a necessidade de partilhar o nome dos retalhistas em questão e, por uma questão de Regulamento Geral sobre a Proteção de Dados [1](RGPD), decidimos denominar as duas lojas como Loja A e Loja B.

A proposta da metodologia não se trata apenas de dar a conhecer alternativas na construção de um novo sentido aos dados (reutilização de dados abertos), mas sim criar condições para a obtenção de conhecimentos básicos e desenvolver habilidades necessárias ao serviço de extração de dados, armazenamento e visualização dos mesmos transversal a quaisquer fontes abertas, como sites web, meios de comunicação social, jornais de notícias online, informação do domínio público relacionada

[1] Sabe mais sobre RGPD aqui https://www.sg.pcm.gov.pt/sobre-nos/regulamento-geral-de-prote%C3%A7%C3%A3o-de-dados.aspx

com dados oficiais de organismos governamentais, publicações académicas disponibilizadas pelas instituições, entre outras.

Este estudo é composto por 5 capítulos. Apresenta um breve capítulo introdutório à temática em estudo, sobre a sua motivação, descreve o problema e identifica os macro e micro-objetivos desta estratégia. Segue-se uma abordagem do setor de retalho nas suas dimensões demográfica, económica e financeira bem como uma abordagem a todos os conceitos necessários para a compreensão do tema estudado. O elemento central do presente documento, o terceiro capítulo, apresenta uma referência aos principais aspetos metodológicos do desenho e da implementação da plataforma e analisa também os principais indicadores passíveis de se extrair da recolha de dados efetuada. No quarto capítulo, apresentam-se todas as análises provenientes da implementação do projeto. Por fim são apresentadas algumas limitações do estudo e perspetivas futuras.

2. REVISÃO DA LITERATURA

O presente capítulo procura através de análises minuciosas aos trabalhos de outros investigadores que precederam sobre todos os componentes que constituem o propósito final da investigação, abordar os conceitos necessários e intrínsecos para um maior entendimento do objetivo da dissertação. Cada subcapítulo abordará uma determinada área relevante para um conhecimento mais aprofundado de todos os componentes que constituem o propósito final: comprovar a criação de valor através da reutilização de dados abertos no mercado retalhista online.

Através de informação prévia que fora localizada, analisada minuciosamente, sintetizada, e interpretada a partir de trabalhos de outros investigadores acerca do propósito da dissertação, o presente capítulo procura justificar a necessidade de acompanhar as estratégias implementadas no mercado/concorrência em tempo real na perspetiva do retalhista através da reutilização de dados abertos.

Assim, o presente capítulo, para uma leitura mais clara, encontra-se dividido em quatro subcapítulos:

- 2.1 Dados Abertos – neste subcapítulo poderemos encontrar literatura relacionada com a definição, o processo de recolha e exemplos de plataformas baseadas em reutilização.
- 2.2 Técnicas utilizadas para a recolha e extração de dados – neste subcapítulo encontraremos literatura direcionada a uma vertente mais técnica.
- 2.3 Factualidades sobre o panorama das lojas online – Oportunidades de explorar os dados dos concorrentes – foca-se essencialmente em estudar e investigar o panorama atual do e-commerce em Portugal e qual a importância de monitorizar o mercado e a concorrência online.
- 2.4 Arquiteturas Serverless e serviços Amazon Web Services (AWS) – neste subcapítulo abordaremos o *Core* e os *Providers* em arquiteturas Serverless, sistematizando o impacto das arquiteturas Serverless nos Sistemas de Informação nas pequenas, médias e grandes empresas em Portugal. Além disso, estudaremos em detalhe os serviços da AWS *Provider*.

2.1 DADOS ABERTOS

2.1.1 Classificação das fontes de informação e Dados Abertos

Tendo em conta que a reutilização de dados parte da recolha assertiva e direcionada de informação para um determinado tópico, a identificação da fonte de informação e a elucidação quanto ao acesso da mesma, representam dois tópicos cruciais em todo o processo. Desta forma, primariamente importa caracterizar uma determinada fonte de informação quanto as características mencionadas para obtermos uma maior qualidade na recolha de informação.

De acordo com o livro "Introdução à cibersegurança – A internet, os aspetos legais e a análise digital forense", quanto à autoridade para aceder explicitamente à informação, as fontes utilizadas podem ser classificadas como Fechadas ou Abertas.

Do mesmo livro, os autores afirmam que "O acesso a uma fonte fechada de informação implica a obtenção de uma autorização formal ou judicial explícita, ou através de uma delegação implícita de competências." (Introdução à cibersegurança – A internet, os aspetos legais e a análise digital forense, Mário Antunes & Baltazar Rodrigues (2018)).

Assim, na prática, "a caracterização de uma fonte como fechada fá-la ficar fora dos limites legais de recolha de informação por iniciativa própria da polícia. Pretende-se com isso, manter a privacidade do cidadão, evitar a inadmissibilidade da utilização da informação assim obtida na investigação criminal e, no limite, evitar que sobre o agente policial ou a Organização para a qual trabalha, possa recair responsabilidade civil, disciplinar e criminal." (O Conceito de "Fontes Abertas " na Investigação do Cibercrime, Rogério Bravo (2014))

Neste sentido, podemos encontrar vários exemplos de fontes fechadas, como sejam:

- Instituto de Registos e Notariado (IRN)
- Instituto da Mobilidade e dos Transportes (IMT)
- Autoridade Tributária e Aduaneira (AT)
- Segurança Social (SS)

Por outro lado, uma fonte diz-se aberta "se for totalmente acessível por terceiros, que seja de origem individual ou coletiva, independentemente de ser ou não possível a recolha e o processamento automático da informação recolhida" (Introdução à cibersegurança – A internet, os aspetos legais e a análise digital forense, Mário Antunes & Baltazar Rodrigues (2018)).

Posto isto, e analisando esta definição de fonte aberta podemos reter que ao analisar dados das lojas retalhistas online em Portugal (para o caso prático o Website da Loja A e Loja B), estas se caracterizam como fontes abertas uma vez que a sua informação é totalmente acessível por qualquer utilizador Web. Assim, acedendo a qualquer loja online, o seu acesso é livre não sendo exigido qualquer permissão e limitação de acesso para consultar o seu conteúdo: ver produtos, marcas de produtos, preços, a aplicabilidade de todos os descontos, ou seja, se o desconto é aplicado em percentagem ou se é aplicado em euros, campanhas do momento, imagens dos produtos, o marketing das categorias, ou seja, no fundo toda a informação que é partilhada e totalmente disponível em cada loja online.

Ainda em relação à informação que é disponibilizada em fontes abertas, importa numerar as vantagens e desvantagens da sua utilização. Com base no livro "Introdução à cibersegurança – A internet, os aspetos legais e a análise digital forense", destacamos:

a) Vantagens da sua utilização:
    a. Custos reduzidos
    b. Acesso rápido à informação
    c. Diversidade e quantidade de dados disponíveis

b) Desvantagens da sua utilização:
    a. Estas fontes podem ser alvo de desinformação, requerendo, por isso, uma rigorosa validação
    b. Necessitam de especialistas em vários domínios
    c. O recurso à Internet implica o acesso a um gigantesco volume de informação disponível. Tal facto implica necessariamente o recurso a aplicações especificas com potencialidades de pesquisa avançada, tratamento automático e inteligência na recolha e processamento da informação obtida.

É com base nas vantagens e desvantagens acima descritas, que iremos trabalhar e estudar o conceito da extração de dados abertos. Assim, procuraremos responder e entender com profundidade as seguintes questões: Qual o nível de complexidade associado ao acesso de forma automatizada aos dados disponibilizados nas lojas online? Qual a quantidade de dados extraídos diariamente? Como automatizar o processamento e tratamento dos dados extraídos de forma diária?

Assim, e neste seguimento, importa então perceber a definição de dados abertos. Dados Abertos, de acordo com a *Open Definition*[2], são "dados que qualquer pessoa pode aceder, utilizar, modificar e partilhar, para qualquer propósito." (DEFINING OPEN IN OPEN DATA, OPEN CONTENT AND OPEN KNOWLEDGE, Open Definition)

Quando pensamos no conceito de dados abertos, facilmente retomamos ao início da era tecnológica: o Windows é uma ferramenta fechada. Contudo, o Linux introduziu ferramentas abertas com a possibilidade de manipular os dados e de inserir melhorias. Além disso, através de fontes abertas só é possível extrair os dados abertos e reutilizá-los, mas sem a possibilidade de manipular a fonte original, tal como acontece num *Website* direcionado a uma loja online. Podemos extrair os dados abertos, manipulá-los e transformá-los com foco na reutilização, mas sem nunca manipular o *Website* que apresenta os dados originais.

Neste sentido, o movimento dos dados abertos, que é parte integral das políticas dedicadas ao Governo Aberto (Open Government)[3], combina os princípios da transparência, participação e colaboração, assim como o potencial de desenvolvimento económico que o digital trouxe.

De acordo com o Portal de dados abertos da Administração Pública em Portugal, "O grande desafio (e a maior preocupação das iniciativas de dados abertos como o dados.gov) passa por facilitar o seu acesso e reutilização, beneficiando vários grupos e sectores da sociedade." Nomeadamente, "o setor

[2] Open Definition é um documento publicado pela Open Knowledge Foundation. Sabe mais aqui https://opendefinition.org/

[3] Sabe mais aqui https://ogp.eportugal.gov.pt/inicio

empresarial, que pode reutilizar informação pública para criar aplicações, plataformas ou serviços com elevado potencial comercial" (Portal de dados abertos da Administração Pública). Tendo por base este desafio, e como já mencionado anteriormente, entender, perceber e comprovar a criação de valor através da reutilização de dados disponibilizados e acessíveis nas lojas de retalho online em Portugal é o foco de todo o trabalho da presente investigação. Dada a complexidade de acesso e extração dos dados, o Portal de dados abertos da Administração Pública, afirma que "Esse desafio passa por disponibilizar os dados em formatos passíveis de serem lidos por mecanismos automatizados, através de formatos e ferramentas abertas, para que possam ser reutilizados, transformados ou integrados por qualquer cidadão ou entidade, por norma disponibilizados sob a forma de conjuntos de dados." Ou seja, será que criar uma plataforma com os dados disponibilizados em lojas online como o nome dos produtos, marcas, preço, descontos, entre outros, de diferentes retalhistas, não representará um enorme poder de conhecimento acerca do mercado com a possibilidade de tornar tais dados como base na tomada de decisões nas empresas?

- **Maturidade de Dados Abertos em Portugal e na Europa**

O estudo sobre a maturidade dos dados abertos "Maturidade dos dados abertos – Relatório de 2021", serve de ponto de referência para perspetivar o desenvolvimento alcançado na área dos dados abertos na Europa. Avalia o grau de maturidade em relação a quatro dimensões: política, portal, impacto e qualidade. O estudo reúne os países em quatro grupos diferentes: líderes, seguidores rápidos, seguidores e principiantes, desde o mais maturo ao menos. O relatório de 2020 sobre o Valor Económico dos Dados Abertos estuda o valor criado pelos dados abertos na Europa. É o segundo estudo realizado pelo Portal Europeu de Dados, depois do relatório de 2015. A dimensão do mercado dos dados abertos é estimada em 184 mil milhões de euros e calcula-se que ascenda a entre 199,51 e 342,21 mil milhões de euros em 2025. O relatório considera ainda como a dimensão deste mercado está distribuída pelos diferentes setores e quantas pessoas estão empregadas devido aos dados abertos. Os ganhos de produtividade dos dados abertos, em possíveis vidas salvas, tempo economizado, benefícios ambientais, e melhoria de serviços linguísticos, bem como potenciais economias associadas são explorados e quantificados sempre que possível. Por fim, o relatório também considera exemplos e informações da reutilização de dados abertos nas organizações. Os principais resultados do relatório são resumidos seguidamente:

1. A especificação e a implementação de conjuntos de dados de valor acrescentado como parte da nova diretiva relativa aos dados abertos representam uma oportunidade promissora de abordar as exigências de qualidade e quantidade dos dados abertos.
2. É importante abordar as exigências de qualidade e quantidade dos dados abertos, embora tal não seja suficiente para concretizar todo o potencial dos dados abertos.
3. Os reutilizadores de dados abertos devem estar informados e serem capazes de compreender e explorar todo o potencial.
4. A criação de valor dos dados abertos faz parte do desafio mais alargado da transformação das competências e dos processos: um processo moroso cuja alteração e impacto nem sempre é fácil de observar e medir.
5. As iniciativas setoriais e a colaboração nos setores público e privado, quer isoladamente, quer transversalmente, fomentam a criação de valor.
6. Combinar dados abertos com dados pessoais, partilhados ou colaborativos, é vital para a consecução de maior crescimento do mercado dos dados abertos.
7. Para os diferentes desafios, devemos explorar e melhorar os diversos métodos de reutilização dos dados que sejam éticos, sustentáveis e adequados.

2.1.2 Ciclo da recolha de informação

"A recolha de informação em fontes abertas e consequentemente a técnica utilizada para esse fim é usualmente designada por Open Source Intelligence e abreviada para o acrónimo OSINT." (Introdução à cibersegurança – A internet, os aspetos legais e a análise digital forense, Mário Antunes e Baltazar Rodriguez, (2018)). Podemos ler ainda que "A capacidade de sistematização das informações recolhidas e a sua correlação confere um grau de automatização e inteligência à recolha de informação efetuada e à sua consequente análise." A NATO (North Atlantic Treaty Organization) define OSINT como "[…] a informação não classificada que foi deliberadamente descoberta, discriminada, destilada e disseminada para uma audiência selecionada, de modo a responder a uma questão especifica." (NATO Open Source Intelligence Handbook, (2001)).

As motivações para o uso destas técnicas de recolha são várias. Concretamente, no estudo da criação de valor através de dados disponíveis em lojas online, pretende-se obter informação sobre o mercado retalhista online em Portugal, com vista à monitorização da concorrência e apoio na tomada de decisões das empresas, ao nível de princing, campanhas de maketing, estratégias de descontos, entre outras vertentes.

Analisando e estudando a metodologia genérica para a recolha de informação proposta no livro "Introdução à cibersegurança – A internet, os aspetos legais e a análise digital forense", a seguinte imagem – Figura 1 - Ciclo da recolha de informação - ilustra o ciclo da recolha de informação a adotar.

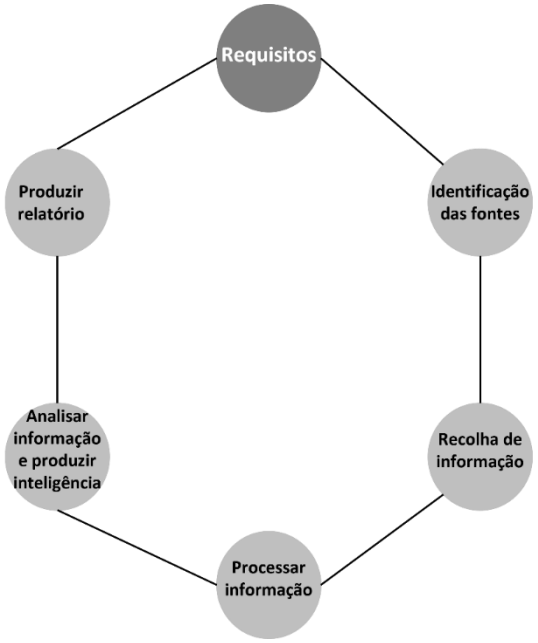

Figura 1 - Ciclo da recolha de informação

De acordo, com o ciclo da recolha de informação partilhado, este inicia-se com a análise de requisitos (ou quesitos) para a procura e recolha de informação. Assim, para a presente investigação é importante estruturar os seguintes requisitos:

- Perceber em que sentido os dados abertos em lojas de retalho online, possibilitam novas formas de reutilizar os dados
- Analisar a possível recolha e o processamento automático da informação recolhida acerca da informação disponível em lojas de retalho online
    - Desenvolver um *bot* para a recolha de informação previamente identificada
- Estudar o preço/desconto/promoções dos produtos: correlações, *outliers*, identificação de estratégias, etc
- Estudar e analisar todo o processo em volta da reutilização de dados disponíveis em lojas online

"De seguida, identificam-se as fontes de informação que melhor poderão satisfazer os requisitos anteriormente definidos. Para cada uma dessas fontes é realizada a recolha, o processamento e a análise da informação recolhida que seja relevante para o caso em concreto." (Introdução à cibersegurança – A internet, os aspetos legais e a análise digital forense, Mário Antunes e Baltazar Rodriguez, (2018)).

Desta forma, e mediante as etapas aqui descritas, a identificação das fontes bem como os dados de interesse serão espelhados de forma mais detalhada no subcapítulo 3.1 Contexto e objetivos do processo metodológico.

"A fase de análise tem em vista a produção de inteligência sobre a informação recebida, designadamente através da correlação automática e produção de nova informação que esteja relacionada com o caso." (Introdução à cibersegurança – A internet, os aspetos legais e a análise digital forense, Mário Antunes e Baltazar Rodriguez, (2018)). Neste sentido, será então aplicada análises sobre os dados extraídos diariamente com a finalidade de obter insights valiosos sobre as práticas do mercado, padrões aplicados de forma temporal com foco na descoberta de informação nos dados.

Por fim, o ciclo da recolha de informação sugere que "da análise deverá resultar a elaboração de um relatório com as informações relevantes que foram obtidas, podendo saí serem extraídos novos requisitos para uma nova recolha." (Introdução à cibersegurança – A internet, os aspetos legais e a análise digital forense, Mário Antunes e Baltazar Rodriguez, (2018)).

2.1.3 Reutilização de dados abertos

Num artigo publicado pela Oxford Univertity Press, titulado como Big Data and data reuse: a taxonomy of data reuse for balancing big data benefits and personal data protection e publicado em 2016, este clarifica que a reutilização de dados no seu sentido mais lato sugere que existe uma utilização inicial (primária) dos dados e uma utilização subsequente (secundária) dos dados, ou seja, a reutilização dos dados. Além disso, podemos ler que a reutilização de dados é um fator importante quando se trata de materializar os benefícios de Big Data e quando feita adequadamente é benéfica tanto a nível social como a nível económico.

A reutilização de dados no contexto que aqui abordamos pode ser literalmente interpretada como transformar informação (informação que objetiva a venda de produtos de retalho no canal online) num novo valor de informação: utilizar estes dados para o estudo das estratégias praticadas no mercado retalhista online em Portugal.

Assim entendemos que a reutilização de dados (utilização secundária) só pode acontecer após a utilização de dados (utilização primária).

Os benefícios da reutilização de dados residem nas possibilidades de deter novos valores de informação, correlações e padrão nos dados que podem derivar de diferentes fontes.

Na área da recolha e reutilização de dados abertos, existem já muitos repositórios que disponibilizam dados abertos. O trabalho destes repositórios, através do desenvolvimento de aplicações, traduz-se em recolher, processar, homogeneizar, armazenar e disponibilizar dados. Estas plataformas simplificam o acesso, aglomerando dados de diferentes origens e estandardizando-os para que os utilizadores possam, com simples *downloads*, ou consultas rápidas, aceder à informação que pretendem de uma forma muito mais direta e gratuita. Torna-se simples, rápido e eficaz. Exemplos disso em Portugal são plataformas como dados.gov, a central de dados, *Open Data*, entre outros. Podemos saber mais sobre este tema no subcapítulo 2.1.1 Classificação das fontes de informação e Dados Abertos.

Fazendo um levantamento destas plataformas, podemos concluir, que existe uma oferta muito escassa, limitada e com pouca variedade nas áreas de informação o que leva a que o processo de obtenção da informação seja mais complexo na medida em que é exigido aos interessados na informação que adquiram conhecimentos técnicos não só no processo de categorização e organização de dados, como também nos processos de recolha, análise e visualização da informação.

Segue abaixo descrito uma breve descrição e a especificação de como são utilizados os dados abertos em algumas aplicações/plataformas implementadas em Portugal que procuram explorar os dados abertos em inteligência tecnológica:

- Portugal – Fogos (FOGOS.pt):
    - Breve descrição: Fogos de mapas em Portugal sobre um mapa. Os utilizadores podem ver a intensidade de um incêndio, um *feed* do *twitter* sobre o incêndio, dados meteorológicos da área afetada, uma linha temporal do surto de incêndio, bem como datas sobre o alcance do destacamento dos bombeiros (número de pessoas e veículos de combate a incêndios). Os dados são derivados da Proteção Civil Portuguesa.
    - Como são usados os dados abertos: é uma plataforma, no setor de Justiça, sistema judiciário e segurança pública que reúne dados abertos da Proteção Civil Portuguesa e visualiza-os de uma forma clara através de um mapa. Além disso, combina dados meteorológicos, dados aeroespaciais e dados públicos tais como *feeds* do Twitter.
- SNS – Serviço Nacional de Saúde
    - Breve descrição: O Portal da Transparência agrega dados de saúde de várias entidades em Portugal e torna estes conjuntos de dados disponíveis ao público. O seu objetivo é criar transparência e aumentar a eficiência do Serviço Nacional de Saúde (SNS) e permitir o acesso a dados de saúde de alta qualidade a instituições, cidadãos e profissionais de saúde.
    - Como são usados os dados abertos: O Portal do SNS combina informações publicadas por diversos serviços nacionais de saúde instituições e os SPMS (*Shared Services of the Ministry of Health*), que está disponível ao público.
- Portugal – SmartCity Hub
    - Breve descrição: *Bitcliq* criou as plataformas *SmartCity Hub* com o objetivo de fornecer informações úteis para apoiar a mobilidade de turistas e cidadãos em Caldas da Rainha.
    - Como são usados os dados abertos: *SmartCity Hub* recolhe dados urbanos de bases de dados nacionais e locais para fornecer aos cidadãos e turistas informações sobre Caldas da Rainha. Os dados recolhidos são visualizados num mapa para mostrar onde se podem encontrar os pontos de interesse.

2.2 TÉCNICAS UTILIZADAS PARA A RECOLHA E EXTRAÇÃO DE DADOS

2.2.1 Extração de dados e os processos ETL e ELT

Temos hoje acesso a mais dados que nunca. Se pensarmos numa loja de retalho online, diariamente esta é atualizada: sabemos que os preços oscilam com frequência e se um conjunto de novos produtos entram em campanha um outro conjunto de produtos deixa de estar em promoção. A questão é: se queremos monitorizar esta informação diariamente para um conjunto alargado de retalhistas, ou seja, monitorizar esta informação em mais que um Website, como podemos tirar o máximo partido da atualização diária dos dados?

Segundo o artigo "*What is Data Extraction? Data Extraction Tools & Techniques*", um dos desafios reside em encontrar uma ferramenta de integração de dados que possa gerir e analisar muitos tipos de dados a partir de um conjunto de fontes em constante evolução. Mas antes que esses dados possam ser analisados ou utilizados, devem ser primeiro extraídos. É neste sentido, que examinaremos o significado do termo "extração de dados".

De acordo com o mesmo artigo, "a extração de dados é o processo de obter dados em bruto de uma fonte e replicar esses dados noutro local. Os dados em bruto podem derivar de várias fontes, tais como uma base de dados, uma folha de cálculo Excel, uma plataforma SaaS, Web Scraping, entre outras" (What is Data Extraction? Data Extraction Tools & Techniques, Stich A Talend Product (2023)). Assim, para a presente dissertação serão extraídos todos os dados em bruto de duas lojas de retalho online, ou seja, a nossa fonte de dados serão os dois Websites - Web Scraping[4]. "O *web scraping* é o processo de recolha de dados estruturados da web de uma forma automatizada" (Web Scraping – saiba o que é e para que serve, Pplware (2020, 11 dezembro)).

Segundo o artigo mencionado anteriormente, "os dados em bruto extraídos podem ser replicados para um destino, tal como uma base de dados, concebido para apoiar o processamento analítico online (OLAP). Isto pode incluir dados não estruturados, tipos de dados díspares, ou simplesmente dados mal-organizados. Uma vez os dados consolidados, processados e refinados, podem ser armazenados - localmente, na Cloud ou num sistema híbrido - para futuros processamentos ou transformações posteriores".

Isto diz-nos que para criar valor através de dados abertos em lojas de retalho online teremos de seguir os seguintes passos:

1. Extrair dados brutos das duas lojas de retalho online – fonte de dados Web
2. Replicar os dados e guardá-los após cada extração numa base de dados, por exemplo
3. Trabalhar os dados puros armazenados e que podem apresentar diferentes formas: dados não estruturados, tipos de dados díspares ou simplesmente dados mal-organizados
4. Depois de consolidar, processar e refinar os dados de interesse, armazenar os "novos" dados para futuros processamentos e transformações necessárias

[4] Sabe mais sobre Web Scraping (o que é e para que serve aqui) - https://pplware.sapo.pt/internet/web-scraping-saiba-o-que-e-e-para-que-serve/

Através do mesmo artigo, são dados alguns exemplos para tirar proveito da extração de dados, nomeadamente:

a. Uma organização que quer monitorizar a sua reputação no mercado. Isto pode exigir muitas fontes diferentes de dados, incluindo análises online em páginas web, estatísticas nas redes sociais e transações online. Uma ferramenta ETL (*Extract*, *Transform*, *Load*) pode extrair dados destas várias fontes e carregá-los para uma base de dados onde podem ser analisados e extraídos para se ter uma ideia da perceção da marca.
b. Recolha de vários tipos de dados de clientes para obter uma imagem mais clara dos clientes, dados financeiros para ajudar as empresas a acompanhar o desempenho e ajustar a estratégia, e dados de desempenho, que podem ajudar a melhorar os processos ou a monitorizar as tarefas.

É importante reter então que a extração de dados está relacionada com os processos de ETL (*Extract*, *Transform*, *Load*) e ELT (*Extract*, *Load*, *Transform*). Assim, "a extração de dados é o primeiro passo de ingestão de dados nos dois processos ETL e ELT. Estes processos fazem parte de uma estratégia completa de integração de dados, com o objetivo de preparar dados para análise ou Business Intelligence (BI)." – What is Data Extraction? Data Extraction Tools & Techniques. (Stich A Talend Product, (2023))[5].

Segundo mesmo o artigo "as principais diferenças entre ETL e ELT registam-se na quantidade de dados guardados e quando os dados são transformados". As figuras abaixo apresentadas - Figura 2 - Processo ETL – e a Figura 3 - Processo ELT, retratam as principais diferenças nos processos.

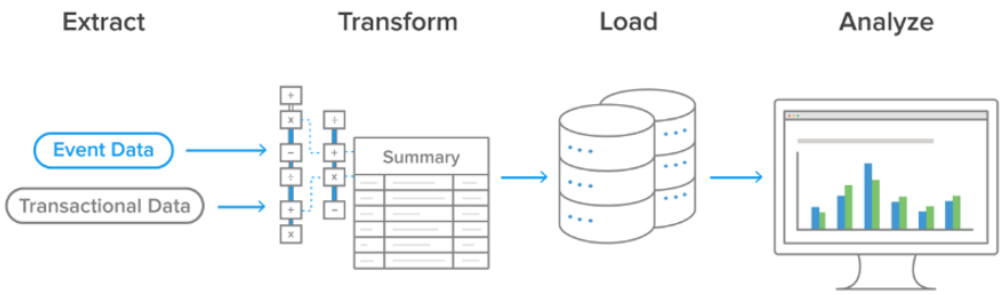

Figura 2 - Processo ETL

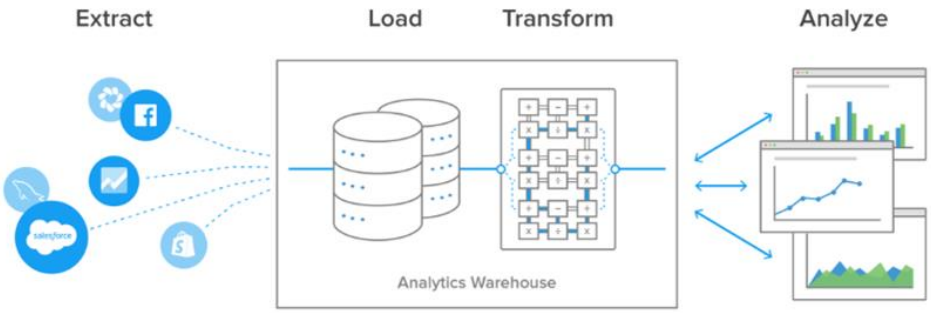

Figura 3 - Processo ELT

[5] Empresa pioneira em lançar o primeiro produto para integração de dados em modelo *open source*

Com ETL, a transformação de dados é feita antes de ser carregada para uma base de dados. Isto permite aos analistas e utilizadores empresariais obter os dados de que necessitam mais rapidamente, sem construir transformações complexas ou tabelas persistentes nas suas ferramentas de inteligência empresarial. Por outro lado, na abordagem ELT, os dados são guardados num repositório exatamente como são extraídos, sem qualquer transformação antes de serem carregados. Isto torna os trabalhos mais fáceis de configurar, porque apenas requer uma fonte de dados e um destino de dados.

No mesmo artigo, podemos analisar que as abordagens ETL e ELT no tema de integração de dados diferem em várias formas fundamentais, tais como:

- Tempo de extração - Demora significativamente mais tempo a obter dados dos sistemas fontes para o sistema de destino com abordagens ETL.
- Tempo de transformação – As abordagens ELT executam transformação de dados específicos, utilizando o poder computacional do sistema alvo, reduzindo os tempos de espera para transformações.
- Complexidade - As ferramentas ETL têm tipicamente uma GUI fácil de usar que simplifica o processo. O ELT requer um conhecimento profundo das ferramentas BI, quantidades de dados brutos, e uma base de dados que possa transformá-lo eficazmente.
- Suporte de armazenamento de dados – as abordagens ETL ajustam-se de forma mais prática a armazenamentos de dados locais direcionados a dados estruturados. O ELT é concebido para a escalabilidade de serviços em Cloud.
- Manutenção - ETL requer uma manutenção significativa para a atualização de dados no armazenamento de dados. Com o ELT, os dados estão sempre disponíveis em tempo quase real.

Importa referir ainda que segundo o artigo, “Tanto os processos ETL como ELT têm o seu lugar no panorama competitivo atual, e compreender as necessidades e estratégias únicas de uma empresa é fundamental para determinar qual o processo que irá proporcionar os melhores resultados.”

Neste sentido, abordaremos no processo metodológico uma abordagem ELT:

- *Extract* os dados das lojas online: Loja A e Loja B
- *Load*: guardaremos os dados puros/brutos
- *Transform*: Transformaremos os dados de forma a obter novos insights/conhecimentos

2.3 FACTUALIDADES SOBRE O PANORAMA DAS LOJAS ONLINE – OPORTUNIDADES DE EXPLORAR OS DADOS DOS CONCORRENTES

2.3.1 Evolução, representatividade e caracterização das empresas no e-commerce em Portugal

O presente subcapítulo pretende justificar a necessidade da importância de acompanhar e monitorizar as políticas praticadas no mercado retalhista online em Portugal. Para isso, primeiramente, é necessário conhecer os números concretos do valor da representatividade das empresas para a qual este estudo se direciona, ou seja, que percentagem de empresas registam presença em Portugal no mercado retalhista online. Assim, neste subcapítulo apresentamos informação disponível sobre o e-commerce nas empresas, em particular sobre as vendas por redes eletrónicas.

Segundo o inquérito da *CE*, *Information and Communication Technologies in Enterprises* de 2021 (os dados relativos ao comércio eletrónico pelas empresas respeitam ao ano anterior ao do inquérito. Por exemplo, no caso do inquérito de 2021, a informação refere-se ao ano de 2020), e analisando a figura abaixo apresentada - Figura 4 - Percentagem de empresas que receberam encomendas por redes eletrónicas e peso no volume de negócios - cerca de 16% das empresas portuguesas com 10 ou mais pessoas ao serviço receberam encomendas e venderam bens e serviços através de redes eletrónicas [6]a particulares (B2C), outras empresas (B2C) ou do Estado (B2G) durante o ano de 2020[7]. Ainda assim, é de registar menos três pontos percentuais do que a média da UE27 e menos 4 pontos percentuais que no ano anterior.

Além disso, podemos ainda concluir que as encomendas através de redes eletrónicas representaram 17% do volume de negócios das empresas em 2020, menos três pontos percentuais que no ano anterior.

[6] Encomendas recebidas através de um website ou app ou através de intercâmbio eletrónico de dados (EDI). O EDI é um conjunto de protocolos surgidos nos anos 70 que permite o intercâmbio de documentos (anteriormente existentes apenas em papel) entre empresas (peer-to-peer), recorrendo a serviços de transmissão de dados. Trata-se de meios utilizados sobretudo para B2B.

[7] Nesta análise considera-se somente as encomendas que representam pelo menos 1% do total do volume de negócios.

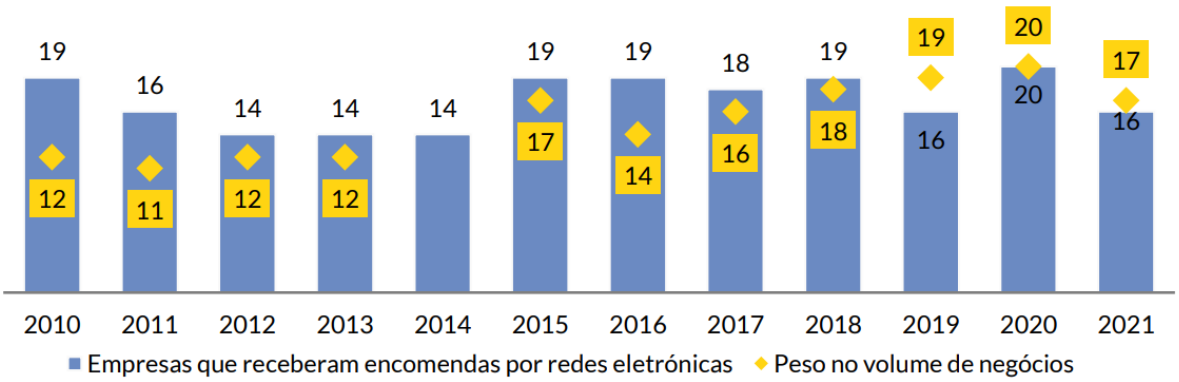

Figura 4 - Percentagem de empresas que receberam encomendas por redes eletrónicas e peso no volume de negócios

Outras notas importantes a reter de acordo com o inquérito mencionado, é que durante o ano de 2020, cerca de 13% das empresas receberam encomendas através de website/app, representando 5% do seu volume de negócios, como podemos ver na figura abaixo apresentada - Figura 5 - Percentagem de empresas que receberam encomendas por redes eletrónicas e tipo de receção e peso no volume de negócios. Embora as encomendas recebidas por websites/app estejam mais presentes nas empresas analisadas (13%), tendem a representar uma menor percentagem do seu volume de negócios (5%) quando comparadas com as EDI.

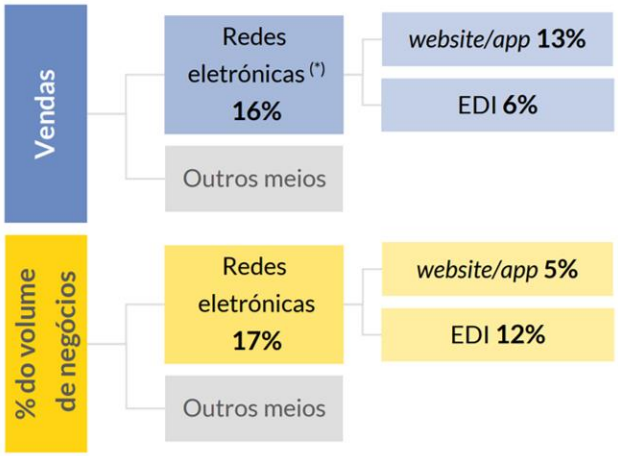

Figura 5 - Percentagem de empresas que receberam encomendas por redes eletrónicas e tipo de receção e peso no volume de negócios

Durante 2020, cerca de 12% das empresas receberam encomendas de particulares (B2C) e 7% realizaram negócios com outras empresas (B2B) ou com o Estado (B2G) – ver Figura 6 - Percentagem de empresas que receberam encomendas via website/app por segmento e peso no volume de negócios. O peso das encomendas recebidas por website no volume de negócios no segmento B2C foi de 2% e no segmento B2BG de 3%.

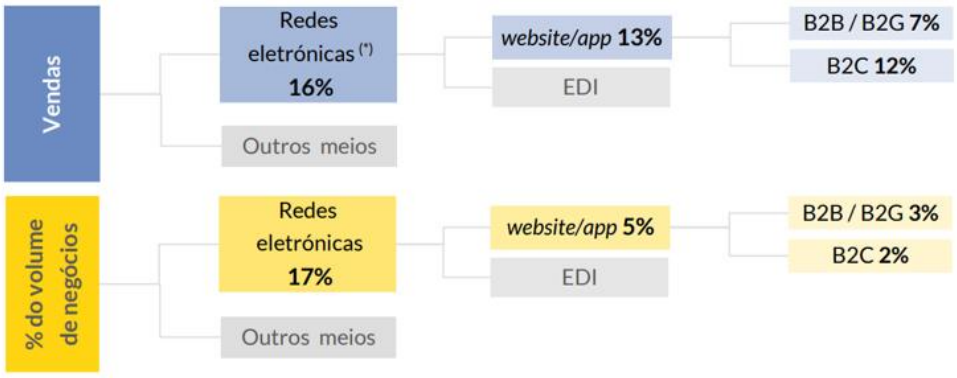

Figura 6 - Percentagem de empresas que receberam encomendas via website/app por segmento e peso no volume de negócios

Embora a maioria das empresas efetue vendas através do seu website/app, cerca de 6% das empresas analisadas rececionaram encomendas através de portais de comércio eletrónico ou plataformas digitais (via apps) utilizadas por várias empresas, como por exemplo Booking, hotels.com, eBay, Amazon, Amazon Business, Alibaba, Rakuten, Showroomprive, TimoCom – ver Figura 7 - Percentagem de empresas que receberam encomendas via website/app por tipologia.

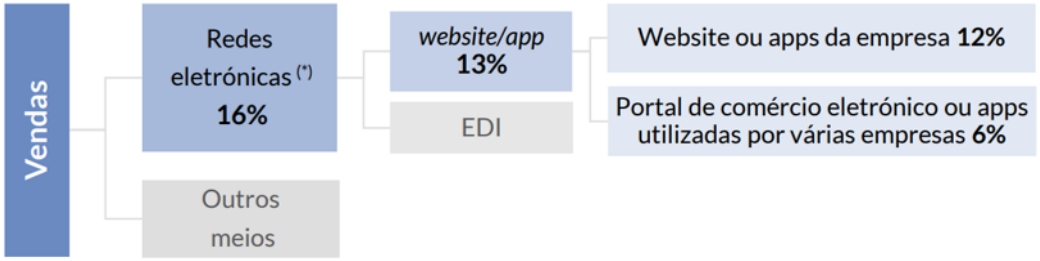

Figura 7 - Percentagem de empresas que receberam encomendas via website/app por tipologia

Por outro lado, importa analisar ao nível da dimensão empresarial e o setor de atividade.

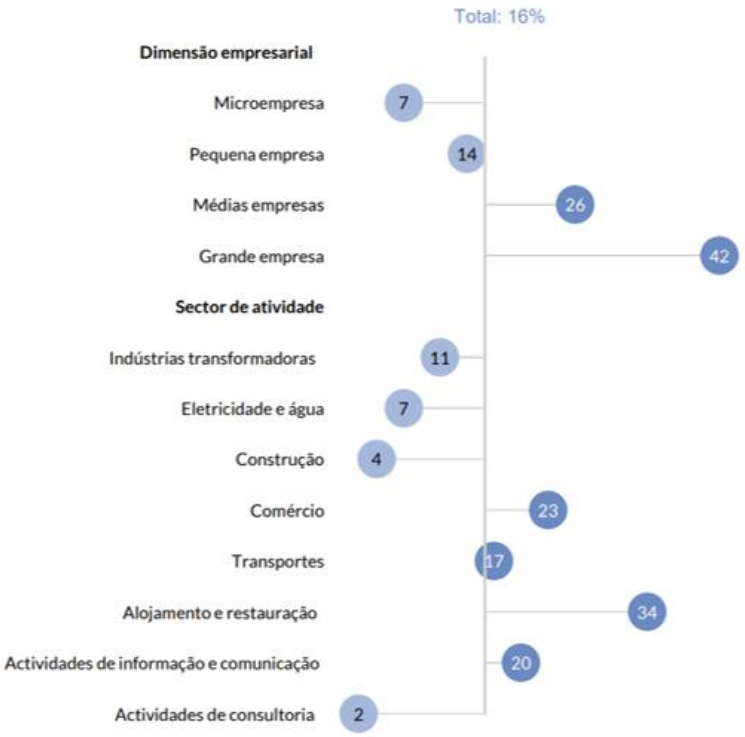

Figura 8 - Grupos de empresas que mais receberam encomendas por redes eletrónicas

Assim, concluímos da figura acima apresentada - Figura 8 - Grupos de empresas que mais receberam encomendas por redes eletrónicas - que a nível dimensional empresarial são as grandes empresas que registam valores mais altos e, por outro lado, a nível de setor de atividade é o setor do alojamento e restauração e o setor do comércio (justificação do tema da presente investigação) que apresentam valores mais altos.

2.3.2 A monitorização do mercado retalhista no canal online

É necessário entendermos quais são as necessidades e desafios que um retalhista precisa de ter quanto ao conhecimento quando tem como um canal de vendas o canal do e-commerce. Assim no que diz respeito à monitorização da concorrência, junto da SONAE foram recolhidos os seguintes insights, através de um questionário feito sobre as políticas que a empresa pratica no que diz respeito ao mercado online:

1. Relativamente às estratégias implementadas na loja online ao nível dos preços, promoções, entradas e saídas dos produtos, estas não são as mesmas que são implementadas nas lojas físicas

2. Na implementação de estratégias, a monitorização da concorrência apenas é feita nas seguintes vertentes
    a. Preço
    b. Descontos/Promoções nos produtos
    c. Grandes campanhas

3. A monitorização da concorrência é feita de forma diferente entre o mercado online e físico

4. Seguem as diferenças e as estratégias de monitorização implementadas no mercado físico e no mercado online e que diferem:
    a. "Os processos de shopping online são obtidos através de *service providers* especializados neste tipo de operações, enquanto o processo físico é suportado por uma equipa de profissionais que recolhem *in loco* os preços da concorrência."

5. A monitorização dos preços/descontos/campanhas é feita diariamente pela empresa

6. Técnicas/tecnologias utilizadas para cruzar a informação da concorrência com a da empresa
    a. São utilizados "tradutores" para mapear os artigos da concorrência à gama dos nossos produtos. Nos artigos com o mesmo EAN não é necessário desenvolver qualquer tipo de mapeamento. Todo o processo é suportado em bases de dados e algoritmos sofisticados que nos permitem medir o posicionamento competitivo.

7. Peso tem a informação da concorrência nas estratégias implementadas ao nível de preços, promoções, entradas e saídas dos produtos no mercado online – 5 numa escala de 1 a 5 (sendo o 5 o mais elevado)

8. Grau de maturidade das tecnologias (i.e.: nível de uso que a empresa já adota) na visualização de dados que a empresa tem implementadas para o suporte na tomada de decisão nas estratégias de pricing e marketing do mercado online – 4 numa escala de 1 a 5 (sendo o 5 o mais elevado)

Da revisão de literatura referente ao questionário respondido retiramos os seguintes insights:

- Justificação de criar mecanismos de análise às estratégias definidas pela concorrência e pelo mercado como um todo
- Possibilidade de estender as análises à oferta de produtos
- Já é implementada esta ou outras soluções idênticas à solução apresentada nesta dissertação, ou seja, é uma mais-valia o enriquecimento da partilha de conhecimento
- Percebemos que as respostas obtidas vão ao encontro do que é definido na arquitetura do use case apresentado
- Propostas para trabalhos futuros e como extensão do trabalho desenvolvido com este processo metodológico
- Mostra ser muito relevante
- Mostra que ainda existe margem de melhoria

Assim, o questionário (as questões e as respetivas respostas) pode ser consultado nos anexos - ANEXO A – Questionário SONAE.

2.4 ARQUITETURAS SERVERLESS E SERVIÇOS AMAZON WEB SERVICES (AWS)

O presente subcapítulo refere-se a um subcapítulo mais técnico. Suporta as ferramentas de desenvolvimento, processamento e armazenamento.

A computação *Serverless* representa uma mudança na estrutura de desenvolvimento da web, uma vez que permite a despreocupação sobre muitas questões "tradicionais" relacionadas com o armazenamento aplicacional.

Ao contrário do armazenamento Web tradicional, onde são necessários controlos sobre os servidores, configurações e manutenção, e em seguida, procede-se à implementação do código, em aplicações *Serverless*, não é necessário a gestão de servidores. Em contrapartida, apenas fornecemos o código e define-se apenas quando deverá ser executado.

Sem gerir qualquer servidor, normalmente existe uma redução de custos significativa, uma vez que não existe a capacidade de sobrecarga, os picos de tráfico de inputs tornam-se mais simples de contornar e não existe preocupação com a segurança do servidor da parte do desenvolvedor.

Assim, em 2019, investigadores da Universidade de Berkeley, num relatório técnico "*Cloud Programming Simplified: A Berkeley View on Serverless Computing*", já apontavam o modelo como o paradigma da computação em Cloud para o futuro.

Por todas estas razões, a computação *serverless* continua em ascensão, com cada vez mais empresas interessadas em integrá-la nas suas soluções tecnológicas.

No fundo, o desenvolvimento de aplicações baseadas na arquitetura *serverless* foca-se em fornecer grandes aplicações sem existir a preocupação com o fornecimento de servidores.

Apesar das ofertas que a arquitetura *serverless* oferece, há que ter em conta que esta também impõe limitações relativamente restritas sobre as linguagens de execução do código e bibliotecas disponíveis, sobre a dimensão (quer em memória quer em tempo de execução) dos processos de ingestão e processamento, afetando a pipeline da aplicação, e ainda sobre as formas de armazenamento e

consulta dos dados – os dados ficam armazenados em bases de dados que a empresa de *serverless* oferece. Sistemas de *Platform-as-a-Service* como Docker [8]oferecem a capacidade de ultrapassar algumas destas limitações.

As aplicações com arquiteturas serverless, carecem das seguintes características:

1. Automatização: esta é a palavra-chave
2. Serverless como *FaaS* (função como um serviço): Baseia-se na ideia de que os desenvolvedores passem a responsabilidade da gestão de um servidor para o provedor do serviço cloud (no nosso caso AWS) e apenas se concentrem no desenvolvimento das suas aplicações
3. Hoje os principais *players* no mercado de cloud computing já oferecem opções de serverless:
    - A Microsoft com o Azure Functions [9]
    - A AWS com o Lambda[10]
4. Principais vantagens ao utilizar serverless:
    - Tempo de processamento
    - Orçamento dispensado
    - Armazenamento de dados
    - Redução de custos
        - Recursos humano ao poupar o tempo dos profissionais que estariam ocupados com a arquitetura que armazena a aplicação
        - Consumo do provedor cloud escolhido – passa a pagar por event e não por tempo de execução da máquina
        - Tempo para o mercado: desenvolvedores têm maior dinamismo e facilidade na realização dos testes com a ferramenta
        - Energia elétrica, espaço físico e outros recursos necessários para sustentar
        - Pagamos apenas o código executado
    - A lógica que permite armazenar no servidor é feita pelo desenvolvedor da aplicação, mas ela apenas é executa quando um event a invoca
    - Arquitetura do serverless é totalmente gerida pelo seu provedor
    - Automatização de todo o código desenvolvido
5. Principais dificuldades, desafios, desvantagens ao utilizar soluções serverless
    - Bibliotecas de programação especificas
    - IPs (Internet Protocol) da AWS facilmente "conhecidos"

A oferta *serverless* explorada nesta dissertação será a Amazon Web Services AWS.

Assim, a arquitetura *Serverless*, particularmente o serviço AWS Lambda, regista uma ascensão de adoção por parte dos desenvolvedores uma vez que se torna uma maneira acessível e escalável de construir software. A framework Serverless, abstrai a configuração complexa que a AWS requer e permite que os desenvolvedores se foquem apenas no código.

[8] Docker é uma tecnologia de *containers open source* usada para empacotar, entregar e executar aplicações em containers Linux. Consulta mais informação aqui: https://www.docker.com/

[9] Sabe mais sobre Azure Functions aqui https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview

[10] Sabe mais sobre AWS Lambda aqui https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

Com a integração da framework Serverless com os serviços de AWS Lambda, torna-se mais simples desenvolver e implementar as funções AWS Lambda. Através desta *framework* é possível configurar todos os aspetos necessários para a interação com a AWS via conta associada; por exemplo, escrever a função lambda, determinar os limites de tempo da função, as configurações de memória e as permissões de IAM.

De facto, a integração deste *stack* tecnológico traduz potencialidades significativas, tais como:

- O desenvolvimento e implementação de funções AWS Lambda corretamente com o Framework Serverless
- O conhecimento dos fundamentos e das opções avançadas da AWS Lambda
- A implementação de soluções que interagem com S3, API Gateway, EC2 & CloudWatch e Amazon RDS
- A configuração do ficheiro YAML[11] (YAML Ain't Markup Language), bem como a gestão de todo o seu desenvolvimento através de código

[11] Sabe mais sobre ficheiros YAML aqui https://www.treinaweb.com.br/blog/o-que-e-yaml

3. METODOLOGIA

O presente capítulo visa a implementação de um processo metodológico qualitativo[12] dado que se objetiva a compreensão do valor de dados abertos. No âmbito da implementação de um caso prático, pretende-se desenvolver uma ferramenta de recolha e reutilização de dados abertos, especificando o desenho da arquitetura e o seu desenvolvimento. Com este trabalho pretende-se aprofundar o conhecimento na área e caracterizar quer o processo quer o produto final com vista a compreender a sua utilidade como ferramenta sistemática de larga escala.

Assim, o presente capítulo, para uma leitura mais clara, encontra-se dividido em três subcapítulos:

- *3.1. Contexto e objetivos do processo metodológico* - contexto e propósito da pesquisa metodológica inserido no tema da investigação, recorrendo ao caso de uso
- *3.2. Arquitetura da ferramenta* - desenho da arquitetura e fundamentos da *stack* tecnológica utilizada
- *3.3. Desenvolvimento da ferramenta* - *know-how* técnico de todos os componentes necessários que constituem a ferramenta

3.1 CONTEXTO E OBJETIVOS DO PROCESSO METODOLÓGICO

A metodologia da presente investigação procura desenvolver uma ferramenta de recolha e reutilização de dados abertos, documentando assim todo o processo e mostrando soluções para os problemas característicos da área.

Analisando a imagem abaixo - Figura 9 - Transformar dados de uma loja online em novas análises do mercado - esta representa, de uma forma alegórica, o que temos (1), onde queremos chegar (3) e como chegaremos (2).

Figura 9 - Transformar dados de uma loja online em novas análises do mercado

[12] Ver diferença de pesquisas Quantitativas e pesquisas Qualitativas em www.diferenca.com/pesquisa-quantitativa-e-pesquisa-qualitativa

Assim, analisando a imagem apresentada, de uma loja online que apresenta os seus produtos iremos implementar uma ferramenta capaz de extrair os dados, tratá-los e reutilizá-los para análises do mercado e assim apresentar por exemplo, a análise ao preço por Kg e desconto num determinado período temporal. Para isso é precisa a extração do campo preço por kilo dos produtos e o desconto nesse mesmo período de análise.

Como vimos no subcapítulo 2.3.2 A monitorização do mercado retalhista no canal online**,** os retalhistas com negócios online, monitorizam o mercado e os seus concorrentes a partir dos dados abertos disponibilizados nas lojas dos concorrentes, construindo análises sobre esses dados. Além disso, com base na revisão da literatura realizada, sabemos que recorrem a estes serviços através de *services providers* e que existem alguns produtos/serviços fora de Portugal. Assim, torna-se evidente a ausência de uma rede de conhecimento nesta área dentro das empresas com negócios online.

Neste seguimento, a construção de uma plataforma de extração de dados abertos e a criação de valor através da reutilização dos mesmos, espelha uma oportunidade de negócio direcionado ao mercado retalhista online em Portugal.

Para o desenvolvimento do caso prático, a recolha de dados em duas lojas online de retalho em Portugal (Loja A e a Loja B), representa vantagens no seu processo, nomeadamente:

- conhecemos as variantes do negócio de retalho em Portugal e estamos mais próximos de como funciona o negócio
- o facto de podermos aceder aos Websites de cada retalhista com IP's portugueses, pode representar uma abertura de acesso mais simples e com menos custos associados aos dados publicamente disponíveis online

Posto isto, e já contextualizada a necessidade de desenvolver aplicações dentro desta temática, com o presente processo metodológico, objetivamos:

- Elencar todos os processos necessários ao *Know-how* (conhecimento, saber fazer)
- Apresentar o nível de complexidade no desenvolvimento e desmistificar os conceitos das tecnologias associadas em abordagens de recolha e reutilização (transformação e construção de análises) de dados abertos
- Analisar como é que as empresas estão predispostas a tornar acessível a recolha da informação que disponibilizam nas lojas online, ou seja que barreiras é que criam no acesso à recolha de dados para que estes sejam usados, reutilizados e redistribuídos por qualquer pessoa[13].
- Adquirir o dito conhecimento técnico
- Entender o nível de paridade no mercado e o valor económico que os dados abertos conferem

Com foco no objetivo primordial da investigação - comprovar a criação de valor através da reutilização de dados abertos - iremos implementar o ciclo da recolha de informação presente na figura - Figura 1 - Ciclo da recolha de informação.

[13] Dados abertos em Portugal

De salientar que o ciclo da recolha de informação estudado reflete todas as áreas de conhecimento que irão ser abordados no processo metodológico da investigação.

Como referido anteriormente, o processo é transversal e independente do contexto do tema, contudo para concretizar de uma forma mais clara cada componente iremos detalhar cada processo recorrendo ao caso de uso em questão.

Contextualizando o ciclo da recolha de informação para o caso prático em questão, primeiramente são então reunidos os requisitos. De notar que este foram espelhados no mesmo subcapítulo que o estudo do ciclo da recolha de informação.

Assim, importa a nível técnico começar pela "Identificação das fontes" de interesse. A Loja A e a Loja B, representam as nossas fontes de informação de interesse e, para uma correta identificação do(s) websites(s) de interesse deve-se conseguir efetuar uma análise rígida as seguintes variáveis:

- Veracidade da informação do website: importância de entendermos se estamos perante uma fonte de informação que disponibiliza informação verídica
    - Como já mencionado anteriormente no capítulo 1. Introdução, a Loja A e a Loja B, representam retalhistas em Portugal de renome, com uma presença forte e uma expressão coesa no mercado de retalho em Portugal
    - Desta forma, podemos concluir que a sua informação é verídica. Ou seja, corresponde à informação das práticas reais praticadas no setor
- Caracterização da informação: Localização da informação, autoridade para aceder à informação, website estático ou dinâmico[14], se for dinâmico com que frequência é atualizada a informação
    - Localização da informação: dois Websites target
    - Fonte aberta e dados abertos, ou seja, são totalmente acessíveis por terceiros e disponibilizados a qualquer pessoa sem qualquer tipo de restrição
    - Os dois Websites caracterizam-se como dinâmicos, uma vez que estes sofrem constantes alterações de conteúdo como a criação de novas páginas com produtos, alteração de características dos produtos, entre outros. Isto sem a necessidade de alteração no código fonte por parte do profissional responsável pela monitorização da página
- A forma de estruturação e organização da informação no *Website*: importância de perceber como iremos receber os dados
    - Um produto tem a si associado atributos como o nome, marca, preço, entre outros.

[14] Ver diferenças entre websites dinâmicos e estáticos em https://www.agenciamacan.com.br/blog/site-dinamico-vs-estatico-qual-a-diferenca-entre-eles

- Identificação dos dados/informação alvo a extrair: identificar as características do produto que são de interesse extrair, como descrito na Tabela 1 - Características de interesse do produto a extrair:

Característica de interesse do produto a extrair	Exemplo
Nome	Bolacha maria doces
Marca	Vieira
Quantidade	-
Conteúdo	Embalagem de 300g
Medida de unidade	Kg
Descrição	-
Imagem	
EAN	5601008100539
Categorias	Mercearia; bolachas biscoitos e bolos; bolachas maria e torrada
Preço	1.79
Preço sem desconto	2.54
Preço por medida de unidade	5.97

Tabela 1 - Características de interesse do produto a extrair

Nota: é importante referir que este é um processo meramente inicial, uma vez que quando efetuamos pedidos de informação à fonte, a resposta ao pedido pode ser feita de várias formas e por isso limita ou não a seleção de informação a extrair. Poderemos ver mais sobre este tópico no subcapítulo 3.3 Desenvolvimento da ferramenta.

Assim, depois de analisar a fonte de informação é necessário passar à "Recolha de informação". Este tópico exige conhecimentos técnicos, nomeadamente conceitos e processos que envolvem a recolha/extração/pedido de dados a websites. Serão espelhados no subcapítulo 3.3.2 Processo de Recolha dos Dados e Processo Tratamento de Dados.

Com os dados extraídos na sua forma mais "pura" (tal como se encontram no *Website*), é necessário fazer uma triagem dos ficheiros recolhidos de forma a identificar gralhas, problemas durante a recolha ou alterações de organização nos Websites – referíamo-nos assim ao step do ciclo da recolha de informação "Processar a informação".

Neste processo é dividido o texto em blocos e convertida a informação relevante em variáveis dentro da plataforma. Abordaremos este tópico em detalhe nos capítulos seguintes: arquitetura e desenvolvimento da ferramenta.

De seguida passamos a "analisar informação e produzir inteligência" onde é definida uma estruturação de armazenamento dos dados. É importante referir que o presente step e o step anterior "processar a informação", dada a sua complexidade, ditam o potencial e o sucesso de transformar os dados em novos insights. Muitas vezes os dados brutos apresentam má qualidade e formas de leitura complexas, pelo que o processo de limpeza, transformação e modelação não é uma tarefa simples e daí ser de extrema relevância prestar atenção a estes dois steps.

Por último, depois de termos a informação devidamente triada, estandardizada e direcionada para as áreas de interesse – denominado o step de "produzir relatório" – é dar asas à criatividade e criar processos de reutilização de dados, por exemplo, aplicar *Data Science* utilizando Python para analisar os dados, obter novos conhecimentos, desenvolver novos produtos, entre outros.

De forma sucinta, a imagem abaixo apresentada, Figura 10 – Processo metodológico na transformação de dados abertos para a criação de valor, retrata o processo metodológico experimental para desenvolver uma ferramenta passível de capitalizar em outras indústrias (todas as que tenham websites de lojas online), focando-se essencialmente em dois grandes processos técnicos: Recolha dos dados e Tratamento dos dados.

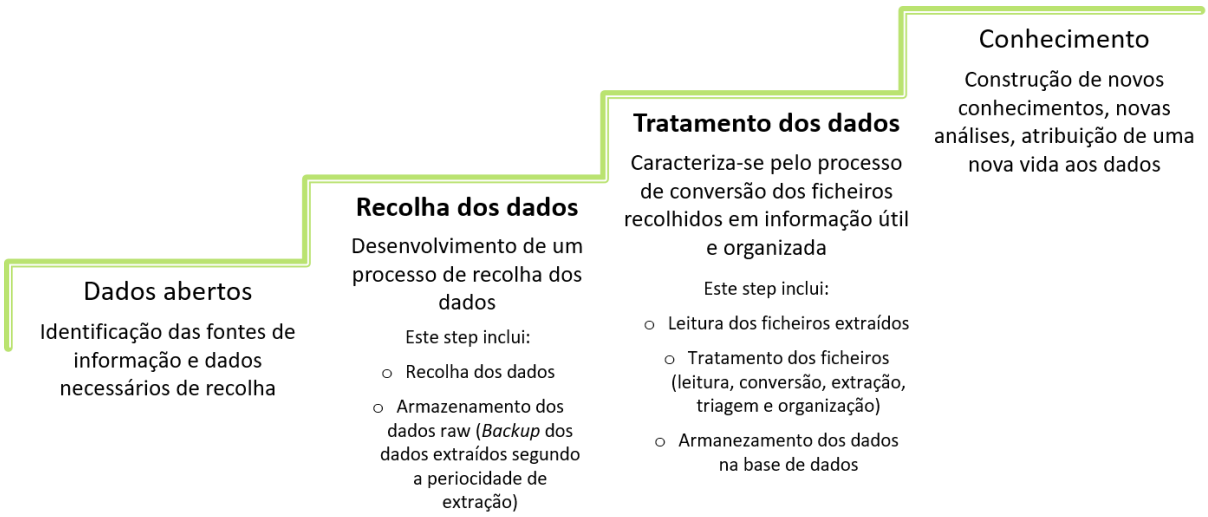

Figura 10 – Processo metodológico na transformação de dados abertos para a criação de valor

Analisando a imagem descrita, os steps realçados “Recolha dos dados” e “Tratamento dos dados” representam os dois grandes desenvolvimentos técnicos que procuraremos abordar na presente investigação:

1. Recolha dos dados
    a. Recolha da informação dos produtos dos *Websites*; por exemplo: nome, preço, marca, imagem, preço com promoção, tipo de promoção, capacidade, categorias (hierarquia), sem/com glúten, entre outros.
    b. Armazenamento da informação extraída (sem processos de tratamentos, segundo a periocidade de extração). Ou seja, sempre que se extrai informação esta é devidamente armazenada numa pasta indicada pelo mês, dia ou hora/minutos do momento de extração. Este processo é baseado numa arquitetura de processamento ELT. Para saber mais sobre esta abordagem deve consultar o subcapítulo 2.2.1 Extração de dados e os processos de ETL e ELT.
2. Tratamento dos dados
    a. *Parse* da informação extraída. Após a informação extraída, esta deve ser devidamente trabalhada para um armazenamento de dados mais limpo e estruturado a fim de os dados estarem prontos a serem reutilizados. Este é um processo de interpretação, digitalização e conversão dos dados extraídos uma vez que a informação vem tipicamente codificada sob a forma de texto, strings.
    b. Armazenar a informação identificando a data de extração para possíveis análises temporais.

Concluindo a introdução ao contexto e objetivos do processo metodológico, é importante realçar que bibliotecas de recolha, processamento e armazenamento de dados em *Python*, serviços de *Serverless Computing AWS,* lógica e serviços de fluxo de trabalho, construção de análises e visualização aos dados representam os conceitos mais trabalhados no presente processo metodológico e que suportam ambos os desenvolvimentos mencionados.

3.2 ARQUITETURA DA FERRAMENTA

A arquitetura pretende especificar a forma de como é que o processo lógico funcionará, suportado pelas tecnologias da área.

O uso de arquitetura para representar soluções de software foi incentivado principalmente por duas tendências (GARLAN e PERRY, 1995; KAZMAN, 2001): (1) o reconhecimento por parte dos líderes de projetos que o uso de abstrações facilita a visualização e o entendimento de certas propriedades do software, e (2) a exploração cada vez maior de *framework* visando diminuir o esforço de construção de produtos através da integração de partes previamente desenvolvidas.

Com este pressuposto, um dos objetivos no desenho da arquitetura é delinear um sistema de processos robusto e que permita uma experiência de desenvolvimento sem grandes altos e baixos, seguindo uma linha de raciocínio lógica.

Um dos pressupostos iniciais foi o de construir uma aplicação *Serverless* em *Cloud AWS* e, com persistência de dados numa base de dados *Amazon RDS*.

Esta solução, caracteriza-se como um projeto suportado por um *Software As a Service (SaaS)*, desenvolvendo um *bot* de recolha automática de dados, conseguindo obter um *dataset* rico para reutilização da informação. Além da criação de processos escaláveis, adaptáveis e ágeis (tanto na recolha como no armazenamento dos dados), a disponibilidade dos dados para possíveis reutilizações tornar-se-á mais ampla, rica e precisa; a construção de análises para um melhor conhecimento do estado do mercado - num contexto de monitorização dos concorrentes com lojas online – é um dos exemplos de reutilização.

Deste modo, a figura abaixo Figura 11 - Arquitetura Alto Nível Recolha dos Dados e Tratamentos dos Dados apresenta: o processo de recolha dos dados (através de Web Scraping e Web Crawling[15]), armazenamento dos dados *Raw (*ficheiros de texto e HTML*)*, tratamento dos dados através de processos de *Parse* aos ficheiros extraídos, obtendo como output o *dataset* pretendido.

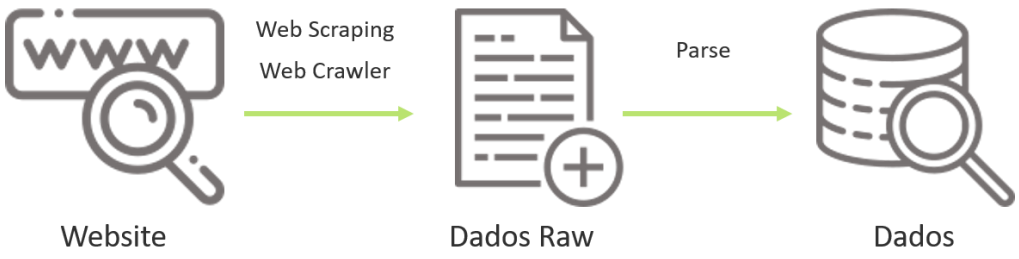

Figura 11 - Arquitetura Alto Nível Recolha dos Dados e Tratamentos dos Dados

[15] Podes saber mais sobre a diferença entre Web Scraping e Web Crawling aqui: https://brightdata.com/blog/leadership/web-crawling-vs-web-scraping

Por outro lado, para obtermos conhecimento, poderemos aceder aos dados e construir análises através de Bibliotecas Python, como podemos na figura abaixo Figura 12 - Arquitetura Alto Nível Atribuir Conhecimento aos Dados:

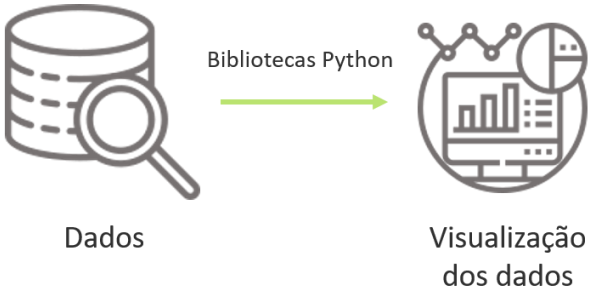

Figura 12 - Arquitetura Alto Nível Atribuir Conhecimento aos Dados

Posto isto, quando tentamos solucionar um problema, é possível identificar diversas soluções passíveis de ser utilizadas com a finalidade de o resolver. Porém, outros fatores como o custo e eficiência influenciam o desenho da solução a ser adotada. Neste sentido, no contexto de desenvolver a aplicação capaz de dar resposta ao problema identificado, é crucial analisar os requisitos da construção do software. De acordo com o artigo “Fundamentos de Arquitetura de Software”, várias soluções computacionais podem ser definidas para atender aos requisitos, mas é feita uma análise para definir a mais adequada no contexto do desenvolvimento da aplicação.

Assim, começaremos por definir todo o stack tecnológico envolvente no desenvolvimento da aplicação. A tabela que se segue - Tabela 2 - Stack tecnológico da ferramenta – apresenta as especificações das tecnologias:

Especificação	Tecnologia
Linguagem de programação	Python
Framework de desenvolvimento	PyCharm
Páginas Web	HTML, ASP.NET, ASPX
Framework de deployment	Serverless Framework
Armazenamento da aplicação	AWS Lambda e AWS Step Function
Armazenamento dos dados *Raw*	Amazon S3
Sistemas de base de dados	PostgreSQL
Armazenamento da base de dados	Amazon RDS
Framework da base de dados	DBeaver
Segurança	IAM
Monitorização	AWS Cloud Watch e AWS SNS

Tabela 2 - Stack tecnológico da ferramenta

Assim, toda a ferramenta será desenvolvida na linguagem *Python*[16]**.** Existem variadas razões para justificar a escolha desta linguagem de programação, mas podemos centrar-nos apenas nas seguintes e validaremos o potencial de optar por esta linguagem:

- Fonte livre e aberta
- Linguagem de alto nível
- Grande comunidade e, por isso, grande leque de portais de ajuda entre os desenvolvedores
- Portátil e extensível (Python é suportado pela maioria das plataformas presentes no mercado, desde o Windows, Linux, Macintosh, Solaris, Playstation, entre outras)
- Grande variedade de bibliotecas desenvolvidas (recolha de dados, processamento de dados, inteligência artificial, entre outras)
- Bibliotecas direcionadas a tema de *Big Data* (Pydoop, Dask, Pyspark, entre outras)
- É a linguagem líder na disciplina Ciência de Dados (bibliotecas como Pandas, Numpy, Matplotbib, entre outras)

[16] Poderá saber mais sobre Python em: https://www.python.org/

Relativamente à framework de desenvolvimento a aplicação foi programada no ambiente de desenvolvimento *PyCharm*[17]. Trata-se de um ambiente de desenvolvimento integrado usado em programação de computadores, especificamente para a linguagem de Python.

No que diz respeito à *framework* de *deployment*, foi utilizada a *Serverless Framework*[18], sendo esta responsável pela comunicação com a *cloud AWS*, através de ficheiros de configuração *YAML*, conseguindo especificar todos os recursos necessários que devem ser criados do lado da *AWS*. Neste sentido, aplica-se então o conceito de *serverless*, onde nos compete apenas desenvolver código do nosso lado, configurar o arquivo *YAML*, e a *Serverless Framework* comunicará o necessário a nível de recursos (servidores, base de dados, entre outros) com a *Cloud AWS*.

Já do lado da *AWS*, relativamente ao armazenamento da aplicação, o serviço de *Step Function* [19]será responsável pelo fluxo de trabalho que "lançará" as *Funções Lambda* [20]que contêm o código necessário para a execução da aplicação.

Os dados em bruto (*raw*), ou seja, aqueles extraídos diretamente das fontes de dados serão armazenados no serviço Amazon S3 [21]sob a forma de bases de dados standard.

Estas bases de dados (depois de processos de parse, estruturação e organização dos dados) serão construídas com base na ferramenta PostgreSQL [22]que permite armazenar e aceder aos dados de forma expedita e otimizada. Quanto ao armazenamento das tabelas (tabela dos produtos da Loja A e a tabela dos produtos da Loja B), o *Amazon Relational Database Service* (*Amazon RDS*[23]), facilitará a configuração, a operação e a escalabilidade da base de dados relacional em *Cloud AWS*.

Relativamente à segurança, gestão de acesso dos utilizadores e comunicação entre serviços, estas tarefas são realizadas utilizando o serviço Identity and Access Management (Amazon IAM[24]).

Por último, tarefas de monitorização e notificações serão da responsabilidade dos serviços de *Amazon CloudWatch* [25] e *Amazon SNS*[26], respetivamente. Como mencionado anteriormente, o detalhe das especificações de como os serviços funcionarão entre si, veremos nos tópicos seguintes.

[17] Poderá saber mais sobre PyCharm em https://www.jetbrains.com/pycharm/

[18] Poderá saber mais sobre Serverless Framework em https://www.serverless.com/

[19] Poderá saber mais sobre AWS Step Functions em https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html

[20] Poderá saber mais sobre AWS Lambda em https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

[21] Poderá saber mais sobre Amazon S3 em https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html

[22] Poderá saber mais sobre PostgreSQL em https://www.postgresql.org/docs/

[23] Poderá saber mais sobre Amazon RDS em https://aws.amazon.com/pt/rds/

[24] Poderá saber mais sobre IAM AWS em https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

[25] Poderá saber mais sobre Amazon CloudWatch em https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html

[26] Poderá saber mais sobre Amazon SNS em https://docs.aws.amazon.com/sns/latest/dg/welcome.html

Por conseguinte, a figura abaixo apresentada - Figura 13 - Relação entre a Framework Serverless e a plataforma de desenvolvimento - retrata um diagrama do fluxo da arquitetura no que diz respeito aos serviços necessários para a execução da ferramenta interligados com a Framework Serverless. Este mostra os setores críticos da aplicação e as etapas que os serviços percorrerão, dando relevância ao software que foi desenvolvido bem como a análise que foi realizada para entender se a arquitetura desenhada soluciona as questões que se pretendem ver respondidas.

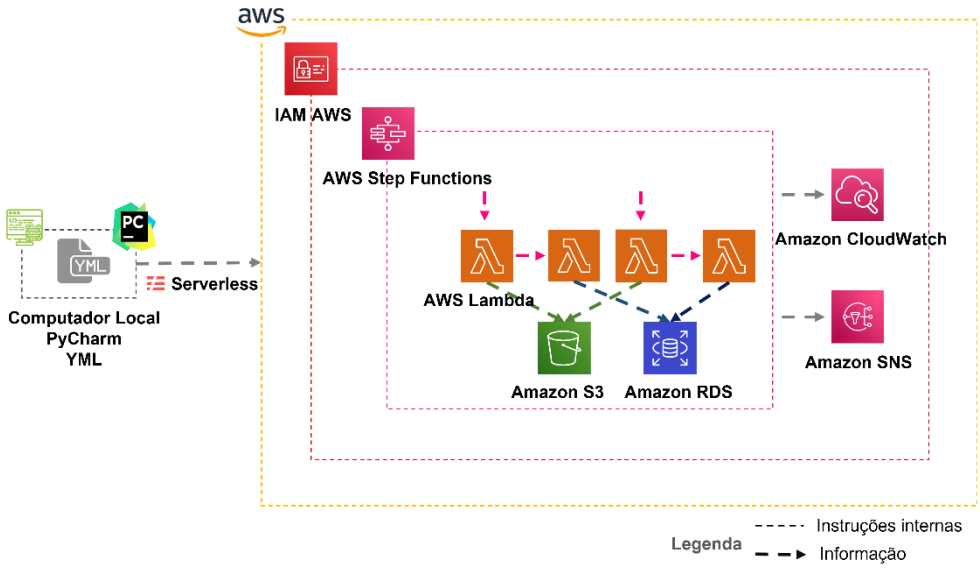

Figura 13 - Relação entre a Framework Serverless e a plataforma de desenvolvimento

Assim, o diagrama acima ajuda-nos a entender de forma sucinta como é que os serviços da AWS comunicam e se interligam. De salientar que cada bloco representa um domínio ou característica técnica e que pode ser encontrada na maioria das arquiteturas *Serverless*.

A tabela seguinte - Tabela 3 - Especificação dos recursos numa perspetiva de Arquitetura Alto Nível - procura detalhar como é que os recursos e serviços da AWS se interligam entre si:

Recursos e serviços AWS	Descrição
Computador Local e PyCharm	Localmente, a partir do nosso computador, programamos toda a nossa ferramenta a partir do ambiente de desenvolvimento *PyCharm*
Ficheiro *YAML*	Configuramos todos os recursos necessários no arquivo *YAML* e, utilizando *Serverless* Framework, esta consegue comunicar com o *Provider da Cloud Serverless* (AWS) e dar instruções para que sejam criados todos os componentes e serviços previamente especificados no arquivo YAML
Serverless Framework e Cloud AWS	Como esta comunicação entre a Framework Serverless e a Cloud AWS são criados então todos os recursos programados
Amazon IAM	Gestão de acessos a utilizadores, segurança e permissões são definidos através do serviço Amazon IAM
Amazon Step Function e Amazon Lambda	O serviço de Amazon Step Function funcionará como a forma de automatizar os processos de negócio nomeadamente, a recolha dos dados e tratamento dos dados. De salientar, e como podemos ver através da arquitetura acima descrita, existirão duas funções para cada processo: cada função representa o processo associado a cada fonte de informação. Assim, temos uma função para a recolha de dados e tratamento dos dados direcionada para a Loja A e, por outro lado, existirá uma função para a recolha de dados e tratamento dos dados direcionada para a Loja B. Justificaremos em detalhe estas especificações na Arquitetura de Baixo Nível
Amazon Lambda e Amazon S3	Processos de negócio associados à recolha dos dados, armazenam os dados raw (puros) no serviço Amazon S3
Amazon Lambda e Amazon RDS	Processos de negócio associados ao tratamento dos dados, armazenam os dados estruturados organizados e limpos no serviço Amazon RDS
Amazon CloudWatch e Amazon SNS	Todas as tarefas de monitorização (tarefas de gestão de logs) e SMS/emails serão da responsabilidade dos serviços Amazon CloudWatch e Amazon SNS

Tabela 3 - Especificação dos recursos numa perspetiva de Arquitetura Alto Nível

Uma vez detalhado o stack tecnológico (todas as tecnologias necessárias à implementação) bem como a especificação dos recursos e serviços da AWS e como se interligam entre si, importa agora entender a sua Arquitetura Baixo Nível.

A imagem abaixo – Figura 14 - Arquitetura Baixo Nível - retrata a sua arquitetura. São ilustrados os dois processos principais (Recolha dos dados e Tratamento dos dados) por retalhista e, em blocos, os serviços que os sustentam: ao nível da monitorização e tarefas de notificações, processo do fluxo de trabalho, armazenamento de dados raw e armazenamento dos dados (posterior a processos de tratamentos de dados - Parse e estruturação dos dados). Além dos processos principais, e dos serviços que os suportam, podemos acompanhar a imagem abaixo da arquitetura com a legenda que procura dar ao leitor uma visão e explicação da arquitetura mais clara e direta.

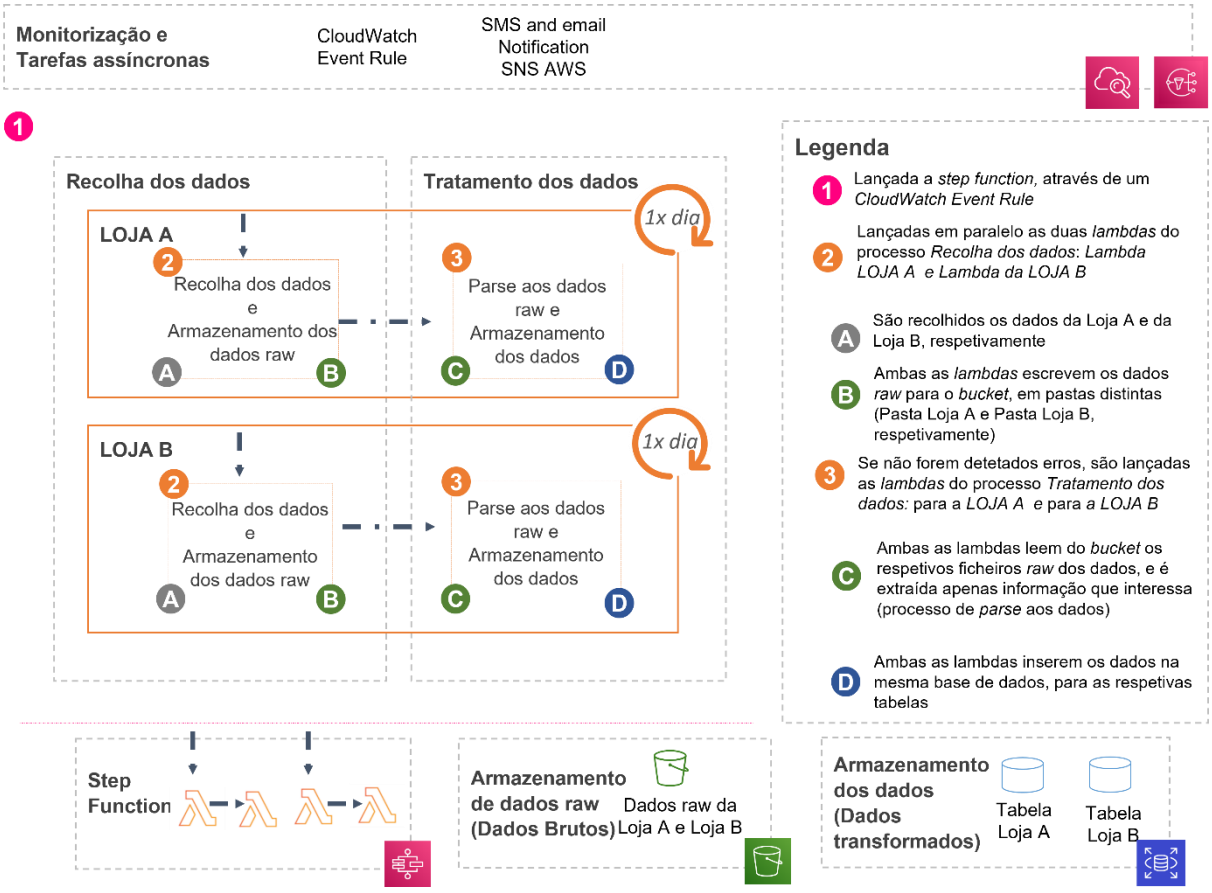

Figura 14 - Arquitetura Baixo Nível

Sequenciando a arquitetura da ferramenta, teremos os seguintes passos:

1. Com o suporte do serviço AWS Step Function, é criada uma máquina de estados que automatizará os processos de negócio, nomeadamente a recolha dos dados e o tratamento dos dados. Através do *CloudWatch* é definido um *event rule* que desencadeia diariamente a máquina de estado criada para a execução da aplicação

2. São lançadas em paralelo as duas lambdas que executarão sobre as páginas retalhistas, respetivamente. A função de cada lambda (2) passa por recolher todos os dados dos produtos e armazená-los

    a. É executado o código de *Scraping* aos dados (acesso às lojas online e recolha dos dados)
    b. Os dados dos produtos (em formato de *JSON ou HTML*) são guardados num determinado *bucket* da *AWS S3* (denominadas dados *raw* ou dados brutos/puros) (escrita no *bucket*)
3. Mediante o estado do *Tratamento de Erros* na máquina de estados, é ou não lançada a segunda lambda de cada loja: as lambdas responsáveis pelo tratamento dos dados
    c. É executado o código que procederá ao *Parsing* sobre os ficheiros recolhidos (*JSON* ou *HTML*), com a finalidade de adicionar os produtos na base de dados. Na prática os dados dos ficheiros serão extraídos e divididos em dados individuais (leitura do Bucket de S3)
    d. É executado o código onde cada produto é analisado numa estrutura de dados *Python* e a informação do produto é guardada na AWS RDS (escrita na base de dados *PostgreSQL*)

De salientar que durantes todos os processos, são sempre registados os logs através do CloudWatch. AWS CloudWatch será usada tanto para *Schedule* (neste caso, será feito uma execução uma vez por dia) como para tarefas de *logs* (gestão de eventos).

Por outro lado, notificações relativas a erros serão enviadas por SMS e emails através no serviço Amazon SNS.

Da presente arquitetura, é importante ainda especificar a necessidade de existir uma lambda para cada loja online (fonte de informação). Esta necessidade prende-se ao facto de cada *Website* estar estruturado naturalmente de forma diferente e, por isso, influencia a forma como se extrai a informação. O código varia quer na aquisição da página HTML e dos dados na sua forma mais "pura" (fundamental ou *raw*), quer no processamento destas páginas HTML (conhecido como *Parse*) em objetos que só contêm as variáveis de interesse (nome, marca, preço, desconto, tipo, categoria, etc.). E neste sentido, que quando trabalhamos com a extração de dados abertos em Websites necessariamente é criado um algoritmo de ingestão por fonte de informação uma vez que dificilmente existem dois Websites estruturados exatamente da mesma forma.

3.3 DESENVOLVIMENTO DA FERRAMENTA

O presente subcapítulo procurará detalhar em pormenor todas as etapas do desenvolvimento ao longo da ferramenta, bem como partilhar os desafios e as soluções encontradas para os contornar. Para uma explicação objetiva, clara e simples, considerando o nível de complexidade de minuciar todos os procedimentos da ferramenta, o presente subcapítulo encontra-se dividido da seguinte forma:

- No subcapítulo 3.3.1 Pré-configurações necessárias no desenvolvimento da plataforma, começaremos por abordar as pré-configurações necessárias, antes de desenvolver código, e que dizem respeito ao ambiente de desenvolvimento, ambiente de deployment e a configuração à ferramenta de administração de base de dados *DBeaver* para uma integração com a base de dados *AWS RDS*. Além disso será discutida a comunicação entre a *Serverless Framework* e a *cloud AWS.*
- No subcapítulo 3.3.2.1 Processo de Recolha dos dados, procuraremos abordar a informação relativa a todas as etapas relacionadas com a recolha dos dados às duas lojas online (*Websites*): Loja A e Loja B. De realçar que é este o processo mais crítico, uma vez que é um processo que depende da estrutura própria de cada *Website*, e que dita o nível de complexidade em extrair dados e o nível de abertura do *Website* em relação à disponibilidade de recolha dos dados abertos (podendo criar limitações ao longo do processo)
- No subcapítulo 3.3.2.2 Processo de Tratamento dos dados, detalharemos a informação relativa a todas as etapas relacionadas com o tratamento dos dados que inclui a análise aos ficheiros extraídos dos websites, parse da informação, organização e individualização dos dados para os armazenar numa base de dados, prontos a serem reutilizados
- Por último, no subcapítulo 3.3.3 Fluxo de trabalho entre os dois processos: Recolha dos dados e Tratamento dos dados & Monitorização e Notificações, estudaremos o fluxo para automatizar os dois processos: Recolha dos dados e Tratamento dos dados, bem como questões de monitorização e notificações

3.3.1 Pré-configurações necessárias no desenvolvimento da plataforma

Para dar início ao processo de desenvolvimento é necessário configurar o ambiente de desenvolvimento (para que seja possível desenvolver o código no computador local), o ambiente de deployment (implementar todos os recursos, código e configurações na cloud AWS), a configuração da ferramenta DBeaver bem como entender como é feita a comunicação entre a Serverless Framework e a cloud AWS.

Desta forma, começamos por explicar a configuração necessária para o ambiente de desenvolvimento. De salientar que a ordem deve ser respeitada para que os processos de configuração sejam concluídos com sucesso.

- **Ambiente de desenvolvimento**:

1. Como mencionado anteriormente a ferramenta será desenvolvida com base na linguagem de programação *Python*. Assim, antes de ser criada a diretoria para o projeto deve-se instalar o ambiente

de desenvolvimento *PyCharm*[27] e a versão *Python 3.7.9*[28]. Esta especificação é relevante considerando as bibliotecas disponíveis e as suas especificações para esta versão.

2. Deve ser disponibilizado o acesso ao sistema de gestão de pacotes de *software* escritos em *python pip* através do seguinte comando. Este comando estabelece um *path* mas não instala o pip. Prepara o acesso ao pip que já vem com o *Python.*:

#[29] ***setx PATH "%PATH%;C:\Pthon37\Scripts"***

3. Instalar a *virtualenv*. O objetivo é criar um ambiente que tem os seus próprios diretórios de instalação, que não partilha bibliotecas com outros ambientes *virtualenv* (e opcionalmente também não acede às bibliotecas instaladas globalmente). Para proceder à sua instalação deve-se executar os seguintes comandos:

 # pip install virtualenv

 # virtualenv venv –python python

 # venv\Scripts\activate.bat

4. Considerando que para a configuração de ambientes de desenvolvimento *Python* será necessário instalar dependências, é necessário configurar executando o seguinte comando:

 # pip install -r app/requirements.txt

5. Instalar *Docker Desktop*. Para isso é necessário:

a. Fazer o download do Docker Desktop[30]

b. Ao implementar a aplicação serverless, permite a inicialização do Docker Container e a implementação é automatizada

Quando configuramos o Docker, é normal que ocorram alguns erros de sistema. Abaixo ficam descritos alguns comandos que podem resolver esses erros:

 # docker-machine stop default

 # docker-machine start default

 # docker-machine upgrade

 # docker-machine regenerate-certs default

---

[27] Para uma correta instalação do PyCharm siga as instruções em: https://www.jetbrains.com/pycharm/download/#section=windows

[28] Para uma correta instalação de python siga as instruções em: https://www.python.org/downloads/

[29] Sempre que for apresentado um comando, este será sempre apresentado depois de um "#"

[30] Para uma correta instalação do Docker desktop siga as instruções em: https://www.docker.com/products/docker-desktop

 # @FOR /f "tokens=" %i IN ('docker-machine env --shell cmd default') DO @%*i

Uma vez concluídos com sucesso todos as configurações necessárias ao ambiente de desenvolvido, abordaremos de seguida, em detalhe, as configurações necessárias para o ambiente de *deployment*. Na mesma lógica que as configurações anteriores, a ordem apresentada deve ser respeitada para que não existam erros de configuração.

- **Ambiente de deployment**:

1. A ideia de trabalhar com a *framework Serverless* surge da necessidade de acelerar o processo de desenvolvimento e implementação da infraestrutura de recolha de dados. Como já referido, utilizou-se a plataforma AWS, sendo um dos maiores fornecedores de produtos *Cloud* da atualidade. Ao estabelecer uma ligação entre a *Cloud AWS* com a *Serverless Framework*, conseguimos implementar aplicações usando tecnologias *Serverless* em *Cloud* ganhando assim a capacidade de fácil escalamento automático.

Para proceder ao desenvolvimento do código recorrendo a uma *framework Serverless*, é necessário que a *framework* esteja instalada e configurada com uma conta AWS. Para instalar esta *framework*, será necessária a instalação da última versão de *Node.js*[31]. Depois da instalação *Node.js*, basta executar o seguinte comando no terminal:

 # - npm install -g serverless

É ainda importante referir que para a configuração de ambientes de desenvolvimento *npm* será necessário instalar dependências. Para isso basta adicionar no final da variável *Path* na secção *User variable* das Variáveis de Ambiente nas Propriedades do Sistema. Podemos fazê-lo da seguinte forma:

 # ;C:\Program Files\nodejs

2. Assim que o processo de instalação estiver concluído, devemos configurar as chaves de acesso para criar os recursos pretendidos na conta AWS associada. A configuração deve ser feita substituindo as credenciais da conta AWS, nomeadamente: *AWS_ACCESS_KEY_ID* e *AWS_SECRET_ACCESS_KEY_ID,* através do seguinte comando:

 # serverless config credentials --provider aws --key <AWS_ACCESS_KEY_ID> --secret <AWS_SECRET_ACCESS_KEY_ID>

Na ausência de problemas, a *Serverless Framework* está configurada com sucesso. Quer isto dizer que agora a preocupação para qualquer desenvolvedor é apenas o desenvolvimento do código (conceito de serverless).

3. Posto isto, é necessário aceder à diretória do projeto através do seguinte comando (substituir <*location*> com a diretória onde se encontra o projeto):

cd <location>/nomeProjeto

[31] Para uma correta instalação do Node.js poderá aceder a: https://nodejs.org/en/download/

4. É importante salientar que para a configuração de ambientes de desenvolvimento *Python* será necessário instalar dependências. Para isso é necessário executar os seguintes comandos:

 # serverless plugin install –name serverless-python-requirements

 # serverless plugin install –name serverless-glob-merge-yaml

 # serverless plugin install –name serverless-iam-roles-per-function

5. Dentro da diretoria onde se encontra o projeto e onde foi instalado a *Serverless Framework*, podemos então fazer o *deploy* do projeto para a *AWS Computing Services* através da execução do seguinte comando:

 # sls deploy

Além do deploy total da aplicação, é possível apenas fazer deploy apenas a uma função (lambda), executando o seguinte comando:

 # sls deploy function -f <function_name>

Depois destes passos o *deploy* está executado e a aplicação passa a ser da responsabilidade da cloud *AWS*.

Uma vez detalhadas as configurações necessárias para o ambiente de desenvolvimento da aplicação bem como o ambiente de *deployment* para a *cloud AWS*, detalharemos em seguida a ferramenta de administração de base de dados, necessária para uma interação com a base de dados AWS RDS, onde serão armazenados os dados e onde poderá ser possível consultar, validar e analisar os dados a partir de *queries* e com suporte a esta ferramenta.

- **Configuração da ferramenta DBeaver:**

Para poder controlar as bases de dados remotamente deve-se proceder à instalação da ferramenta[32] DBeaver. No menu principal, criar uma configuração, preenchendo os seguintes campos:

  - RDS Server host
  - EC-2 host
  - Password para o utilizador postgres
  - A chave SSH.pem

De salientar que a ligação à instância RDS a partir do exterior do *VPC* só é permitida através da execução de um túnel *SSH* com uma instância EC-2 responsável por estabelecer uma ponte entre a base de dados e o endereço IP. Caso se verifique que através destas configurações ainda não seja possível uma ligação à base de dados, é necessário adicionar o endereço IP ao grupo de segurança da instância EC-2, através do serviço *IAM*.

[32] Para uma correta instalação do DBeaver siga as instruções em: https://dbeaver.io/download/

- **Comunicação entre a *Serverless Framework* e a *cloud AWS*:**

A aplicação serverless tem, de alguma forma, de ser suportada por meio de *front*-end. Para facilitar o nosso desenvolvimento de *front-end*, tiramos partido do ambiente de desenvolvimento integrado em programação *PyCharm*. Com base nas boas práticas de implementação, com a organização e estruturação das várias pastas de código, existe o ficheiro *YAML*. É neste ficheiro *YAML* que especificamos todas as configurações serverless em cloud AWS. A título de exemplo: são especificados, o *Provider* do serviço (neste caso e como já referido *AWS*), a versão da linguagem de programação (devido a especificações de bibliotecas, por exemplo), neste caso Python 3.7, o ambiente da aplicação, ou seja, se estamos a desenvolver um código num ambiente de desenvolvimento ou num ambiente de produção, a região onde se localiza a nossa aplicação (neste caso foi selecionada Norte da Virgínia, pelo facto de ser mais acessível), entre outras especificações.

É realmente importante configurar o ficheiro *YAML* com excelência, uma vez que este arquivo é utilizado pelo *Serverless Framework*[33] para que este saiba que recursos são necessários de serem criados do lado da AWS, com que especificações e como é que comunicam entre si, por exemplo, ao nível da segurança. Palavras-chaves neste processo:

- Automatização: não é necessário criar os serviços de forma manual na plataforma da AWS
- Eficácia: quando necessário, as alterações são feitas com simples comandos
- Compacto: descrição de todos os elementos necessários descritos num só lugar

Assim, para a integração da camada de desenvolvimento (através dos nossos computadores locais) em cloud AWS, já entendemos que é utilizada a *Serverless Framework*[34] que suportará toda a gestão da criação da aplicação. Não é demasiado repetir que todos os recursos necessários para a "vida" da aplicação podem ser configurados através do arquivo *YAML* e que com um simples comando de *deploy* todos os recursos são criados na plataforma AWS.

[33] Referência para exemplo de um arquivo serverless.YAML em: https://www.serverless.com/framework/docs/providers/aws/guide/serverless.yml/

[34] Toda a documentação de Serverless Framework em: https://www.serverless.com/

3.3.2 Processo de Recolha dos Dados e Processo Tratamento de Dados

No presente subcapítulo, abordaremos todos os passos técnicos necessários para o processo de Recolha dos Dados e o Processo de Tratamento de Dados como sugere a figura abaixo apresentada - Figura 15 - Processo de desenvolvimento Recolha de Dados e desenvolvimento Tratamento de Dados. De referir, que os componentes da figura serão detalhadamente discutidos no presente documento.

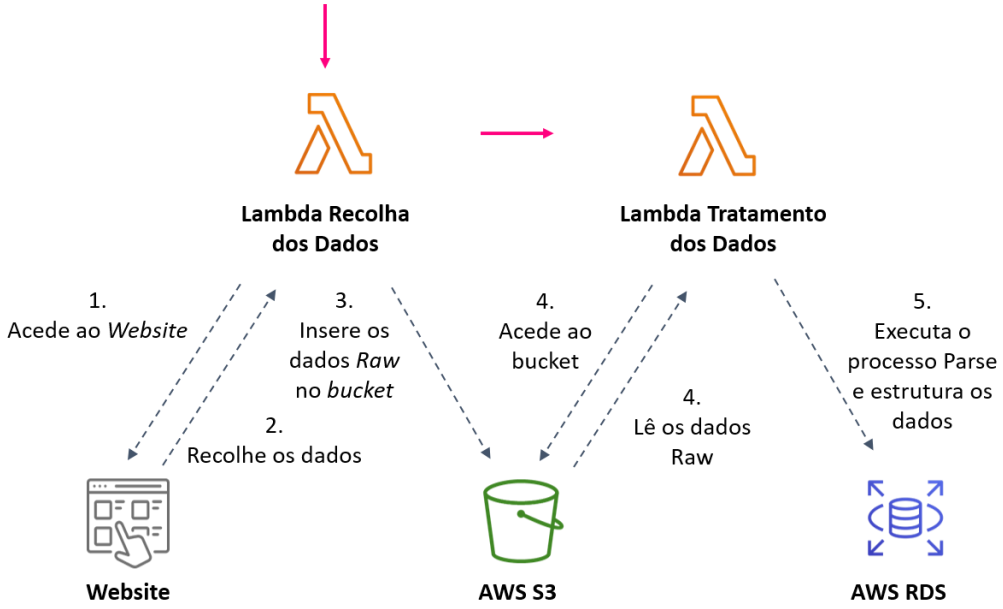

Figura 15 - Processo de desenvolvimento Recolha de Dados e desenvolvimento Tratamento de Dados

3.3.2.1 Processo de Recolha dos Dados

A grande questão do presente capítulo "Desenvolvimento da plataforma" foca-se em: como é que se obtém dados de *Websites*? Neste caso, *Websites* direcionados para lojas online que constantemente alteram os dados que disponibilizam, ou seja, *Websites* caracterizados como fontes de dados dinâmicos.

De realçar que os procedimentos explicativos seguem uma lógica de raciocínio que vai ao encontro de que é implementado sequencialmente em muitas aplicações de recolha de dados abertos, nomeadamente: scraping a *websites*, aplicações em *python* (linhas de código e abordagem de bibliotecas direcionadas para determinadas funções), aplicações com recurso à *Serverless Framework* e aplicações *serverless* em *cloud AWS* utilizando os serviços referidos na arquitetura da aplicação, podendo ser utilizados como consulta para futuras implementações associados a estas temáticas.

A figura abaixo – Figura 16 - Processo de desenvolvimento Recolha de Dados - foca os componentes trabalhados no Processo de Recolha de Dados e que serão abaixo apresentados.

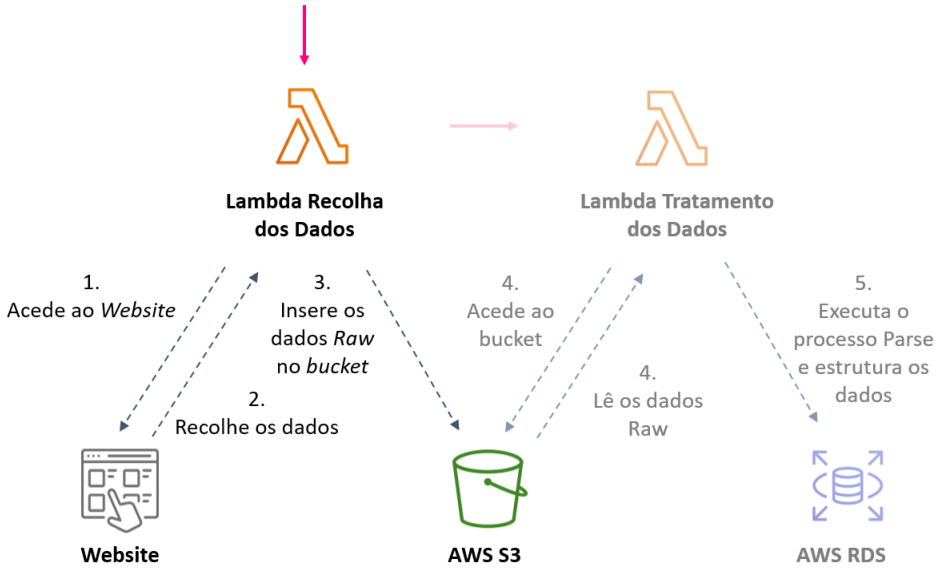

Figura 16 - Processo de desenvolvimento Recolha de Dados

Quando iniciamos um projeto de recolha de dados devemos começar por sistematizar os seguintes pontos, apresentados na tabela abaixo - Tabela 4 – Esquematização da informação para a Recolha dos Dados

1. Identificação dos dados	2. Identificação dos websites alvo	3. Análise à estrutura do website
Exemplo de um produto e dos dados de interesse a extrair: • Nome: Azeite Virgem Extra Suave Biológico • Marca: Gallo • Embalagem: bem. 75cl • Preço por produto: €4,89/un • Preço: €6,52/lt • Desconto: 30% • Preço sem Desconto: €6,99/un	Fontes de informação (lojas online) que contenham informação do nosso interesse: • Loja A • Loja B	Entender como é que as páginas são estruturadas: • Conhecimento básico de como são desenvolvidos os websites • Análise ao HTML, CSS e Java Script • Analisar se existem API's disponíveis que comunicam com o website, ou seja, que contém os dados alvo a extrair

Tabela 4 – Esquematização da informação para a Recolha dos Dados

Analisando a tabela acima apresentada, a primeira etapa passa sempre por perceber que conjunto de dados queremos obter. No mercado retalhista é importante analisar informação como o preço por medida, que tipo de desconto são praticados – promoção imediata, cartão, parcerias – em que categoria se insere o produto, qual é a frequência de promoções, entre outras variáveis. Em termos técnicos, todas estas variáveis representam o conjunto de dados que queremos extrair. Depois de uma análise a algumas lojas online, conseguimos perceber que o mercado de distribuição alimentar online é um negócio que armazena uma grande quantidade de dados - em média cada retalhista disponibiliza 20000 [35]produtos no *website*, e em média, cada produto tem a si associado cerca de 10 atributos.

Assim, em termos práticos são cerca de 200000 dados por retalhista, 400000 dados no total diariamente a extrair com o desenvolvimento da presenta aplicação (considerando a Loja A e a Loja B).

Da análise efetuada aos produtos disponibilizados nos *Websites* e mediante o negócio em questão, conseguimos entender com facilidade que os produtos disponibilizados sofrem alterações com bastante frequência. O preço por produto pode alterar de um dia para o outro, são apresentados novos descontos, existem produtos que saem do mercado, outros que entram, ou até alterações devidos a falhas na base de dados que monitoriza o *website* e, por isso, necessariamente a informação disponibilizada sofre também alterações seguindo as estratégias implementadas por cada retalhista. Por existir esta dinâmica tão significativa nos dados, os dados serão extraídos uma vez por dia. Contudo é aconselhável que com a monitorização da aplicação se consiga perceber se a base de dados do retalhista sofre alterações frequentes durante o dia. No presente caso de estudo, não foi feita esta validação, pelo que foram apenas feitas extrações 1 vez por dia em cada loja online.

De seguida, será necessário identificar com precisão que *websites* representarão as *webpages* sob as quais agirá o nosso *scraper*. E segue-se assim uma análise ao website. Esta é uma tarefa fundamental. Exige que se analise de uma forma detalhada a página que se identificou como possível fonte de dados na qual o *scraper* vai atuar. Realizar uma *checklist* que categorize a página na sua complexidade na extração dos dados é fulcral. Isto ajuda a que o tempo estimado para o desenvolvimento do projeto seja mais preciso. Além disso, no momento de implementar o *scraper* conhecem-se as limitações e proteções que a página apresenta e, desta forma, são desenvolvidos programas capazes de contornar essas limitações. Estas estratégias não impossibilitam situações inesperadas. A qualquer momento os administradores dos *websites* podem alterar a sua estrutura, implicando alterações na forma como o *scraper* é desenvolvido; ou seja, este tipo de situações inesperadas modifica a forma como os dados devem ser acedidos e deve ser considerado no momento de determinar o tempo estimado para o processo.

Quando estamos a realizar esta análise estamos a recorrer à forma tradicional de executar *scraping* a um *website,* ou seja, analisando o próprio *HTML* e tentar extrair a informação que necessitamos para o nosso conjunto de dados.

[35] Verificados 20000 produtos numa loja online de um retalhista em Portugal. Consultando a sua loja online e validando o número total de produtos por categoria, conseguimos facilmente perceber o número total de produtos do retalhista. Dados verificados a setembro de 2022

Para isso, devemos inspecionar a página utilizando as ferramentas disponibilizadas pela maioria dos *browsers (Chrome, Firefox,* entre outros*).* Basta clicar com o botão direito em qualquer lugar da página e selecionar "Inspecionar". Depois disso, poderemos aceder ao *HTML* que estrutura a *webpage* selecionada. É útil existir um conhecimento intermédio de como são as estruturas das *webpages*. De forma direta e simples, as páginas são construídas com base no conteúdo que se quer disponibilizar e constrói-se a página, com marcações e codificações em HTML, criam-se estilos em CSS e programa-se as ações necessárias em JavaScript.

A figura abaixo - Figura 17 - Exemplo de um bloco de HTML de um produto disponível numa loja online de retalho - retrata o exemplo de um bloco de HTML de um produto. Sublinhado com a cor vermelha podemos identificar as características de interesse na loja online bem como a correspondência no bloco em HTML desse produto.

Figura 17 - Exemplo de um bloco de HTML de um produto disponível numa loja online de retalho

Analogamente, *web scraping* é exatamente o contrário: retiramos o JavaScript, o CSS e o HTML da página, separando a forma e ficando apenas com o conteúdo. Segundo esta forma tradicional de *scraping*, a maneira de explicitar qual é o tipo de conteúdo que pretendemos filtrar é através de seletores de *CSS*. Isso passa por indicar as *tags*, *id* (únicos), *class* (podem existir classes iguais na mesma página) e *atributtes*. A arte do *web scraping* passa pela seleção do melhor seletor para aquele o conteúdo a extrair.

Porém, e com a evolução tecnológica relativamente à forma como os dados são disponibilizados na *internet* facilmente encontramos sites de complexidade elevada e torna-se necessário passar não pela examinação da página HTML, mas antes por comunicar diretamente com as Interfaces de Programação de Aplicações (application programming interface - *API)* privadas que disponibilizam a informação que queremos extrair, por exemplo, a extração dos produtos de uma determinada categoria no mercado retalhista. É possível analisar a existência ou não destas *API's* no *website* através da *tab Network* e examinando o tráfego de pedidos HTTP entre o site e o nosso browser.

Assim, e de forma sucinta, todo este processo de início à arte de scraping e todas as etapas pela qual devemos percorrer para entender como chegamos à extração dos dados estão refletidos na imagem abaixo – Figura 18 - Processo de desenvolvimento à arte de Web Scraping:

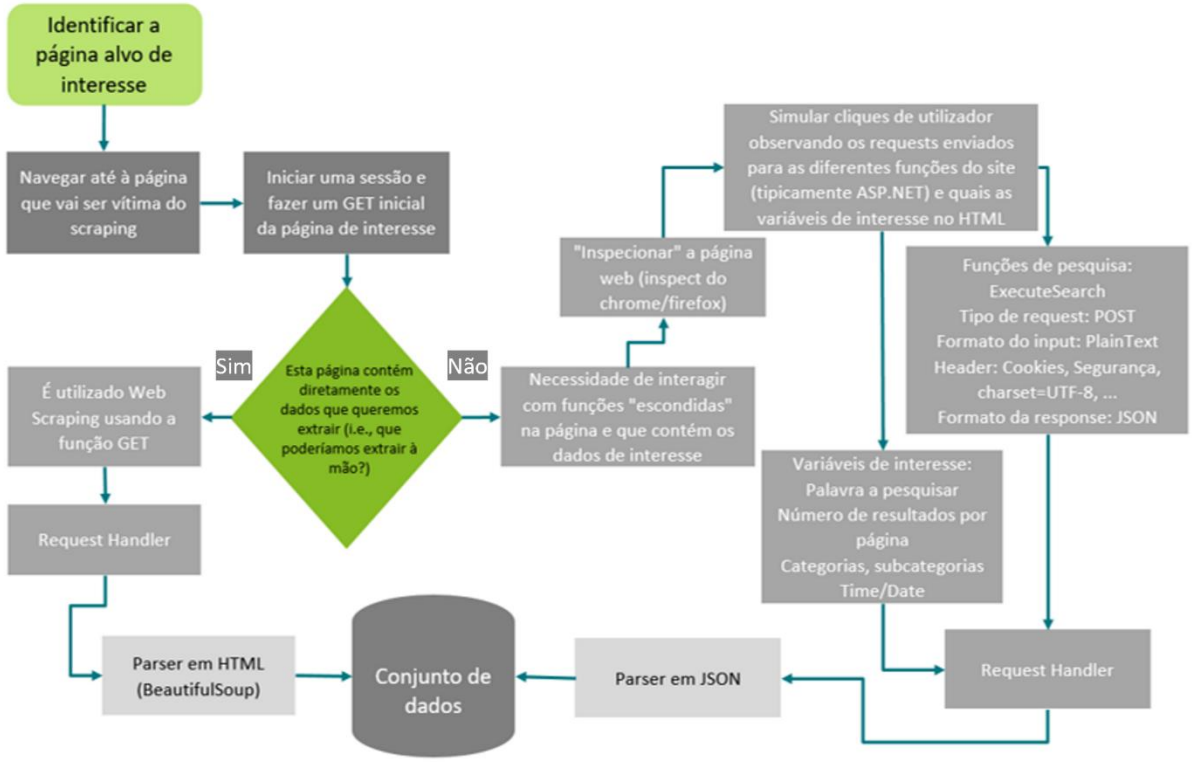

Figura 18 - Processo de desenvolvimento à arte de Web Scraping

Posto isto, e uma vez que cada *website* é construído individualmente e por isso a sua estrutura é única, como já mencionado na arquitetura da ferramenta, o processo da recolha de dados é consequentemente implementado individualmente para cada fonte de informação, mesmo que a forma de desenvolvimento para se chegar aos dados seja igual.

Assim, o processo explicativo da recolha de dados que se segue refere-se ao processo da Loja A.

O processo de desenvolvimento inicia-se mediante as etapas identificadas da figura acima descrita. No decorrer das etapas conseguimos entender que o *website* da Loja A é caracterizado por um sistema de aplicações web complexas, nomeadamente conseguimos encontrar duas *API's* para diferentes ações.

Uma *API* que disponibiliza os produtos em destaque e uma outra para disponibilizar os produtos por categoria nas diferentes páginas. Por vezes estas interfaces não são simples API's, mas sim scripts de JavaScript ou PHP que recebem comandos e retornam porções do conteúdo apresentado na página. Para o caso de uso em questão, importa-nos apenas a *API* que devolve os produtos por categoria, que engloba todos os produtos que o retalhista disponibiliza bem como toda a informação de cada produto. Não nos é importante extrair os produtos em destaque uma vez que isso representa apenas um indicador na construção de insights valiosos sobre a estratégia implementada (por exemplo,

perceber que marcas estão com mais frequência nos produtos em destaque) mas com o processo de extração igual à extração de todos os produtos por categoria.

Na secção dos anexos - ANEXO B – exemplo de uma API que contêm e disponibiliza dados no *website*, poderá ser consultado visualmente um exemplo de um *print* da API que contêm e disponibiliza dados no *website*.

O próximo passo é então entender como é que a *API* em questão funciona, que *inputs* recebe e em que formato retorna o *output*. Esses requisitos podem ser consultados também no anexo mencionado anteriormente.

Como podemos verificar sempre que pedimos produtos a esta *API*, é necessário enviar dois *inputs*: a nome da categoria em específico e o número de produtos por página. Relativamente ao *input* da categoria é implementado através de um *array* que contem os nomes de todas as categorias da Loja A, e em ciclo, esta função envia o nome de uma categoria e recebe os produtos dessa categoria até percorrer todas as categorias. Relativamente ao número de produtos por páginas é pedido o número máximo que esta API consegue devolver nomeadamente, 80 produtos.

Por outro lado, um exemplo de um *output* de um *script* de uma categoria pode ser consultado na secção dos anexos - ANEXO C – exemplo de um *output* de um *script* de uma categoria, sendo que obtemos um *output* com um ficheiro no mesmo formato para todas as categorias. Neste sentido, conseguimos verificar que a *API* devolve para cada categoria um ficheiro *JSON*, que contem todos os produtos e os seus atributos.

Aspetos importantes a reter do *output* desta *API*:

- O *output* é devolvido em formato *JSON* com uma estrutura muito específica e, por isso, existe a necessidade de trabalharmos o ficheiro de forma a extrair os dados limpos, organizados e estruturados
- O output devolvido surpreendentemente devolve mais informação que aquela que disponibiliza. Atributos como: *IsANewProduct*, *IsATopTenProduct*, *IsAnExclusiveProduct*, *IsALactoseFreeProduct*, *IsAGlutenFreeProduct* são atributos que são disponibilizados e que representam um *dataset* rico em informação e conhecimento acerca do produto e permite desenvolver análises ricas e detalhadas sobre o mercado

Com isto, conseguimos criar um *script* em python que acede ao *website* da Loja A, faz um pedido a uma *API* que disponibiliza os produtos e devolve esses produtos e os seus atributos por meio de um ficheiro *json*. Segue o exemplo desse script (pedido) nos anexos - ANEXO D – exemplo de um pedido Loja A.

Mas nem tudo correu sempre bem, e nesse sentido importa realçar os desafios durante este processo. Como já foi referido quando fazemos pedidos a esta *API*, temos de enviar como *input* o nome de uma categoria e o número de produtos por página. Inicialmente para obtermos o nome correto de todas as categorias a ser enviado para esta *API*, encontramos de igual forma uma *API* que nos devolvia o nome de todas as categorias no formato correto. Porém, após algumas semanas a fazer pedidos a essa *API*, percebemos que o acesso nos foi negado. Era nos devolvido um resultado vazio. O impacto desta alteração está relacionado com o *input* que enviamos para a *API* que devolve os produtos todos. Caso o nome de uma categoria se altere - por exemplo, a categoria “animais” é escrita como *input* para a

API dos produtos como "*animais-1"* - não enviaremos da forma correta o nome da categoria animais e consequentemente a *API* que nos devolve os produtos, retorna um erro uma vez que não reconheceu o *input* enviado.

Percebemos assim que os *websites* procuram cada vez mais proteger-se contra ações de *scraping* (demonstrando não estarem disponíveis para a maturidade de trabalhar e incentivar a economia circular de dados abertos) e uma dessas formas é a de não devolver os resultados de uma pesquisa feita por *bots* (programas que automatizam tarefas).

Este tipo de atitude esta refletido no seu ficheiro 'robots.txt'. Conseguimos consultar este ficheiro em qualquer *website* através do link www.NomeWebSite.pt/robots.txt.

A criação deste arquivo provém do Protocolo de Exclusão de Robôs. Este protocolo é um método desenvolvido pelos administradores de sistemas para informarem os bots visitantes quais os diretórios a que o *bot* não pode aceder. Neste caso, o *bot* representa o programa desenvolvido para percorrer automaticamente a *webpage* do retalhista selecionado. Com a finalidade de controlar as atividades desses *bots* durante as suas pesquisas, opcionalmente, os desenvolvedores das páginas criam o arquivo robots.txt na diretória raiz do seu website. Este aquivo é escrito em formato texto (.txt) que funciona como "filtro" para bots, permitindo ou bloqueando o acesso a partes ou à totalidade de um determinado site.

Ao consultar o ficheiro 'robots.txt' da Loja A, verificamos o seguinte:

```
User-agent: *
Disallow: */private
Disallow: */search
```

Ao encontrar esta informação, o *bot* saberá que serviços ou pastas poderá ou não consultar naquela *webpage*.

Porém, existe uma outra maneira de se conseguir algo semelhante através de tags, colocadas estrategicamente nos cabeçalhos de páginas HTML:

```
<META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">
```

No decorrer desta metodologia este problema não foi ultrapassado por uma questão de limitação de tempo. Limitamo-nos a escrever manualmente o nome de cada categoria como visualizamos no link quando acedemos à página que devolve todos os produtos por categoria. Por exemplo: https://www.LojaA.pt/animais-by-zu/ e assim entendemos que nome está a ser usado para enviar como input à API que devolve os produtos. Contudo, esta solução não é a mais fiável uma vez que sempre que exista uma alteração nos nomes das API, surgirá um erro na aplicação e a alteração terá de ser feita manualmente.

Assim sendo, a solução mais inteligente passaria por perceber como conseguiríamos extrair o nome das categorias dos produtos de outra forma e depois desse processo usar esses nomes extraídos das categorias como input para a API.

De salientar que a fonte de informação continua a ser caracterizada como uma fonte de informação aberta, uma vez que é totalmente acessível por terceiros, independentemente de ser ou não possível a recolha e o processamento automático da informação disponível[36] e, por isso, não estamos a implementar práticas ilegais ao encontrar soluções e implementá-las para contornar estes bloqueios.

Posto isto, os ficheiros raw são então armazenados no serviço *Amazon S3.* Para isso é criado um bucket, denominado por exemplo como *DadosAbertosRaw*, com duas pastas distintas: Loja A e Loja B. Neste sentido os ficheiros extraídos no formato JSON da Loja A e que contêm os produtos e os seus atributos de uma determinada categoria são armazenados na pasta da Loja A. Para uma correta organização no bucket dos ficheiros extraídos, estes devem ser armazenados por datas de extração e por categoria (sendo que cada ficheiro devolve os produtos de apenas uma categoria). Desta forma, teremos os dados organizados por Loja, por data de extração e os ficheiros dos produtos por categoria.

De salientar que todos os processos mencionados nomeadamente, o pedido à *API*, a recolha do seu *output* e o *insert* dos dados *raw* nas devidas pastas no bucket, são executados através de um *script* em *python*. Assim, este *script* com início, meio e fim representa o algoritmo de recolha dos dados da loja A. Quando nos referimos ao termo *script* não é mais que uma função que num contexto de *serverless cloud AWS* representa o código a ser executado por meio do serviço *AWS Lambda*[37]. Ou seja, é executado o código desenvolvido e sem pensar em servidores (conceito de serverless).

Assim, temos construída a primeira *lambda*. Podemos ver o código associado à lambda em anexos - ANEXO E – Exemplo de código Lambda.

Posto isto, importa agora detalhar o processo explicativo da recolha dos dados da Loja B e, com isto, recolher *insights* valiosos sobre diferentes métodos de recolha dos dados segundo as diferentes estruturações dos *websites*. O conhecimento e o *know-how* nestes métodos conseguem-se apenas com a experiência técnica nos desenvolvimentos destes projetos.

Para o processo de *scraping* da Loja B, este passou pelas mesmas etapas num processo de scraping como descrita na Figura 18 - Processo de desenvolvimento à arte de Web Scraping. Contudo, para esta loja, verifica-se a necessidade de expansão em técnicas a utilizar em trabalhos de *scraping* a fontes de informação, uma vez que não conseguimos obter os dados de forma direta com pedidos usando a função *GET* nem encontrar nenhuma *API* disponível para a recolha dos dados, como conseguimos para a Loja A. E é perante este desafio que conseguimos perceber o nível de complexidade que o mundo da recolha dos dados abertos representa e as oportunidades que existem na expansão de conhecimento técnico direcionado a esta área.

Neste sentido, surge assim o conceito de *web crawler*. O *web crawler* é um programa que recolhe o conteúdo disponível visualmente no *website* de forma sistematizada através do protocolo *HTTP*/*HTTPS*.

[36] Significado de fonte aberta segundo o livro Introdução à cibersegurança
[37] Toda a documentação sobre *AWS Lambda* em: https://AWS.amazon.com/pt/lambda/

Um web crawler navega recursivamente os conteúdos da página para o qual é preparado (seguir links, imagens, não entrar na zona de login, etc) e permite aceder a uma grande porção da página web de forma semiautomática, guardando o conteúdo alvo e tipicamente requer uma menor manutenção do scraper. Infelizmente estes crawler têm a contrapartida de por vezes recolherem informação de forma menos seletiva, resultando num maior esforço do lado da triagem e filtragem dos dados recolhidos. Estas ferramentas por vezes podem ficar presas em certas zonas de sites se o desenvolvedor do site criar *loops* de links, fazendo o crawler andar a seguir links que o mandam sempre para o mesmo conjunto de 2 a 3 páginas. Notoriamente, a Google [38]é um enorme proponente da tecnologia de web crawling e scraping dado que todo o seu produto do Motor de Busca é fruto do trabalho destes crawlers.

Assim, através de *web crawler* conseguimos obter toda informação que está visualmente disponível por meio de um rastreador criado em *python*, especificamente utilizado a biblioteca *Scrapy*[39]. De forma resumida, é uma biblioteca de código aberto colaborativa desenhada para extrair os dados de *websites* de forma rápida, simples e escalável. Representa um pacote em *python* e está disponível a partir do *pip*. Para dar início a um projeto de *scrapy*, começa-se pela instalação da própria biblioteca:

```
# pip install scrapy
```

Após a instalação, importa criar o projeto *Scrapy*, com o seguinte comando:

```
# scrapy startproject RecolhaDadosLojaB
```

Um projeto implementado com base nesta biblioteca herda necessariamente arquivos e diretórias que estão relacionados com o fluxo de trabalho do crawler, nomeadamente: items.py, middlewares.py, pipelines.py e settings.py.

A documentação bem como a explicação da existência destes arquivos é bastante enriquecida e completa na documentação da biblioteca[40].

Depois do projeto criado, e dentro da pasta *spiders*, é então criado o *script* para a recolha dos dados. Do *script* é importante referir a importância de criar um *crawler.* O processo de *crawler* é criado com base na extensão da biblioteca *scrapy.crawler* e que permite especificar as seguintes variáveis: *'USER_AGENT', 'FEED_FORMAT', 'FEED_EXPORT_ENCODING', 'FEED_URI', 'FEED_OVERWRITE'*

Poderá consultar um exemplo de um processo de *crawler* criado num *script python* em anexos - ANEXO F – Exemplo de um runner Crawler.

[38] Poderá consultar informação aqui https://www.google.com/search/howsearchworks/crawling-indexing/

[39] Poderá consultar toda a documentação da biblioteca *Scrapy* em: https://scrapy.org/

[40] Poderá consultar toda a documentação de um projeto desenvolvido com base em Scrapy em: https://docs.scrapy.org/en/latest/intro/tutorial.html

As configurações especificadas na variável denominada *crawler,* permitem identificar:

- *'USER_AGENT'*: nome do agente
- *'FEED_FORMAT'*: tipo de formato para receber os dados extraídos
- *'FEED_EXPORT_ENCODING'*: tipo de codificação
- *'FEED_URI'*: onde armazenará os ficheiros com a informação extraída
- *'FEED_OVERWRITE'*

Através da variável *crawler* é lançado o evento que permite rastrear o website, nomeadamente com a seguinte linha de código:

 # crawler.crawl(spider=event['LojaB'], storeUrl=event['storeUrl'])

Assim, o spider é responsável pela recolha de todo o conteúdo a extrair no *website*, seguindo uma lógica sequencial criado no *script* de url. O processo começa sempre com a lista de *URL's* a visitar, imitando a consulta de todos os *links* que o ser humano precisaria de visitar para a visualização da informação pretendida. Ou seja, visitar todos os *links* existentes das categorias de produtos (uma vez que não existe uma só página que carregue todos os produtos de uma só vez), dentro de todas as categorias visitar a página que nos devolve todos os produtos dessa categoria e seguidamente visitar todas as páginas que existem dessa consulta com a finalidade de visualizar (para o spider extrair) todos os produtos. Em alguns casos, caso o objetivo seja extrair toda a informação de detalhe de um produto, é adicionada outra carga de trabalho, ou seja, além de visitar as páginas que contêm todos os produtos por categoria, é necessário, visitar todas as páginas individuais de cada produto.

Perante esta complexidade, de facto conseguimos validar com o desenvolvimento da aplicação, que recorrer a processos de scraping com base em programas de *web crawler* eleva de facto o nível de dificuldade, esforço envolvido e conhecimento técnico relativo ao desenvolvimento do código.

A necessidade de especificar todos os *url* que o *bot* tem de percorrer para aceder à informação requer uma análise profunda e complexa de como o site está estruturado e desenvolvido. Além disto, qualquer alteração na página, é refletida na aplicação e por isso, exige manutenção frequente.

O propósito da presente metodologia é abordar as questões técnicas no desenvolvimento de recolha dos dados e discutir métodos, desafios, potencialidades que se encontraram ao longo do desenvolvimento. Dessa forma, não se justifica explicar detalhadamente o script em python que executa a recolha de dados à Loja B uma vez que cada página é construída de forma única e por isso, cada código é um código. Contudo, existe um leque muito grande e variado de tutoriais relativos a web crawler que podem facilmente ser encontrados na internet.

Tal como se sucedeu no processo de recolha dos dados à Loja A, durante o processo de recolha dos dados à Loja B deparamo-nos com técnicas de bloqueio do lado do *website* da Loja B. Neste caso, não se verificou um acesso negado à *API*, uma vez que não era essa a prática de scraping implementada, mas sim a proibição de IP's ao acesso ao *website*. O que se verificou é que depois de um longo período de extrações diárias com a execução a correr nos servidores da AWS, todos os IP's de acesso deixaram de ter permissão ao website da Loja B. Quando dizemos todos os IP's, significa que os processos a executar colocados em cloud AWS podem ser colocados em diferentes regiões e necessariamente são lhes atribuídos deferentes IP's. Quando nos deparamos com este bloqueio, tentamos mudar o servidor da aplicação para outras regiões, mas mesmo assim o acesso ao website era negado.

Contudo, com testes locais (a partir de IP's privados) o processo era executado sem ocorrer nenhum bloqueio nem nenhuma ocorrência.

A solução encontrada foi a de implementar servidores proxies. Em termos práticos, é uma tecnologia intrínseca e utilizada em redes de computadores e a sua utilização neste conceito pretende desbloquear ou evitar a proibição de IP's das configurações de privacidade que alguns *websites* implementam. A nível de arquitetura o que acontece está ilustrado na imagem que se segue - Figura 19 - Servidor Proxy:

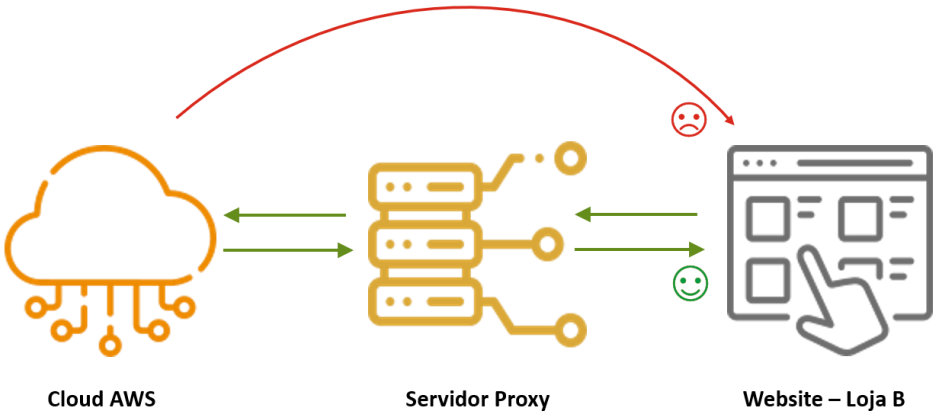

Figura 19 - Servidor Proxy

Além da implementação de proxies são boas práticas implementar outras técnicas para extrair dados sem que existam bloqueios de *IP's* por exemplo, diminuição do tempo de extração de dados, uso de diferentes *user-agents* e alterações no horário dos processos que executam as extrações diárias dos dados.

Uma vez desenvolvido o código de recolha dos dados com recurso a web crawler para a Loja B e abordadas técnicas de bloqueios por parte dos *websites* bem como práticas para contornar e/ou evitar esses mesmos bloqueios, é importante abordar o processo seguinte: o armazenamento dos dados recolhidos.

Assim, os dados extraídos são armazenados de igual forma ao que acontece na Loja A. Para uma visualização mais clara, segue a estrutura de armazenamento dos dados *raw* para as duas lojas (Loja A e Loja B) no serviço *Amazon S3* - Figura 20 - Armazenamento dos dados raw no Bucket Amazon S3:

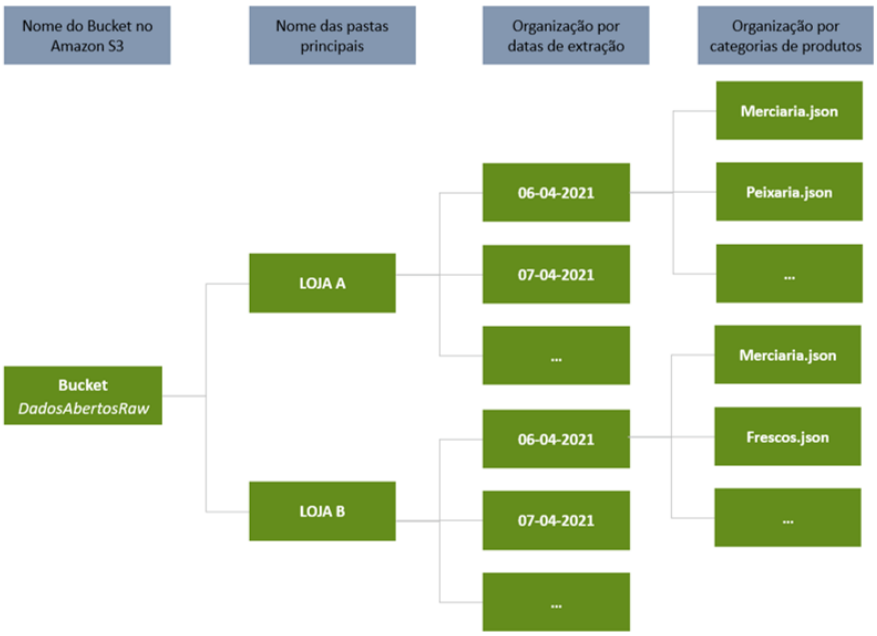

Figura 20 - Armazenamento dos dados raw no Bucket Amazon S3

Contextualizando os processos de recolha dos dados da Loja A e da Loja B nos serviços da AWS, cada script explicado para cada loja (o acesso ao website, a recolha dos dados e o armazenamento do serviço Amazon S3) representa uma *Lambda*.

Em suma temos:

- A função *lambda da Loja A* recolhe todos os dados dos produtos através da comunicação com uma *API* que disponibiliza os dados no website. Envia-lhe como argumento e em ciclo o nome das categorias e o número de produtos por página. Esta retorna-lhe um ficheiro, por categoria, em formato *json* com todos os produtos e os seus atributos. De seguida, acede ao amazon s3, ao *bucket DadosAbertosRaw*, na pasta da Loja A, cria uma pasta com a data da extração (o dia atual) e coloca o ficheiro json com o nome da categoria. Este processo acontece para todas as outras categorias
- A função lambda da Loja B, tem um procedimento idêntico com a exceção do processo de recolha de dados. Como no website da Loja B, não foram encontradas API's, com a possibilidade de aceder aos produtos, a solução passou por criar um spider através da extensão *crawler,* da biblioteca *scrapy* e assim conseguimos recolher toda a informação que se encontra visualmente disponível no website.

Insights de conhecimento valiosos adquiridos durante o processo de desenvolvimento:

1. Sobre o desenvolvimento do processo de recolha dos dados

A imagem abaixo identificada - Figura 21 - Aspetos importantes a reter do desenvolvimento da Recolha de Dados - espelha o desenvolvimento do processo de recolha de dados dividido em *steps* e que pretende demonstrar os principais insights que se deve conhecer quando trabalhamos com recolha de dados abertos. O *step* 1 pretende resumir o conhecimento necessário sobre a validação da fonte de informação do que diz respeito à autoridade do seu acesso. Sequencialmente, o *step* 2 objetiva descrever as técnicas de recolha de dados a *websites*, indicando as bibliotecas usadas no desenvolvimento da presente aplicação, bem como indicar a biblioteca no processo *parse* da informação. Por último, no *step* 3, são descritos alguns dos desafios onde é muito provável de se encontrar em desenvolvimentos desta temática.

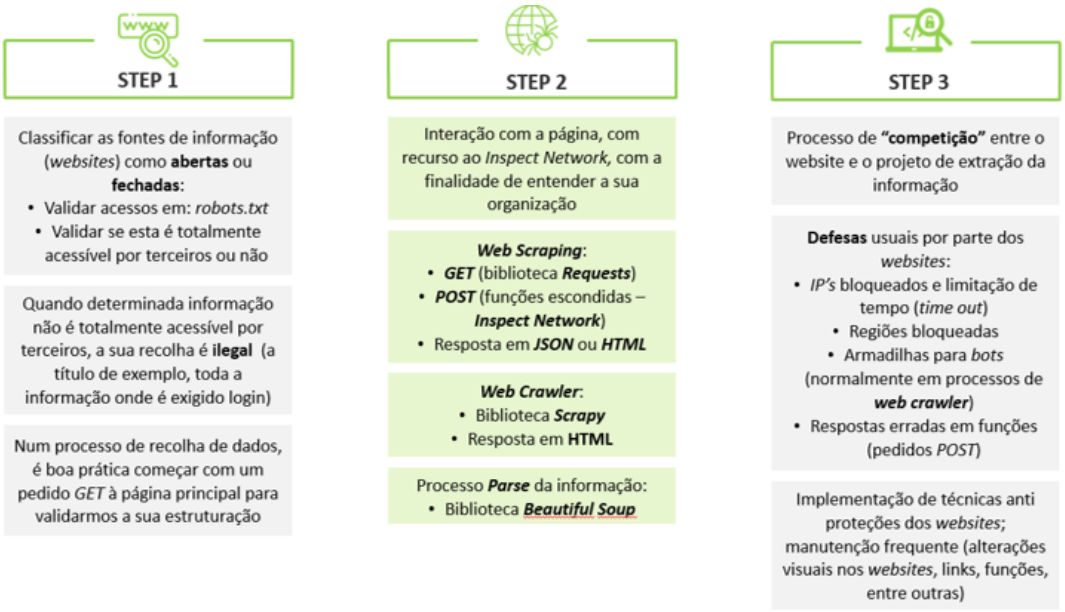

Figura 21 - Aspetos importantes a reter do desenvolvimento da Recolha de Dados

2. Sobre as diferenças mais relevantes entre web scraping (acesso a API's do website) e web crawler em tarefas de recolha de dados, a tabela abaixo - Tabela 5 - Diferenças encontradas em Web Scraping e Web Crawling no desenvolvimento - representa as diferenças mais notórias durante o processo de desenvolvimento:

Web Scraping (acesso a uma *API* do website)	**Web Crawler**
Acesso direto aos dados (produtos)	Necessária uma análise detalhada à informação no website, para um maior entendimento sobre a disposição dos dados (produtos)
Acesso a outros atributos não disponíveis no website e que enriquece o *dataset* extraído	Acesso apenas à informação disponibilizada visualmente no *website*
Poucas linhas de código, logo pouco processamento	Nível de complexidade elevado em relação ao código desenvolvido
Dados recebidos em formato JSON, devidamente organizados	Necessidade de trabalhar o formato em que o crawler extrai os dados

Tabela 5 - Diferenças encontradas em Web Scraping e Web Crawling no desenvolvimento

3. Sobre benefícios de recolha de dados abertos e armazenamento dos dados *raw* num serviço *Amazon Bucket S3:*

    - Recolha dos dados com liberdade na escolha da periocidade de extração, na quantidade de dados a extrair e na seleção destes
    - Considerando o seguinte cenário: sabemos que existem campanhas que são publicadas na página num específico dia da semana; portanto, é criado um processo de extração de informação direcionado apenas para aquele conteúdo (os produtos em campanha e que normalmente encontram-se em destaque nas páginas). Para isto, será executado com um *Schedule* que vai ao encontro do momento da disponibilização dos dados (por exemplo, uma vez por semana), evitando assim processos que ocorrem desnecessariamente, com cargas elevadas na máquina.
    - Análise aos dados Raw disponíveis e acessíveis em qualquer momento (valor do armazenamento dos dados raw sempre que é feita uma extração - abordagem de processamento de dados ELT)

De forma sucinta, são descritas abaixo boas práticas e indicadores a ter em conta quando trabalhamos com recolha e armazenamento de dados abertos:

- Identificar a necessidade de recolha de dados, a sua periocidade e quantidade de dados para evitar cargas de trabalhos desnecessárias
- Entender a necessidade de existir um bucket de armazenamento de dados *raw* (dados puros, sem manipulação de processos de tratamento)
    a. Quando se trata de trabalhar, analisar e/ou interpretar dados provenientes de fontes secundárias ou terciárias (análises construídas sobre os dados recolhidos, relatórios, *etc*) deve-se sempre recorrer a validações de dados quando necessário
    b. Para algumas destas fontes é possível ter a fonte de informação primária como uma fonte de validação dos dados. Mas apenas quando se verifica que a fonte original dos dados é estática em relação à informação que disponibiliza (ou seja, os dados não sofrem alterações com o tempo)

Atribuir sempre que faça sentido o dia de extração dos dados

3.3.2.2 Processo de Tratamento dos dados

O processo de tratamento dos dados caracteriza-se essencialmente por duas etapas:

- A processo de *Parse* aos ficheiros extraídos através de um *script python*, com o objetivo de trabalhar e extrair dos ficheiros recolhidos os dados de interesse
- Organizar a informação, estruturá-la em dados e armazená-los na *amazon RDS*

A figura abaixo – Figura 23 - Processo de desenvolvimento Tratamento de Dados - foca os componentes trabalhados no Processo de Tratamento dos Dados e que serão abaixo apresentados.

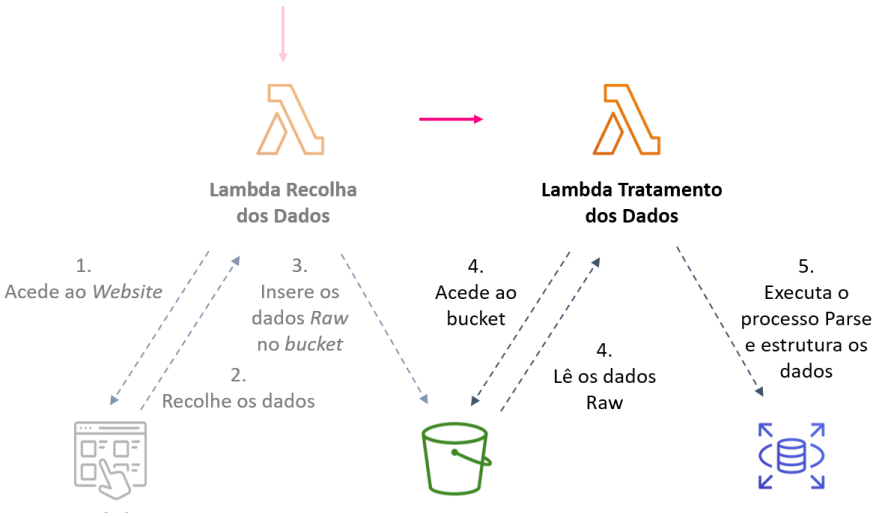

Figura 22 - Processo de desenvolvimento Tratamento de Dados

Ambos os processos relativos ao tratamento dos dados para a Loja A e para a Loja B, foram executados seguindo a mesma lógica de raciocínio, com a diferença na tarefa de *parse* uma vez que o ficheiro com

os dados extraídos não é, naturalmente, estruturado de forma igual. Mas como esta é uma tarefa muito individual de ficheiro para ficheiro e como pode tomar variadas formas, formatos ou estruturas, o processo de tratamento dos dados vai ser explicado em conjunto e de forma generalizada, sublinhando os *insights* valiosos no processo de desenvolvimento que foram necessários para ambas as lojas.

Assim, o script em python, através da biblioteca *boto3* com a extensão *boto3.client* inicia a sessão de um *client S3* especificando o nome do *Bucket* que se pretende aceder (*Bucket DadosAbertosRaw*), o nome da pasta (Loja A ou Loja B) e o nome da subpasta (o dia da extração que coincidirá com o dia atual da leitura dos dados para o processo do *parse*). Segue uma linha de código exemplo para este processo:

```
# lista_ficheiros = boto3.cliente('s3').list_objects_v2(Bucket='DadosAbertosRaw'), Prefix='LojaA/06-11-2022'
```

Quando acedemos a cada pasta, por exemplo, à pasta do dia '06-04-2021' da Loja A, obtemos o seu conteúdo e, para cada ficheiro, por exemplo *Mercearia.json*, lemos o seu corpo e executamos o parse da informação. Posteriormente é invocada a função que executa o parse do corpo do ficheiro e é devolvido o conjunto de dados. Um exemplo dessa função segue em anexo - ANEXO G – Exemplo de uma função *Parse*.

Devolvido o conjunto de dados, importa agora criar uma conexão com a base de dados que armazenará os ficheiros. Assim, na conexão à base de dados deve ser especificado:

- Host: host da base de dados na cloud AWS
- User: o nome de um *user* atribuído e com acessos à base de dados
- Password: a password criado para o user criado
- Name_db: o nome da base de dados criado na amazon RDS
- Conn: com suporte à extensão psycopg2.connect da biblioteca psycopg2 é possível estabelecermos uma comunicação com a base de dados

De seguida, precederemos então ao *INSERT* dos resultados devolvidos da função responsável pelo parse e é construída uma linha de comando em SQL para cada resultado, executando o *commit* dos dados.

Aspetos importante no processo:

- A base de dados amazon RDS para *PostgreSQL* foi uma necessidade sentida durante o processo como a ferramenta de base de dados que armazenaria os dados. PostgreSQL mostrou ser mais adaptável às necessidades de armazenamento dos atributos dos produtos extraídos uma vez que existia a possibilidade de caracterização de um atributo como uma lista. A carência surge do facto de ter de se atribuir como lista o atributo categoria a um produto, sendo que um produto tem a si associado uma categoria principal, mas pode ainda ter duas, três ou em alguns casos quatro subcategorias.

3.3.3 Fluxo de trabalho entre os dois processos: Recolha dos dados e Tratamento dos dados & Monitorização e Notificações

O presente subcapítulo pretende abordar as necessidades sentidas a nível técnico e as limitações com as quais nos deparamos ao longo do desenvolvimento quando trabalhamos em *Cloud AWS*. Além disso, discutiremos como é que a aplicação comunica entre si quando é executada na cloud e que procedimentos temos de garantir para que consigamos monitorizar a aplicação sem problemas uma vez que a responsabilidade está do lado dos servidores *AWS*. Desta forma, abaixo seguem estruturados alguns dos pontos essenciais a partilhar:

- Começaremos por explicar a necessidade de separar a processo de recolha de dados e tratamento de dados de cada loja em duas lambdas distintas.

Porque que não colocamos os dois processos numa só lambda para cada loja? Assim, em vez de termos 4 lambdas teríamos apenas 2 e assim existiria uma carga menor de recursos a usar na cloud. Pois bem, quando colocamos código em lambdas o processo de execução é limitado apenas a 15 minutos. É um indicador muito limitado uma vez que processos mais longos exigem a implementação de cadeia de lambdas e por isso o nível de complexidade da aplicação aumenta.

Por exemplo, se pensarmos em expandir a aplicação adicionando-lhe um componente de match onde o objetivo será encontrar match entre todos os produtos, certamente que o algoritmo não o consegue fazer em 15 minutos. Obriga a que exista processos mais complexos onde sejam introduzidos intervalos de *inputs* nas lambdas e uma contagem sem falhas do número de produtos processados em cada lambda.

- Como é que é lançada a primeira lambda de cada loja?

O serviço Step Function da AWS garante o serviço de fluxo de trabalho da aplicação. Quer isto dizer que são estruturados todos os steps necessários para que aplicação inicie com a lambda da Recolha dos dados e posteriormente a isso seja executada a lambda do Tratamento dos dados. Para isto, é criada uma *state machine* com a finalidade de criar o fluxo lógico da aplicação. Em termos práticos reflete o serviço de fluxo de trabalho visual utilizado para automatizar processos de negócio. A execução das lambdas da recolha dos dados à Loja A e Loja B, que embora representem processos distintos espelham tarefas idênticas como visto anteriormente. Para que possamos otimizar a performance e a eficiência do fluxo de tralhado, a execução das tarefas da recolha dos dados acontece em paralelo (execução da lambda recolha de dados da Loja A e execução da lambda recolha de dados da Loja B). Isto permite obter tempos de execução consistentes e aprimorar a utilização de recursos para reduzir custos operacionais.

- Como é garantida a qualidade da execução de cada lambda?

Com tratamento de erros: as funções de *parse* e armazenamento dos dados, são executadas apenas se as funções de recolha dos dados terminarem com sucesso; caso não terminem são manuseados os erros usando uma sofisticada funcionalidade de captura e tentativa do erro.

- Como é definido um Schedule para aplicação?

Para definirmos um Schedule para a aplicação, ou seja, para que seja lançada a execução da aplicação num determinado momento com a periocidade que se pretende, recorremos ao serviço CloudWatch. Em *Rules*, selecionamos a opção *Create Rule* e selecionamos como *target* a state machine criada. Depois disso definimos de forma simples a periocidade da aplicação (uma vez por dia) e o horário. Relativamente ao horário, este deve ser um aspeto com a qual os desenvolvedores devem prestar alguma atenção.

Como vimos anteriormente é uma boa prática em projetos de scraping a web sites, não definir um padrão de Schedule (pelo menos no horário) para que não sejam reconhecidos como *bots* de informação e assim se consiga evitar bloqueios. Um aspeto relevante é entender a velocidade de procedimento. Esta não deve ser excessiva para não corrermos o risco de sermos identificados como um "bot" e assim a página bloquear o acesso.

- Como é monitorizada a aplicação? Como são notificados os erros que possam ocorrer?

Durante todo o fluxo de trabalho, desde o momento que inicia, teremos os serviços de Amazon CloudWatch e *Amazon SNS* a acompanhar todas as tarefas nos momentos adequados e críticos. Através do CloudWatch conseguiremos monitorizar toda a aplicação através de *logs* no início e no fim de cada *lambda*. Entre muitos benefícios evidencia-se a extração de insights acionáveis com base em logs, permitindo a exploração, análise e visualização dos logs para solucionar problemas operacionais com facilidades. Por outro lado, o serviço de *Amazon Simple Notification Service*[41] *(AWS SNS),* permitirá enviar notificações de mensagens de texto para números de telemóveis configurados e/ou e-mails de texto simples para endereços de e-mail. Estas notificações divulgaram mensagens com um determinado *tópico*, relacionado com erros nas *lambdas*.

- Quais são as mais valias em registar o log de início e o log de fim de cada tarefa?

Além de termos acesso à hora de quando começou uma determinada ação, podemos consultar o tempo que foi necessário para essa execução. Ao longo do tempo, podemos avaliar a performance da execução. Sempre que existir um problema e o programa retornar erro, conseguimos perceber a tarefa exata onde ocorreu o erro e de forma rápida resolvê-lo. Além disso, existe a possibilidade de criação de *dashboards* para os *logs* e assim fazer comparações de execução das tarefas ao longo do tempo de forma clara. No fundo analisar o desempenho do programa.

- Onde são definidas questões de segurança?

Num ambiente de *Amazon Web Services (AWS)*, todas as questões de segurança, ou seja, gestão de acessos para os recursos AWS, são definidos através do *Identity and Access Management (IAM).* Gerir utilizadores, grupos de utilizadores, regras para os vários serviços ou até políticas de acesso de escrita ou leitura é feito através deste serviço.

[41] Toda a documentação sobre Amazon SNS poderá ser consultada em: https://aws.amazon.com/pt/sns/

4. RESULTADOS E DISCUSSÃO

4.1 CRIAÇÃO DE VALOR ATRAVÉS DA REUTILIZAÇÃO DE DADOS

Após o processo do desenvolvimento concluído e depois de algum tempo a extrair dados dos *websites*, procede-se à análise de dados.

Antes de qualquer trabalho e transformação aos dados com a finalidade de os transformar em *valor*, é oportuno validar a qualidade dos mesmos. No processo metodológico desenvolvido, torna-se necessário validar a qualidade dos dados em dois momentos distintos:

1. Por um lado, avaliar sempre que possível se os produtos disponíveis foram todos extraídos, contabilizando o número de produtos existentes na página face à quantidade de produtos extraídos nos ficheiros raw. Assim, muitas vezes, quando consultamos determinadas páginas de categorias de produtos temos a indicação de quantos números existem no total, e nesse caso conseguimos comparar o número de produtos extraídos para os ficheiros raw face ao número indicado na página;
2. Por outro lado, avaliar a veracidade dos dados que são sujeitos ao processo de transição dos ficheiros raw para a base de dados. Ou seja, validar se a quantidade de produtos inseridos na base de dados (depois das transformações) é igual ao número de produtos extraídos nos ficheiros raw.

Estes processos ditam não só a qualidade dos dados como a sua veracidade nas análises. Uma vez que no desenvolvimento da aplicação não foram desenvolvidos os processos de validação mencionados, não garantimos que todos os dados disponíveis nos websites num determinado dia tenham sido extraídos na sua totalidade para os ficheiros raw, e do mesmo modo, não garantimos que todos os dados guardados no serviço Amazon S3, tenham sido inseridos na Amazon RDS. A qualidade destes processos pode falhar por variadas razões, entre elas:

- No processo de extração dos dados:
    - Acesso negado ao processo de scraping ou ao processo de crawler no momento de extração
    - O acesso e os pedidos ao website estarem mais demorados que o habitual e por essa razão a lambda não conseguir terminar a extração de todos os dados em 15 minutos
    - O nome de uma determinada categoria ter-se alterado e por isso o processo de extração de dados daquela categoria ser inexistente
    - Falhas técnicas nas lambdas
    - Entre outras

- No processo de tratamento de dados:
    - O tempo para a lambda executar o processo de parse a todos os produtos pode não ser suficiente para a lambda correr em 15 minutos; considerando, por exemplo, que num determinado dia o ficheiro *raw* contem um número elevado de dados extraídos
    - O fluxo de trabalho (state machine do caso de uso) falhar no processo da recolha dos dados e, por isso, não executa o processo de tratamento de dados para os dados que conseguiu extrair
    - O ficheiro *raw* apresentar estruturas diferentes face aquela que foi desenvolvida para o parse dos produtos
    - Queries de inserção de dados falhar por algum motivo, por exemplo, tipo de dados
    - Entre outras

Assim, face às análises apresentadas de seguida, não é garantida a qualidade dos dados para as questões mencionadas.

Antes de se desenvolver qualquer análise aos dados, é importante definir o período de análise. Para vermos a potencialidade destas plataformas, foi selecionado um período curto de análise de quinze dias seguidos entre o dia 15 novembro de 2022 a 30 de novembro de 2022. Neste sentido, e para que o valor dos indicadores a desenvolver faça sentido, é importante restringir o mesmo período para ambas as lojas.

De realçar ainda que todas as análises desenvolvidas serão observadas de forma isolada para a Loja A e para a Loja B, de forma que se permita uma comparação entre as duas lojas e se obtenha valor acrescentado.

De realçar que quando queremos analisar o universo de produtos no mercado, em termos práticos queremos fazer uma correspondência entre os produtos da Loja A e da Loja B. Ou seja, identificar todos os produtos únicos no mercado e, para isso, temos que necessariamente estabelecer um match entre produtos e assim obter o dataset do mercado. Este processo pode ser executado de diversas formas, destacando:

a. Através do código de ean
b. Através da construção de algoritmos que procuram encontrar um match entre produtos
c. Manualmente comparando produtos, através de queries

Se considerarmos a hipótese a. Através do código de ean, implica necessariamente que na extração de dados nas duas lojas, na Loja A e na Loja B, terá de ser extraído o atributo ean de cada produto. Assim e como vimos anteriormente, a Loja A, por proceder à extração de dados através do acesso à API de dados do retalhista que disponibiliza no seu website (um dos processos do Web Scraping), não conseguimos aceder a este campo. Para contornar esta limitação poderíamos através de Web Crawler extrair o campo de ean caso este fosse apresentado na loja online. Porém, além deste campo não ser disponibilizado na API do retalhista este também não está disponível na loja online/website.

Por outro lado, se considerarmos a hipótese b. Através da construção de algoritmos que procuram encontrar um match entre produtos, seria necessário construir um algoritmo capaz de estabelecer comparações para regras previamente estabelecidas de forma a garantirmos com uma mínima margem de erro o match entre produtos. Devido à complexidade do tema, este é uma das recomendações para trabalhos futuros.

Por último, se o objetivo passa por ocasionalmente encontrar match entre produtos das duas lojas, pode-se sempre recorrer a queries em PostgreSQL.

Posto isto, passamos então às análises realizadas:

- **Número de produtos de cada retalhista no período de análise**

Para determinarmos o universo de produtos de cada retalhista, é importante definirmos qual o(s) campo(s) que dita(m) que um produto é único. Desta forma e da análise feita aos dados concluímos que um produto é único através da coluna *productcode* que define o código único de cada produto no universo de produtos de cada retalhista em questão.

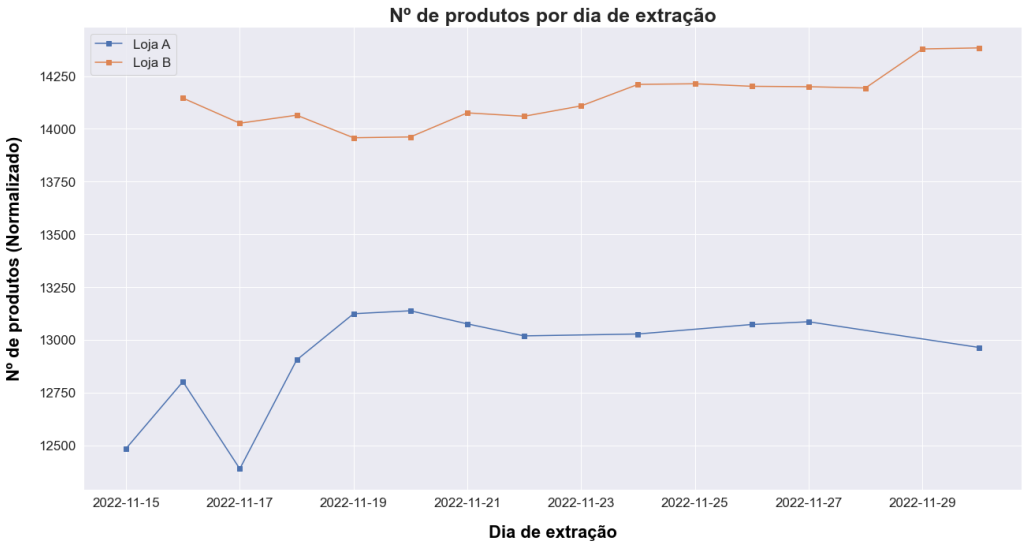

Figura 23 - Resultados: Nº de produtos por dia de extração

Analisando a figura acima apresentada - Figura 23 - Resultados: Nº de produtos por dia de extração – através de um histograma, conseguimos perceber que o processo falhou em alguns dias:

- Dia 15 de novembro: Para a Loja B
- Dia 23 de novembro: Para a Loja A
- Dia 25 de novembro: Para a Loja A
- Dia 28 e 29 de novembro: Para a Loja A

A correção do processo não foi efetuada para obtermos esta monitorização. As possíveis razões que levaram a estas falhas já foram mencionadas acima. Importa assim sublinhar que é relevante a ferramenta ganhar a sua maturidade e existir uma forte manutenção para evitar que o processo não seja executado num determinado período temporal.

Outro insight que podemos tirar da análise efetuada é que com base nos dados extraídos e apresentados, efetivamente a Loja B apresenta uma oferta maior que a Loja A. A Loja B tem disponível na loja online, em média, 14500 produtos e, por outro lado, a Loja A apresenta, em média, 12750 produtos disponíveis.

Um outro aspeto relevante a ter em conta é o facto de diariamente se registarem entradas e/ou saídas de produtos em ambos os retalhistas. Comprova-se assim, que as páginas sofrem uma alteração de dados de forma diária, e que a informação desta nunca é estática realçando-se assim aplicações neste sentido.

O código proveniente da presente análise pode ser consultado em anexo - ANEXO H – Código da análise Número de produtos de cada retalhista no período de análise.

- **Distribuição de Preços até 40€ com intervalos de 1€**

Para determinarmos a distribuição de preços por número de produtos é necessário, agrupar por retalhista, o número de produtos e a sua segmentação por preço aplicado. Neste sentido, foram necessárias as colunas *productcode* e *price,* ambos disponibilizados nas lojas online.

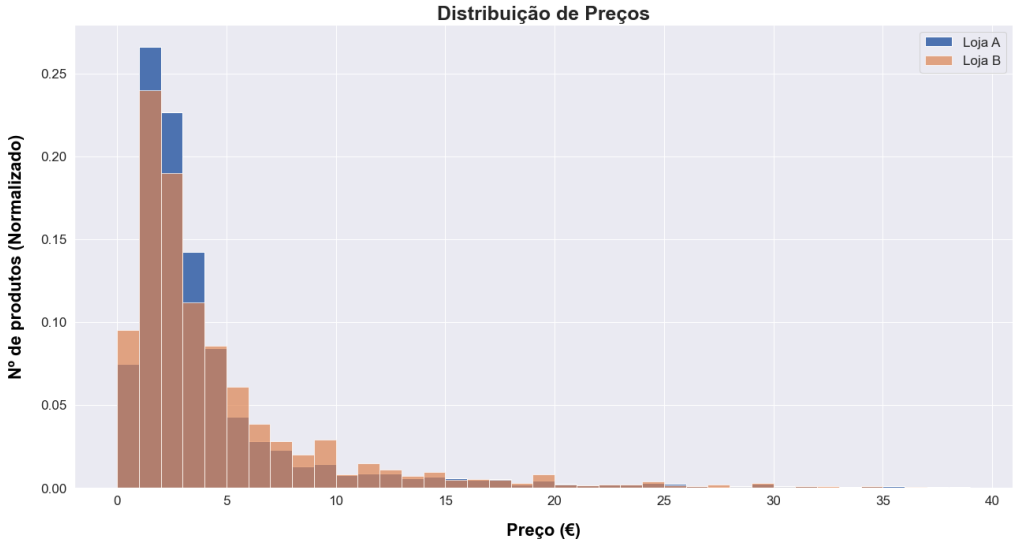

Figura 24 - Resultados: Distribuição de Preços até 40€ com intervalos de 1€

Analisando a figura acima apresentada - Figura 25 - Resultados: Distribuição de Preços até 40€ com intervalos de 1€ - através de um histograma, conseguimos avaliar a distribuição de preços até 40€ com intervalos de 1€. A analise é feita até aos 40€, uma vez que representa a maior conjunto de produtos. Assim, pela análise desenvolvida, conseguimos perceber os seguintes insights:

- A densidade de produtos concentra-se na distribuição de preços até 10€, aproximadamente, ou seja, os produtos apresentados são vendidos na sua maioria até 10€.

- Quando comparadas, a Loja A regista uma maior densidade de produtos até 5€, aproximadamente, que a Loja B. Contrariamente, a Loja B concentra uma maior densidade de produtos entre os 5€ e 20€ quando comparada com a Loja A. Numa primeira perspetiva parece-nos que a Loja B pratica valores mais altos que a Loja A.

Desta forma, desenvolvemos um Box plot, que nos permitirá comparar a política de preços praticada nas duas lojas e assim, entender qual a loja que pratica preços mais altos.

O código proveniente da presente análise pode ser consultado em anexo - ANEXO I – Código da análise Distribuição de Preços até 40€ com intervalos de 1€.

- **Distribuição de Preços por Loja**

Para determinarmos a distribuição de preços por loja é necessário, agrupar por retalhista, a sua segmentação por preço aplicado. Neste sentido, foram necessárias as colunas *productcode* e *price,* ambos disponibilizados nas lojas online.

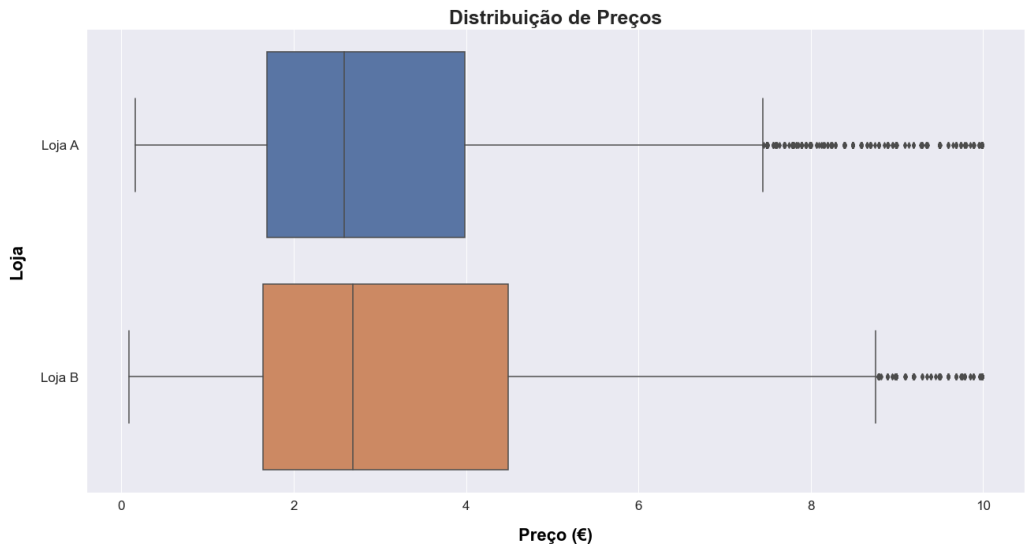

Figura 25 - Resultados: Distribuição de Preços por Loja: Loja A e Loja B

Analisando a figura acima apresentada – Figura 25 – Resultados: Distribuição de Preços por Loja: Loja A e Loja B - através de um Box plot, conseguimos avaliar a distribuição de preços até 10€. Focamos as nossas análises neste intervalo, uma vez que, na análise anterior conseguimos perceber que é o intervalo onde se concentra uma grande parte dos produtos extraídos.

Assim, através da análise apresentada podemos retirar os seguintes insights:

- A densidade dos produtos é distribuída por 4 quartis. Sendo que para cada Loja os quartis foram distribuídos de forma distinta:
  - 1º quartil: aproximadamente de 0,20€ a 1,80€ (tanto na Loja A como na Loja B)
  - 2º quartil: aproximadamente de 1,80€ a 2,60€ (no caso da Loja A) e aproximadamente de 1,80€ a 2,70€ (no caso da Loja B)
  - 3º quartil: aproximadamente de 2,60€ a 4€ (no caso da Loja A) e aproximadamente de 2,70€ a 4,30€ (no caso da Loja B)
  - 3º quartil: aproximadamente de 4€ a 7,50€ (no caso da Loja A) e aproximadamente de 4,30€ a 9€ (no caso da Loja B)
- Os *outliers* na distribuição de preços (valor atípico no intervalo de 0 a 10 €) no caso da Loja A registam-se a partir dos 7,50€ (aproximadamente)
- Os *outliers* de preços no caso da Loja A registam-se a partir dos 9€ (aproximadamente)
- No 2º e no 3º quartil é onde se registam maior densidade de produtos
- Consegue-se perceber pelos valores descritos que a Loja B, tem uma maior densidade de produtos em valores mais altos, ou seja, pratica preços mais altos quando comparados com a Loja A

O código proveniente da presente análise pode ser consultado em anexo - ANEXO J – Código da análise Distribuição de Preços por Loja.

- **Distribuição de Preços até 1€ com intervalos de 0,01€**

Com uma visão mais estratégica, desenvolvemos a seguinte análise numa perspética de entender a psicologia dos preços que é aplicada no mercado. Assim, para determinarmos a distribuição de preços por número de produtos é necessário, agrupar por retalhista, o número de produtos e a sua segmentação por preço aplicado. Neste sentido, foram necessárias as colunas *productcode* e *price,* ambos disponibilizados nas lojas online.

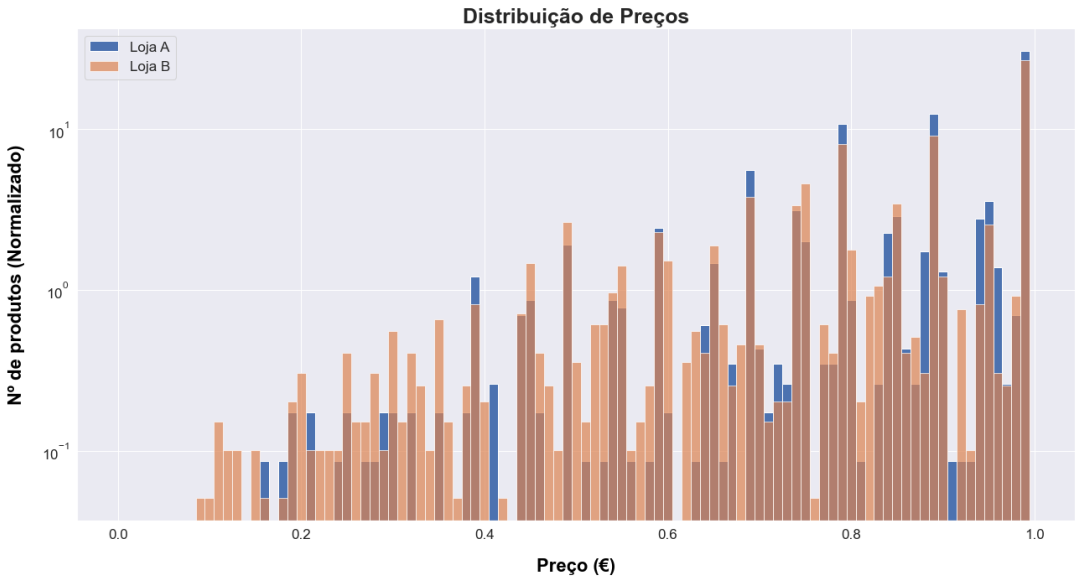

Figura 26 - Resultados: Distribuição de Preços até 1€ com intervalos de 0,01€

Analisando a figura acima apresentada - Figura 26 - Resultados: Distribuição de Preços até 1€ com intervalos de 0,01€ - através de um histograma, conseguimos avaliar a distribuição de preços até 1€ com intervalos de 0,01€. Neste caso, o foco é entender os preços acabados em 9 – por exemplo, 0,39; 0,89, etc) e se isso se reflete nos dados. Assim, pela análise desenvolvida, conseguimos perceber os seguintes insights:

- Efetivamente, analisando a figura acima, é praticada a psicologia dos preços em valores que terminem nos 0,99€. Os "picos", registados no gráfico refletem exatamente isso.
- São praticados com mais significância os preços quando terminam em 0,99€
- Verifica-se com mais frequência esta psicologia na Loja A (comprovada anteriormente como sendo aquela que pratica preços mais baixos)

O código proveniente da presente análise pode ser consultado em anexo - ANEXO K – Código da análise Distribuição de Preços até 1€ com intervalos de 0,01€.

- **Distribuição de Preços até 10€ com intervalos de 0,1€**

Com o mesmo objetivo que a análise anterior, construímos uma análise com a distribuição de preços até 10€ com intervalos de 0,1€, para validar se registamos o mesmo fenómeno. Assim, da mesma forma, para determinarmos a distribuição de preços por número de produtos é necessário, agrupar por retalhista, o número de produtos e a sua segmentação por preço aplicado. Neste sentido, foram necessárias as colunas *productcode* e *price,* ambas disponibilizadas nas lojas online.

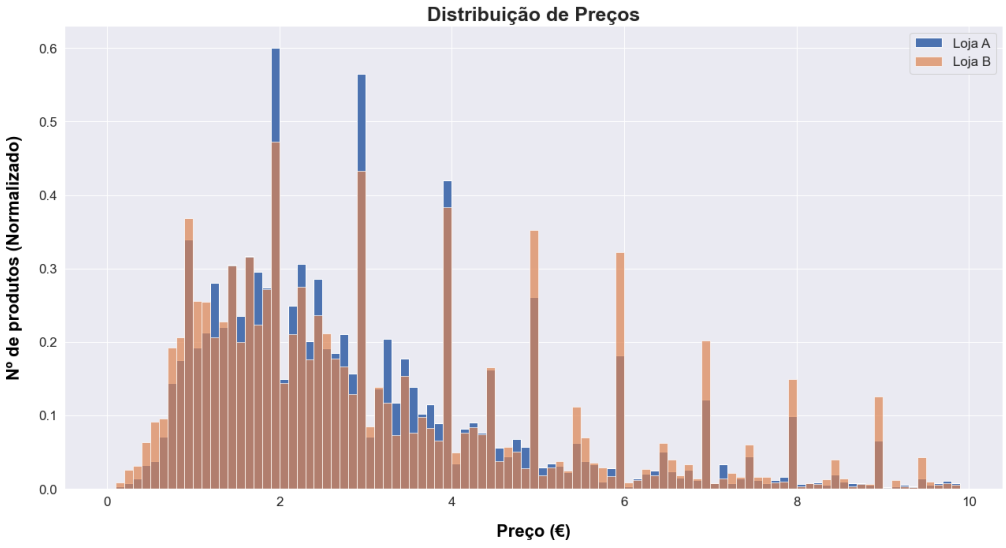

Figura 27 - Resultados: Distribuição de Preços até 10€ com intervalos de 0,1€

Analisando a figura acima apresentada - Figura 27 - Resultados: Distribuição de Preços até 10€ com intervalos de 0,1€ - através de um histograma, conseguimos avaliar a distribuição de preços até 10€ com intervalos de 0,1€. Neste caso, o foco é entender os preços acabados em 9 – por exemplo, 0,39; 0,89, etc) e se isso se reflete nos dados.

Assim, pela análise desenvolvida, conseguimos perceber os seguintes insights:

- Efetivamente, conseguimos validar com maior destreza, que a psicologia dos preços acabarem em 0,99€ é efetivamente praticada pelos dois retalhistas
- Neste gráfico, vemos também com mais clareza que a Loja B pratica esta política de estratégia com mais significância que a Loja A, em preços com valores mais altos que 5€
- Por outro lado, em preços inferiores, é a Loja A, quem pratica mais esta psicologia

O código proveniente da presente análise pode ser consultado em anexo – ANEXO L – Código da análise Distribuição de Preços até 10€ com intervalos de 0,1€.

- **Distribuição de Promoções por Preço Original**

Não limitando as análises aos preços praticados, foram também desenvolvidas análises aos descontos praticados pelas duas lojas. Para analisarmos as promoções praticadas, primeiramente analisaremos se existe alguma correlação entre os produtos que estão em promoção e o seu preço original.

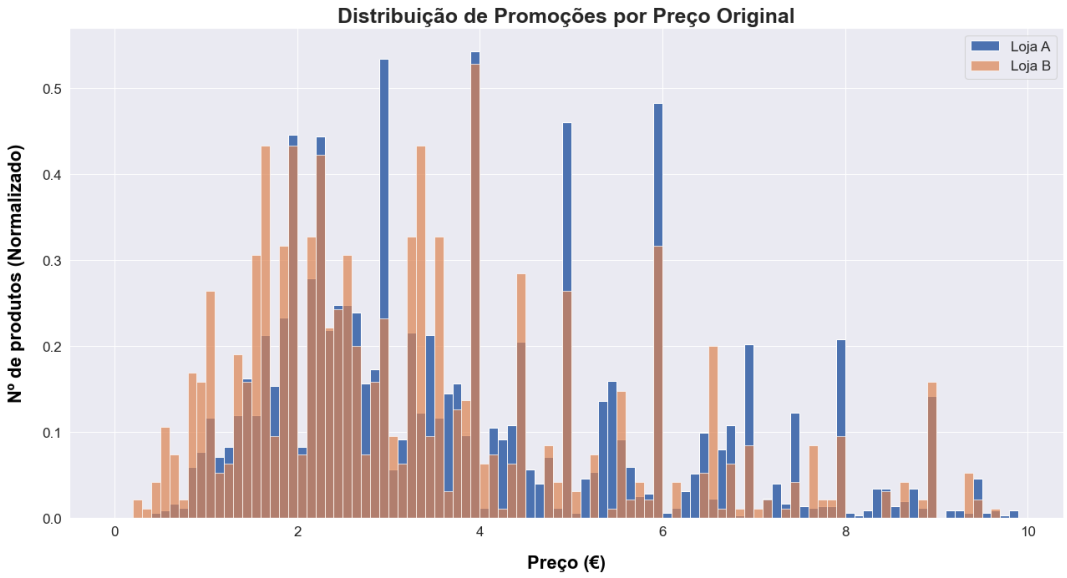

Figura 28 - Resultados: Distribuição de Promoções por Preço Original

Analisando a figura acima apresentada - Figura 28 - Resultados: Distribuição de Promoções por Preço Original - através de um histograma, conseguimos avaliar a distribuição de promoções por Preço Original.

Assim, pela análise desenvolvida, conseguimos perceber os seguintes insights:

- Que as promoções praticadas se concentram na sua maioria nos produtos com preços até os 5€, aproximadamente
- Aparentemente, a Loja A pratica muitas mais promoções que uma Loja B
- Existe uma correlação entre os produtos que terminam em 0,99€ e as promoções aplicadas? É levantada esta questão.

O código proveniente da presente analise pode ser consultado em anexo - ANEXO M – Código da análise Distribuição de Promoções por Preço Original.

- **Distribuição de Promoções Percentuais**

Por último, analisaremos a distribuição de promoções a nível percentual.

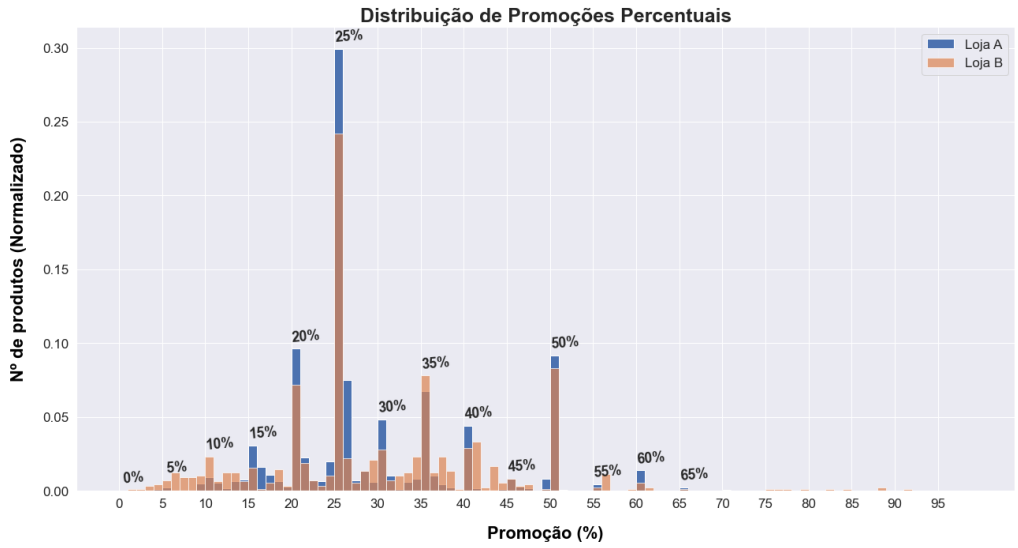

Figura 29 - Resultados: Distribuição de Promoções Percentuais

Analisando a figura acima apresentada - Figura 29 - Resultados: Distribuição de Promoções Percentuais - através de um histograma, conseguimos avaliar a distribuição de promoções percentuais por retalhista. Assim, pela análise desenvolvida, conseguimos perceber os seguintes insights:

- A promoção mais praticada por ambos os retalhistas é sem dúvida a dos 25% (sendo que a Loja A, pratica-a numa maior densidade de produtos quando comparada com a Loja B)
- Aparentemente, é praticada também uma psicologia de promoções, praticando com mais frequência promoções na casa dos 25% (como mencionado), 35%, 20%, 50%
- A Loja B regista mais promoções na casa dos 5% a 15%, aproximadamente, quando comparadas as duas Lojas

O código proveniente da presente analise pode ser consultado em anexo - ANEXO N – Código da análise Distribuição de Promoções Percentuais.

Uma vez as análises desenvolvidas e interpretadas, importa reter o seguinte:

- Todas as análises desenvolvidas, comtemplaram um período de análise de 15 de novembro de 2022 a 30 de novembro de 2022
- Embora o período de análise seja reduzido, conseguimos criar valor através da reutilização de dados extraídos
- As análises apresentadas podem servir diferentes finalidades:
  - Numa ótica de regulamentação, é possível analisar as políticas praticadas no mercado retalhista

- Numa ótica de acompanhamento, é possível construir análises descritivas e preditivas sobre os dados
- Numa ótica de retalhista com negócio online, é possível monitorizar a concorrência e tomar decisões mais assertivas, mediante o que é praticado no mercado
- Numa ótica de fornecedor, é possível avaliar o que é praticado no mercado e discutir novas parcerias com novos retalhistas

4.2 MODELO DA ECONOMIA CIRCULAR DE DADOS

Do processo desenvolvido, através da implementação do caso de uso e da abordagem aos conceitos envolventes analisando e discutindo as suas potencialidades, surge a necessidade de abordar um modelo da economia circular dos dados.

Analogicamente, se pensarmos na economia circular, existe um caminho a percorrer desde o momento em que um par de calças é depositada em repositórios de reutilização até ao momento de serem um novo produto. Assim, existe também um longo caminho desde que o dado é disponibilizado para uma finalidade até servir outro fim.

Assim, o processo metodológico para dar resposta ao objetivo da investigação *A criação de valor através de dados abertos*, além das análises construídas e interpretadas, teve como resultado a construção de um modelo da economia circular de dados e que desperta novos conhecimentos e novas abordagens para a área científica de dados. No fundo, permitirá um ciclo de vida dos dados maior e, por isso, permite um maior prolongamento da informação.

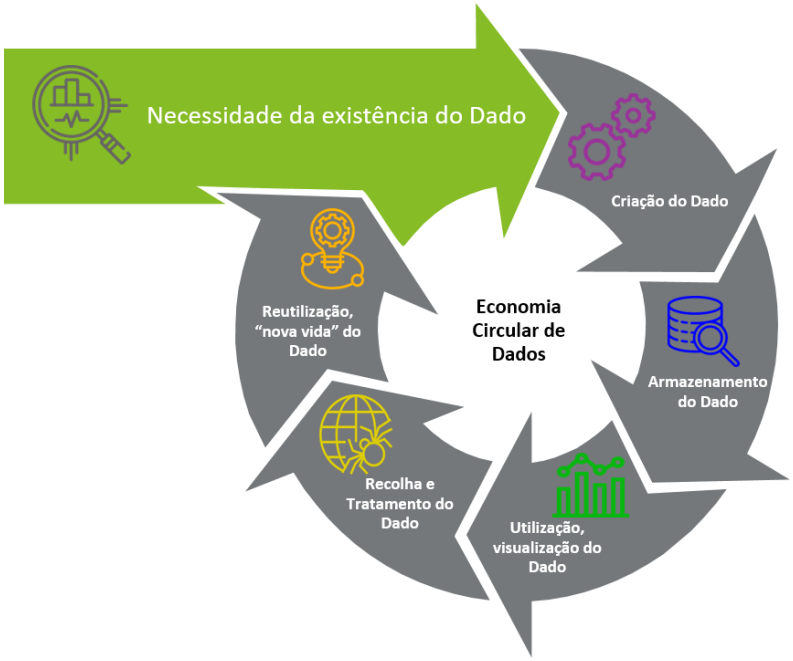

Figura 30 - Resultados: Modelo da economia circular dos dados

Através da figura apresentada - Figura 30 - Resultados: Modelo da economia circular dos dados – e analisando o modelo descrito, não só é possível a expansão de temáticas de dados abertos, recolha e a sua reutilização como a potencialização da natureza económica e intelectual, como contribui para

uma sustentabilidade mais consciente e eficaz na medida em que contribui para a redução de recursos. Caso não exista a possibilidade de recolha de dados abertos, existe naturalmente outras formas de obter dados (o que podemos observar no processo metodológico) e muitas das vezes exigem mais recursos, mais energia elétrica, mais recursos humanos.

Do modelo apresentado, identificam-se as seguintes vantagens:

- Nível de paridade em relação aos concorrentes (em mercados online)
- Recolha de informação de forma gratuita e a qualquer momento
- Informação estruturada e organizada
- Informação sempre atualizada
- Tempo e eficiência na obtenção de dados
- Contributo para a economia sustentável na medida em que existe uma redução de recursos para a aquisição dos dados (por exemplo, recursos humanos que consequentemente exige mais capital de natureza económica)
- Capacidade infinita de reutilizar dados de forma a estimular e a prolongar a vida dos dados
- Mais poder de conhecimento
- Estimular redes de conhecimento
- Questões éticas na obtenção de dados

5. CONCLUSÕES, LIMITAÇÕES E RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Dada a conclusão do projeto é importante refletir sobre o trabalho desenvolvido e os resultados obtidos. Como projeto final, para o grau de mestre em Gestão de Informação, propôs-me a desenvolver uma ferramenta altamente tecnológica capaz de criar valor no mercado de retalho online em Portugal.

Foi um projeto longo, com muitos recuos e avanços, mas que no final obtive o esperado: através de análises aos dados recolhidos, obtivemos conhecimento sobre os preços e descontos praticados por dois retalhistas a operar em Portugal, mesmo apenas com um curto período em análise. Sabemos que o mercado de retalho vai muito além de apenas 2 retalhistas e, por isso, o presente trabalho caracteriza-se como uma prova conceito de que é possível escalar a ferramenta a outros retalhistas e negócios online. Como prova conceito, sabemos, neste momento, que através do conhecimento das tecnologias corretas podemos tornar maduro o tema dos abertos abertas nas organizações e efetivamente criar valor de diversas formas sobre os dados disponíveis e obtidos.

Assim, de forma global os resultados do projeto vão ao encontro do que foi definido desde o início. É um projeto/serviço que tem sempre em atenção as boas práticas de programação, segurança e confidencialidade dos dados.

Relativamente às limitações nos dados abertos, recolha e reutilização, observados do processo metodológico desenvolvido na presente investigação, específico os seguintes aspetos:

- Complexidade em encontrar e implementar soluções que visam ultrapassar medidas de proteção da informação por parte de algumas fontes de informação
- Bloqueamentos de *IP's* durante os processos que derivam de proteções das fontes de informação
- Informação que é disponível apresentar erros, podendo não estar correta (por exemplo, a fonte de informação primária à partida carrega a sua informação através de outros sistemas de dados – base de dados, sistemas de ERP, entre outros – tais sistemas, podem em algum momento apresentar inconsistências por derivadas razões e a informação que apresenta, neste caso, na loja online, consequentemente será impactada com esses erros.
- Sempre que existem alterações na estrutura como os websites estão desenhados, implica a alteração à forma como se acedem aos dados; por sua vez, este tempo de refazer o programa implicará um aumento no tempo estimado para o processo de web scraper. Mesmo que o bot já tenha sido desenvolvido. Mediante o nível da dimensão das alterações, pode representar um nível de complexidade avançado para as modificações no programa.
- Para estabelecer um nível de complexidade às alterações nas fontes de informação fontes deve-se considerar as diferentes naturezas pela qual podem ser sido causadas.
- Por um lado, os desenvolvedores podem proteger-se contra técnicas de anti scraping e, por isso, exige que se façam alterações à forma como acedemos ao conteúdo da página. Normalmente, é necessária uma alteração à forma como implementamos o *bot* e acrescentar-lhe processos que se pareça o máximo possível com um utilizador normal a tentar aceder à

página. Por outro lado, os desenvolvedores das páginas podem apenas fazer alterações ao nível da estrutura e desenho do web site, como por exemplo, o nome das classes ou a estrutura dos produtos nas páginas. À partida, estas alterações implicam adaptações do bot muito menos complexas e fáceis de resolver.

Por fim, importa perceber o que podemos fazer com a ferramenta construída como sugestão para trabalhos futuros:

- Desenvolvimento de algoritmos match para comparação de produtos entre vários concorrentes
- Implementação de modelos de previsão de preços, descontos, entre outros
    - Ver retalhistas internacionais e tendências por país, clima, estação do ano, religião, eventos sociais (regresso às aulas, semana do animal, festival da canção, concertos, entre outros)

Em suma, espero efetivamente que o presente trabalho espelhe todo o conhecimento que adquiri ao longo do processo de desenvolvimento e do processo de escrita e que ajude todos os leitores que buscam conhecimento.

6. BIBLIOGRAFIA

Mário Antunes & Baltazar Rodrigues. (2018). Introdução à cibersegurança.

Agência para a Modernização Administrativa, IP. Dados.Gov. (2016). Guia de Introdução aos dados abertos. https://www.ama.gov.pt/documents/24077/24804/guia_introdu__o_dados_abertos_ama.pdf/9b40b98c-4935-471b-af5d-f6f6a656edc0

European Data Portal. (2020). The Economic Impact of open Data - Opportunities for value creation in Europe. https://data.europa.eu/sites/default/files/the-economic-impact-of-open-data.pdf

European Data Portal. (2020). Open Data Maturity. https://data.europa.eu/sites/default/files/open_data_maturity_report_2019.pdf

Rui Barros. (2019, 3 dezembro). Portugal na cauda da Europa na acessibilidade a dados públicos. https://rr.sapo.pt/2019/12/04/pais/portugal-na-cauda-da-europa-na-acessibilidade-a-dados-publicos/noticia/173978/

Rui Barros. (2019, 4 dezembro). Governo responde às críticas de relatório europeu sobre acesso a dados públicos. "Estamos a trabalhar numa estratégia nacional". https://rr.sapo.pt/2019/12/04/pais/governo-responde-as-criticas-de-relatorio-europeu-sobre-acesso-a-dados-publicos-estamos-a-trabalhar-numa-estrategia-nacional/especial/174114/

European Data Portal. (2022). Open Data Maturity. https://data.europa.eu/sites/default/files/data.europa.eu_landscaping_insight_report_n8_2022_1_0.pdf

Bart Custers & Helena Ursic. (2022). Big Data and Data Reuse - A Taxonomy of Data Reuse for Balancing Big Data Benefits and Personal Data Protection. https://deliverypdf.ssrn.com/delivery.php?ID=033064094066086090103094116065077125028059005053019029122064096125112099107002091100056013012059041007046066096127065120019018112048062045006096000122125096111030079022001018104107072086090118008096064080114006031110022126008069001117124106009011080009&EXT=pdf&INDEX=TRUE

Stich A Talend Product. (2022). What is Data Extraction? Data Extraction Tools & Techniques. https://www.stitchdata.com/resources/what-is-data-extraction/

Pplware. (2020, 11 dezembro). Web Scraping – saiba o que é e para que serve. https://pplware.sapo.pt/internet/web-scraping-saiba-o-que-e-e-para-que-serve/

Craig S. Mullins & Mullins Consulting. (janeiro 2020). Extract, Load, Transform (ELT). https://www.techtarget.com/searchdatamanagement/definition/Extract-Load-Transform-ELT

Instat Institute of Statistic. (2021). Usage of Information and Communication Technologies in Enterprises. http://instat.gov.al/media/9227/ict-2021_final_eng.pdf

Eric Jonas, Johann Schleier-Smith, Vikram Sreekanti, Chia-Che Tsai, Anurag Khandelwal, Qifan Pu, Vaishaal Shankar, Joao Menezes Carreira, Karl Krauth, Neeraja Yadwadkar, Joseph Gonzalez, Raluca Ada Popa, Ion Stoica & David A. Patterson. (2019, 10 Fevereiro). Electrical Engineering and Computer Sciences. https://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-3.pdf

Rogério Bravo. (2014). "OPEN SOURCES" NA INVESTIGAÇÃO DO CIBERCRIME: conceito e implicações.

https://www.academia.edu/5906230/O_Conceito_de_FonteO%20Conceito%20de%20%22Fontes%20Abertas%20%22%20na%20Investiga%C3%A7%C3%A3o%20do%20Cibercrime._Abertas_na_Investiga%C3%A7%C3%A3o_do_Cibercrime

7. ANEXOS

- **ANEXO A – Questionário SONAE**

Caracterização do retalho alimentar online

O presente questionário enquadra-se numa produção cientifica acerca do retalho alimentar online no âmbito de uma dissertação como requisito parcial para obtenção do grau de Mestre em Gestão de Informação, com especialização em Gestão do Conhecimento e Business Intelligence, pela NOVA Information Management School – Universidade Nova de Lisboa, com a autoria de Cláudia Maria Silva Pimenta e com a orientação do professor Pedro Manuel Carqueijeiro Espiga da Maia Malta.

Os resultados obtidos serão utilizados apenas para fins académicos (inclusão na dissertação), sendo realçado que as respostas representarão algumas conclusões acerca da monitorização da concorrência por parte dos retalhistas.

Assim sendo, muito obrigada pelo seu tempo. Se quiser fornecer alguns dados pessoais (nome) irá constar nos agradecimentos da tese pelo seu contributo (além do agradecimento à própria empresa SONAE).

1. **As estratégias implementadas na loja online ao nível dos preços, promoções, entradas e saídas dos produtos são as mesmas que são implementadas nas lojas físicas?**
   (Resposta única)

   [ ] Sim [x] Não

2. **Na implementação de estratégias, é feita uma monitorização da concorrência em algumas das seguintes vertentes? Selecione as práticas que são monitorizadas.**
   (Resposta múltipla)

   [x] Preço

   [x] Descontos/Promoções nos produtos

   [x] Grandes campanhas

   [ ] Oferta de produtos

   [ ] Outra. Por favor, indique qual: _____

   [ ] Nenhuma (Se selecionou esta hipótese, por favor, siga para a questão 6)

3. Sendo feita uma monitorização da concorrência, esta é feita de forma diferente entre o mercado online e físico?
(Resposta única)

[x] Sim [ ] Não

3.1. Se sim, quais são as diferenças e que estratégias de monitorização são implementadas no mercado físico e no mercado online? Se possível, explicite em traços gerais quais as estratégias implementadas em cada um dos mercados (i.e.: processo Shopping, serviços de consultoria, software…).
(Resposta aberta)

Os processos de shopping online são obtidos através de service providers especializados neste tipo de operações, enquanto o processo físico é suportado por uma equipa de profissionais que recolhem *in loco* os preços da concorrência.

3.2. Se não, que estratégias de monitorização são implementadas? Se possível, explicite em traços gerais quais as estratégias implementadas (i.e.: processo Shopping, serviços de consultoria, software…).
(Resposta aberta)

4. Num mercado onde os preços/descontos/campanhas variam constantemente com que frequência fazem essa monitorização?
(Resposta múltipla)

[x] Diária
[ ] Semanal
[ ] Mensal
[ ] Anual
[ ] Outra. Por favor, indique qual: _____

5. Que técnicas/tecnologias utilizam para cruzar a informação da concorrência com a vossa?
(Resposta aberta)

São utilizados "tradutores" para mapear os artigos da concorrência à gama dos nossos produtos. Nos artigos com o mesmo EAN não é necessário desenvolver qualquer tipo de mapeamento. Todo o processo é suportado em bases de dados e algoritmos sofisticados que nos permitem medir o posicionamento competitivo.

6. Que peso tem a informação da concorrência nas estratégias implementadas ao nível de preços, promoções, entradas e saídas dos produtos no mercado online?
(Resposta de escala)

[ ] 1 [ ] 2 [ ] 3 [ ] 4 [x] 5

7. Qual o grau de maturidade das tecnologias (i.e.: nível de uso que a empresa já adota) na visualização de dados que a empresa tem implementadas para o suporte na tomada de decisão nas estratégias de pricing e marketing do mercado online?
(Resposta de escala)

[ ] 1 [ ] 2 [ ] 3 [x] 4 [ ] 5

- **ANEXO B – exemplo de uma API que contêm e disponibiliza dados no *website***


```
{
   "request":{
      "Refiners":[{"Name":"originallistprice","Value":""},{ "Name":"primaryparentcategoryname","Value":""},{"Name":"brand","Value":""} ],
      "SearchText":"",
      "SearchScope":"1",
      "CategoryId":"{0}"".format(search_term)",
      "NumberOfItemsToReturn":"{0}"".format(results_per_page)",
      "CurrentPage":"{0}"".format(pagenum)",
      "Sort":[],
      "PropertiesToReturn":[
         "ProductId",
         "DisplayName",
         "Brand",
         "WebPackageContentInfo",
         "PriceCapacityRatio",
         "PriceCapacityratioUnit",
         "Listprice",
         "OriginalListPrice",
         "ProductCode",
         "CatalogId",
         "OriginalSalesUnit",
         "OriginalAlternativeSalesUnit",
         "VariantInfoXml",
         "UnitConversionRate",
         "SalesUnit",
         "ContentType",
         "IsAManufacturerClubProduct",
         "IsOwnBrandProduct",
         "IsOwnBrandSelectedProduct",
         "IsAGourmetProduct",
         "IsABiologicProduct",
         "IsANationalProduct",
         "IsAGlutenFreeProduct",
         "IsALactoseFreeProduct",
         "IsANewProduct",
         "HasABuyXGetYFreeDiscount",
         "AlternativeSalesUnit",
         "IsInPromotion",
         "IsATopTenProduct",
         "IsADonationEnabledProduct",
         "IsACheapProduct",
         "IsAnExclusiveProduct",
         "AvailabilityInDays",
         "WebDiscountStartDate",
         "WebDiscountEndDate",
         "WebDiscountValue",
         "WebDiscountIcon",
         "RetekDiscountStartDate",
         "RetekDiscountEndDate",
         "WebDiscountType",
         "RetekDiscountValue",
         "RetekDiscountIcon",
         "RetekDiscountType",
         "WebDisplayName",
         "DefinitionName",
         "CanonicalCategoriesAggregated",
         "AdditionalSaleOptions",
         "DeferredDelivery",
         "WebDiscountQuantity",
         "TaxCode",
         "ProvisioningStore",
         "MaxNumberOfUnitsPerSale",
         "MinNumberOfUnitsPerSale"
      ],
      "SpecialCategoryKey":"",
      "ContentTypes":[
         "Product"
      ],
      "PreventProductsInBasket":"false"
   }
}
```

- **ANEXO C – exemplo de um *output* de um *script* de uma categoria**


```
[[{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'AdditionalSaleOptions', 'Value': ''}, {'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'AlternativeSalesUnit', 'Value':
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'Brand', 'Value': 'lojaA'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'CanonicalCategoriesAggregated', 'Value': '<r><ri pid="74" cid="6891" n="Mercearia(eCsf_WebProductCatalog_MegastoreLojaAOnline_LojaA_EUR_Colombo_PT)" v="Me
tCatalog_MegastoreLojaAOnline_LojaA_EUR_Colombo_PT)" v="Cápsulas" d="WebCategoryDefinition" e="1"/><ri pid="14689" cid="95741" n="mercearia-doce-capsulas-nespresso(eCsf_WebProductCatalog_MegastoreLojaAOnline_LojaA_EUR_Colombo_PT)" v="Nes
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'DisplayName', 'Value': 'CAFÉ LojaA GOA NSP 10CAP'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'HasABuyXGetYFreeDiscount', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsABiologicProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsACheapProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsADonationEnabledProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsAGlutenFreeProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsAGourmetProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsALactoseFreeProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsAManufacturerClubProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsANationalProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsANewProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsAnExclusiveProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsATopTenProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsInPromotion', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsOwnBrandProduct', 'Value': 'true'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsOwnBrandSelectedProduct', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'Listprice', 'Value': '2390.0'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'MaxNumberOfUnitsPerSale', 'Value': '50'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'MinNumberOfUnitsPerSale', 'Value': '1'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'OriginalAlternativeSalesUnit', 'Value': ''},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'OriginalListPrice', 'Value': '2.39'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'OriginalSalesUnit', 'Value': 'Unit'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'PriceCapacityRatio', 'Value': '0.239'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'PriceCapacityratioUnit', 'Value': 'Unit'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'ProductCode', 'Value': '5183370'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'ProductId', 'Value': '5183370(eCsf_RetekProductCatalog_MegastoreLojaAOnline_LojaA)'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'ProvisioningStore', 'Value': ''},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'RelevanceScore', 'Value': '31.9'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'RetekDiscountEndDate', 'Value': '2020-01-20T23:59:59Z'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'RetekDiscountIcon', 'Value': 'superPriceIcon'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'RetekDiscountStartDate', 'Value': '2020-01-14T00:00:00Z'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'RetekDiscountType', 'Value': 'Value'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'RetekDiscountValue', 'Value': '0.7200'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'SalesUnit', 'Value': 'Unit'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'StoreOrder', 'Value': '1'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'TaxCode', 'Value': '1'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'UnitConversionRate', 'Value': ''},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'VariantInfoXml', 'Value': ''},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'WebDiscountEndDate', 'Value': '2015-11-08T23:59:59Z'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'WebDiscountIcon', 'Value': 'superPriceIcon'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'WebDiscountQuantity', 'Value': ''},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'WebDiscountStartDate', 'Value': '2015-10-19T00:00:00Z'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'WebDiscountType', 'Value': 'Value'}, {
'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'WebDiscountValue', 'Value': '0,63'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'WebDisplayName', 'Value': 'Café Goa Cápsulas'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'WebPackageContentInfo', 'Value': 'emb. 10 un -compatível com Nespresso*'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'Score', 'Value': '29.0'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'ChannelId', 'Value': '1'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'ChannelName', 'Value': 'LojaA'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'ChannelDisplayName', 'Value': 'LojaA Online'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'InventoryCondition', 'Value': '0'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'OriginalListPriceTax', 'Value': '0.4469'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'PriceCapacityRatioTax', 'Value': '0.0447'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'RetekDiscountValueTax', 'Value': '0.1346'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'WebDiscountValueTax', 'Value': '11.7805'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'IsAlreadyInBasket', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'HasRetekLoyaltyDiscounts', 'Value': 'false'},
{'__type': 'Property:urn:Sonae.eCsf.Presentation.Services.DataContracts', 'Name': 'HasWebLoyaltyDiscounts', 'Value': 'false'}],[...]
```

- **ANEXO D – exemplo de um pedido Loja A**

```
################################################
#
# function that return the JSON of all the products for each existent category
#
################################################
def GetAllProductsByCategoryFromSite(search_term, results_per_page=80):
    URL = 'https://www.LojaA.pt/stores/continente/pt-pt/public/Pages/searchResults.aspx'
    payload = {
        'globalSearch': search_term
    }
    session = requests.session()
    page = session.get(URL, params=payload)
    if (page.status_code != requests.codes.ok):
        print('[WEBSCRAPPER] There was an issue loading the page {0}'.format(page.url))
        exit(-1)
    URL_EXECUTE_SEARCH = "https://www.LojaA.pt/stores/LojaA/_vti_bin/eCsfServices/SearchServices.svc/ExecuteSearch"
    session.headers['Content-type'] = 'application/json;charset=UTF-8'
    session.headers['Accept'] = 'application/json'
    pagenum = 0
    RESULTS = []
    while page.status_code == requests.codes.ok:
        pagenum += 1
        post_payload = {"request":
            {"Refiners":
                [{"Name": "originallistprice","Value": ""},
                    {"Name": "primaryparentcategoryname","Value": ""},
                    {"Name": "brand", "Value": ""}
                ],
                "SearchText": "",
                "SearchScope": "1",
                "CategoryId": "{0}",
                "NumberOfItemsToReturn": "{0}",
                "CurrentPage": "{0}",
                "Sort": [],
                "PropertiesToReturn":
                    ["ProductId", "DisplayName", "Brand", "WebPackageContentInfo", "PriceCapacityRatio",
                     "PriceCapacityratioUnit", "Listprice"
                     "OriginalListPrice", "ProductCode", "CatalogId", "OriginalSalesUnit",
                     "OriginalAlternativeSalesUnit", "VariantInfoXml",
                     "UnitConversionRate", "SalesUnit", "ContentType", "IsAManufacturerClubProduct",
                     "IsOwnBrandProduct",
                     "IsOwnBrandSelectedProduct", "IsAGourmetProduct", "IsABiologicProduct", "IsANationalProduct",
                     "IsAGlutenFreeProduct", "IsALactoseFreeProduct", "IsANewProduct", "HasABuyXGetYFreeDiscount",
                     "AlternativeSalesUnit", "IsInPromotion", "IsATopTenProduct", "IsADonationEnabledProduct",
                     "IsACheapProduct", "IsAnExclusiveProduct", "AvailabilityInDays", "WebDiscountStartDate",
                     "WebDiscountEndDate",
                     "WebDiscountValue", "WebDiscountIcon", "RetekDiscountStartDate", "RetekDiscountEndDate",
                     "WebDiscountType",
                     "RetekDiscountValue", "RetekDiscountIcon", "RetekDiscountType", "WebDisplayName", "DefinitionName",
                     "CanonicalCategoriesAggregated", "AdditionalSaleOptions", "DeferredDelivery",
                     "WebDiscountQuantity", "TaxCode",
                     "ProvisioningStore", "MaxNumberOfUnitsPerSale", "MinNumberOfUnitsPerSale"
                     ],
                "SpecialCategoryKey": "",
                "ContentTypes":
                    ["Product"],
                "PreventProductsInBasket": "false"
            }}
        page = session.post(URL_EXECUTE_SEARCH, json=post_payload)
        json_page = page.json()
        N = len(json_page["d"]["SearchResultItems"])
        if N != 0:
            RESULTS.extend(json_page["d"]["SearchResultItems"])
        if N != results_per_page:
            # print(RESULTS)
            return RESULTS
```

- **ANEXO E – Exemplo de código Lambda**

```
#######################################################
#
# get of the searches to the page of the continent
# send json file from lambda function to bucket
#
#######################################################
lambda_client = boto3.client('lambda',  region_name='us-east-1')
def main(event, context):
  extraction_now = datetime.date.today()
  log.log_message("Starting the ingest of LojaA with event:")
  log.log_message(event)
  log.log_message("Getting highlighted brand")
  mainbrands = iP.GetHighLightedBrands()
  log.log_message("Ending.")
  log.log_message("Getting highlighted products.")
  mainproducts = iP.GetHighLightedProducts()
  contentmainbrands = str(mainbrands)
  contenmainproducts = str(mainproducts)
  log.log_message("Ending.")
  bucket_name = "matchproduct"
  file_name_mainbrands = "highlighted_brands.json"
  file_name_mainproducts = "highlighted_products.json"
  bucket_path = 'rawzone_LojaA/' + str(extraction_now)
  lambda_path_mainbrands = bucket_path + "/" + file_name_mainbrands
  lambda_path_mainproducts = bucket_path + "/" + file_name_mainproducts
  s3 = boto3.resource("s3")
  bucket = s3.Bucket(bucket_name)
  bucket.put_object(Key=lambda_path_mainbrands, Body=str(contentmainbrands))
  bucket.put_object(Key=lambda_path_mainproducts, Body=str(contenmainproducts))
  log.log_message("Getting categories.")
  categories = uf.category_titles
  log.log_message("Ending.")
  for category in categories:
    category_name = category.split("(")[0]
    file_name_category = bucket_path + "/" + category_name + ".json"
    log.log_message("Getting products for category " + category_name)
    products = json.dumps(iP.GetAllProductsByCategoryFromSite(category))
    log.log_message("Ending.")
    bucket.put_object(Key = file_name_category, Body=str(products))
  log.log_message("Finalizing the lambda.")
  return {'message': "sucess", "StatusCode": 200}
```

- **ANEXO F – Exemplo de um runner Crawler**

```
runner = CrawlerProcess({
        'USER_AGENT': 'AppleWebKit/537.36 (KHTML, like Gecko)',
        'FEED_FORMAT': 'json',
        'FEED_EXPORT_ENCODING': 'utf-8',
        'FEED_URI': filename,
        'FEED_OVERWRITE': True
    })
```

- **ANEXO G – Exemplo de uma função *Parse***

```
def ParseProduct(self, product_LojaA):
    if len(self.data_selection) == 0:
        log.log_message("Product being parsed has no selection")
        return {}
    product_event = {}
    valueCategory = []
    listCategories_name = []
    listCategories_id = []
    for selection in self.data_selection:
        Name = product_LojaA[selection]['Name']
        Value = product_LojaA[selection]['Value']
        product_event[Name] = Value
    if 'productimage' in self.EXTRA_SQL_columns:
        image_url = 'https://media.LojaA.pt/Sonae.eGlobal.Presentation.Web.Media/media.axd' \
                    '?resourceSearchType=2&resource=ProductId=' + str(product_LojaA[33]['Value']) + \
                    '(eCsf$RetekProductCatalog$MegastoreLojaAOnline$LojaA)&siteId=1&channelId=1&width' \
                    '=180&height=170&defaultOptions=1'
        product_event['productimage'] = image_url
    if 'categorylevel1' in self.EXTRA_SQL_columns:
        selection = 4  # Canonical Aggregate - value of categories Continete
        soup = BeautifulSoup(product_LojaA[selection]['Value'], 'html.parser')
        valueCategory += soup.find('r')
        for i in valueCategory:
            listCategories_id.append(i['n'])
            listCategories_name.append(i['v'])
        listCategoryId = listCategories_id
        new_listCategoryId = list(dict.fromkeys(listCategoryId))
        result = self.levelCategories_flat(new_listCategoryId)
        product_event['categorylevel1'] = result[0]
        product_event['categorylevel2'] = result[1]
        product_event['categorylevel3'] = result[2]
        product_event['categorylevel4'] = result[3]
    if "discountvalueweb" in self.EXTRA_SQL_columns:
        if product_LojaA[52]['Value'] == '':
            product_event["discountvalueweb"] = None
        else:
            product_event["discountvalueweb"] = float(product_LojaA[52]['Value'].replace(",", "."))
    if "pricewithpromotion" in self.EXTRA_SQL_columns:
        if product_event['IsInPromotion'] == 'true':
            if (not product_event['OriginalListPrice']) or (not product_event['RetekDiscountValue']) or (
                    not product_event['PriceCapacityRatio']):
                product_event['pricewithpromotion'] = None
                product_event['pricecapacityratiowithpromotion'] = None
            elif product_event['RetekDiscountType'] == 'Value':
                pricewithpromotion = float("{0:.2f}".format(float(product_event['OriginalListPrice']) - float(
                    product_event['RetekDiscountValue'])))
                pricecapacityratiowithpromotion = float("{0:.2f}".format((float(pricewithpromotion) * float(
                    product_event['PriceCapacityRatio'])) / float(product_event['OriginalListPrice'])))
                product_event['pricewithpromotion'] = pricewithpromotion
                product_event['pricecapacityratiowithpromotion'] = pricecapacityratiowithpromotion
            elif product_event['RetekDiscountType'] == 'Percentage':
                ValuePromotion = float("{0:.2f}".format(float(product_event['OriginalListPrice']) * (
                        float(product_event['RetekDiscountValue']) / 100)))
                ValuePromotionCapacity = float("{0:.2f}".format(float(product_event['PriceCapacityRatio']) * (
                        float(product_event['RetekDiscountValue']) / 100)))
                pricewithpromotion = float(
                    "{0:.2f}".format(float(product_event['OriginalListPrice']) - float(ValuePromotion)))
                pricecapacityratiowithpromotion = float(
                    "{0:.2f}".format(float(product_event['PriceCapacityRatio']) - float(
                        ValuePromotionCapacity)))
                product_event['pricewithpromotion'] = pricewithpromotion
                product_event['pricecapacityratiowithpromotion'] = pricecapacityratiowithpromotion
            else:
                product_event['pricewithpromotion'] = None
                product_event['pricecapacityratiowithpromotion'] = None
        elif product_event['IsInPromotion'] == 'false':
            product_event['pricewithpromotion'] = None
            product_event['pricecapacityratiowithpromotion'] = None
    return product_event
```

- **ANEXO H – Código da análise Número de produtos de cada retalhista no período de análise**

```
def products(show=True):
    conn = getbdconnection()
    # definição da imagem
    fig = plt.figure(figsize=(16, 10), dpi=72)
    # SQL query
    data_extracted_A = pd.read_sql('SELECT extraction_date, count( distinct productcode) as number_of_products  FROM por.{0} '
                                   'group by extraction_date '.format('raw_LojaA'), conn)
    data_extracted_B = pd.read_sql(
        'SELECT extraction_date, count( distinct productcode) as number_of_products  FROM por.{0} '
        'group by extraction_date '.format('raw_LojaB'), conn)
    # x
    month_extracted_list_A = data_extracted_A['extraction_date'].tolist()[1:]
    month_extracted_list_B = data_extracted_B['extraction_date'].tolist()[2:]
    # y
    number_products_extracted_list_A = data_extracted_A['number_of_products'].tolist()[1:]
    number_products_extracted_list_B = data_extracted_B['number_of_products'].tolist()[2:]
    # títulos
    x_title = "Dia de extração"
    y_title = "Nº de produtos (Normalizado)"
    fig_name ="produtos_por_dia"
    # figure
    plt.title(F"Nº de produtos por dia de extração", fontweight='bold', fontsize=25)
    plt.plot(month_extracted_list_A, number_products_extracted_list_A, '-s', label="Loja A")
    plt.plot(month_extracted_list_B, number_products_extracted_list_B, '-s', label="Loja B")
    plt.legend(fontsize=17)
    # usar o estilo seaborn (podes retirar se preferires simples ou experimentar o comando abaixo comentado)
    sns.set(style="darkgrid")  # default
    # algumas variáveis de controlo
    title_font_size = 22
    label_font_size = 17
    title_padding = 20
    # títulos dos eixos
    plt.xlabel(x_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    plt.ylabel(y_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    # tamanho dos números nos eixos
    plt.tick_params(axis='x', color='black', labelsize=label_font_size)
    plt.tick_params(axis='y', color='black', labelsize=label_font_size)
    plt.savefig(F"{fig_name}.png")
    plt.show()
```

- **ANEXO I – Código da análise Distribuição de Preços até 40€ com intervalos de 1€**

```
def prices_histo_040(show=True):
    conn = getbdconnection()
    # definição da imagem
    fig = plt.figure(figsize=(16, 10), dpi=72)
    # SQL query
    data_extracted_A = pd.read_sql('SELECT extraction_date, productcode, price  FROM por.{0}'.format('raw_LojaA'), conn)
    data_extracted_B = pd.read_sql(
        'SELECT extraction_date, productcode, price  FROM por.{0}'.format('raw_LojaB'), conn)
    # x
    max_price = 40.
    x_A = data_extracted_A[(data_extracted_A['extraction_date']=='2022-11-27') & (data_extracted_A['price'] < max_price)]['price'].tolist()
    x_B = data_extracted_B[(data_extracted_B['extraction_date']=='2022-11-27') & (data_extracted_B['price'] < max_price)]['price'].tolist()
    data = [x_A,x_B]
    # títulos
    x_title = "Preço (€)"
    y_title = "Nº de produtos (Normalizado)"
    fig_name ="distribuicao_precos_0100"
    bins = np.arange(0., max_price, step=1)
    # figure
    plt.title(F"Distribuição de Preços", fontweight='bold', fontsize=25)
    plt.hist(x_A, bins=bins, label="Loja A", density=True, alpha=1)
    plt.hist(x_B, bins=bins, label="Loja B", density=True, alpha=0.7)
    plt.legend(fontsize=17)
    # usar o estilo seaborn (podes retirar se preferires simples ou experimentar o comando abaixo comentado)
    sns.set(style="darkgrid")  # default
    # algumas variáveis de controlo
    title_font_size = 22
    label_font_size = 17
    title_padding = 20
    # títulos dos eixos
    plt.xlabel(x_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    plt.ylabel(y_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    # tamanho dos números nos eixos
    plt.tick_params(axis='x', color='black', labelsize=label_font_size)
    plt.tick_params(axis='y', color='black', labelsize=label_font_size)
    plt.savefig(F"{fig_name}.png")
    plt.show()
```

- **ANEXO J – Código da análise Distribuição de Preços por Loja**

```
def prices_boxplot_010(show=True):
    conn = getbdconnection()
    # definição da imagem
    fig = plt.figure(figsize=(16, 10), dpi=72)
    # SQL query
    data_extracted_A = pd.read_sql('SELECT extraction_date, productcode, price  FROM por.{0}'.format('raw_LojaA'), conn)
    data_extracted_B = pd.read_sql(
        'SELECT extraction_date, productcode, price  FROM por.{0}'.format('raw_LojaB'), conn)
    # x
    max_price = 10.
    x_A = data_extracted_A[(data_extracted_A['extraction_date']=='2022-11-27') & (data_extracted_A['price'] < max_price)]['price'].tolist()
    x_B = data_extracted_B[(data_extracted_B['extraction_date']=='2022-11-27') & (data_extracted_B['price'] < max_price)]['price'].tolist()
    data = [x_A,x_B]
    # títulos
    y_title = "Loja"
    x_title = "Preço (€)"
    fig_name ="distribuicao_precos_010_boxplot"
    bins = np.arange(0., max_price, step=1)
    # figure
    plt.title(F"Distribuição de Preços", fontweight='bold', fontsize=25)
    sns.boxplot(data=data, orient='h')
    # usar o estilo seaborn (podes retirar se preferires simples ou experimentar o comando abaixo comentado)
    sns.set(style="darkgrid")  # default
    # algumas variáveis de controlo
    title_font_size = 22
    label_font_size = 17
    title_padding = 20
    # títulos dos eixos
    plt.xlabel(x_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    plt.ylabel(y_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    plt.gca().set_yticklabels(["Loja A", "Loja B"])
    # tamanho dos números nos eixos
    plt.tick_params(axis='x', color='black', labelsize=label_font_size)
    plt.tick_params(axis='y', color='black', labelsize=label_font_size)
    plt.savefig(F"{fig_name}.png")
    plt.show()
```

- **ANEXO K – Código da análise Distribuição de Preços até 1€ com intervalos de 0,01€**

```
def prices_histo_005(show=True):
    conn = getbdconnection()

    # definição da imagem
    fig = plt.figure(figsize=(16, 10), dpi=72)

    # SQL query
    data_extracted_A = pd.read_sql('SELECT extraction_date, productcode, price  FROM por.{0}'.format('raw_LojaA'), conn)
    data_extracted_B = pd.read_sql(
        'SELECT extraction_date, productcode, price  FROM por.{0}'.format('raw_LojaB'), conn)

    # x
    max_price = 1.
    x_A = data_extracted_A[(data_extracted_A['extraction_date']=='2022-11-27') & (data_extracted_A['price'] < max_price)]['price'].tolist()
    x_B = data_extracted_B[(data_extracted_B['extraction_date']=='2022-11-27') & (data_extracted_B['price'] < max_price)]['price'].tolist()

    data = [x_A,x_B]

    # títulos
    x_title = "Preço (€)"
    y_title = "Nº de produtos (Normalizado)"
    fig_name ="distribuicao_precos_005"

    bins = np.arange(0.005, max_price+0.005, step=0.01)

    # figure
    plt.title(F"Distribuição de Preços", fontweight='bold', fontsize=25)
    plt.hist(x_A, bins=bins, label="Loja A", density=True, alpha=1, log=True)
    plt.hist(x_B, bins=bins, label="Loja B", density=True, alpha=0.7, log=True)
    plt.legend(fontsize=17)

    # usar o estilo seaborn (podes retirar se preferires simples ou experimentar o comando abaixo comentado)
    sns.set(style="darkgrid")  # default

    # algumas variáveis de controlo
    title_font_size = 22
    label_font_size = 17
    title_padding = 20

    # títulos dos eixos
    plt.xlabel(x_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    plt.ylabel(y_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)

    # tamanho dos números nos eixos
    plt.tick_params(axis='x', color='black', labelsize=label_font_size)
    plt.tick_params(axis='y', color='black', labelsize=label_font_size)
    plt.savefig(F"{fig_name}.png")
    plt.show()
```

- **ANEXO L – Código da análise Distribuição de Preços até 10€ com intervalos de 0,1€**

```
def prices_histo_010(show=True):
    conn = getbdconnection()

    # definição da imagem
    fig = plt.figure(figsize=(16, 10), dpi=72)

    # SQL query
    data_extracted_A = pd.read_sql('SELECT extraction_date, productcode, price  FROM por.{0}'.format('raw_LojaA'), conn)
    data_extracted_B = pd.read_sql(
        'SELECT extraction_date, productcode, price  FROM por.{0}'.format('raw_LojaB'), conn)

    # x
    max_price = 10.
    x_A = data_extracted_A[(data_extracted_A['extraction_date']=='2022-11-27') & (data_extracted_A['price'] < max_price)]['price'].tolist()
    x_B = data_extracted_B[(data_extracted_B['extraction_date']=='2022-11-27') & (data_extracted_B['price'] < max_price)]['price'].tolist()

    data = [x_A,x_B]

    # títulos
    x_title = "Preço (€)"
    y_title = "Nº de produtos (Normalizado)"
    fig_name ="distribuicao_precos_010"

    bins = np.arange(0., max_price, step=0.1)

    # figure
    plt.title(F"Distribuição de Preços", fontweight='bold', fontsize=25)
    plt.hist(x_A, bins=bins, label="Loja A", density=True, alpha=1)
    plt.hist(x_B, bins=bins, label="Loja B", density=True, alpha=0.7)
    plt.legend(fontsize=17)

    # usar o estilo seaborn (podes retirar se preferires simples ou experimentar o comando abaixo comentado)
    sns.set(style="darkgrid")  # default

    # algumas variáveis de controlo
    title_font_size = 22
    label_font_size = 17
    title_padding = 20

    # títulos dos eixos
    plt.xlabel(x_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    plt.ylabel(y_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)

    # tamanho dos números nos eixos
    plt.tick_params(axis='x', color='black', labelsize=label_font_size)
    plt.tick_params(axis='y', color='black', labelsize=label_font_size)
    plt.savefig(F"{fig_name}.png")
    plt.show()
```

- **ANEXO M – Código da análise Distribuição de Promoções por Preço Original**

```
def promos_histo_010(show=True):
    conn = getbdconnection()

    # definição da imagem
    fig = plt.figure(figsize=(16, 10), dpi=72)

    # SQL query
    data_extracted_A = pd.read_sql('SELECT extraction_date, productcode, price, pricewithoutpromo  FROM por.{0}'.format('raw_LojaA'), conn)
    data_extracted_B = pd.read_sql(
        'SELECT extraction_date, productcode, price, pricewithoutpromo  FROM por.{0}'.format('raw_LojaB'), conn)

    data_extracted_A = data_extracted_A[~data_extracted_A.loc[:, "pricewithoutpromo"].isna()]
    data_extracted_B = data_extracted_B[~data_extracted_B.loc[:, "pricewithoutpromo"].isna()]

    # x
    max_price = 10.
    x_A = data_extracted_A[(data_extracted_A['extraction_date']=='2022-11-27') & (data_extracted_A['pricewithoutpromo'] < max_price)]['pricewithoutpromo'].tolist()
    x_B = data_extracted_B[(data_extracted_B['extraction_date']=='2022-11-27') & (data_extracted_B['pricewithoutpromo'] < max_price)]['pricewithoutpromo'].tolist()

    data = [x_A,x_B]

    # títulos
    y_title = "Nº de produtos (Normalizado)"
    x_title = "Preço (€)"
    fig_name ="distribuicao_promos_010"

    bins = np.arange(0., max_price, step=0.10)

    # figure
    plt.title(F"Distribuição de Promoções por Preço Original", fontweight='bold', fontsize=25)
    plt.hist(x_A, bins=bins, label="Loja A", density=True, alpha=1)
    plt.hist(x_B, bins=bins, label="Loja B", density=True, alpha=0.7)
    plt.legend(fontsize=17)

    # usar o estilo seaborn (podes retirar se preferires simples ou experimentar o comando abaixo comentado)
    sns.set(style="darkgrid")  # default

    # algumas variáveis de controlo
    title_font_size = 22
    label_font_size = 17
    title_padding = 20

    # títulos dos eixos
    plt.xlabel(x_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    plt.ylabel(y_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)

    # tamanho dos números nos eixos
    plt.tick_params(axis='x', color='black', labelsize=label_font_size)
    plt.tick_params(axis='y', color='black', labelsize=label_font_size)
    plt.savefig(F"{fig_name}.png")
    plt.show()
```

- **ANEXO N – Código da análise Distribuição de Promoções Percentuais**

```
def promos_percent_histo_010(show=True):
    conn = getbdconnection()

    # definição da imagem
    fig = plt.figure(figsize=(16, 10), dpi=72)

    # SQL query
    data_extracted_A = pd.read_sql('SELECT extraction_date, productcode, price, pricewithoutpromo  FROM por.{0}'.format('raw_LojaA'), conn)
    data_extracted_B = pd.read_sql(
        'SELECT extraction_date, productcode, price, pricewithoutpromo  FROM por.{0}'.format('raw_LojaB'), conn)

    data_extracted_A = data_extracted_A[~data_extracted_A.loc[:, "pricewithoutpromo"].isna()]
    data_extracted_B = data_extracted_B[~data_extracted_B.loc[:, "pricewithoutpromo"].isna()]

    data_extracted_A.loc[:, "promo_percent"] = (data_extracted_A["pricewithoutpromo"]-data_extracted_A["price"])/data_extracted_A["pricewithoutpromo"]*100.
    data_extracted_B.loc[:, "promo_percent"] = (data_extracted_B["pricewithoutpromo"] - data_extracted_B["price"])/data_extracted_B["pricewithoutpromo"]*100.

    # x
    max_price = 10.
    x_A = data_extracted_A[(data_extracted_A['extraction_date']=='2022-11-27') & (data_extracted_A['pricewithoutpromo'] < max_price)]['promo_percent'].tolist()
    x_B = data_extracted_B[(data_extracted_B['extraction_date']=='2022-11-27') & (data_extracted_B['pricewithoutpromo'] < max_price)]['promo_percent'].tolist()

    data = [x_A,x_B]

    # títulos
    y_title = "Nº de produtos (Normalizado)"
    x_title = "Promoção (%)"
    fig_name ="distribuicao_promos_percent_010"

    bins = np.arange(0., 100, step=1)

    # figure
    plt.title(F"Distribuição de Promoções Percentuais", fontweight='bold', fontsize=25)
    histo_A = plt.hist(x_A, bins=bins, label="Loja A", density=True, alpha=1, log=False)
    histo_B = plt.hist(x_B, bins=bins, label="Loja B", density=True, alpha=0.7, log=False)
    plt.legend(fontsize=17)

    # usar o estilo seaborn (podes retirar se preferires simples ou experimentar o comando abaixo comentado)
    sns.set(style="darkgrid")  # default

    # algumas variáveis de controlo
    title_font_size = 22
    label_font_size = 17
    title_padding = 20

    # títulos dos eixos
    plt.xlabel(x_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)
    plt.ylabel(y_title, fontweight='bold', color='black', fontsize=title_font_size, horizontalalignment='center',
               labelpad=title_padding)

    # tamanho dos números nos eixos
    plt.tick_params(axis='x', color='black', labelsize=label_font_size)
    plt.tick_params(axis='y', color='black', labelsize=label_font_size)
    plt.xticks(np.arange(0., 100., 5.))

    for bin in bins:
        if int(bin)%5 == 0 and bin < 70:
            value_y = max(histo_A[0][int(bin)], histo_B[0][int(bin)])
            plt.text(bin, 0.005 + value_y, "{0:2.0f}%".format(bin), rotation=5, fontweight='bold', fontsize=18)

    plt.savefig(F"{fig_name}.png")
    plt.show()
```