

Faculdade

XPe



RELATÓRIO

PROJETO
APLICADO

PÓS-GRADUAÇÃO

XP Educação
Relatório do Projeto Aplicado

**Banco de dados para cálculo de
inflação de produtos**

Victor Augusto Pereira Burgardt

Orientador(a): Ítalo Lucena

06/12/2023



Victor Augusto Pereira Burgardt

XP EDUCAÇÃO

RELATÓRIO DO PROJETO APLICADO

Banco de dados para cálculo de inflação de produtos

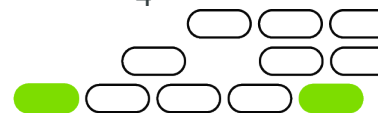
Relatório de Projeto Aplicado
desenvolvido para fins de conclusão do
curso.

Orientador (a): Ítalo Lucena

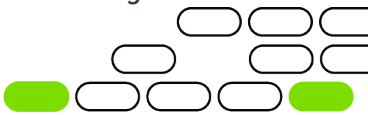


Sumário

1. CANVAS do Projeto Aplicado	4
Desafio	5
1.1.1 Análise de Contexto	5
1.1.2 Personas	6
1.1.3 Benefícios e Justificativas	7
1.1.4 Hipóteses	8
1.2 Solução	9
1.2.1 Objetivo SMART	9
1.2.2 Premissas e Restrições	11
1.2.3 Backlog de Produto	13
2. Área de Experimentação	14
2.1 Sprint 1	16
2.1.1 Solução	16
Evidência do planejamento:	16
Evidência da execução de cada requisito:	16
Evidência dos resultados:	16
2.1.2 Lições Aprendidas	16
2.2 Sprint 2	17
2.2.1 Solução	17
Evidência do planejamento:	17
Evidência da execução de cada requisito:	17
Evidência dos resultados:	17
2.2.2 Lições Aprendidas	17
2.3 Sprint 3	18
2.3.1 Solução	18
Evidência do planejamento:	18
Evidência da execução de cada requisito:	18



Evidência dos resultados:	18
2.3.2 Lições Aprendidas	18
3. Considerações Finais	19
3.1 Resultados	19
3.2 Contribuições	19
3.3 Próximos passos	19



1. CANVAS do Projeto Aplicado

Figura conceitual, que representa todas as etapas do Projeto Aplicado.



1.1 Desafio

1.1.1 Análise de Contexto

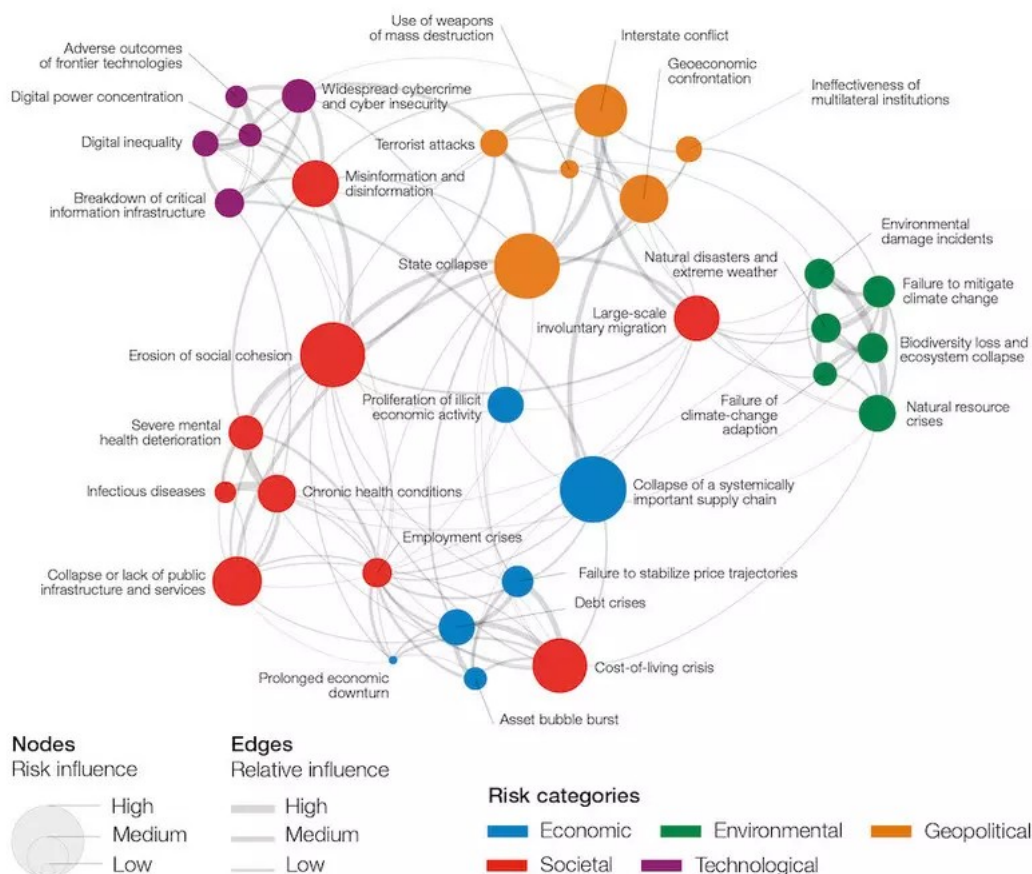
Nos últimos anos, temos vivenciado um período considerável de instabilidade causada por diferentes crises que vão de origens climáticas, humanitárias, tensões políticas, sanitárias, econômicas até pandemias e guerras. Atualmente, pessoas têm até utilizado o termo "poli-crisis" para descrever o período atual, que é denotado como um período em que ocorrem múltiplos eventos catastróficos de diversas naturezas em um certo período. Entre as diversas crises mundiais atuais as que mais estão impactando nosso mundo e estão desestabilizando o panorama internacional, podemos citar as seguintes:

- **Pandemia do Corona Vírus:** A pandemia global do COVID-19 trouxe diversos desafios para a saúde pública e para a economia dos países, mudando completamente a rotina das pessoas e empresas.
- **Crise das redes sociais:** Com a proliferação das notícias falsas e desinformações nas redes sociais hoje temos muita dificuldade de saber o que realmente é verdade na internet, fator foi grande causador de tensões políticas.
- **Guerra da Rússia e Ucrânia:** Os confrontos na Ucrânia e Rússia impactam não só a região europeia mas também países que faziam comércio com eles, como Brasil que comprava fertilizantes.
- **Guerra de Israel e Palestina:** Recentemente esse conflito está gerando diversos protestos em todos países e está aumentando a tensão na região do Oriente Médio.
- **Crises climáticas:** Mudanças climáticas têm gerado diversos impactos globais como má colheitas, incêndios, inundações, etc.
- **Crise de imigração:** Fluxos migratórios massivos têm causado uma grande preocupação humanitária e econômica para países que estão lidando com o desafio.

Para compreender melhor essas crises e o cenário atual, abaixo podemos observar a figura 1 da *World Economic Forum*, que fez uma boa representação de como está o cenário atual das crises e riscos que estamos enfrentando e como elas estão ligadas entre si.



Global risks landscape: an interconnections map



Source: World Economic Forum, Global Risks Perception Survey 2022-2023

Figura 1: Global risks landscape: an interconnections map

Essas crises estão interligadas, não possuem fronteiras e causam impacto duradouro na vida das pessoas. A globalização e a tecnologia desempenham um papel importante na transformação de crises locais em crises globais e também na amplificação de seus efeitos de pequenos para uma escala global. O Brasil não é imune a crises e atualmente tem passado por diversas, talvez a maior recentemente é a tensão política atual, que afeta diversos aspectos do país.

As crises trouxeram consigo uma questão que tem ganhado bastante destaque mundialmente, que é a inflação crescente de produtos e alimentos, a qual representa um desafio significativo para a sociedade como um todo. Esta questão se mostra particularmente relevante em um país como o Brasil, que possui uma grande

parcela de sua população em situação de vulnerabilidade e cuja economia é fortemente dependente do agronegócio. À medida que a economia oscila, a preocupação com a inflação de alimentos tem se tornado cada vez mais importante, passando de apenas uma preocupação doméstica para a uma reflexão sobre nossa economia, demandando uma análise aprofundada.

Para melhor entender esse cenário é necessário uma base de dados que centralize informações atualizadas e históricas relevantes sobre a flutuação de preços de itens dentro do mercado interno do país, para assim gerar indicadores econômicos relevantes para futuras análises da economia do país. Entretanto, hoje no Brasil cientistas de dados e jornalistas têm muita dificuldade de gerar boas análises pela falta de dados históricos, atualizados e corretos tornando todo esse processo muito devagar, difícil e desencorajador

Dado esse cenário, a empresa X, em conjunto com o governo brasileiro, tem apresentado diversas propostas para combater essa situação por meio de sistemas de informação e processos de coleta de dados de informações de produtos de mercado. Entretanto, até o momento, nenhuma delas retornou informações com um alto nível de granularidade e atualização em tempo real, em vez disso, a coleta de dados ainda é realizada de forma periódica em grandes intervalos e em alguns casos até manualmente.

O objetivo central deste artigo é desenvolver um sistema que automatize a extração de dados de maneira programada, permitindo-nos coletar informações de uma ampla variedade de produtos e alimentos anunciados em sites de supermercados em todo o território nacional. Essa abordagem visa centralizar as informações em uma única base de dados focada em análises históricas. Dessa forma, estaríamos garantindo o acesso a uma fonte de dados rica, precisa e atualizada, proporcionando assim, um ambiente que facilite a construção de indicadores de inflação mais precisos gerando assim, relatórios de análise econômica mais significativos a respeito da nossa atual situação econômica e até possibilitando a construção de novos projetos para empresa.

Para melhor contextualizar o problema foi utilizado a ferramenta matriz CSD (Certezas, Suposições e Dúvidas) que é uma ferramenta que auxilia contextualização do problema através de uma tabela com diferentes percepções a respeito da situação abordada.





Figura 2: Matriz CSD

Outra ferramenta utilizada para auxiliar no entendimento do contexto do problema foi a matriz POEMS (People, Objects, Environments, Messages and Services) que ajuda a identificar oportunidades de melhoria através de tabela com cinco elementos a serem analisados.

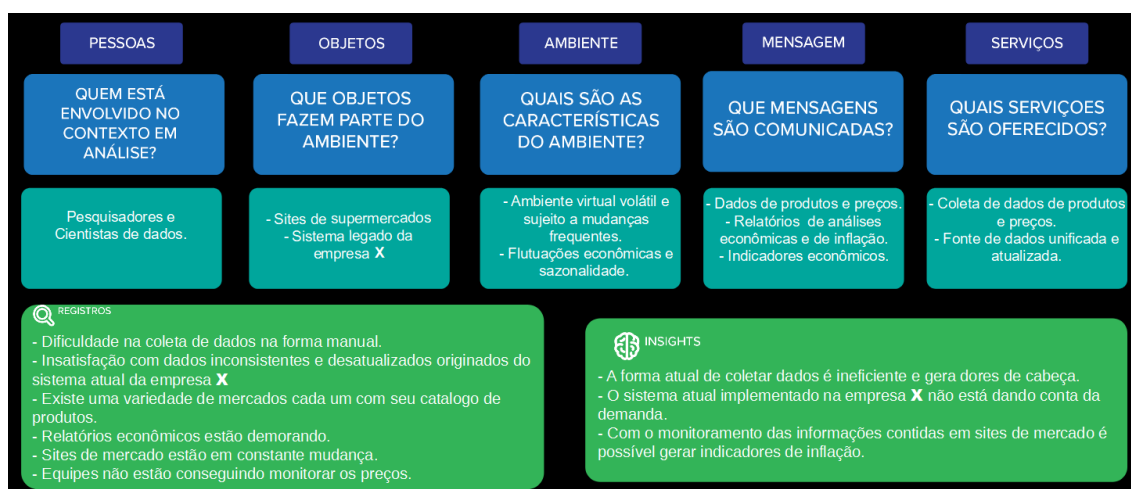


Figura 3: Matriz POEMS

1.1.2 Personas

Para melhor exploração do contexto apresentado será feita a descrição de duas Personas baseadas em pessoas reais envolvidas diretamente com o problema exposto na seção de análise de contexto. Nesta análise, detalharemos uma série de aspectos extraídos durante entrevistas feitas com pessoas que trabalham na empresa X para assim detalhar o máximo possível suas características, comportamento, sentimentos, dores entre outros. Assim aumentando nosso conhecimento sobre a situação atual e seus agentes.

Persona 1 - Andrei Bandeira, Cientista de dados

Andrei Bandeira é um profissional experiente no campo da ciência de dados e Inteligência Artificial. Ele tem 40 anos, é casado e também é pai de dois filhos pequenos, ele dedica bastante tempo à sua carreira, mas também ao bem-estar de sua família. Ele trabalha na empresa X há 8 anos e atualmente está no cargo de cientista de dados sênior do time de Inteligência artificial, possui ensino superior completo e é mestre em ciência da computação. Além disso, detém um sólido conhecimento em ferramentas de *Business Intelligence* e auxilia diversos times a utilizá-las. Andrei atualmente tem o objetivo de expandir a equipe de inteligência artificial e seus projetos, ele reconhece a importância da construção de novas bases de dados e extração de dados com alto potencial exploratório, particularmente na coleta de dados econômicos do país e tem grande paixão por utilizar modelos de aprendizagem de máquina em seus projetos, mas percebe que atualmente algumas bases de dados não estão tão ricas para esses tipos de projeto. Andrei acredita que a atual base de dados de preços de mercado é insuficiente para seus projetos e lhe dá muita dor de cabeça pela falta de informações e pela desestruturação dessas informações dentro da empresa. Ele tem uma meta clara que é reunir informações econômicas robustas e utilizá-las para desenvolver produtos e relatórios de excelente qualidade, impulsionando ainda mais a inovação e a eficácia de sua equipe.

Persona 2 - Daniel Ribeiro, Professor de Estatística

Daniel Ribeiro é um professor de estatística, com 57 anos de idade e com uma carreira acadêmica rica em experiência. Daniel faz pesquisas acadêmicas há mais de 30 anos e possui participação em diversos projetos governamentais, atualmente ele faz diversas pesquisas com foco na situação econômica do país, sempre gerando relatórios para aumentar a transparência da situação econômica no país como para o bem-estar público. Atualmente ocupa o cargo de professor-assistente na universidade federal de pernambuco no centro de informação. Além disso, exerce o papel de professor pesquisador na empresa X, na qual ele possui acesso a diversas bases de dados da empresa para fazer seus relatórios econômicos. Além disso, ele também realiza reuniões periódicas com diferentes times dentro da empresa para dar conselhos e direcionamentos em projetos. Ele possui um bom background em computação, entretanto possui maior familiaridade com ferramentas voltadas para projetos estáticos com a linguagem R. Como cidadão e professor de



estatística ele tem grandes preocupações com a economia do país e acompanha atentamente as notícias e tendências econômicas. O desejo de Daniel é que a empresa amplie seus horizontes e colete mais dados em diferentes fontes de dados para aumentar o potencial da empresa, gerando um ambiente rico e diversificado de dados. Dessa forma, os pesquisadores conseguiram monitorar aspectos financeiros do país, gerando assim melhores documentos e *insights* mais precisos, o que no momento atual não está conseguindo fazer por conta da falta de qualidade de dados coletados. Um dos principais projetos que o mesmo tem acompanhado é o coletor de preços de mercado, o qual tem para si um grande peso, pois além de professor pesquisador, Daniel é um pai e um cliente habitual de diversos mercados e deseja como todo cidadão brasileiro saber se os produtos que gosta estão ficando mais caros ou mais baratos.

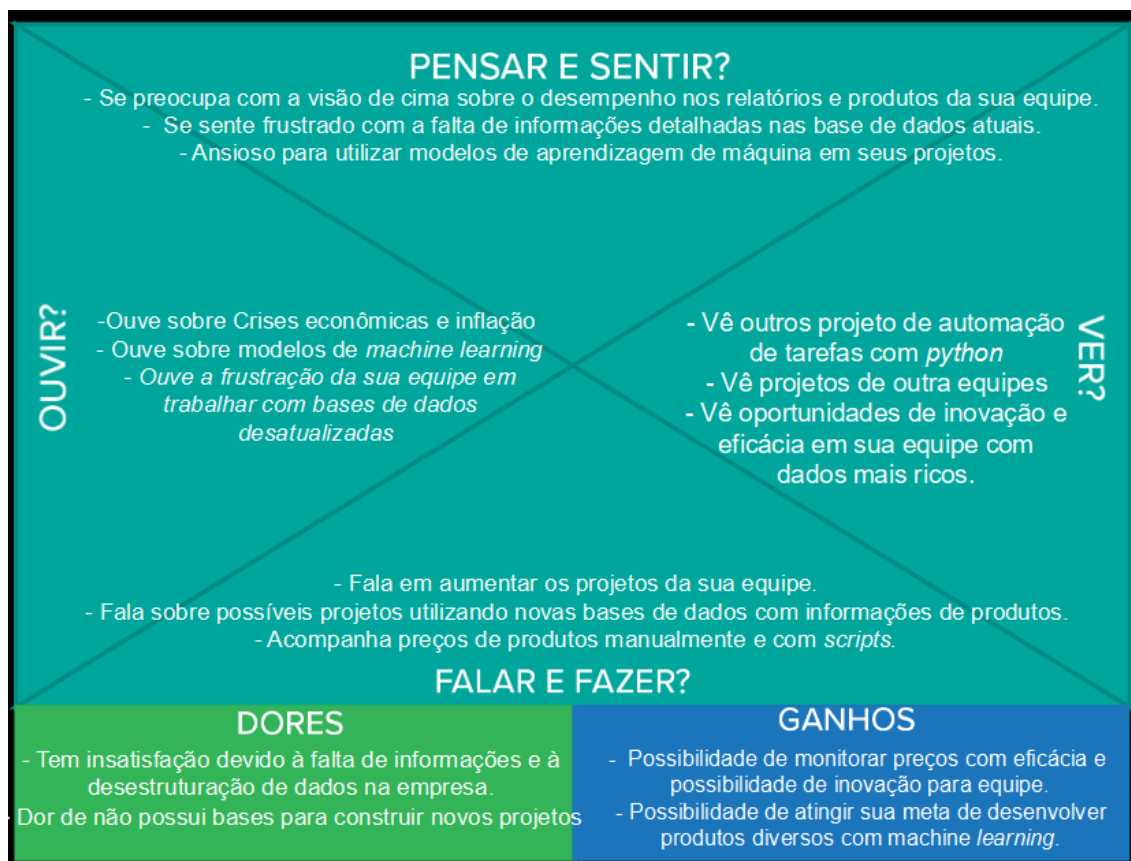


Figura 4: Mapa empatia Andrei Bandeira (Persona 1)

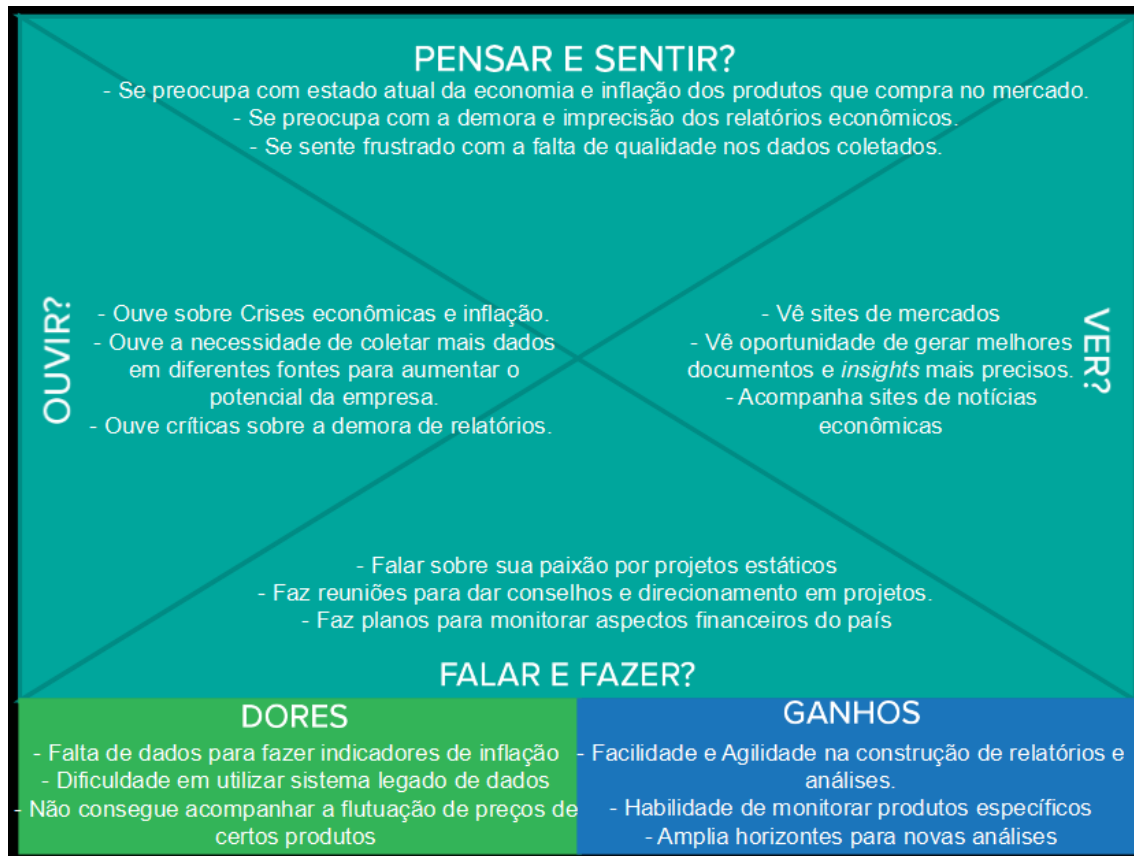
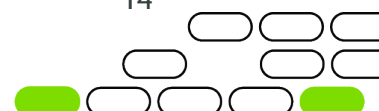


Figura 5: Mapa empatia Daniel Ribeiro (Persona 2)

1.1.3 Benefícios e Justificativas

Nesta seção, serão explorados os benefícios e justificativas que fazem do projeto de coleta de dados de preços de produtos de supermercados uma iniciativa fundamental para a empresa, expandindo as bases de dados sobre a economia brasileira. Será exposto os impactos positivos da adoção desse projeto e as motivações dos agentes por trás da construção desse sistema. Para realizar esta análise, foi utilizado duas técnicas para facilitar a compreensão dos dados obtidos. A primeira é o *Blueprint*, uma ferramenta que permite analisar a rotina de agentes com o objetivo de encontrar possíveis *insights* para construir uma solução mais assertiva. A segunda técnica utilizada foi o Canvas de Proposta de Valor, que faz uma representação visual para facilitar o entendimento da interação das dores e ações do público envolvido com as ações e vantagens da solução proposta. As duas técnicas aplicadas no projeto podem ser observadas abaixo respectivamente na tabela 1 e na figura 6.

Ações do Cliente	<ul style="list-style-type: none"> • Realizar coleta de dados disponíveis em bases incompletas. • Faz coleta manual de dados em sites de mercado quando se deseja saber sobre dados faltantes.
Objetivos do Cliente	<ul style="list-style-type: none"> • Fazer indicadores de inflação em diversos produtos. • Desenvolver softwares em cima de uma base de dados completa de informações de produtos/alimentos. • Produzir relatórios sobre estado atual da economia do país
Atividades do Cliente	<ul style="list-style-type: none"> • Procurar dentro de diversos bancos de dados as informações necessárias para análise. • Construção de <i>scripts</i> para extrair dados de forma pontual. • Consultas nos bancos de dados para extração de informação relevante. <ul style="list-style-type: none"> • Pré Processamento de dados extraídos
Questões do Cliente	<ul style="list-style-type: none"> • Os dados disponíveis nas bases de dados da empresa X estão atualizados? • Quando foi a última vez que verificaram os dados da base da empresa X? • Existe uma forma de unificar esses dados em algum lugar? • Os dados disponíveis na empresa X estão corretos?
Barreiras do Cliente	<ul style="list-style-type: none"> • Falta de informação nas bases da empresa X



	<ul style="list-style-type: none"> • Lentidão na hora de fazer coleta manual de dados • Dificuldade na construção de <i>scripts</i> de extração de dados e <i>scripts</i> de pré-processamento.
Saída desejável da solução	<ul style="list-style-type: none"> • Construção de uma base de dados que centraliza as informações sobre produtos e alimentos para construção de indicadores de inflação.
Funcionalidades da solução	<ul style="list-style-type: none"> • Coleta periodicamente informações sobre produtos dentro de diversos sites de supermercado. • Faz o pré-processamento dos dados extraídos. • Centraliza as informações obtidas em tabelas de fácil acesso.
Interação com a solução	<ul style="list-style-type: none"> • Os usuários interagem com a solução principalmente consultando as tabelas do banco de dados para acessar informações relevantes.
Mensagem solução	<ul style="list-style-type: none"> • Informações coletadas de produtos dentro sites de mercados, proporcionando dados confiáveis para análises econômicas.
Onde ocorre a solução	<ul style="list-style-type: none"> • A solução é executada em um ambiente de nuvem, especificamente na plataforma AWS
Tarefas aparentes da solução	<ul style="list-style-type: none"> • Usuários podem acessar um banco de dados RDS na nuvem.
Tarefas escondidas da solução	<ul style="list-style-type: none"> • Coletores <i>web-scraping</i> rodando periodicamente extraindo informações dos sites

Tabela 1: Blue print

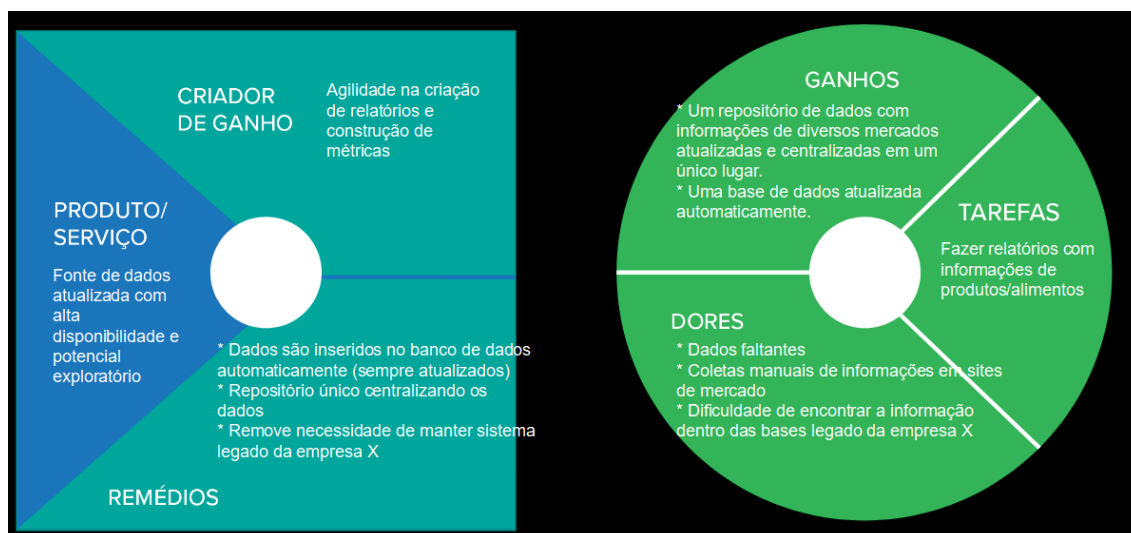


Figura 6: Canvas de proposta de valor

Em suma pode-se notar que o projeto traz diversos impactos para a empresa sendo o principal a criação de uma base de dados atualizada e correta com grande potencial de exploração. Isso não só atende às demandas imediatas, mas também proporciona um ambiente para criação de novos projetos para as partes interessadas no sistema. Além disso, o novo sistema pode substituir sistemas velhos e bases de dados legado dentro da empresa.

1.1.4 Hipóteses

Para melhor entendimento da solução proposta, nesta seção introduziremos uma série de hipóteses analisadas durante o processo de entendimento do contexto do problema, que foram fundamentais para o direcionamento e formação do sistema proposto. Para auxiliar a documentação de análise de hipóteses, utilizamos duas técnicas: Matriz de observações e hipóteses e tabela de priorização de ideias. Podemos observar as duas aplicadas ao nosso contexto nas tabelas abaixo, respectivamente.

Observações	Hipóteses
Dados sobre produtos e alimentos no Brasil podem ser acompanhados em sites de supermercados	É possível extrair essas informações através de <i>web-scrapers</i> .
Alimentos e produtos em mercado estão em constante mudança	É possível extrair dados diretamente dos sites para monitorar a flutuação nos valores
Sites de mercado estão em constante atualização	É possível mapear a estrutura de um site para descobrir a melhor forma de consumir seus dados para acompanhar mudanças
A Solução atual apresenta muitos dados incorretos e faltantes	É possível criar rotinas garantindo o funcionamento contínuo e pré-processamento efetivo
É necessário uma forma de consultar os dados sem a necessidade de pré-processamento extensivo e <i>scripts</i> .	Podemos criar um banco de dados estruturado para facilitar o consumo das informações e a organização da informação
Alguns sites podem utilizar tecnologias de proteção contra web-scraping.	É possível contornar esse problema através de algumas ferramentas.
Cada site de mercado possui sua variedade de produtos e preços	É possível centralizar essas informações em um único banco de dados
A coleta de dados manual é lenta, cara e produz dados duvidosos	A automação da extração de dados pode economizar tempo e recursos em comparação com a coleta manual.

Tabela 2: Matriz de observações para hipóteses



Ideias:

- Banco de dados estruturado com preços de produtos de supermercados.
- Data Lake com dados de supermercados.
- Parceria com Supermercados para Coleta de Dados em Tempo Real.
- Continuar sistema atual da empresa X.

Ideia	B	A	S	I	C	O	Somatório
Banco de dados estruturado com preços de produtos de mercado	5	5	5	4	4	3	26
Data Lake com dados de mercado	5	5	4	4	3	1	23
Parceria com Supermercados para Coleta de Dados em Tempo Real	5	4	5	4	3	2	23
Continuar sistema atual da empresa X	4	3	3	5	2	1	18

Tabela 3: Priorização de ideias graduação BASICO.

1.2 Solução

1.2.1 Objetivo SMART

Após exposto o detalhamento completo do contexto do projeto, nesta etapa, apresentaremos os objetivos do projeto. Para isso, utilizamos a metodologia SMART em que são definidos objetivos respeitando cinco critérios: Específico (*Specific*), Mensurável (*Measurable*), Atingível (*Attainable*), Relevante (*Relevant*) e Temporal (*Time-based*). Dessa forma, os objetivos do projeto são os seguintes:

Construir um banco de dados estruturado com informações sobre a variedade de produtos e seus preços em vários supermercados em todo o país. Tudo deve ser feito, através de um pipeline de dados formado de coletores de dados (*web-scrapers*) que serão acionados diariamente para percorrer os sites fazendo a extração da informação do site. Logo após, os dados serão tratados e armazenados em um banco de dados para usuários acessarem e consumirem. Dessa forma, será possível criar diversos produtos e relatórios a partir desse banco de dados, além de fornecer aos pesquisadores uma forma de monitorar os preços de produtos dentro do país.

Lista de Objetivos:

- O banco de dados deve estar em um ambiente com alta disponibilidade (nuvem).
- O banco deve ser projetado para facilitar as pesquisas sobre preços de itens.
- O banco de dados deve sempre ter os preços atuais dos produtos.
- Os dados devem ser extraídos de fontes confiáveis.
- Os coletores de dados devem percorrer os sites diariamente.
- O banco de dados deve conter informações sobre os supermercados explorados.
- Os dados coletados devem ser armazenados de forma segura.
- O projeto deve ser concluído até o fim de novembro de 2023.
- Os dados extraídos devem ser processados pelos coletores para depois repassar para o banco.
- O sistema deve incluir mecanismos de notificação de problemas ou falhas na coleta de dados.

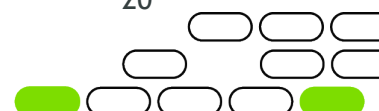


1.2.2 Premissas e Restrições

Para melhor curso no desenvolvimento de projetos é recomendo fazer uma análise de premissas descobertas e restrições associadas à construção de um sistema. As premissas são fatos que orientam nossas ações, já as restrições são delimitadores do escopo da solução. Nesta seção, detalharemos essas premissas e restrições através de uma matriz de riscos, que descreve os riscos identificados, impactos possíveis, ações para remediá-los e evitá-los. Nossa matriz de risco pode ser observada na tabela.

Risco identificado	Impacto potencial	Ações preventivas	Ações corretivas
Sites com bloqueio de <i>web-scrapers</i>	Alto	Fazer mapeamento de estrutura de sites extensivo.	-Trocar sites -Utilizar outras técnicas de <i>scrapping</i>
Estrutura de site mudou durante o projeto	Alto	Monitorar constantemente os coletores	Atualizar coletor de dados
Sites apresentando instabilidades	Alto	-Não depender de um único site -Adicionar fluxo quando o site está fora do ar	-Trocar sites. -Executar o coletor quando o site voltar ao ar.
Agendador de atividades não funcionando	Médio	-Fazer validação do agendador de tarefas. -Implementação de monitoração.	-Executar o coletor manualmente -Resolver a causa do problema
Coletores extraindo dados inconsistentes para o RDS.	Médio	-Validar coleta de dados antes de armazenar-los. -Implementar verificação de dados.	-Remover dados inconsistentes do banco e atualizar coletor.
Custos do projeto excedendo o orçamento	Alto	-Realizar análise de custos do projeto. -Verificar periodicamente o custo da solução.	-Identificar custos de cada parte do sistema e cortar partes não essenciais. -Utilizar outros produtos nuvem ou trocar de infraestrutura.

Tabela 4: Matriz de riscos



1.2.3 Backlog de Produto

Após feita a delimitação da solução, definindo seu escopo, objetivo e documentado suas características, detalharemos nesta seção os passos para a desenvolvimento da solução proposta, através da construção de um *backlog* de atividades, detalhando cada atividade e qual *sprint* de desenvolvimento ela pertence. Para facilitar o acompanhamento do *backlog* de atividade, será utilizado o Trello. Essa ferramenta permitirá o rastreo e o gerenciamento das atividades, nos garantindo que todas sejam executadas no momento certo do desenvolvimento. Abaixo podemos observar uma tabela com as *sprints* e um *snapshot* do painel Trello que foi utilizado durante o desenvolvimento do projeto, no qual as atividades listadas foram divididas em 3 *sprints*.

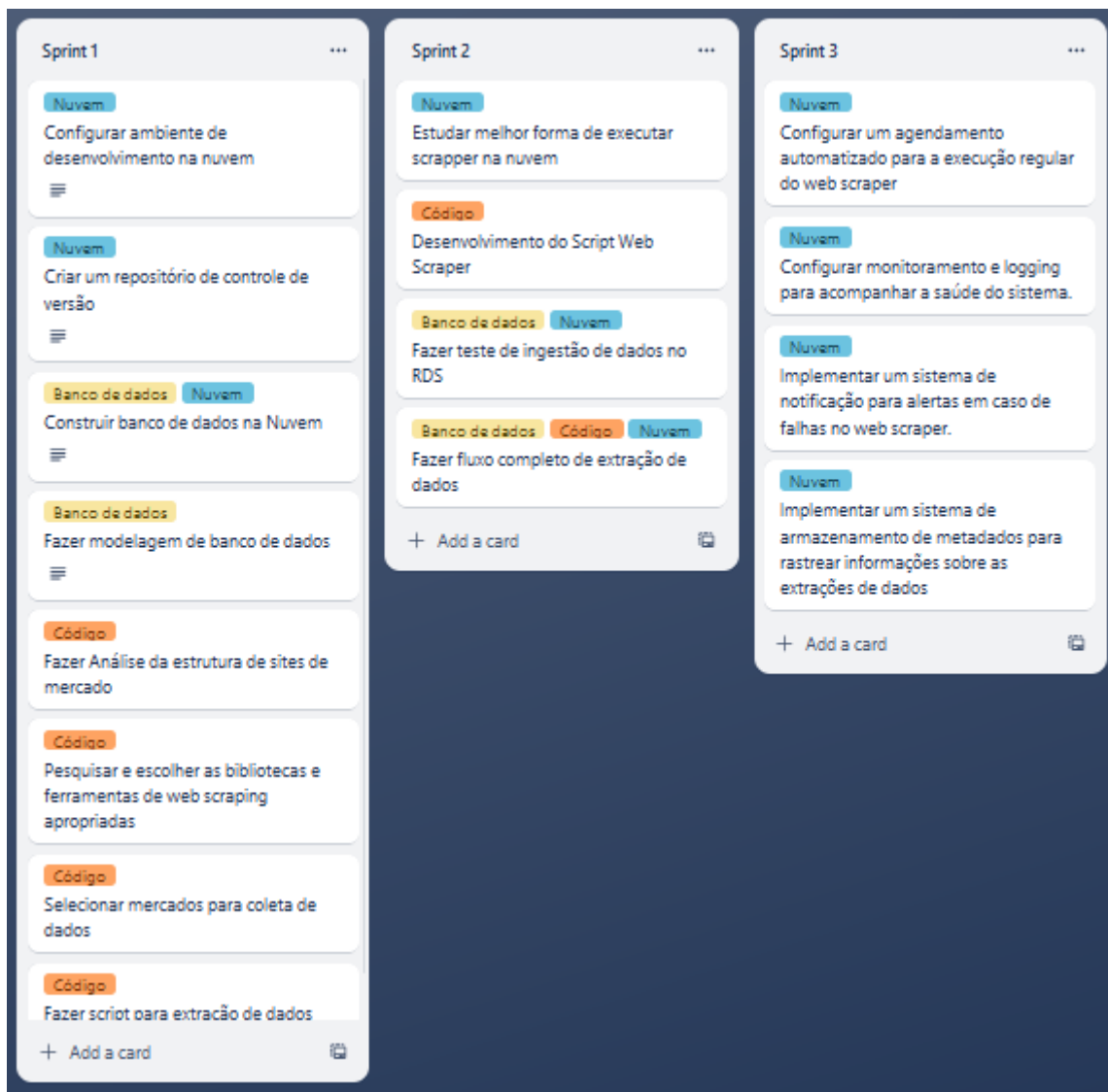


Figura 7: Trello Dashboard

Sprint 1
Configurar ambiente de desenvolvimento na nuvem
Criar um repositório de controle de versão
Fazer modelagem de banco de dados
Criar instância de banco de dados relacional na Nuvem
Fazer Análise da estrutura de sites de mercado
Selecionar mercados para coletar dados
Pesquisar e escolher as bibliotecas e ferramentas de <i>web-scraping</i> apropriadas
Fazer script <i>proof of concept</i> para extração de dados (<i>jupyter notebook</i>)
Criar tabelas no banco de dados
Sprint 2
Estudar melhor forma de botar <i>scraper</i> na nuvem
Desenvolvimento do <i>Script Web Scraper</i>
Fazer teste de ingestão de dados no RDS
Fazer fluxo completo de extração de dados
Sprint 3
Configurar um agendamento automatizado para a execução regular do web scraper
<i>Configurar monitoramento e logging para acompanhar a saúde do sistema.</i>
Implementar um sistema de notificação para alertas em caso de falhas no web scraper.
Implementar um sistema de armazenamento de metadados para rastrear informações sobre as extrações de dados

Tabela 5: Backlog de atividades



2. Área de Experimentação

Nesta seção, apresentaremos uma descrição detalhada das três sprints propostas para o desenvolvimento da solução. Abordaremos o planejamento de cada sprint, apresentando evidências do planejamento, do desenvolvimento, das dificuldades enfrentadas e dos resultados alcançados em relação a cada requisito da sprint. Essa documentação nos permite avaliar se a solução proposta está caminhando na direção inicial da solução proposta ou se estão surgindo desvios que possam exigir a implementação de uma solução alternativa.



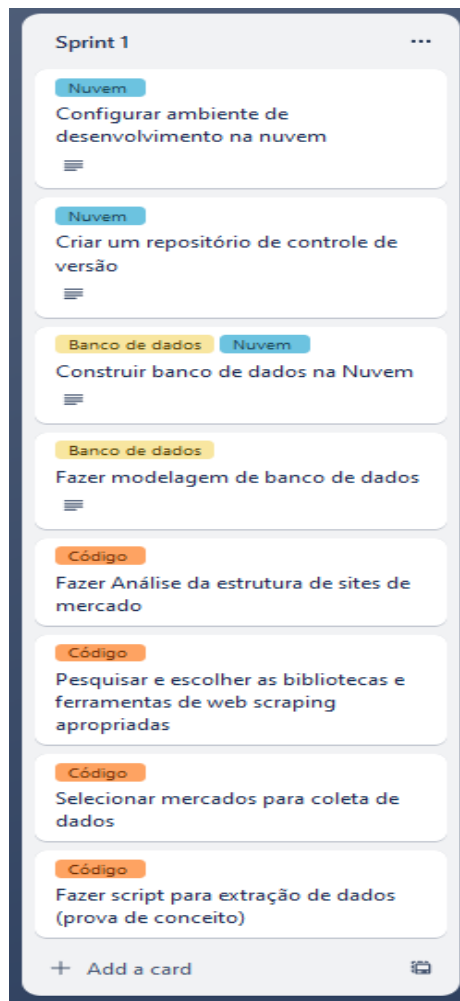
2.1 Sprint 1

Na primeira sprint, deu-se início ao desenvolvimento do projeto, marcando uma etapa de alta importância para delimitação, orientação e direcionamento da solução do projeto. Essa fase teve um foco maior em configurações iniciais do ambiente nuvem, estruturação da forma de armazenamento de dados coletados e na realização de provas de conceitos para extração de informações de produtos de supermercados. Para isso, foi necessário analisar e selecionar as ferramentas e tecnologias que melhor atendessem aos requisitos do projeto. Além disso, foi necessário aprofundar o entendimento do contexto dos supermercados e identificar as informações relevantes a serem extraídas, entre outras tarefas.

2.1.1 Solução

Evidência do planejamento:

Abaixo, podemos observar a evidência do planejamento da primeira sprint.



Evidência da execução de cada requisito:

1.1 Configurar ambiente de desenvolvimento nuvem

Nesta etapa, concentramos nossos esforços na configuração de um ambiente de desenvolvimento na nuvem. A escolha da AWS (Amazon Web Services) como nossa plataforma de nuvem foi uma decisão estratégica, que trouxe benefícios e desafios específicos. A AWS oferece um ecossistema rico em serviços, que nos permite escolher as ferramentas mais adequadas para as demandas do projeto.

Abaixo podemos observar a captura de tela da nossa infraestrutura na nuvem nova.

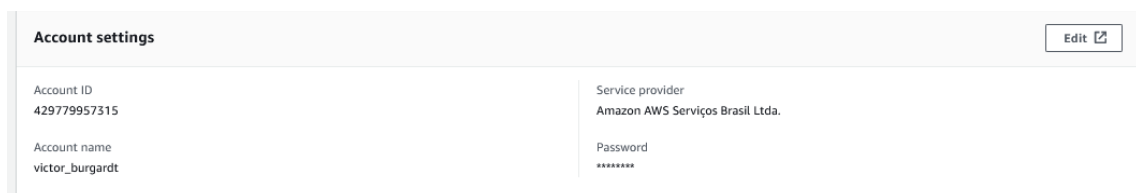


Figura 8: Conta AWS

1.2 Criar um repositório de controle de versão

Nesta etapa, foi criado de um repositório de controle de versão, desempenhando um papel crucial na organização e no desenvolvimento do projeto. O repositório proporciona um ambiente adequado para o gerenciamento das versões do projeto, permitindo o acompanhamento das mudanças ao longo do tempo e proporcionando um ambiente de colaboração eficaz entre membros da equipe. Para atender esse requisito, foi escolhido o GitHub como plataforma de controle de versões. Neste contexto, foi criado um repositório público no GitHub, que oferece flexibilidade aos colaboradores e desenvolvedores, garantindo visibilidade da equipe.

A imagem abaixo ilustra a captura de tela do repositório no GitHub, demonstrando sua existência e configuração:



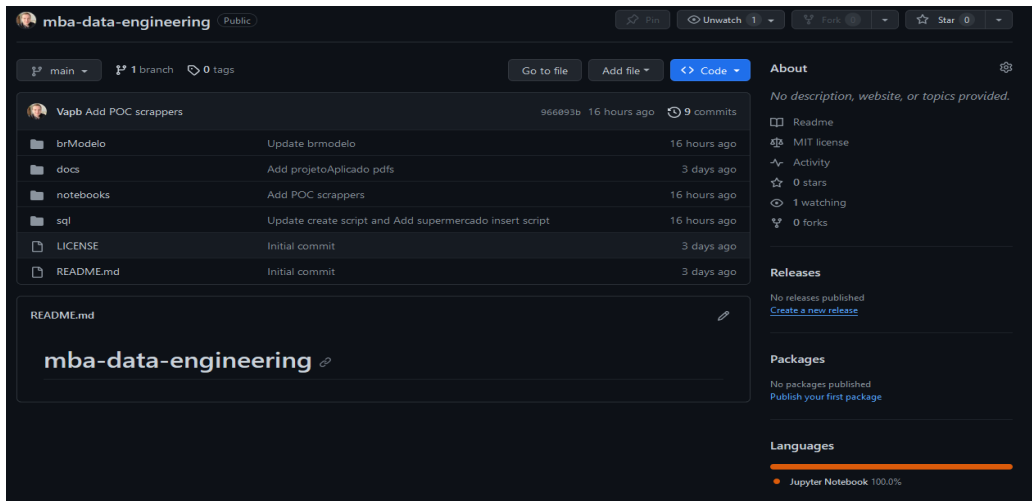


Figura 9: Repositório github

Além disso, você pode acessar o repositório de controle de versão no GitHub através do seguinte link: <https://github.com/Vapb/mba-data-engineering>.

1.3 Construir banco de dados na Nuvem

Nesta etapa, focamos na criação de um banco de dados na nuvem. Optamos por utilizar o Amazon RDS (Relational Database Service) da AWS com PostgreSQL, que oferece um ambiente robusto e escalável para o armazenamento e gerenciamento de dados. O Amazon RDS proporciona uma solução gerenciada que simplifica a administração do banco de dados, garantindo alta disponibilidade, segurança e desempenho.

Além disso, durante esta etapa, levamos em consideração a opção de uso do nível gratuito (free tier) do Amazon RDS, que nos permite aproveitar recursos sem custos iniciais. Dessa forma, teremos uma escolha econômica e eficiente para o projeto, sem grandes pesos no orçamento do projeto.

Abaixo podemos observar uma captura de tela do RDS construído.

Instance		
Configuration	Instance class	Storage
DB instance ID db-mba-dataengineering	Instance class db.t3.micro	Encryption Enabled
Engine version 15.3	vCPU 2	AWS KMS key aws/rds
DB name -	RAM 1 GB	Storage type General Purpose SSD (gp2)
License model Postgresql License	Availability	Storage 20 GiB
Option groups default:postgres-15 In sync	Master username [REDACTED]	Provisioned IOPS -
Amazon Resource Name (ARN) arn:aws:rds:us-east-1:429779957315:db:db-mba-dataengineering	Master password *****	Storage throughput -
Resource ID db-KW6Z62EVD7JCQ4OX85LKK5XBJ4	IAM DB authentication Not enabled	Storage autoscaling Disabled
Created time November 06, 2023, 19:47 (UTC-03:00)	Multi-AZ No	Storage file system configuration Current
DB instance parameter group default.postgres15 In sync	Secondary Zone -	
Deletion protection Disabled		

Figura 10: Instancia RDS

1.4 Fazer modelagem do banco de dados

Nesta etapa, foi feito o fluxo completo de modelagem de banco de dados para o projeto, que desempenha um papel fundamental para o estabelecer a estrutura que armazenará e organizará os dados coletados no projeto. O objetivo desse processo consiste em definir como os dados serão estruturados e relacionados, permitindo a eficácia nas consultas e análises futuras. O Banco de dados proposto no projeto tem o foco em possibilitar análises históricas detalhadas dos preços de produtos de supermercados, o que exige um modelo sólido que capture todas as informações relevantes. O fluxo de modelagem do banco de dados seguiu a seguinte abordagem: modelo conceitual, modelo lógico e modelo físico. Para a construção dos modelos conceitual e lógico foi utilizado a ferramenta brModelo. Podemos observar cada uma das etapas abaixo.

1.4.1 Modelagem conceitual com Modelo Entidade-Relacionamento (ER)

A partir do domínio do contexto do projeto, foi feita a modelagem conceitual na qual utilizamos o Modelo Entidade-Relacionamento (ER) para representar abstratamente as principais entidades, atributos e seus relacionamentos. Esse modelo representa uma base sólida para o projeto enfatizando a compreensão do negócio sem se prender a detalhes técnicos.

A imagem abaixo ilustra o modelo entidade-relacionamento resultante do projeto.

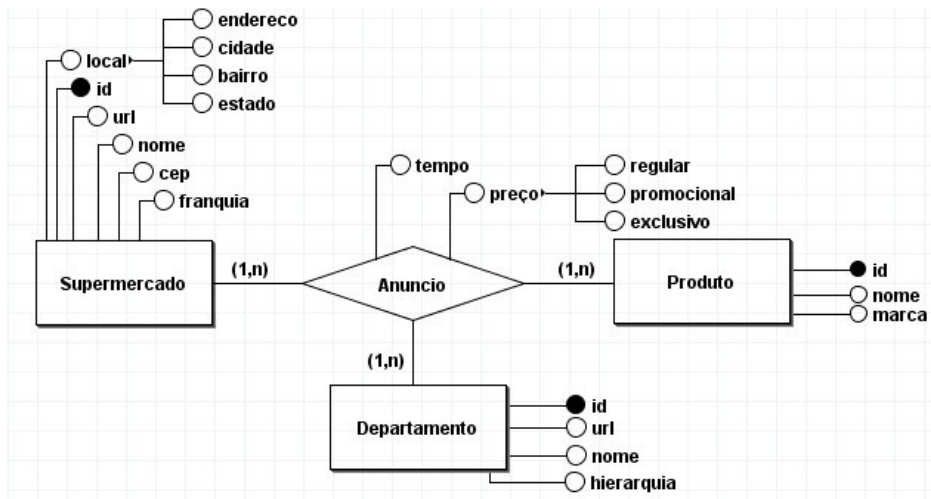


Figura 11: Modelagem conceitual

1.4.2 Modelagem lógica

Após a modelagem conceitual ER foi realizada a modelagem lógica, em que os conceitos abstratos foram traduzidos em estruturas de dados concretas, como tabelas e campos, levando em consideração as necessidades de consulta e integração dos dados.

A imagem abaixo ilustra o modelo lógico do banco de dados do projeto.

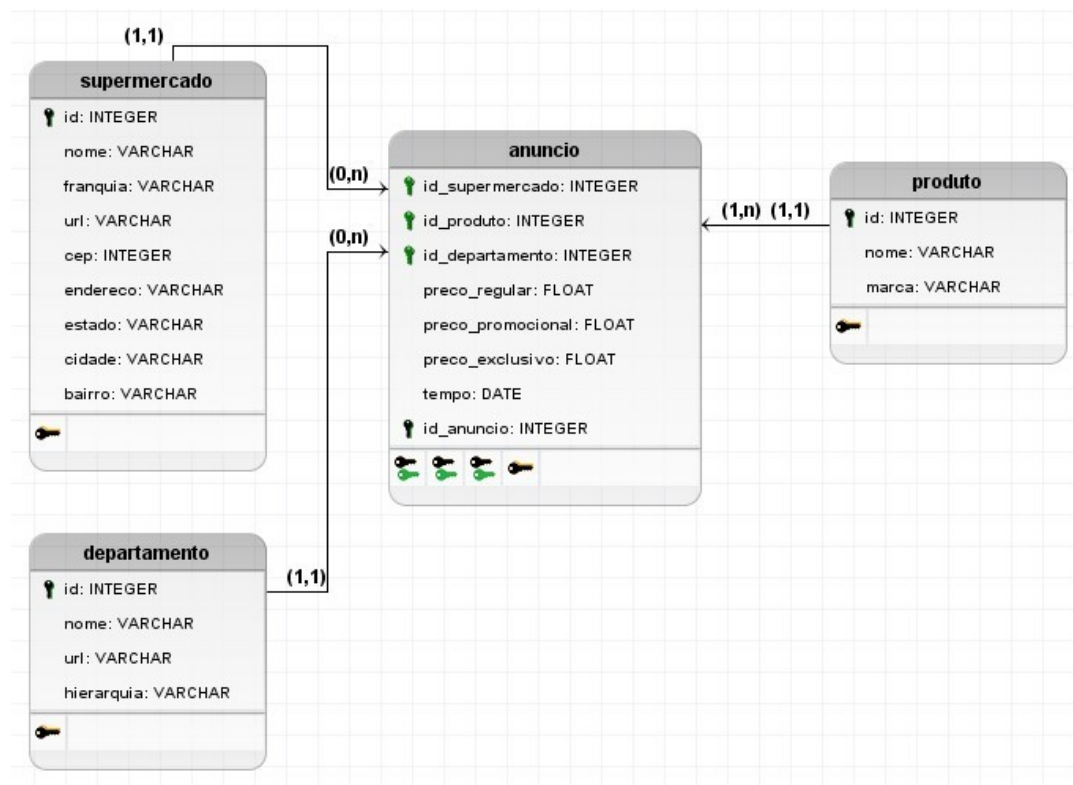


Figura 12: Modelagem lógica

1.4.3 Modelagem física

Com o modelo lógico completo deu-se início a modelagem física, que converteu o modelo lógico em um esquema de banco de dados real, incluindo detalhes como tipos de dados, índices e chaves primárias, preparando a infraestrutura para armazenar e gerenciar os dados. Foi escolhido o PostgreSQL como o Sistema de Gerenciamento de Banco de Dados (SGBD) para este projeto, devido à sua robustez e suporte a recursos avançados de manipulação de dados.

A imagem abaixo ilustra a captura de tela do script de construção de tabelas em PostgreSQL

```
create table if not exists supermercado(  
    id SERIAL primary key,  
    nome VARCHAR(60) unique not null,  
    franquia VARCHAR(30) not null,  
    endereco VARCHAR(90),  
    estado VARCHAR(30),  
    cidade VARCHAR(30),  
    bairro VARCHAR(30),  
    url VARCHAR(60) not null,  
    cep INTEGER  
);  
  
create table if not exists departamento(  
    id SERIAL primary key,  
    nome VARCHAR(30) not null,  
    url VARCHAR(120) not null,  
    hierarquia VARCHAR(90) not null  
);  
  
create table if not exists produto(  
    id SERIAL primary key,  
    nome VARCHAR(90) not null,  
    marca VARCHAR(60)  
);  
  
create table if not exists anuncio(  
    id_anuncio SERIAL primary key,  
    id_supermercado INTEGER not null,  
    id_departamento INTEGER not null,  
    id_produto INTEGER not null,  
    preco_regular FLOAT,  
    preco_exclusivo FLOAT,  
    tempo DATE,  
    FOREIGN KEY (id_supermercado)  
        references supermercado(id),  
    FOREIGN KEY (id_departamento)  
        references departamento(id),  
    FOREIGN KEY (id_produto)  
        references produto(id)  
);
```

Figura 13: Modelagem Física

1.5 Análise da Estrutura de Sites de Supermercados

Nesta etapa, realizamos uma análise detalhada da estrutura de sites de supermercados. Essas análises possuem grande importância para desenvolvimento do código de coleta de dados pois diferentes sites podem adotar abordagens distintas para disponibilizar informações, seja por meio de APIs, tokens de autenticação,



solicitações HTTP ou outros métodos. Com essas informações podemos escolher a melhor forma de coletar essas informações. Durante a análise da estrutura dos sites, identificamos as informações essenciais que poderiam ser coletadas, incluindo diferentes tipos de preços de produtos, características de produtos e datas da coleta.

Para realizar essas análises, utilizamos a ferramenta Burp Suite, que oferece recursos poderosos para inspeção, interceptação e modificação de solicitações HTTP. Essa ferramenta permitiu identificar tokens de autenticação, examinar os fluxos de dados e entender as respostas dos servidores web ligados aos sites de supermercados.

A imagem abaixo ilustra a captura de tela de uma das análises com a ferramenta Burp suite.

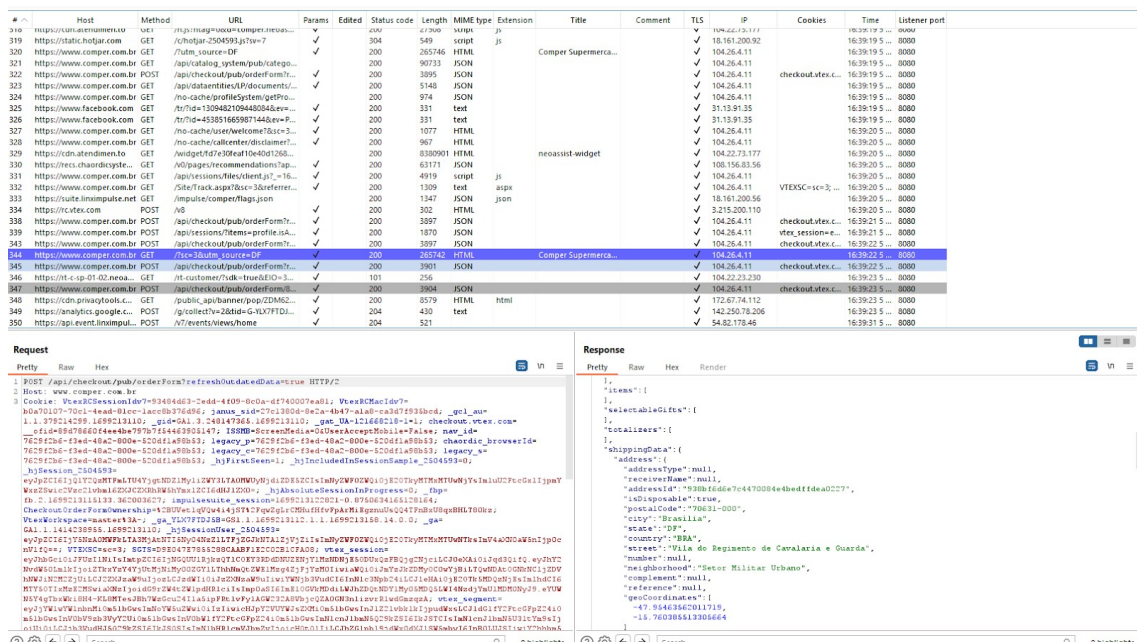


Figura 14: Análise com burpSuite

1.6 Pesquisar e escolher bibliotecas

Nesta etapa do projeto, dedicamos tempo à pesquisa e seleção das bibliotecas Python necessárias para implementar coletores de dados e web scraping de maneira eficiente e eficaz. A escolha dessas bibliotecas é crucial para o sucesso do projeto, pois influencia diretamente a capacidade de extrair informações valiosas dos sites de supermercados.

Após a análise da estrutura dos sites, fizemos uma extensa pesquisa para identificar as bibliotecas mais adequadas para o web scraping das estruturas vistas. Analisamos diferentes opções, levando em consideração fatores como desempenho, funcionalidades, documentação, comunidade de suporte e familiaridade da organização com as ferramentas.

Após uma extensa avaliação, selecionamos as bibliotecas `re`, `requests`, `lxml`, `urllib`, e `unidecode` para cumprir nossos requisitos de coleta de dados. Cada uma dessas bibliotecas desempenha um papel específico em nossa estratégia de web scraping:

- **`re`**: Utilizada para realizar expressões regulares, o que é útil para buscar e extrair padrões de texto dos sites.
- **`requests`**: Essa biblioteca é usada para fazer solicitações HTTP e interagir com os servidores web dos supermercados.
- **`lxml`**: É uma poderosa biblioteca para processamento de HTML e XML, permitindo-nos analisar a estrutura das páginas da web e extrair informações de forma estruturada.
- **`urllib`**: Utilizada para manipulação de URLs e construção de links, o que é importante para navegar nas páginas dos supermercados.
- **`unidecode`**: Essa biblioteca é usada para lidar com a normalização de caracteres e codificação, garantindo que os dados coletados sejam consistentes e legíveis.

1.7 Seleção de sites de supermercados

Após a análise detalhada da estrutura dos sites de supermercados, é iniciado o processo para a seleção dos mercados a serem incluídos no nosso escopo. Foram analisados uma variedade de opções, considerando fatores como a amplitude, facilidade de acesso aos dados e potencial de coleta de informações relevantes.

Foram analisados diversos supermercados para integrar nosso projeto. Os sites com melhor aderência a nosso escopo foram Arasuper e Comper. Ambos foram escolhidos por suas características específicas que facilitam a coleta de informações. O Comper oferece uma API que, se for necessário, pode ser adaptada para acessar informações de outras localidades, expandindo assim nossa capacidade de coleta de dados para até seis lugares distintos. O Arasuper, embora opere em dois estados, apresenta uma estrutura de dados acessível e oferece informações do norte do país, o que o torna valioso para nossas análises regionais.

A seleção cuidadosa dos supermercados garante que nosso projeto tenha uma base sólida e diversificada para a coleta de dados de preços de produtos, permitindo análises abrangentes e insights relevantes para os consumidores e a equipe do projeto.

1.8 Fazer script de extração de dados (Prova de conceitos)

Nesta etapa, desenvolvemos scripts de extração de dados para coletar informações sobre produtos de supermercados. Esses scripts desempenham um papel fundamental na nossa estratégia, permitindo fazer uma prova de conceito sobre a



captura eficiente de dados relevantes. Essa etapa é essencial pois proporciona uma validação na metodologia proposta para o projeto.

Cada script foi projetado com funções específicas que se encaixam em um fluxo de dados bem definido. O objetivo é fazer o web scraping de forma que o código percorre os sites dos supermercados, acessa os departamentos correspondentes e coleta informações sobre vários produtos de cada departamento. Essa estrutura facilita a expansão para coletar informações de um grande número de produtos em diferentes categorias.

Para realizar essas atividades, foram desenvolvidas funções personalizadas que manipulam as solicitações HTTP, analisam o conteúdo das páginas web, extraem informações relevantes e armazenam os dados de maneira organizada.

É importante esclarecer que existem outras funções suporte como definição de tokens e seções além de diferenças nas funções entre sites diferentes de supermercados, porém a estrutura de coleta segue a mesma ideia. Abaixo podemos observar capturas de tela com as principais funções.

Função de Recuperação de Departamentos: Esta função retorna todos os departamentos com URLs e hierarquias de departamento

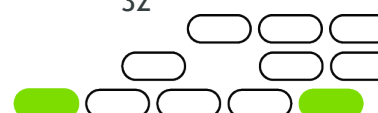
```
def get_all_departments(json_response):
    all_departments = {}
    for department in json_response:
        department_name = department['name']
        if department['hasChildren']:
            for sub_department in department['children']:
                sub_department_name = sub_department['name']
                if sub_department['hasChildren']:
                    for sub_sub_department in sub_department['children']:
                        sub_sub_department_name = sub_sub_department['name']
                        category_name = process_text(f"{department_name}/{sub_department_name}/{sub_sub_department_name}")
                        all_departments[sub_sub_department['url']] = category_name
                else:
                    category_name = process_text(f"{department_name}/{sub_department_name}")
                    all_departments[sub_department['url']] = category_name
            else:
                category_name = process_text(f"{department_name}")
                all_departments[department['url']] = category_name
    return all_departments
```

Figura 15: Função `get_all_departments`

Função de Navegação em Páginas de Produtos: Esta função recebe a URL de um departamento e itera pelas páginas de produtos de cada departamento.

```
def get_pages(url):
    i = 1
    while True:
        new_url = url + '?page=' + str(i)
        tree = get_tree_from_url(session, new_url)
        page_empty = tree.xpath('//div[@class="produto-lista empty-content"]')
        if page_empty != []:
            break
        else:
            yield tree
        i = i + 1
```

Figura 16: Função `get_pages`



Função de extração e formatação de produtos: Esta função retorna uma lista de dicionários com informações de produtos formatadas a partir do HTML da página.

```
def extract_products(page):
    products_info = []
    for product in page.xpath(PRODUCT_XPATH):
        product_url = BASE_URL + product.xpath('@href')[0]
        product_name = product.xpath(NAME_XPATH)[0].text_content()
        product_brand = product.xpath(BRAND_XPATH)[0].text_content()

        if product.xpath(PRODUCT_ONE_XPATH): # One price
            regular_price = None
            price = product.xpath(PRODUCT_ONE_XPATH)[0].text_content()
            price = extract_price(price)

        elif product.xpath(PRODUCT_TWO_XPATH): # Regular Price and Price
            prices = product.xpath(PRODUCT_TWO_XPATH)[0]
            regular_price = prices.xpath(REGULAR_PRICE_XPATH)[0].text_content()
            regular_price = extract_price(regular_price)
            price = prices.xpath(PRICE_XPATH)[0].text_content()
            price = extract_price(price)

        else: # Only price
            prices = product.xpath(PRODUCT_LEVE_XPATH)[0]
            regular_price = prices.xpath(REGULAR_PRICE_XPATH)[0].text_content()
            regular_price = extract_price(regular_price)
            price = prices.xpath(PRICE_XPATH)[0].text_content()
            price = extract_price(price)

        products_info.append(
            {
                'url': product_url,
                'name': product_name,
                'brand': product_brand,
                'price': price,
                'regular_price': regular_price
            }
        )

    return products_info
```

Figura 17: Função `extract_products`

Evidência dos resultados:

Nesta sprint, atingimos importantes marcos que fortalecem a base do projeto e validam nossa abordagem. Construímos um banco de dados na nuvem com uma estrutura sólida, pronto para armazenar os dados coletados dos supermercados. Seleccionamos as ferramentas necessárias para a construção do projeto de forma metódica. Além disso, realizamos com sucesso provas de conceito e funções que validam nossa proposta de coleta de informações de mercado por meio do web scraping. Essas funções estão prontas para serem adaptadas e implementadas na nuvem.



Abaixo, podemos observar o banco de dados PostgreSQL na nuvem, já inicializado, com alguns dados de supermercados inseridos.

select * from supermercado | Enter a SQL expression to filter results (use Ctrl+Space)

id	nome	franquia	endereco	estado	cidade	bairro	url	cep
1	comper - cuiaba - jardim italia	comper	av. gov. dante martins de oliveira, 1093	mato grosso	cuiaba	jardim italia	https://www.comper.com.br/?sc=1	78.050,170
2	comper - campo grande - itanhangá park	comper	r. joaquim murtinho, 1679	mato grosso do sul	campo grande	itanhangá park	https://www.comper.com.br/?sc=2	79.003,027
3	comper - brasil - águas claras	comper	r. 36 norte, s/n - lt. 05	distrito federal	brasil	águas claras	https://www.comper.com.br/?sc=3	71.919,180
4	comper - brasil - sobradinho	comper	lote 12/16, q 14	distrito federal	brasil	sobradinho	https://www.comper.com.br/?sc=4	73.050,140
5	comper - dourados - vila alba	comper	av. marcelino pires, 3855	mato grosso do sul	dourados	vila alba	https://www.comper.com.br/?sc=5	79.801,002
6	comper - rondonópolis - centro	comper	av. rui barbosa, 1859	mato grosso	rondonópolis	centro	https://www.comper.com.br/?sc=6	78.700,130
7	arasuper - rio branco - aviação	arasuper	est. do aviação, 122	acre	rio branco	aviário	https://www.arasuper.com.br/	69.900,854

Figura 18: Tabela Supermercados

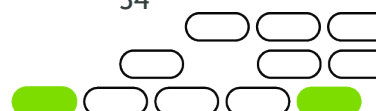
Logo abaixo, apresentamos alguns dos resultados obtidos dos scripts de prova de conceito de extração de dados, fornecendo uma evidência clara da viabilidade da nossa metodologia.

Extração de departamentos através de script

```
{
  'https://www.arasuper.com.br/c/massas-resfriadas/580/': 'Alimentos > Massas Resfriadas',
  'https://www.arasuper.com.br/c/torradas/562/': 'Alimentos > Torradas',
  'https://www.arasuper.com.br/c/diet/547/': 'Alimentos > Sal > Diet',
  'https://www.arasuper.com.br/c/queijo-ralado/530/': 'Alimentos > Queijo Ralado',
  'https://www.arasuper.com.br/c/soja/490/': 'Alimentos > Óleo > Soja',
  'https://www.arasuper.com.br/c/milho/489/': 'Alimentos > Óleo > Milho',
  'https://www.arasuper.com.br/c/girassol/488/': 'Alimentos > Óleo > Girassol',
  'https://www.arasuper.com.br/c/coco/487/': 'Alimentos > Óleo > Coco',
  'https://www.arasuper.com.br/c/composto/486/': 'Alimentos > Óleo > Composto',
  'https://www.arasuper.com.br/c/canola/485/': 'Alimentos > Óleo > Canola',
  'https://www.arasuper.com.br/c/mostarda/481/': 'Alimentos > Mostarda',
  'https://www.arasuper.com.br/c/polpa-de-tomate/475/': 'Alimentos > Polpa de Tomate',
  'https://www.arasuper.com.br/c/molho-tomate/474/': 'Alimentos > Molho Tomate',
  'https://www.arasuper.com.br/c/organico/462/': 'Alimentos > Manteigas > Orgânico',
  'https://www.arasuper.com.br/c/maionese/457/': 'Alimentos > Maionese',
  'https://www.arasuper.com.br/c/frutas/453/': 'Alimentos > Frutas',
  'https://www.arasuper.com.br/c/leite-de-coco/429/': 'Alimentos > Leite de Coco',
  'https://www.arasuper.com.br/c/diet/426/': 'Alimentos > Leite Condensado > Diet',
  'https://www.arasuper.com.br/c/instantaneas/455/': 'Alimentos > Massas > Instantâneas',
  'https://www.arasuper.com.br/c/sem-gluten/454/': 'Alimentos > Massas > Sem Gluten',
  'https://www.arasuper.com.br/c/granola/398/': 'Alimentos > Granola',
  'https://www.arasuper.com.br/c/geleias/391/': 'Alimentos > Geléias',
  'https://www.arasuper.com.br/c/diet/390/': 'Alimentos > Gelatinas > Diet',
  'https://www.arasuper.com.br/c/vermelho/369/': 'Alimentos > Feijão > Vermelho',
  'https://www.arasuper.com.br/c/rajado/368/': 'Alimentos > Feijão > Rajado',
  ...
  'https://www.arasuper.com.br/c/paes-producao-propria/499/': 'Padaria e Confeitaria > Pães Produção Própria',
  'https://www.arasuper.com.br/c/paes/279/': 'Padaria e Confeitaria > Pães',
  'https://www.arasuper.com.br/c/bolos/210/': 'Padaria e Confeitaria > Bolos',
  'https://www.arasuper.com.br/c/paes-industrializados/207/': 'Padaria e Confeitaria > Pães Industrializados',
  'https://www.arasuper.com.br/c/biscoitos-e-torradas/197/': 'Padaria e Confeitaria > Biscoitos e Torradas'
}
```

Output is truncated. View as a [scrollable element](#) or open in a [text editor](#). Adjust cell output [settings](#).

Figura 19: Resultado departamentos arasuper



Extração de informações de produtos através de script

```

PAGE : <Element html at 0x1a1632a8180>
PRODUCT : <Element a at 0x1a1632a8180>
{'url': 'https://www.arasuper.com.br/p/bebida-gin-rocks-sunset-1l/19551/', 'name': 'Bebida Gin Rocks Sunset 1L', 'brand': 'ROCKS', 'price': 48.9, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351720>
{'url': 'https://www.arasuper.com.br/p/bebida-vodka-natural-absolut-750ml/19263/', 'name': 'Bebida Vodka Natural Absolut 750ML', 'brand': 'ABSOLUT', 'price': 79.98, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351180>
{'url': 'https://www.arasuper.com.br/p/bebida-aguardente-segredo-chacara-ouro-900ml/18585/', 'name': 'Bebida Aguardente Segredo Chacara Ouro 900ML', 'brand': 'SEGREDO DA CHACARA', 'price': 24.98, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351310>
{'url': 'https://www.arasuper.com.br/p/bebida-aguardente-segredo-chacara-900ml/18587/', 'name': 'Bebida Aguardente Segredo Chacara 900ML', 'brand': 'SEGREDO DA CHACARA', 'price': 24.98, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351640>
{'url': 'https://www.arasuper.com.br/p/bebida-whisky-silverhorn-700ml/18424/', 'name': 'Bebida Whisky Silverhorn 700ML', 'brand': 'SILVERHORN', 'price': 158.99, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351b30>
{'url': 'https://www.arasuper.com.br/p/bebida-whisk-cutty-sark-1l/18429/', 'name': 'Bebida Whisk Cutty Sark 1L', 'brand': 'CUTTY SARK', 'price': 76.75, 'regular_price': 79.69}
PRODUCT : <Element a at 0x1a1643516d0>
{'url': 'https://www.arasuper.com.br/p/bebida-whisky-american-bison-700ml/18420/', 'name': 'Bebida Whisky American Bison 700ML', 'brand': 'A.BISON', 'price': 229.0, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351f40>
{'url': 'https://www.arasuper.com.br/p/bebida-tequilero-del-leste-prata-com-2-copos-750ml/18421/', 'name': 'Bebida Tequilero Del Leste Prata Com 2 Copos 750ML', 'brand': 'TEQUILERO DO LESTE', 'price': 72.98, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351f90>
{'url': 'https://www.arasuper.com.br/p/bebida-whisky-johnnie-walker-double-label-1l/18418/', 'name': 'Bebida Whisky Johnnie Walker Double Label 1L', 'brand': 'JOHNNIE WALKER', 'price': 279.9, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351ef0>
{'url': 'https://www.arasuper.com.br/p/bebida-whisky-chivas-scoth-750ml/18415/', 'name': 'Bebida Whisky Chivas Scoth 750ML', 'brand': 'SEAGRAM', 'price': 215.99, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351d60>
{'url': 'https://www.arasuper.com.br/p/bebida-whisky-chivas-18-anos-750ml/18414/', 'name': 'Bebida Whisky Chivas 18 Anos 750ML', 'brand': 'CHIVAS', 'price': 549.99, 'regular_price': None}
PRODUCT : <Element a at 0x1a163369d0>
{'url': 'https://www.arasuper.com.br/p/bebida-whisky-johnnie-walker-gold-reserve-750ml/18416/', 'name': 'Bebida Whisky Johnnie Walker Gold Reserve 750ML', 'brand': 'JOHNNIE WALKER', 'price': 480.99, 'regular_price': None}
...
PRODUCT : <Element a at 0x1a163353220>
{'url': 'https://www.arasuper.com.br/p/cachaca-caninha-6l-garrafa-970ml/6926/', 'name': 'Cachaca Caninha 6L Garrafa 970ML', 'brand': '6L', 'price': 13.98, 'regular_price': None}
...
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings.
PAGE : <Element html at 0x1a164351d60>
PRODUCT : <Element a at 0x1a163316630>
{'url': 'https://www.arasuper.com.br/p/bebida-gin-rocks-1l/6777/', 'name': 'Gin Rocks 1L', 'brand': 'ROCKS', 'price': 48.9, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351ef0>
{'url': 'https://www.arasuper.com.br/p/whisky-white-horse-garrafa-1l/6778/', 'name': 'Whisky White Horse Garrafa 1L', 'brand': 'WHITE HORSE', 'price': 112.9, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351f90>
{'url': 'https://www.arasuper.com.br/p/cachaca-cabare-ouro-garrafa-700ml/5784/', 'name': 'Cachaca Cabare Ouro Garrafa 700ML', 'brand': 'CABARE', 'price': 45.99, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351d60>
{'url': 'https://www.arasuper.com.br/p/gin-gordons-750ml/5661/', 'name': 'Gin Gordons 750ML', 'brand': 'GORDONS', 'price': 99.99, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351d40>
{'url': 'https://www.arasuper.com.br/p/cachaca-velho-barreiro-910ml/4966/', 'name': 'Cachaca Velho Barreiro 910ML', 'brand': 'VELHO BARREIRO', 'price': 19.99, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351b30>
{'url': 'https://www.arasuper.com.br/p/bebida-catuba-felina-500ml/4921/', 'name': 'Bebida Catuba Felina 500ML', 'brand': 'FELINA', 'price': 5.99, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351180>
{'url': 'https://www.arasuper.com.br/p/vodka-leonoff-900ml/4901/', 'name': 'Vodka Leonoff 900ML', 'brand': 'LEONOFF', 'price': 17.99, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351d60>
{'url': 'https://www.arasuper.com.br/p/whisky-grand-old-parr-12-anos-1l/4838/', 'name': 'Whisky Grand Old Parr 12 Anos 1L', 'brand': 'OLD PARR', 'price': 229.9, 'regular_price': None}
PRODUCT : <Element a at 0x1a1643516d0>
{'url': 'https://www.arasuper.com.br/p/ice-skarloff-frutas-vermelhas-275ml/4865/', 'name': 'Ice Skarloff Frutas Vermelhas 275ML', 'brand': 'SKARLOFF', 'price': 6.98, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351f40>
{'url': 'https://www.arasuper.com.br/p/ice-skarloff-limao-275ml/3358/', 'name': 'Ice Skarloff Limão 275ML', 'brand': 'SKARLOFF', 'price': 6.98, 'regular_price': None}
PRODUCT : <Element a at 0x1a163369d0>
{'url': 'https://www.arasuper.com.br/p/bebida-aguardente-6l-600ml/1359/', 'name': 'Bebida Aguardente 6L 600ML', 'brand': '6L', 'price': 9.99, 'regular_price': None}
PRODUCT : <Element a at 0x1a164351d60>
{'url': 'https://www.arasuper.com.br/p/vodka-skarloff-965ml/3356/', 'name': 'Vodka Skarloff 965ML', 'brand': 'SKARLOFF', 'price': 19.98, 'regular_price': None}
...
PRODUCT : <Element a at 0x1a163369d0>
{'url': 'https://www.arasuper.com.br/p/ice-birlnight-limao-1l/3136/', 'name': 'Ice Birlnight Limão 1L', 'brand': 'BIRLNGHT', 'price': 11.69, 'regular_price': None}

```

Figura 20: Resultado produtos arasuper

2.1.2 Lições Aprendidas

Nesta sprint, obtivemos valiosas lições que enriquecem nossa jornada de desenvolvimento. Essas experiências nos forneceram um conhecimento mais profundo sobre o processo de extração de dados de supermercados e a preparação do banco de dados na nuvem. Embora tenhamos progredido na criação de scripts para coleta de informações, ainda temos que garantir que eles estejam otimizados para operar de maneira escalável na nuvem. Outro aprendizado importante é a necessidade de uma avaliação cuidadosa dos custos associados ao Amazon RDS, percebemos que os preços de operação variam bastante, sendo necessário uma maneira de monitorar mais de perto. Além disso, alguns mercados possuem uma quantidade significativa de produtos, levando-nos a refletir sobre estratégias eficientes para gerenciar essa demanda.



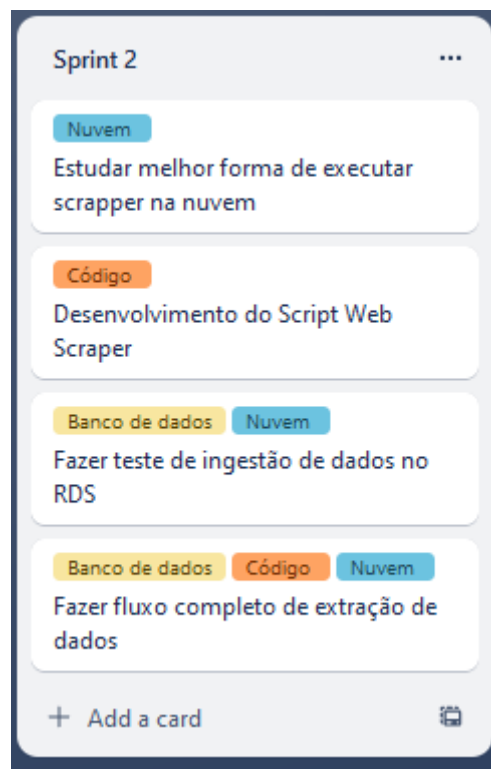
2.2 Sprint 2

Na segunda sprint, avançamos na consolidação do fluxo de código, movendo-o do ambiente local para a nuvem. Esta etapa teve como objetivo transformar as provas de conceito desenvolvidas na sprint 1 em um código robusto e eficiente que funcione e interaja com ambiente na nuvem construído previamente. Para alcançar esse objetivo, realizamos uma análise criteriosa das melhores tecnologias para execução e transição do código para a nuvem, adaptamos as funções de prova de conceito, estabelecemos conexões entre os serviços dentro da nuvem e validamos o fluxo do código no novo ambiente.

2.2.1 Solução

Evidência do planejamento:

Abaixo, podemos observar a evidência do planejamento da segunda sprint.



Evidência da execução de cada requisito:

2.1 Estudar melhor forma de executar scraper na nuvem

Nesta etapa, dedicamos tempo ao estudo das opções disponíveis para executar os web scrapers na nuvem AWS. Duas soluções principais foram avaliadas: Amazon EC2 e AWS Lambda.

O Amazon EC2, uma instância virtual escalável, foi comparado ao AWS Lambda, um serviço serverless. Embora ambas as opções ofereçam vantagens distintas, nossa análise revelou que o EC2 seria mais favorável para as demandas específicas do projeto, proporcionando maior flexibilidade e controle sobre o ambiente de execução que é fundamental para a complexidade do nosso problema.

Para evidenciar as comparações feitas, fizemos alguns testes com coletores simples utilizando AWS Lambda e EC2, as imagens abaixo evidenciam os testes.

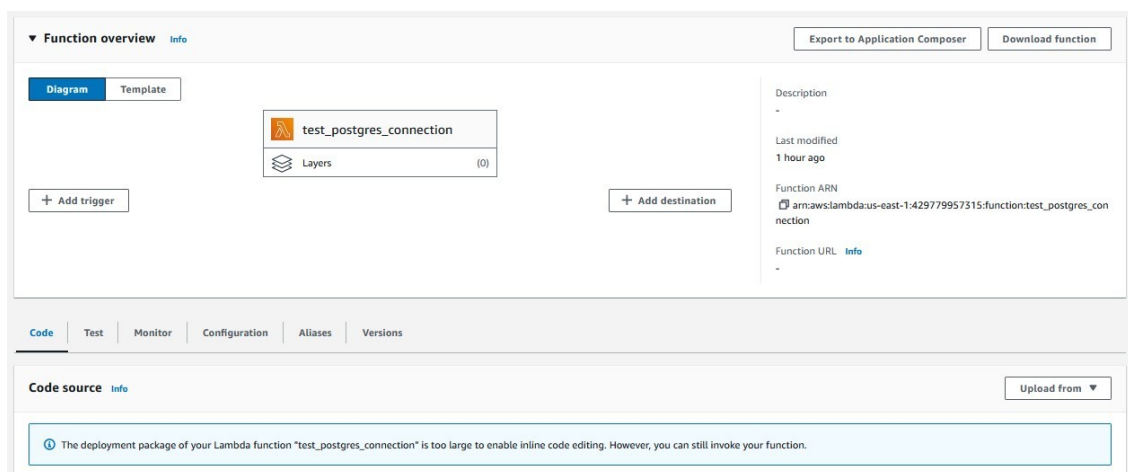


Figura 21: Teste coletor simples lambda com conexão postgres

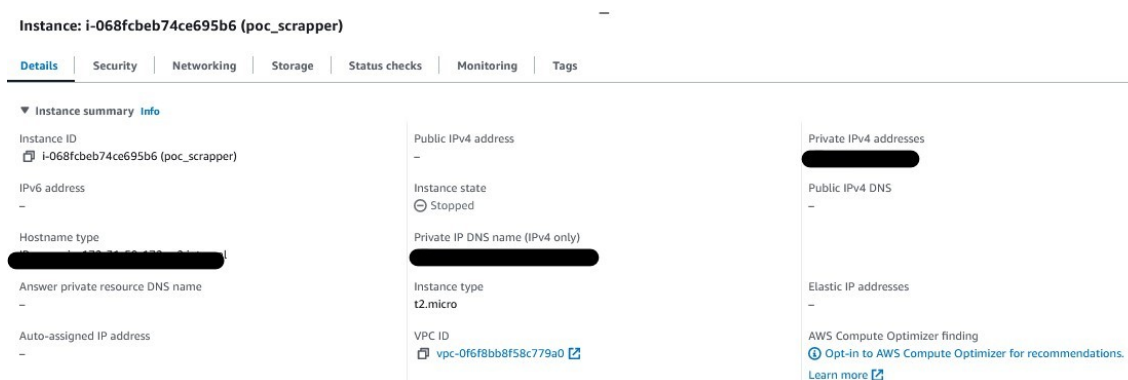


Figura 22: Teste coletor simples EC2

2.2 Desenvolvimento do Script Web Scraper

Nesta etapa, desenvolvemos os scripts para fazer o *web scraping* dos sites de supermercados. Para atingir esse objetivo, atualizamos as funções de prova de conceito desenvolvidas anteriormente, ajustando-as para funcionarem de maneira

sinérgica. Dessa forma consolidando um bloco de código robusto e coeso, capaz de extrair dados dos supermercados de forma eficiente. É importante ressaltar que esse processo de atualização e consolidação foi feito duas vezes para lidar com as peculiaridades de cada mercado, Comper e Arasuper. Além disso, foram criadas funções de pré-processamento auxiliares que são utilizadas pelos dois códigos.

Para ilustrar essas realizações, fornecemos snippets do código.

```
def main():
    session = Session()
    for department in get_departments(session):
        print('DEP: ', department)
        for page in get_pages(session, department['url']):
            products = extract_products(page)
            for product in products:
                print(product)
            sleep(15) # 15 Seconds for Page
    return None
```

Figura 23: Função main do script Arasuper

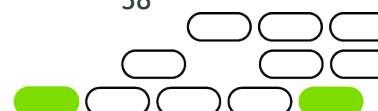
```
def main(store_id):
    session = Session()
    tokens = set_token(session, store_id)
    for department in get_departments(session):
        print('DEP: ', department)
        for page in get_pages(session, department['url'], tokens):
            products = extract_products(page)
            if len(products) == 0:
                break
            for product in products:
                print(product)
            sleep(15) # 15 Seconds for Page
    return None
```

Figura 24: Função main script Comper

2.3 Fazer teste de ingestão de dados nuvem

Nesta etapa, conduzimos um teste de ingestão de dados para o postgres-rds na nuvem AWS. Para isso, adaptamos o script de web scraper de cada site, introduzindo uma classe python que simplifica a conexão com o PostgreSQL. Essa classe, projetada para facilitar a inserção de dados em tabelas, oferece métodos intuitivos, como connect, insert_department, insert_product, close entre outros, dessa forma, simplificando o fluxo de trabalho durante o processo de ingestão de dados.

Abaixo fornecemos alguns *snippets* da classe PostgreSQLConnection.



```
def main():
    collection_date = datetime.now()
    session = Session()
    conn = PostgreSQLConnection(
        dbname="postgres",
        user=os.getenv("POSTGRES_USER"),
        password=os.getenv("POSTGRES_PASSWORD")
    )
    conn.connect()
    for department in get_departments(session):
        print('DEP: ', department)
        conn.insert_department(
            department['id'], department['url'], department["hierarchy"]
        )
        for page in get_pages(session, department['url']):
            products = extract_products(page)
            for product in products:
                print(product)
                conn.insert_product(product['sku'], product["name"], product["brand"])
                conn.insert_advertisement(
                    BASE_MKP,
                    department['id'],
                    product['sku'],
                    product['price'],
                    product['regular_price'],
                    collection_date
                )
            sleep(5) # 10 Seconds for Page
    conn.close_connection()
    return None
```

Figura 26: Nova função main script Arasuper

entrada
configu
possibil
ajustar

```
class PostgreSQLConnection:
    def __init__(self, dbname, user, password, host="localhost", port=5432):
        self.dbname = dbname
        self.user = user
        self.password = password
        self.host = host
        self.port = port
        self.connection = None
        self.cursor = None

    def connect(self):
        try:
            self.connection = psycopg2.connect(
                dbname=self.dbname,
                user=self.user,
                password=self.password,
                host=self.host,
                port=self.port
            )
            self.cursor = self.connection.cursor()
            print("Connected to PostgreSQL database.")
        except psycopg2.Error as e:
            print(f"Unable to connect to the database. Error: {e}")
```

as regras de
o RDS. Essa
e na nuvem,
oi necessário
bem-sucedida
s.

Figura 25: Classe python para conectar com o postgres

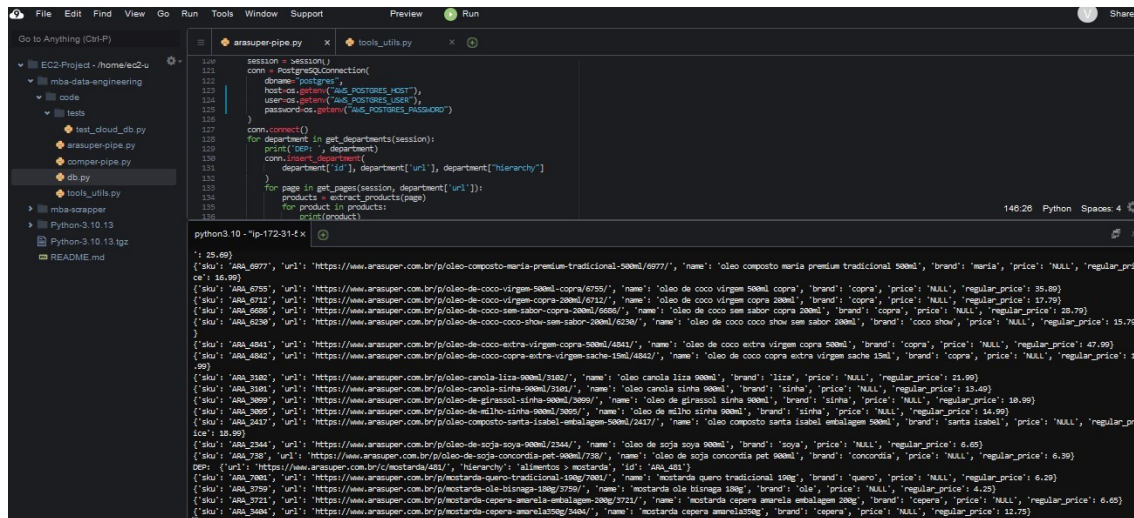
postgres - localhost:5432

- Databases
 - postgres
 - Schemas
 - public
 - Tables
 - anuncio
 - departamento
 - produto
 - supermercado

688K
56K
928K
40K

Figura 27: Postgres local

Abaixo podemos observar o código rodando dentro do EC2 inserindo dados no postgres-rds.



```

109 session = session()
110 conn = postgresql_connection(
111     dbname='postgres',
112     host=os.getenv("AWS_POSTGRES_HOST"),
113     user=os.getenv("AWS_POSTGRES_USER"),
114     password=os.getenv("AWS_POSTGRES_PASSWORD")
115 )
116
117 conn.connect()
118 for department in get_departments(session):
119     print(CEP, department)
120     department['id'], department['url'], department['hierarchy']
121
122 for page in get_pages(session, department['url']):
123     products = extract_products(page)
124     for product in products:
125         print(product)
126
127 python3.10 -i ip-172-31-1-1
128
129 ('sku': 'ARA_6977', 'url': 'https://www.arasuper.com.br/p/oleo-de-coco-virgem-tradicional-500ml/6977/', 'name': 'oleo de coco virgem tradicional 500ml', 'brand': 'maria', 'price': 'NULL', 'regular_price': 15.99)
130 ('sku': 'ARA_6755', 'url': 'https://www.arasuper.com.br/p/oleo-de-coco-virgem-copra-200ml/6755/', 'name': 'oleo de coco virgem copra 200ml', 'brand': 'copra', 'price': 'NULL', 'regular_price': 35.80)
131 ('sku': 'ARA_6712', 'url': 'https://www.arasuper.com.br/p/oleo-de-coco-virgem-copra-200ml/6712/', 'name': 'oleo de coco virgem copra 200ml', 'brand': 'copra', 'price': 'NULL', 'regular_price': 17.79)
132 ('sku': 'ARA_6886', 'url': 'https://www.arasuper.com.br/p/oleo-de-coco-sem-sabor-copra-200ml/6886/', 'name': 'oleo de coco sem sabor copra 200ml', 'brand': 'copra', 'price': 'NULL', 'regular_price': 15.79)
133 ('sku': 'ARA_6138', 'url': 'https://www.arasuper.com.br/p/oleo-de-coco-coco-show-sem-sabor-200ml/6138/', 'name': 'oleo de coco coco show sem sabor 200ml', 'brand': 'coco show', 'price': 'NULL', 'regular_price': 15.79)
134
135 ('sku': 'ARA_4841', 'url': 'https://www.arasuper.com.br/p/oleo-de-coco-extra-virgem-copra-500ml/4841/', 'name': 'oleo de coco extra virgem copra 500ml', 'brand': 'copra', 'price': 'NULL', 'regular_price': 47.99)
136 ('sku': 'ARA_4842', 'url': 'https://www.arasuper.com.br/p/oleo-de-coco-copra-extra-virgem-sache-150g/4842/', 'name': 'oleo de coco copra extra virgem sache 150g', 'brand': 'copra', 'price': 'NULL', 'regular_price': 15.99)
137
138 ('sku': 'ARA_3182', 'url': 'https://www.arasuper.com.br/p/oleo-carola-liza-900ml/3182/', 'name': 'oleo carola liza 900ml', 'brand': 'liza', 'price': 'NULL', 'regular_price': 21.99)
139 ('sku': 'ARA_3181', 'url': 'https://www.arasuper.com.br/p/oleo-carola-sinha-900ml/3181/', 'name': 'oleo carola sinha 900ml', 'brand': 'sinha', 'price': 'NULL', 'regular_price': 15.49)
140 ('sku': 'ARA_3089', 'url': 'https://www.arasuper.com.br/p/oleo-de-girassol-sinha-900ml/3089/', 'name': 'oleo de girassol sinha 900ml', 'brand': 'sinha', 'price': 'NULL', 'regular_price': 18.99)
141 ('sku': 'ARA_3085', 'url': 'https://www.arasuper.com.br/p/oleo-de-milho-sinha-900ml/3085/', 'name': 'oleo de milho sinha 900ml', 'brand': 'sinha', 'price': 'NULL', 'regular_price': 14.99)
142 ('sku': 'ARA_3417', 'url': 'https://www.arasuper.com.br/p/oleo-composto-santa-isabel-embalagem-500ml/3417/', 'name': 'oleo composto santa isabel embalagem 500ml', 'brand': 'santa isabel', 'price': 'NULL', 'regular_price': 18.99)
143
144 ('sku': 'ARA_2344', 'url': 'https://www.arasuper.com.br/p/oleo-de-soja-soja-900ml/2344/', 'name': 'oleo de soja soja 900ml', 'brand': 'soja', 'price': 'NULL', 'regular_price': 6.65)
145 ('sku': 'ARA_738', 'url': 'https://www.arasuper.com.br/p/oleo-de-soja-concordia-pet-900ml/738/', 'name': 'oleo de soja concordia pet 900ml', 'brand': 'concordia', 'price': 'NULL', 'regular_price': 6.39)
146 CEP: ('url': 'https://www.arasuper.com.br/p/mostarda-oleo', 'hierarchy': 'alimentos > mostarda', 'id': 'ARA_4818')
147 ('sku': 'ARA_7885', 'url': 'https://www.arasuper.com.br/p/mostarda-queiro-tradicional-190g/7885/', 'name': 'mostarda queiro tradicional 190g', 'brand': 'queiro', 'price': 'NULL', 'regular_price': 6.29)
148 ('sku': 'ARA_3759', 'url': 'https://www.arasuper.com.br/p/mostarda-ole-bisnaga-180g/3759/', 'name': 'mostarda ole bisnaga 180g', 'brand': 'ole', 'price': 'NULL', 'regular_price': 4.25)
149 ('sku': 'ARA_3721', 'url': 'https://www.arasuper.com.br/p/mostarda-copra-amarela-embalagem-200g/3721/', 'name': 'mostarda copra amarela embalagem 200g', 'brand': 'copra', 'price': 'NULL', 'regular_price': 6.65)
150 ('sku': 'ARA_3084', 'url': 'https://www.arasuper.com.br/p/mostarda-copra-amarela-embalagem-200g/3084/', 'name': 'mostarda copra amarela embalagem 200g', 'brand': 'copra', 'price': 'NULL', 'regular_price': 12.75)

```

Figura 29: EC2 no Ambiente cloud9

Evidência dos resultados:

Nesta sprint, alcançamos resultados significativos para a consolidação da solução proposta de coleta de dados de supermercados. Em que, conseguimos transformar as provas de conceitos da sprint 1 em um código conciso que é capaz de extrair dados dos dois mercados selecionados. Além disso, realizamos a transição desses novos scripts para uma máquina EC2 robusta, que possibilita a execução remota de código e alta escalabilidade. Também foi realizado com sucesso a conexão da instância EC2 com o banco de dados postgres no ambiente nuvem, garantindo assim um fluxo de dados inteiramente na nuvem. Em suma, nesta sprint atingimos a construção do núcleo essencial do projeto, um passo crucial rumo ao aperfeiçoamento contínuo da solução.

Abaixo podemos observar os dados dentro do postgres-rds inseridos pelo EC2 através do dbeaver. Neste exemplo construímos uma query para calcular a quantidade de produtos coletados em cada departamento com a média de preço do departamento.

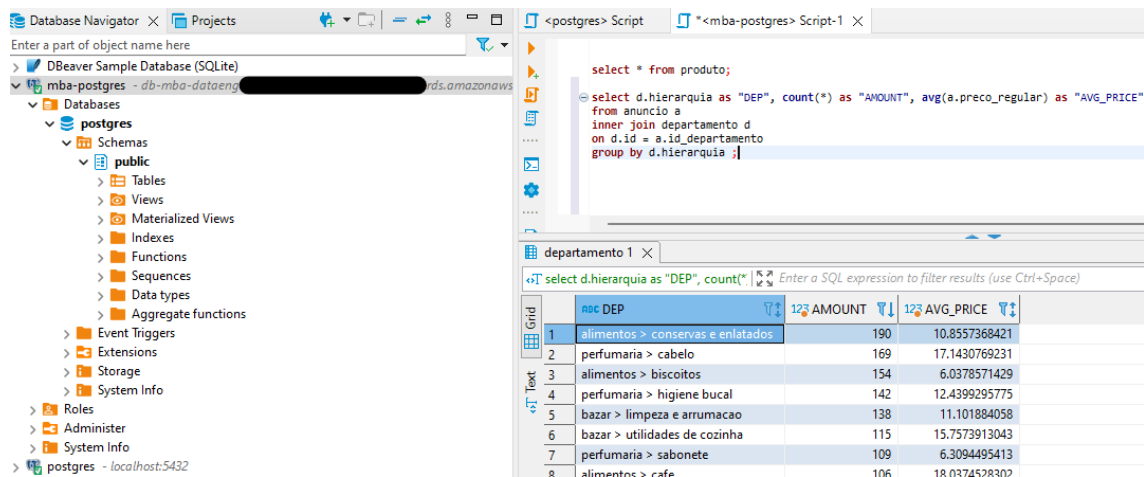


Figura 30: DBeaver query exemplo

Outra ilustração que comprova o funcionamento do sistema é o snapshot do console do EC2 interagindo com o postgres-rds, podemos o observar abaixo.



Figura 31: Console EC2 script arasuper rodando

2.2.2 Lições Aprendidas

Durante esta sprint, absorvemos valiosas lições sobre a implementação, execução e integração de scripts de web scraping em um ambiente de nuvem, conectando-se a um sistema de armazenamento de dados, como o RDS. Essa experiência proporcionou uma compreensão mais profunda do funcionamento prático da solução proposta. No entanto, mesmo com o sucesso na construção do núcleo da solução, identificamos áreas que demandam aprimoramento. Dentre os pontos observados temos, a necessidade de estabelecer um sistema robusto de monitoramento para os logs dos scripts, necessidade de implementar estratégias de gestão de instâncias para otimizar custos, entre outros.

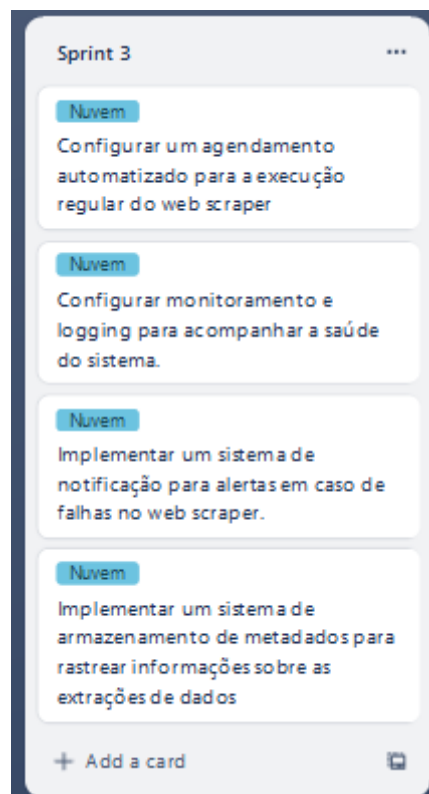
2.3 Sprint 3

Na terceira sprint, concentramos nossos esforços em automatizar a solução, implementando agendamentos para os coletores de preços de supermercados. Nosso foco foi garantir que esses coletores operassem de maneira automatizada e eficiente, sem a necessidade de intervenção humana constante. O objetivo principal desta etapa foi a automação da solução construída na sprint 2, buscando criar uma infraestrutura que permitisse o monitoramento autônomo das atividades na nuvem. Para alcançar esse objetivo, realizamos uma análise minuciosa das soluções disponíveis para automatizar a execução da solução. Implementamos registros (logs) específicos para a aplicação na nuvem e estabelecemos alertas para notificar quando intervenções de manutenção nos scripts fossem necessárias.

2.3.1 Solução

Evidência do planejamento:

Abaixo, podemos observar a evidência do planejamento da terceira sprint.



Evidência da execução de cada requisito:

3.1 Configurar um agendamento automatizado para a execução regular do web scraper

Nesta etapa, focamos na configuração de um agendamento automatizado para a execução regular dos coletores de preços de supermercados. Para isso, foi feita uma análise sobre as melhores práticas para agendamento de atividades no ambiente nuvem. Optamos por implementar o agendamento por meio de cron jobs dentro da instância EC2, reconhecendo a eficácia e facilidade que esse método oferece. Sendo o cron job, uma ferramenta padrão no ambiente Unix é uma escolha intuitiva, simples e eficiente para agendar tarefas recorrentes.

Abaixo, apresentamos os comandos cron utilizados dentro da instância EC2:

```

crontab - "ip-172-31-59-1" x
victor-dev:~/environment $ crontab -l
0 0 * * * /usr/bin/python3 /home/ec2-user/environment/project/arasuper_pipeline.py
0 2 * * * /usr/bin/python3 /home/ec2-user/environment/project/comper-pipeline.py --store_code 1
0 4 * * * /usr/bin/python3 /home/ec2-user/environment/project/comper-pipeline.py --store_code 2
0 6 * * * /usr/bin/python3 /home/ec2-user/environment/project/comper-pipeline.py --store_code 3
0 8 * * * /usr/bin/python3 /home/ec2-user/environment/project/comper-pipeline.py --store_code 4
0 10 * * * /usr/bin/python3 /home/ec2-user/environment/project/comper-pipeline.py --store_code 5
0 12 * * * /usr/bin/python3 /home/ec2-user/environment/project/comper-pipeline.py --store_code 6
victor-dev:~/environment $

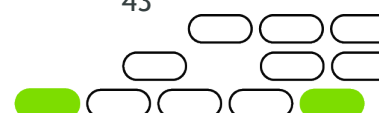
```

Figura 32: Lista de cronjobs

3.2 Configurar monitoramento da solução para acompanhar a saúde do sistema.

Nesta etapa, fizemos a implementação de um sistema de logs que proporcionasse uma visão abrangente e detalhada da execução dos scripts de coleta de preços de mercado. O objetivo principal era estabelecer um monitoramento robusto para garantir a saúde contínua do sistema. Após uma análise criteriosa das opções disponíveis no ambiente de nuvem AWS, decidimos utilizar o Amazon CloudWatch para a configuração do monitoramento que oferece uma solução integrada e eficiente para coleta, armazenamento e análise de logs. Para utilizar as funcionalidades robustas do CloudWatch, adaptamos os scripts de coleta de preços de mercado. Incorporamos informações relevantes nos logs utilizando as bibliotecas boto3 e cloudwatch do Python.

Abaixo, apresentamos alguns snapshots do log construído e seu funcionamento no CloudWatch:



```
import logging
from dotenv import load_dotenv
from datetime import datetime
from cloudwatch import cloudwatch

load_dotenv()

logger_scraper = logging.getLogger(__name__)
logger_scraper.setLevel(logging.DEBUG)

formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')

console_handler = logging.StreamHandler()
console_handler.setLevel(logging.DEBUG)
console_handler.setFormatter(formatter)
logger_scraper.addHandler(console_handler)

now = datetime.now()
current_time = now.strftime("%Y_%m_%d")
log_stream_name = f"{current_time}_logs"


# CloudWatch Handler
cloudwatch_handler = cloudwatch.CloudwatchHandler(
    region='us-east-1',
    log_group = 'scraper_logs',
    log_stream= log_stream_name,
    access_id = os.getenv("AWS_ACCESS_ID"),
    access_key = os.getenv("AWS_ACCESS_PASSWORD")
)
cloudwatch_handler.setLevel(logging.DEBUG)
cloudwatch_handler.setFormatter(formatter)
logger_scraper.addHandler(cloudwatch_handler)
```

Figura 33: Código do logger

scraper_logs

Actions View in Logs Insights Start tailing Search log group

▼ Log group details

<p>Log class - new Info</p> <p>Standard</p> <p>ARN  arn:aws:logs:us-east-1:429779957515:log-group:scraper_logs</p> <p>Creation time 3 days ago</p> <p>Retention 1 week</p> <p>Stored bytes 6 KB</p>	<p>Metric filters 1</p> <p>Subscription filters 0</p> <p>Contributor Insights rules -</p> <p>KMS key ID -</p>	<p>Anomaly detection Configure</p> <p>Data protection -</p> <p>Sensitive data count -</p>
--	---	---

Log streams Tags Anomaly detection - new Metric filters Subscription filters Contributor Insights Data protection

Log streams (3)

Filter log streams or try prefix search ☐ Exact match ☐ Show expired [Info](#)

<input type="checkbox"/>	Log stream	Last event time
<input type="checkbox"/>	2023_11_30_logs	2023-11-30 00:29:03 (UTC-05:00)
<input type="checkbox"/>	2023_11_29_logs	2023-11-29 20:57:41 (UTC-05:00)
<input type="checkbox"/>	2023_11_28_logs	2023-11-28 23:39:09 (UTC-05:00)

Figura 34: Log group com informações dos scrapers

3.3 Implementar um sistema de notificação para alertas em caso de falhas no web scraper.

Nesta etapa, desenvolvemos um sistema de notificação capaz de alertar a equipe em caso de falhas nos scripts de coleta de preços de mercado. A principal meta era estabelecer uma abordagem proativa para identificar e lidar rapidamente com quaisquer problemas que pudessem surgir durante a execução automatizada dos scripts. Optamos por utilizar o Amazon CloudWatch em conjunto com o Simple Notification Service (SNS) da AWS para a criação de alertas. Para isso, aproveitamos o log construído anteriormente, configurando monitoramento contínuo dentro do CloudWatch, analisando o log em intervalos regulares.

Nosso sistema é projetado para gerar alertas instantâneos para o SNS da AWS sempre que detecta erros de execução no log. Esses alertas são então enviados à equipe por meio de e-mails, proporcionando uma notificação imediata e direcionada para a resolução de problemas.

Abaixo podemos observar alguns snapshots do funcionamento do sistema de alerta de erros.

Details			
Name error-messages-log-alarm	State ⏸ Insufficient data	Namespace error-messages-log	Datapoints to alarm 1 out of 1
Type Metric alarm	Threshold error-messages-log >= 5 for 1 datapoints within 15 minutes	Metric name error-messages-log	Missing data treatment Treat missing data as missing
Description There are too many errors happening in the logs.	Last change 2023-11-30 03:36:38	Statistic Sum	Percentiles with low samples evaluate
	Actions 🟢 Actions enabled	Period 15 minutes	ARN arn:aws:cloudwatch:us-east-1:429779957315:alarm:error-messages-log-alarm

► View EventBridge rule

Figura 35: Alarme cloudwatch para detecção de erros no log

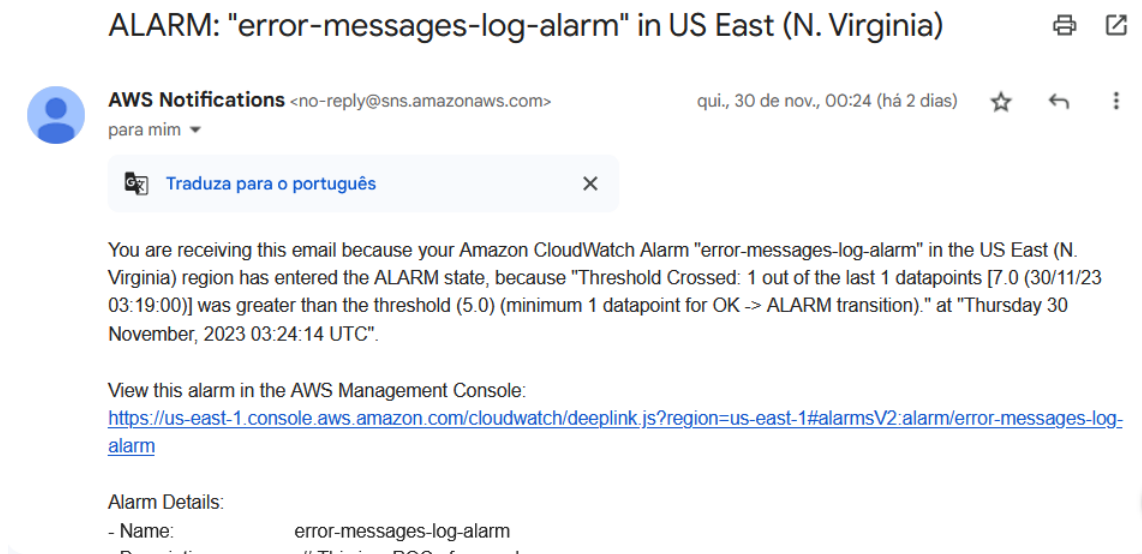


Figura 36: Exemplo de email enviado caso erro encontrado em logs

3.4 Implementar um sistema de armazenamento de metadados para rastrear informações sobre as extrações de dados

Nesta etapa, implementamos uma solução para armazenar metadados cruciais relacionados às execuções dos scripts de coleta de produtos de supermercados. Reconhecendo a importância de rastrear informações sobre cada extração de dados, buscamos e implementamos uma abordagem eficiente para esse propósito. Dentre as opções consideradas para o armazenamento de metadados, optamos por utilizar outro sistema de log específico para registrar exclusivamente essas informações. Essa abordagem se mostrou simples e direta, proporcionando um meio eficaz de armazenar dados essenciais sobre cada execução dos scripts.

Abaixo podemos observar snapshot referentes a construção desse sistema de armazenamento de metadados

```
import os
import logging
from dotenv import load_dotenv
from datetime import datetime
from cloudwatch import cloudwatch

load_dotenv()

logging.addLevelName(25, "METADATA")

logger_metadata = logging.getLogger(__name__)
logger_metadata.setLevel(logging.DEBUG)

formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')

console_handler = logging.StreamHandler()
console_handler.setLevel(logging.DEBUG)
console_handler.setFormatter(formatter)
logger_metadata.addHandler(console_handler)

now = datetime.now()
current_time = now.strftime("%Y_%m")
log_stream_name = f"{current_time}_metadata"

cloudwatch_handler_metadata = cloudwatch.CloudwatchHandler(
    region='us-east-1',
    log_group = 'scraper_metadata',
    log_stream= log_stream_name,
    access_id = os.getenv("AWS_ACCESS_ID"),
    access_key = os.getenv("AWS_ACCESS_PASSWORD")
)
cloudwatch_handler_metadata.setLevel(logging.DEBUG)
cloudwatch_handler_metadata.setFormatter(formatter)
logger_metadata.addHandler(cloudwatch_handler_metadata)
```

Figura 37: Código logger metadados

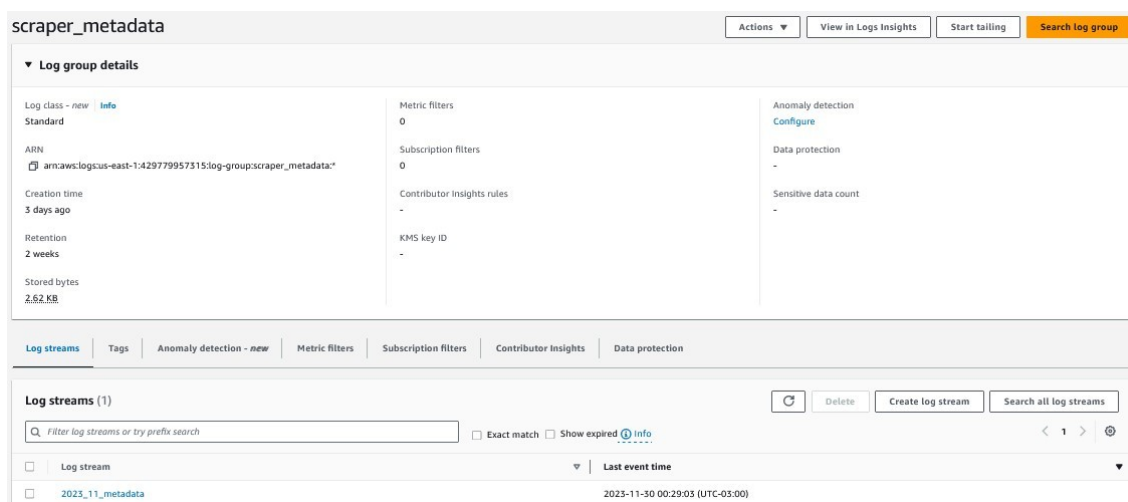


Figura 38: Log group para armazenamento de metadados

Evidência dos resultados:

Na terceira sprint, Conseguimos com sucesso implementar a automação da execução dos scripts responsáveis pela coleta de dados de produtos de supermercados. Através da configuração de agendamentos automatizados, os coletores agora operam de forma regular e independente, assegurando a continuidade da coleta de informações sem intervenção manual. Além disso, introduzimos um sistema de log abrangente que armazena detalhadamente todos os eventos durante a execução dos scripts. Além do mais, também implementamos um sistema de logs focado para o armazenamento de metadados das execuções dos scripts e no que diz respeito ao monitoramento do projeto, conseguimos implementar um sistema eficaz que gera alertas via e-mail sempre que ocorrem erros nos logs.

Abaixo, apresentamos resultados visuais dos logs, evidenciando que os scripts estão operando nos horários programados e gerando os valores esperados para a solução.

Log events
You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Timestamp	Message
No older events at this moment. Retry	
2023-12-01T23:22:40,877-03:00	2023-12-01 23:22:40,876 - INFO - Starting ARASUPER SCRAPER
2023-12-01T23:22:42,392-03:00	2023-12-01 23:22:42,391 - INFO - Connecting to DB
2023-12-01T23:22:45,448-03:00	2023-12-01 23:22:45,447 - DEBUG - Inserting Department https://www.anasuper.com.br/c/massas-resfriadas/500/
2023-12-01T23:22:47,025-03:00	2023-12-01 23:22:47,025 - INFO - Inserting Product : {'sku': 'ARA_16331', 'url': 'https://www.anasuper.com.br/p/massa-de-pastel-grande-romanha-500g/16331/', 'name': 'm...
2023-12-01T23:22:48,139-03:00	2023-12-01 23:22:48,139 - INFO - Inserting Product : {'sku': 'ARA_16324', 'url': 'https://www.anasuper.com.br/p/massa-de-lasanha-romanha-500g/16324/', 'name': 'm...
2023-12-01T23:22:50,204-03:00	2023-12-01 23:22:50,203 - DEBUG - Inserting Department https://www.anasuper.com.br/c/torradas/562/
2023-12-01T23:22:51,932-03:00	2023-12-01 23:22:51,928 - INFO - Inserting Product : {'sku': 'ARA_1871', 'url': 'https://www.anasuper.com.br/p/torrada-marian-magic-toast-original-150g/1871/', ...
2023-12-01T23:22:52,677-03:00	2023-12-01 23:22:52,676 - INFO - Inserting Product : {'sku': 'ARA_1873', 'url': 'https://www.anasuper.com.br/p/torrada-bauducco-tradicional-204g/1873/', 'name': '...
2023-12-01T23:22:53,423-03:00	2023-12-01 23:22:53,421 - INFO - Inserting Product : {'sku': 'ARA_1876', 'url': 'https://www.anasuper.com.br/p/torrada-bauducco-tradicional-142g/1876/', 'name': '...
2023-12-01T23:22:54,192-03:00	2023-12-01 23:22:54,188 - INFO - Inserting Product : {'sku': 'ARA_1874', 'url': 'https://www.anasuper.com.br/p/torrada-bauducco-multigrãos-284g/1874/', 'name': '...
2023-12-01T23:22:55,009-03:00	2023-12-01 23:22:55,004 - INFO - Inserting Product : {'sku': 'ARA_1875', 'url': 'https://www.anasuper.com.br/p/torrada-bauducco-integral-204g/1875/', 'name': 'to...
2023-12-01T23:22:55,756-03:00	2023-12-01 23:22:55,751 - INFO - Inserting Product : {'sku': 'ARA_1877', 'url': 'https://www.anasuper.com.br/p/torrada-bauducco-multigrãos-142g/1877/', 'name': '...

Figura 39: Histórico de eventos do log dos coletores

Log events
You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Timestamp	Message
No older events at this moment. Retry	
2023-12-01T00:47:57,838-03:00	2023-12-01 00:47:57,836 - METADATA - https://www.anasuper.com.br/, Departments : 143, Products : 5162, RunTime: 1 hours : 25 minutes : 16 seconds

Figura 40: Histórico de logs de metadados de coletores

2.3.2 Lições Aprendidas

Durante a terceira sprint, acumulamos valiosas lições que contribuíram significativamente para o aprimoramento contínuo da nossa solução de coleta de dados em supermercados. Conseguimos automatizar com sucesso a execução da solução no ambiente de nuvem, implementamos sistemas de logs para armazenar os acontecimentos dentro das instâncias EC2 e desenvolvemos um sistema de armazenamento para os metadados da solução. Além disso, também conseguimos implementar com êxito alertas no ambiente de nuvem que interagem com os e-mails dos colaboradores. Reconhecemos, contudo, que ainda há diversas possíveis melhorias e oportunidades de refinamento contínuo para a solução desenvolvida.

3. Considerações Finais

Nesta etapa final do relatório, serão apresentadas as reflexões e aprendizados obtidos ao longo do projeto. Além disso, serão explicados os resultados finais, as contribuições da solução para empresa X em comparação aos seus sistemas legados e por fim os próximos passos sugeridos para a solução.

3.1 Resultados

Como resultados finais do projeto com a solução completa, verifica-se o seguinte:

- Implementamos com sucesso a coleta autônoma de dados provenientes de diferentes supermercados, garantindo um amplo escopo de informações.
- Desenvolvemos um sistema automatizado, desde a extração de dados até seu armazenamento, eliminando a necessidade de processos manuais e interação humana.
- Criamos um sistema abrangente de logs e metadados, permitindo um rastreamento detalhado das atividades durante a execução dos coletores.
- Estabelecemos um banco de dados consolidado, capaz de armazenar e organizar os dados coletados, proporcionando uma base sólida para análises futuras e insights.
- Implementamos um sistema de agendamento eficaz, assegurando a execução regular e autônoma dos coletores de preços de mercado.
- Implementamos a solução completamente no ambiente nuvem. Sendo assim, garantimos escalabilidade, flexibilidade e disponibilidade dos dados armazenados.

3.2 Contribuições

A implementação da nova infraestrutura trouxe contribuições significativas para a empresa X, elevando substancialmente a disponibilidade e eficiência das informações sobre preços de produtos em supermercados. O que anteriormente era um processo manual, repleto de obstáculos e dados incompletos, agora se transformou em um procedimento ágil e confiável para geração de relatórios e insights sobre o mercado brasileiro.

Dentre as principais contribuições do novo sistema, destaca-se a introdução do agendamento automatizado para a coleta de dados, em que dados são obtidos de maneira regular e consistente, garantindo a qualidade e completude dos dados coletados. Além disso, a consolidação dos dados em uma base de dados relacional



única proporciona uma visão holística e unificada do mercado brasileiro, o que por conseguinte facilita a geração de relatórios e insights. Outro benefício relevante da implementação da infraestrutura é a capacidade de iniciar novos projetos a partir da base de dados gerada, ampliando assim, as possibilidades de inovação e desenvolvimento dentro da empresa.

3.3 Próximos passos

Abaixo temos os possíveis próximos passos para a solução apresentada neste projeto, são eles:

- Expandir a capacidade dos scripts para incluir uma gama mais abrangente de supermercados
- Desenvolver dashboards interativos no PowerBi para fornecer à equipe de análise uma visualização mais intuitiva
- Investigar a possibilidade de migrar os coletores para um modelo de processamento serverless (AWS lambda).
- Refinar os mecanismos de gestão de erros nos scripts, aprimorando a detecção e tratamento de falhas.
- Adicionar views específicas na base de dados para simplificar consultas e análises por parte da equipe de analytics.
- Investigar a viabilidade de incorporar sistemas de orquestração como Apache Airflow ou Dagster para otimizar o agendamento e a execução de tarefas.

