

Faculdade

XPe



RELATÓRIO

PROJETO
APLICADO

PÓS-GRADUAÇÃO

XP Educação
Relatório do Projeto Aplicado

Automação e validação de transferência de database

Diego Fernando Paranhos

Orientador(a): Daniel Viana

13/02/2024



NOME DO ALUNO

XP EDUCAÇÃO

RELATÓRIO DO PROJETO APLICADO

Automação e validação de transferência de database

Relatório de Projeto Aplicado
desenvolvido para fins de conclusão do
curso Data Science & Machine Learning.

Orientador (a): Daniel Viana

Valinhos - SP

13/02/2024



Sumário

1. CANVAS do Projeto Aplicado	5
1.1 Desafio	5
1.1.1 Análise de Contexto	5
1.1.2 Personas	8
1.1.3 Benefícios e Justificativas	9
1.1.4 Hipóteses	11
1.2 Solução	13
1.2.1 Objetivo SMART	13
1.2.2 Premissas e Restrições	13
1.2.3 Backlog de Produto	14
2. Área de Experimentação	16
2.1 Sprint 1	16
2.1.1 Solução	16
• Evidência do planejamento:	16
• Evidência da execução de cada requisito:	17
• Evidência dos resultados:	19
2.1.2 Lições Aprendidas	21
2.2 Sprint 2	22
2.2.1 Solução	22
• Evidência do planejamento:	22
• Evidência da execução de cada requisito:	23
• Evidência dos resultados:	28
2.2.2 Lições Aprendidas	30
2.3 Sprint 3	31
2.3.1 Solução	31
• Evidência do planejamento:	31
• Evidência da execução de cada requisito:	31
• Evidência dos resultados:	31
2.3.2 Lições Aprendidas	31
3. Considerações Finais	32
3.1 Resultados	32
3.2 Contribuições	32
3.3 Próximos passos	32



1. CANVAS do Projeto Aplicado

Este projeto aplicado compreende as etapas definidas na disciplina Projeto Final do Curso de pós-graduação *Data Science & Machine Learning* da XP Educação.

A figura 1 contém o CANVAS do Projeto Aplicado com os passos sumarizados da execução do projeto.

Figura 1 - Canvas do Projeto Aplicado



1.1 Desafio

1.1.1 Análise de Contexto

A utilização de armazenamento de informações em tabelas e banco de dados transacionais são usados desde o começo da implementação dos recursos computacionais nas empresas, indústria, unidades de ensino e pesquisa.

Com a evolução da informática e a capacidade de geração e armazenamento de dados pelos computadores modernos e recursos em nuvem, cada vez mais a utilização dos bancos transacionais tradicionais não são mais funcional, sendo necessário a busca de novas soluções de armazenamento e estrutura de dados como

processamento compartilhado, estruturação dos *Warehouse* em nuvem, utilização de dados não estruturados.

Porém muitas empresas e organizações já possuíam um sistema robusto de governança de dados, armazenando os mesmos em bancos de dados transacionais, sendo consultados e utilizados via consulta em SQL e disponibilizado de forma visual para os usuários através de ferramentas como Power BI e Tableau.

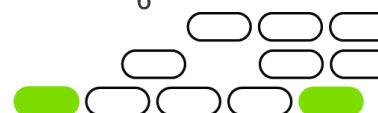
Neste contexto, se faz necessário, para adequação com as arquiteturas de dados atuais, a migração desses bancos de dados transacionais para sistema em nuvem e armazenamento em *data lake* dos dados gerado em BIG DATA.

Uma das arquiteturas mais comuns para essa migração é construir um ecossistema onde as fontes de dados sejam conectadas em um sistema de *data lake*, como exemplo AWS ou Azure, integrado com o Snowflake para disponibilizar os dados para os engenheiros de dados, analistas, arquitetos e desenvolvedores de BI.

Para disponibilização aos usuários finais, os desenvolvedores de BI necessitam conectar essas novas tabelas nos modelos semânticos de Power BI, sem a perda de dados ou confiabilidade do mesmo. Em muitos casos as *VIEW* utilizadas no sistema LEGADO em SQL não estão em esquema estrela, com uma tabela fato contendo somente os dados principais e as tabelas dimensões com as estruturas descritivas utilizadas para compreensão dos dados principais.

É neste contexto que se faz a necessidade do processo de validação das novas tabelas, para verificar se elas possuem os mesmos dados, se não ocorreu mudança de dados históricos, se existem todas as chaves primária e estrangeiras, se não existem dados nulos ou em branco ou dados inconsistentes. Outro ponto muito importante é que na migração, normalmente é criada a estrutura de tabelas e *Views* utilizando a normalização dos dados em esquema estrela e isso gera a necessidade de um estudo detalhado pelo desenvolvedor de BI ou engenheiro de dados para garantir todas as relações necessárias.

O processo de validação feito de forma manual é muito lento e devido a sua complexidade, pode não garantir a confiabilidade 100% dos dados, gerando análises incorretas pelos usuários finais, retrabalho depois de liberado no ambiente produtivo e até quebra do serviço em casos mais críticos. Ainda temos que levar em consideração que normalmente a migração de sistema contém N tabelas e o volume é grande, gerando um alto lead time de entrega e alta carga de hora/homem.



Este trabalho visa criar uma solução para essa dor, criando um algoritmo de automação da validação destas tabelas migradas, gerando um processo estável, padronizado e de rápida execução e obtenção dos resultados finais.

Neste projeto, o nome da empresa, atores serão nomes fantasias e os dados utilizados serão emulados, utilizando a estrutura similar de colunas e informações, porém dados gerados aleatoriamente para preservar a confidencialidade.

O contexto utilizando neste projeto é uma empresa do ramo automotivo que está realizando a migração do seu *Warehouse* em sistema legado, baseado em dados estruturas e Views feitas em SQL Studio e disponibilizadas em Power BI para os usuários para um sistema baseado em Azure integrado com Snowflake utilizando a arquitetura de RAW, GOLD e MART, onde as tabelas fato e dimensão estão liberadas para conexão nos modelos semânticos existentes em Power BI.

A fase inicial do processo de automação será realizada utilizando base de dados em CSV coletados nas tabelas do sistema legado e Snowflake para carregamento em VS Code e análise em Python, para fins deste projeto o escopo de validação se concentrará em dados de uma fato de unidades vendidas, com dimensões de cliente, produto, data e cliente.

A figura 2 contém a Matrix CSD de análise inicial do cenário, pessoas e regras e suas interações. Na figura 3 temos o POEMS

Figura 2 - Matrix CSD (Fonte: Template - Canvas Projeto Aplicado Pós - XP Educação)

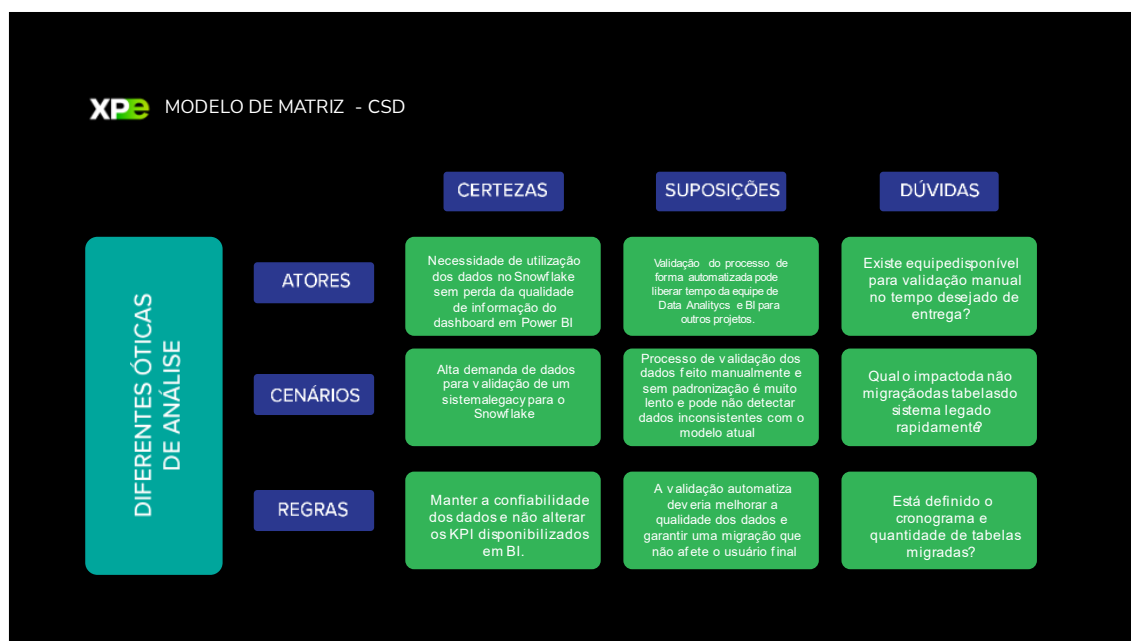


Figura 3 - POEMS (Fonte: Template - Canvas Projeto Aplicado Pós - XP Educação)



1.1.2 Personas

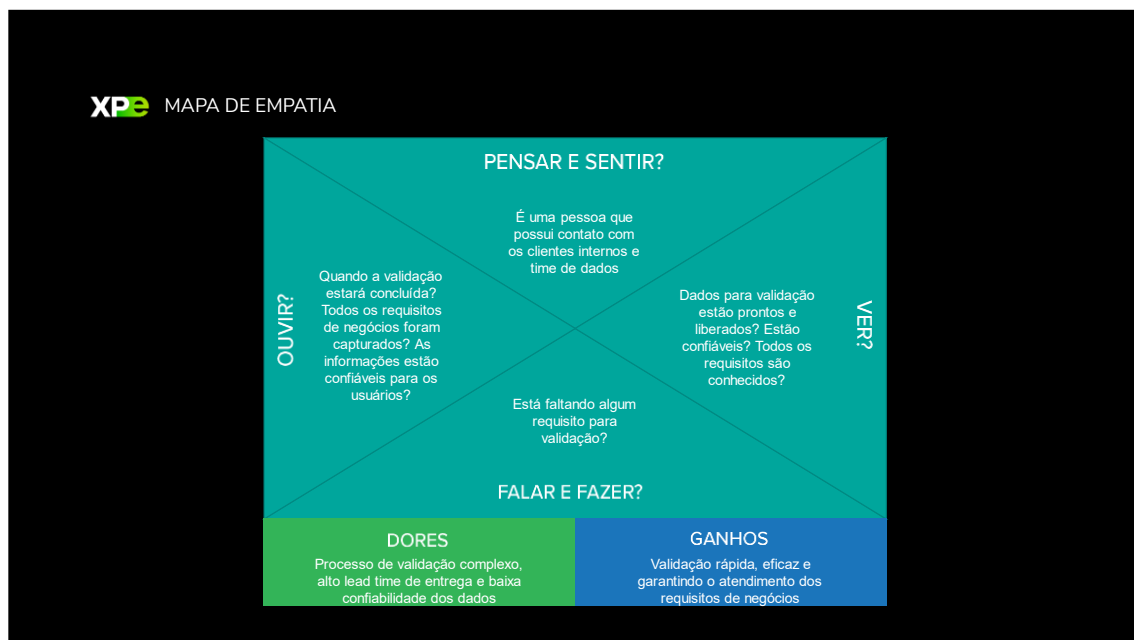
Neste projeto de automação e validação do processo de migração de database, as personas envolvidas são:

- Desenvolvedores de BI: Esses profissionais são os mais afetados e atuantes neste processo de validação, pois são os responsáveis em validar os dados, substituir nos modelos semânticos as conexões e liberar para os usuários finais no ambiente produtivo.
- Product Owner: São os responsáveis por validar o processo de migração, realizar a gestão das entregas e conexão do time de desenvolvedores e negócios e a definição dos requerimentos.
- Arquiteto de dados: Essa pessoa é a responsável por definir a estrutura e arquitetura dos dados no Snowflake, necessita ser alimentado pelo Product Owner dos requisitos de negócios e pelos desenvolvedores de BI das tabelas LEGADO e resultado das validações para aprovação e liberação da migração ou adequações necessárias.
- Engenheiro de dados: É a pessoa que operacionaliza a arquitetura e migração dos dados nas tabelas RAW, GOLD e MART no Snowflake, trabalha muito próxima ao arquiteto de dados e desenvolvedor de BI.

- Usuário final e alta gestão: São as pessoas que vão consumir os dashboard e KPI finais, necessitam receber os dados conforme regras de negócios definidas, no tempo certo e com a confiabilidade alta para tomada de decisões, são as pessoas mais afetadas caso a validação atrase ou seja feita de forma imprecisa.

Com base nesse atores, a figura 4 contém o mapa de empatia do persona “Product Owner” que é o principal cliente deste processo de validação, pois é a pessoa que necessita garantir a aprovação final para liberação aos usuários finais e garantir a realização das atividades conforme cronograma utilizando a metodologia AGILE via SPRINTs.

Figura 4 - Mapa de Empatia



1.1.3 Benefícios e Justificativas

Este projeto tem como benefícios principais para os atores do processo e empresa estabelecer uma rotina automatizada com ganho de velocidade e acuracidade no processo de validação das tabelas migradas para o Snowflake, para conexão nos modelos semânticos do Power BI utilizado pela gestão.

Com a implementação deste projeto, a área de BI poderá reduzir a quantidade desenvolvedores envolvido no processo de validação, disponibilizando esses recursos



para criação de novos dashboard e trabalho com as áreas de clientes de negócio. Eliminar um gargalo no processo de migração do LEGADO para o Snowflake que é a entrega das validações com acurácia na qualidade dos dados.

Para validação desses benefícios e justificada do projeto, foi realizado uma análise no contexto utilizando as ferramentas Business Design Blueprint (Tabela 1).

Tabela 1 - Business Design Blueprint.

Itens	Detalhamento
Objetivos	Automação do processo de validação de migração de database
Atividades	Validar as tabelas migradas do sistema legado para Snowflake e conectar no modelo semântico
Questões	Quais testes são necessários para considerar uma tabela validada e aprovada?
Barreira	Liberação das tabelas no Snowflake pelo time de dados para validação
	Ações do cliente
Funcionalidades	Script para realizar testes comparativos para identificar de forma padronizada e automatizada possíveis bug e inconsistência de dados
Interação	Interação entre os dados liberados pelo time de data analytics e clientes usuários dos dashboard e KPI
Mensagem	Informação para equipe de data analytics dos bugs detectados ou aprovação e para o Product owner realizar a validação final e liberação para usuários.
Onde ocorre	Na empresa a nível global ou rede de clientes
Tarefas pendentes	Otimização dos modelos semânticos com base na nova arquitetura de dados
Tarefas escondidas	Revisão de regras de negócios já estabelecidas nas consultas utilizadas no sistema legado
Processos de suporte	Dataset em Power BI para confirmação dos dados
Saída desejável	Substituição das tabelas nos modelos semânticos de Power BI pelas fontes em Snowflake sem perda de informação ou mudança de KPI para os usuários

Abaixo temos o Canvas Proposta de valor (Figura 5), com a validação e análise da proposta de valor deste projeto, onde podemos evidenciar que a proposta de validação e automação do processo de teste de tabelas de migração de database irá criar valor ao processo da empresa.

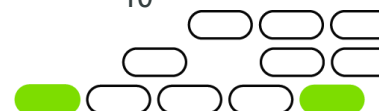
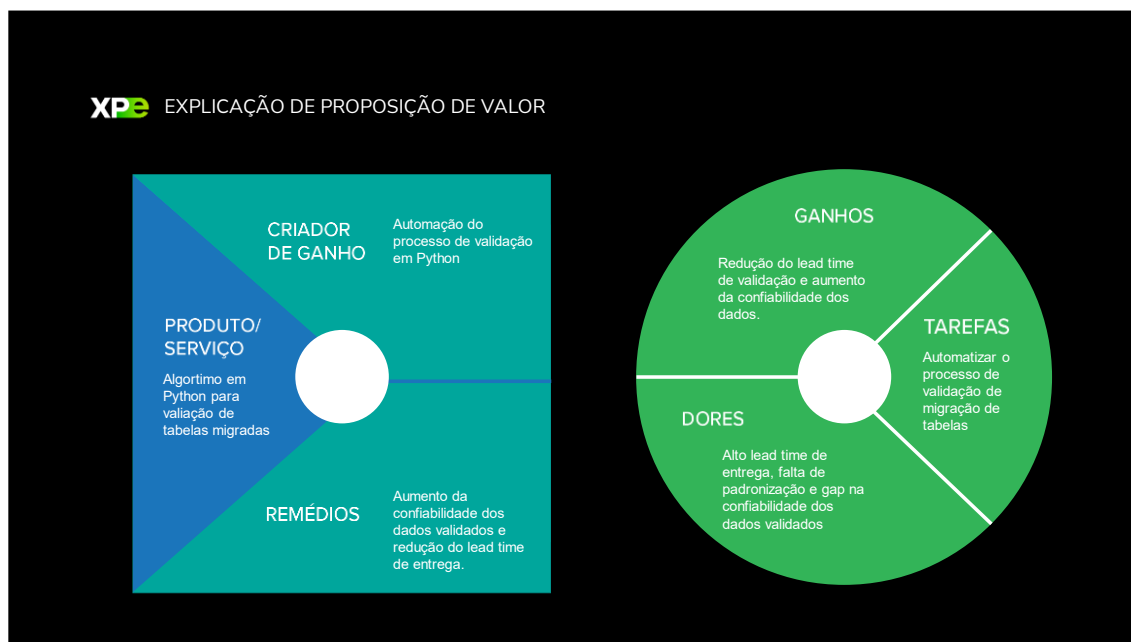


Figura 5 - CANVAS de Proposta de Valor.



1.1.4 Hipóteses

Como principal hipótese deste projeto, a automação permitirá um ciclo de validação menor, gerando mais velocidade e reduzindo custos operacionais, também criará um processo mais estável com maior confiabilidade nos dados.

Porém dentro deste cenário, podemos ter outras hipóteses e observações que podem impactar diretamente na entrega e qualidade do projeto. Para uma análise mais detalhada e profunda as observações e hipóteses, foi utilizado a tabela 2, onde foi levantando as principais observações no processo atual e suas possíveis consequências.

Tabela 2 - Tabela de Observações e Hipóteses.

Observação	Hipótese
Validação das tabelas podem demorar mais que o previsto	Estrutura de dados disponibilizados não está no mesmo padrão que o LEGADO
Dados das tabelas migradas para o Snowflake possuem valores diferentes	Possível diferença na regra de negócio utilizada
Tabela no snowflake com muito valores NULL ou BLANK	Quebra da correlação entre a tabela FATO e Dimensão
Tabelas já migradas e conectadas no modelo semântico gerando bug nos reports	Possível falha na validação que não conseguiu capturar todos os cenários e não garantiu a acuracidade dos dados.

Atraso nas entregas de validação dentro da SPRINT	falta de recurso para realização do processo de validação manual
Mudança de priorização das migrações	Se a validação tem um alto lead time, se ocorrer alguma mudança de priorização o processo pode ficar parado na metade.

Com a hipóteses expostas e todo o contexto do problema analisado, iniciamos o levantamento das ideias para solucionar os principais problemas a serem resolvidos através de um Brainstorm para coleta das possíveis soluções.

Para priorizar as ideias levantadas de solução, foi utilizado uma Matriz de Priorização seguindo a pontuação demonstrada na Tabela 3.

Tabela 3 - Balizadores para notas da Matriz BASICO.

Escala	B - Benefícios	A - Abrangência	S - Satisfação	I - Investimentos	C - Cliente	O - Operacionalidade
5	De vital importância	Total (de 70 a 100%)	Muito grande	Pouquíssimo investimento	Nenhum impacto	Muito fácil
4	Significativo	Muito grande (de 40 a 70%)	Grande	Algum investimento	Impacto pequeno	Fácil
3	Razoável	Razoável (de 20 a 40%)	Média	Médio investimento	Médio impacto	Média facilidade
2	Poucos benefícios	Pequena (de 5 a 20%)	Pequena	Alto investimento	Impacto grande	Difícil
1	Algum benefício	Muito pequena	Quase não é notada	Altíssimo investimento	Impacto muito grande no cliente	Muito difícil

Fonte: <https://melhorianapratica.com.br/matriz-de-priorizacao-de-projetos/>.

Acesso em 16/02/2024.

Tabela 4 - Priorização das ideias.

SOLUÇÕES	B	A	S	I	C	O	Total	Priorização
Criar processo automatizado de validação	5	5	5	5	5	3	28	1
Criação de um algoritmo para padronização do processo de validação	4	4	4	5	4	5	26	2
Contração de mais desenvolvedores de BI	4	5	3	1	5	3	21	3
Contratar consultoria externa para validação	4	5	3	1	5	3	21	4
Refazer 100% os modelos semânticos com as novas tabelas do Snowflake	2	5	3	2	1	1	14	5



1.2 Solução

1.2.1 Objetivo SMART

O objetivo SMART deste projeto é criar um algoritmo em Python para automação e comparação de duas tabelas (Legado vs Snowflake) e retornar como saída se a nova tabela contém as mesmas estruturas de dados para substituição sem impacto ao cliente final no modelo semântico do Power BI.

Esta meta é específica, pois limita o escopo em validação e automação, mensurável, pois estabelece como saída da atividade a não alteração e perdas de dados, atingível pois estabelece um processo que pode ser obtido com as ferramentas e informações disponíveis em mercado e dentro da empresa. Relevante, pois é um processo crítico para empresa e irá eliminar um gargalo operacional, e tangível.

1.2.2 Premissas e Restrições

Premissas do projeto:

- Migração dos dados;
- Utilização do Python para criação do script;
- Conexão das novas tabelas no modelo semântico do PBI de forma que não impacte os usuários;
- 1 hora por dia de disponibilidade para desenvolvimento do algoritmo;
- Utilização de dados emulados para desenvolvimento da ferramenta;
- Garantir a acuracidade dos dados liberados aos clientes.

Restrições do projeto:

- Não usar os dados do ambiente produtivo para garantir a confidencialidade das informações;
- Somente utilizar os dados liberados nos ambientes GOLD e MART do Snowflake;
- Não gerar custo extra



Um ponto importante em todo projeto é analisar e gerenciar os riscos que podem impactar dentro do projeto para a sua entrega. Além de definir as premissas e restrições, precisamos também garantir que todo potencial risco seja mitigado. Para isso é utilizado uma matriz de risco.

Na Tabela 5, temos a matriz levantada neste projeto, com o mapeamento dos potenciais risco, possível impacto e plano de ação para mitigar seu impacto.

Tabela 5 - Matriz de Risco

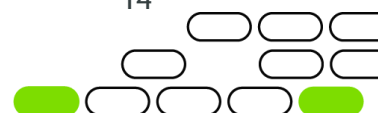
Riscos identificados	Impacto potencial	Ações preventivas	Ações Corretivas
Limite técnico durante a programação do script	Código da automação do processo não entregue	Analisar se o Pandas contém todas as funcionalidades necessárias	Buscar bibliotecas com maior recurso de validação para utilização no Python, caso o Pandas seja um limitante.
Falta de correlação no modelo atual para comparação com as tabelas legado	Impossibilidade de comparação e validação	Realizar uma pré verificação das tabelas	Durante a fase de definição da arquitetura, ter um processo de validação das chaves necessárias
Criação de bugs nos KPI liberados	perda de serviço, tomada de decisão errada pelo usuário	Voltar para o backup da versão anterior	Realizar validações e duplo check antes da liberação para produtivo
Baixa qualidade do código gerado	Geração de bugs e validação incorreta	Volta para validação manual.	Revisão e validação robusta do código antes do uso

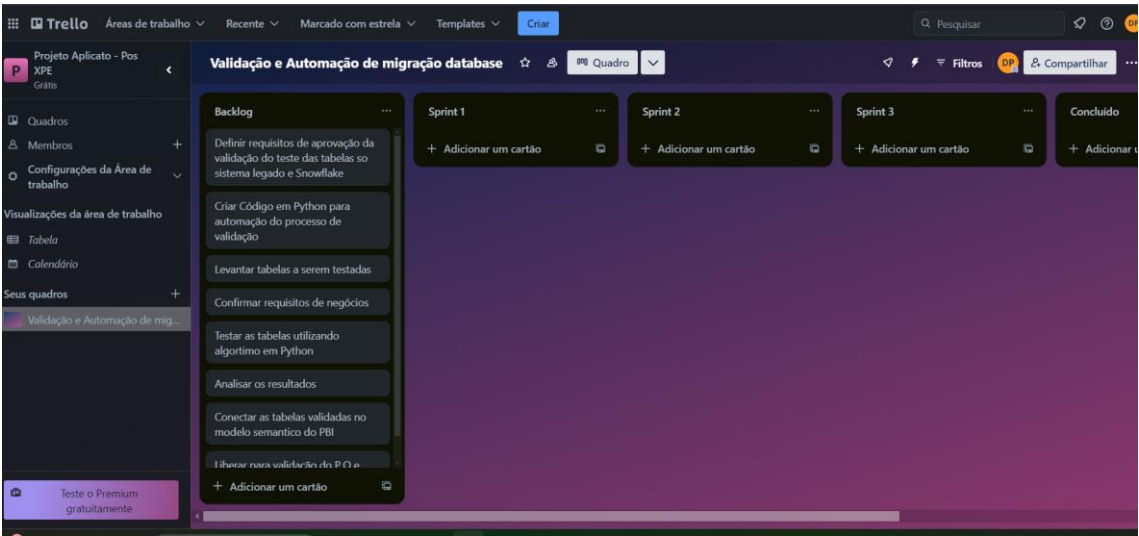
1.2.3 Backlog de Produto

Este projeto será realizado com as seguintes atividades abaixo no backlog do produto, a gestão das atividades será feita no Trello, utilizando a metodologia ágil dividido em 3 Sprints:

Figura 6 - Captura de tela do backlog no Trello

(<https://trello.com/b/EGqBH0Eq/valida%C3%A7%C3%A3o-e-automa%C3%A7%C3%A3o-de-migra%C3%A7%C3%A3o-database>)





2. Área de Experimentação

2.1 Sprint 1

2.1.1 Solução

- Evidência do planejamento:

Para o Sprint 1 do projeto, foi programado atividades:

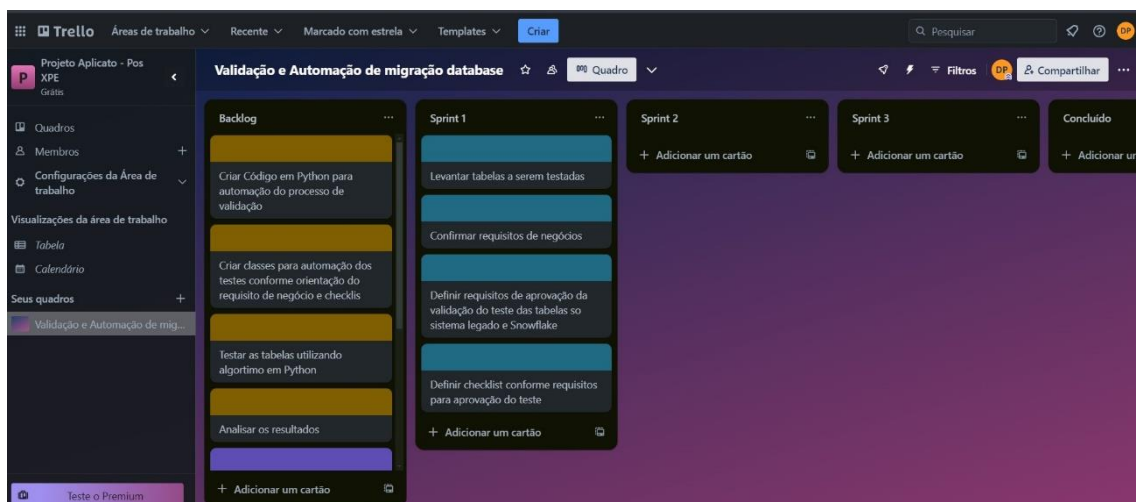
- Levantamento das tabelas a serem testadas
- Confirmação dos requisitos de negócios
- Levantamento dos requisitos de validação e regras de negócio:
- Definir checklist conforme requisitos para aprovação dos testes:

Inicialmente estava programando para este Sprint o início do desenvolvimento do código em python para realização do teste, porém foi analisado a necessidade de definição primeiramente das regras de negócios, checklist e tabelas a serem testadas antes do inicio da definição do script em python.

O controle das atividades e gestão do projeto está utilizando a ferramenta Trello com base na metodologia Agil / SCRUM.

Figura 6 - Captura de tela do backlog no Trello

(<https://trello.com/b/EGqBH0Eq/valida%C3%A7%C3%A3o-e-automa%C3%A7%C3%A3o-de-migra%C3%A7%C3%A3o-database>)



- Evidência da execução de cada requisito:

Para iniciar este Sprint, foi realizado um mapeamento e levantamento das tabelas a serem validadas junto ao Product Owner do time de Data & Analytics, que são usadas nos Super Reports em Power BI.

Neste mapeamento foi identificado 56 tabelas necessárias para migração, tratando dos seguintes tópicos sumarizado:

- Como tabelas fato:
 - Vendas de veículos para concessionárias
 - Vendas de veículos para cliente final
 - Vendas de peças e acessórios para concessionárias
 - Forecast de vendas
 - Inventário nas redes de concessionárias
 - Inventário nas unidades fabris e logística
- Como tabelas dimensão:
 - Cadastro de região
 - Cadastro de concessórias
 - Dados das plantas fabris
 - Dados dos times internos
 - Tabela DATE
 - Tabela Produto
 - Tabela Cliente

Para uso neste projeto aplicado e proteção dos dados confidenciais da empresa, será usado a fim de estudo e validação da metodologia, dados emulados dos temas acima, retirados de banco de dados abertos na internet.

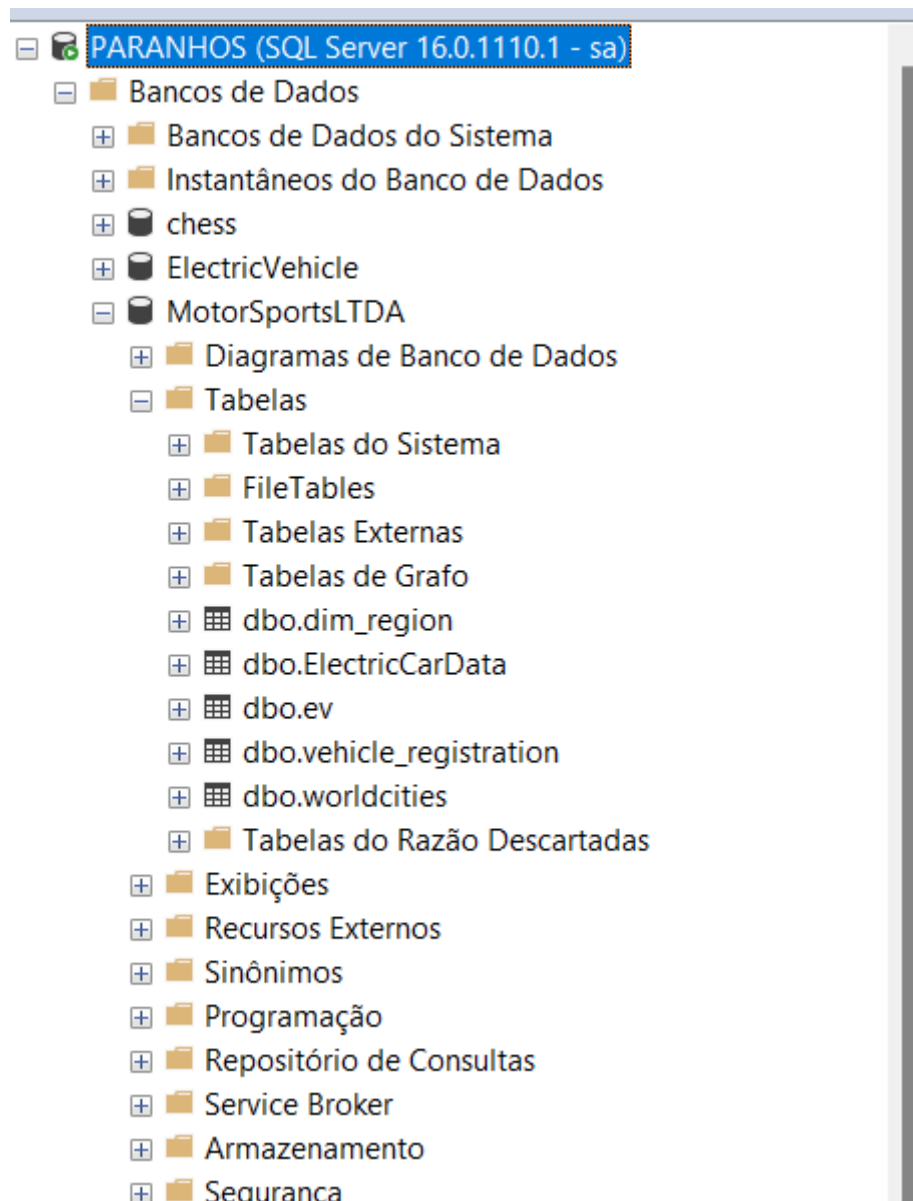
Para o estudo e validação do algoritmo de teste, será utilizado as seguintes tabelas:

- Vehicle registration - registro de vendas na América do Norte de veículos Snowmobile e boat -
<https://www.kaggle.com/datasets/new-york-state/nys-vehicle,-snowmobile,-and-boat-registrations>
- EV - registro de vendas de veículos elétricos -
https://www.kaggle.com/datasets/geoffnel/evs-one-electric-vehicle-dataset?select=ElectricCarData_Norm.csv
- Wordcities e dim_region - tabela dimensão das cidades -
<https://www.kaggle.com/datasets/max-mind/world-cities-database?select=worldcitiespop.csv>
- EletricCarData - tabela de produtos de veículos elétricos fabricados

Na figura 7, temos a evidência da estrutura de dados no legado, no SQL Management Studio. 19, neste estudo a database utilizada é a MotorSportsLTDA.



Figura 7 - Estrutura da base de dados no SQL.



Após a definição das tabelas e estrutura dos dados, foi realizado a etapa de confirmação das regras de negócio a serem aplicadas para validação das tabelas migradas.

Após reunião com os Analista de negócios responsável pela região North America e Internacional (resto dos países, excluindo USA e Canada), ficou definido que para aprovação das tabelas, os dados no Snowflake devem seguir os mesmos filtros e composição de dados igual aos aplicados no sistema legado nas VIEW utilizadas para consumo dos dados.

Abaixo um exemplo da regra que deverá ser seguida para a view 'ev_sedan', figura 8.

Figura 8 - Exemplo da regra de negócio a ser usada para validação, clausura WHERE.

```
SQLQuery5.sql - PA...portsLTDA (sa (64))* SQLQuery3.sql - PA...portsLTDA (sa (66)) SQLQuery2.sql
create view ev_sedan
as
select *
from MotorSportsLTDA.dbo.ElectricCarData as prod
where prod.BodyStyle = 'sedan'
go
```

Depois da confirmação das regras de negócios a serem utilizadas, foi realizado uma reunião com o Tech Lead da área de Business Intelligence, locado no Canadá para criação do checklist com os requisitos básicos para validação das tabelas, esse checklist será usado como base para criação do código.

- Evidência dos resultados:

Os dados, tabelas e repositório deste projeto estão armazenados no endereço abaixo locado no GITHUB:

<https://github.com/diegofp/XPEcienciadosPA>

Abaixo as evidências das criações da base de dados no SQL:

- Vehicle registration

```
SELECT [City],
[State],
[Zip],
[County],
[Model Year],
[Make],
[Body Type],
[Fuel Type],
[Unladen Weight],
[Maximum Gross Weight],
[Passengers],
[Reg Valid Date],
[Reg Expiration Date],
[Color],
[Scofflaw Indicator],
[Suspension Indicator],
[Revocation Indicator]
FROM [MotorSportsLTDA].[dbo].[vehicle_registration]
```

Record Type	VIN	Registration Class	City	State	Zip	County	Model Year	Make	Body Type	Fuel Type	Unladen Weight	Maximum Gross Weight	Passengers	Reg Valid Date
1	VEH	SEVDES1254030577	PAS	BALDWIN	NY 11510	NASSAU	2005	VOLKS	20SD	GAS	2805			2018-12-20TC
2	VEH	SEVDES1254030587	PAS	OLEAN	NY 14760	CATTARAUGUS	2005	VOLKS	20SD	GAS	2855			2018-02-05TC
3	VEH	SEVDES1254030573	PAS	WEST HENRIETTA	NY 14086	MONROE	2005	VOLKS	20SD	GAS	2834			2017-10-05TC
4	VEH	SEVDES1244031154	PAS	BREWSTER	NY 10509	PUTNAM	2004	VOLKS	20SD	GAS	2805			2017-01-01TC
5	VEH	SEVDES1244025825	PAS	SELZEN	NY 11764	SUFFOLK	2004	VOLKS	20SD	GAS	2805			2017-11-15TC
6	VEH	SEVDES1244025970	PAS	CORNING	NY 14830	STEBBEN	2004	VOLKS	20SD	GAS	2771			2018-01-01TC
7	VEH	SEVDES1244025101	PAS	LANCASTER	NY 14085	ERIE	2004	VOLKS	20SD	GAS	2805			2017-03-27TC
8	VEH	SEVDES1244025068	PAS	COPPOQUE	NY 11725	SUFFOLK	2004	VOLKS	20SD	GAS	2804			2017-06-12TC
9	VEH	SEVDES1244016461	PAS	CARMEL	NY 12512	PUTNAM	2004	VOLKS	20SD	GAS	2824			2018-04-03TC
10	VEH	SEVDES1244014278	PAS	BROOKLYN	NY 11238	KINGS	2004	VOLKS	20SD	GAS	2805			2018-02-28TC
11	VEH	SEVDES1244012084	PAS	WEST HURLEY	NY 12861	ULSTER	2004	VOLKS	20SD	GAS	2796			2017-06-28TC
12	VEH	SEVDES1244030383	PAS	MONTOUR FALLS	NY 14885	SCHUYLER	2004	VOLKS	20SD	GAS	3100			2018-04-18TC

- EV

SQLQuery3.sql - PA_portsLTD (sa 64) * X SQLQuery7.sql - PA_portsLTD (sa 69) SQLQuery2.sql - PA_portsLTD (sa 68) SQLQuery1.sql - PA_portsLTD (sa 65)

```

SELECT TOP (1000) (VIN (1-10))
, [Country]
, [City]
, [State]
, [Postal Code]
, [Model Year]
, [Make]
, [Model]
, [Electric Vehicle Type]
FROM [MotorSportsLTD].[dbo].[ev]

```

00 %

Resultados Mensagens

VIN (1-10)	Country	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type
1	100DE14M0V	Groesbeek	WA	98270	1997	CHEVROLET	S-10 PICKUP	Battery Electric Vehicle (BEV)
2	1F1Z1H107W	Chatham	Sequim	WA	98282	1998	FORD	RANGER
3	1F1Z08113K	Pleasant	Graham	WA	98338	1999	FORD	RANGER
4	1F1Z08112K	Pleasant	Graham	WA	98338	1999	FORD	RANGER
5	1F1Z08195K	Whitcomb	Duwamish	WA	98244	1999	FORD	RANGER
6	1F1Z08112K	Blagovest	Mount Vernon	WA	98274	1999	FORD	RANGER
7	1F1Z08161K	Thurston	Olympia	WA	98506	2000	FORD	RANGER
8	1F1Z08161K	King	Shoreline	WA	98133	2000	FORD	RANGER
9	1F1Z08161K	King	Shoreline	WA	98133	2000	FORD	RANGER
10	1F1Z08171K	King	Seattle	WA	98117	2000	FORD	RANGER
11	1F1Z08101K	King	Seattle	WA	98136	2000	FORD	RANGER
12	1F1Z08111K	Groesbeek	Edmonds	WA	98026	2000	FORD	RANGER
13	1F1Z08179K	Whitcomb	Everett	WA	98247	2000	FORD	RANGER
14	1F1Z08101K	San Juan	Friday Harbor	WA	98250	2000	FORD	RANGER
15	1F1Z08111K	Whitcomb	Everett	WA	98247	2000	FORD	RANGER
16	JT30810V32	King	Sammamish	WA	98075	2002	TOYOTA	RAV4

Consultar novamente com êxito.

PARANHOS (16.0 RTM) sa (64) MotorSportsLTD: 00:00:00 1.000 linhas

Wordcities e dim_region

SQLQuery3.sql - PA_portsLTD (sa 69) * X SQLQuery2.sql - PA_portsLTD (sa 68) SQLQuery1.sql - PA_portsLTD (sa 65) SQLQuery3

```

SELECT TOP (1000) ["city"]
, ["city_ascii"]
, ["lat"]
, ["lng"]
, ["country"]
, ["iso2"]
, ["iso3"]
, ["admin_name"]
, ["capital"]
, ["population"]
, ["id"]
FROM [MotorSportsLTD].[dbo].[worldcities]

```

00 %

Resultados Mensagens

city	city_ascii	lat	lng	country	iso2	iso3	admin_name	capital	population	id
1	Tokyo	35.6897	139.6922	Japan	JP	JPN	Tokyo	primary	1392685764	
2	Jakarta	-6.1750	106.8275	Indonesia	ID	IDN	Jakarta	primary	1360771077	
3	Delhi	28.6100	77.2300	India	IN	IND	Delhi	admin	1356872604	
4	Guangzhou	23.1300	113.2600	China	CN	CHN	Guangdong	admin	1156237133	
5	Mumbai	19.0761	72.8775	India	IN	IND	Maharashtra	admin	1356226629	
6	Manila	14.5958	120.9722	Philippines	PH	PHL	Manila	primary	1609818140	
7	Shanghai	31.1667	121.4667	China	CN	CHN	Shanghai	admin	1156073548	
8	São Paulo	-23.5500	-46.6333	Brazil	BR	BRA	São Paulo	admin	1076532519	
9	Seoul	37.5600	126.9900	South Korea	KR	KOR	Seoul	primary	1410836482	
10	Mexico City	19.4333	-99.1333	Mexico	MX	MEX	Ciudad de México	primary	1484247881	
11	Cairo	30.0444	31.2358	Egypt	EG	EGY	Al Qahirah	primary	1818253931	
12	New York	40.6943	-73.9249	United States	US	USA	New York	primary	1840034016	
13	Dhaka	23.7630	90.3800	Bangladesh	BD	BGD	Dhaka	primary	11050259279	
14	Beijing	39.9040	116.4075	China	CN	CHN	Beijing	primary	1156228885	
15	Kolkata	22.5675	88.3700	India	IN	IND	West Bengal	admin	1356060520	
16	Bangkok	13.7525	100.4942	Thailand	TH	THA	Krung Thep Maha Nakhon	primary	180077000	

Consultar novamente com êxito.

PARANHOS (16.0 RTM)

EletricCarData

SQLQuery3.sql - PA_portsLTD (sa 65) * X SQLQuery3.sql - PA_portsLTD (sa 66) SQLQuery3.sql - PA_portsLTD (sa 67)

```

SELECT TOP (1000) [Brand]
, [Model]
, [Accel]
, [TopSpeed]
, [Range]
, [Efficiency]
, [FastCharge]
, [RapidCharge]
, [PowerTrain]
, [PlugType]
, [BodyStyle]
, [Segment]
, [Seats]
, [PriceEuro]
FROM [MotorSportsLTD].[dbo].[ElectricCarData]

```

100 %

Resultados Mensagens

Brand	Model	Accel	TopSpeed	Range	Efficiency	FastCharge	RapidCharge	PowerTrain	PlugType	BodyStyle	Segment	Seats	PriceEuro
1	Tesla	Model 3 Long Range Dual Motor	4.6 sec	233 km/h	450 km	181 Wh/km	940 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	D	55400
2	Volkswagen	ID.3 Pure	10.0 sec	160 km/h	270 km	187 Wh/km	290 km/h	Rapid charging possible	Rear Wheel Drive	Type 2 CCS	Hatchback	C	30000
3	Polestar	2	4.7 sec	210 km/h	400 km	181 Wh/km	620 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	LiBack	D	56440
4	BMW	iX3	6.8 sec	180 km/h	360 km	208 Wh/km	560 km/h	Rapid charging possible	Rear Wheel Drive	Type 2 CCS	SUV	D	68040
5	Honda	e	8.9 sec	148 km/h	170 km	180 Wh/km	180 km/h	Rapid charging possible	Rear Wheel Drive	Type 2 CCS	Hatchback	B	42097
6	Lucid	Air	2.8 sec	250 km/h	610 km	180 Wh/km	620 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	F	105000
7	Volkswagen	e-Golf	8.6 sec	150 km/h	190 km	168 Wh/km	220 km/h	Rapid charging possible	Front Wheel Drive	Type 2 CCS	Hatchback	C	31900
8	Peugeot	e-208	5.1 sec	150 km/h	275 km	184 Wh/km	420 km/h	Rapid charging possible	Front Wheel Drive	Type 2 CCS	Hatchback	B	29652
9	Tesla	Model 3 Standard Range Plus	5.6 sec	225 km/h	310 km	193 Wh/km	650 km/h	Rapid charging possible	Rear Wheel Drive	Type 2 CCS	Sedan	D	46380
10	Audi	Q4 e-tron	6.3 sec	180 km/h	400 km	193 Wh/km	540 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	SUV	D	55000
11	Mercedes	EQC 400 4MATIC	5.1 sec	180 km/h	370 km	216 Wh/km	440 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	SUV	D	69404
12	Nissan	Leaf	7.9 sec	144 km/h	220 km	184 Wh/km	230 km/h	Rapid charging possible	Front Wheel Drive	Type 2 CHAdeMO	Hatchback	C	29234
13	Hyundai	Kona Electric 64 kWh	7.9 sec	167 km/h	400 km	180 Wh/km	380 km/h	Rapid charging possible	Front Wheel Drive	Type 2 CCS	SUV	B	40795
14	BMW	i4	4.0 sec	200 km/h	450 km	178 Wh/km	650 km/h	Rapid charging possible	Rear Wheel Drive	Type 2 CCS	Sedan	D	65000
15	Hyundai	IONIQ Electric	9.7 sec	165 km/h	250 km	193 Wh/km	210 km/h	Rapid charging possible	Front Wheel Drive	Type 2 CCS	LiBack	C	34459
16	Volkswagen	ID.3 Pro S	7.9 sec	180 km/h	440 km	175 Wh/km	590 km/h	Rapid charging possible	Rear Wheel Drive	Type 2 CCS	Hatchback	C	40036
17	Renault	Tesla Model 3	7.9 sec	180 km/h	370 km	170 Wh/km	590 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	D	100781

Abaixo a evidência dos requisitos definidos para o checklist e critérios de validação com o Tech Lead da área.

DPM: Acceptance Criteria

Jan 25, 2024 • 1 min read • 4 people viewed

KPI validations

- Regression tests to make sure same columns have the same value between Legacy and Snowflake
- Validate the same KPI using Legacy and Snowflake for specific period

Field validations

- Any fields used in the Sales Module for Slicer/Relationship has been validated on the snowflake table
 - Field has the right name based on the content of the field
 - Field has the right data type for Power BI
 - Field is not having any wrong values (Accuracy)
 - Field used in relationship needs to be in Snowflake
 - Field used in filters/slicers needs to be in Snowflake

Table validations

- Test the primary key for duplicates
- All records from Legacy are found in Snowflake using the primary key
- All records from Snowflake are found in Legacy using the primary key
- Table has up to date data based on the frequency of the refresh
- Table in the dataset can be reconnect to the Snowflake table

2.1.2 Lições Aprendidas

Neste Sprint 1, considero que a principal lição aprendida foi a necessidade de mudança da ordem da atividade de criação do código com a definição das tabelas e regras de negócios.

Observei que antes do início da codificação, para evitar retrabalhos, foi necessário o levantamento e definição das migrações a serem realizadas e suas regras de negócios, e com isso será possível criar um código mais robusto e assertivo conforme checklist definido.

2.2 Sprint 2

2.2.1 Solução

- **Evidência do planejamento:**

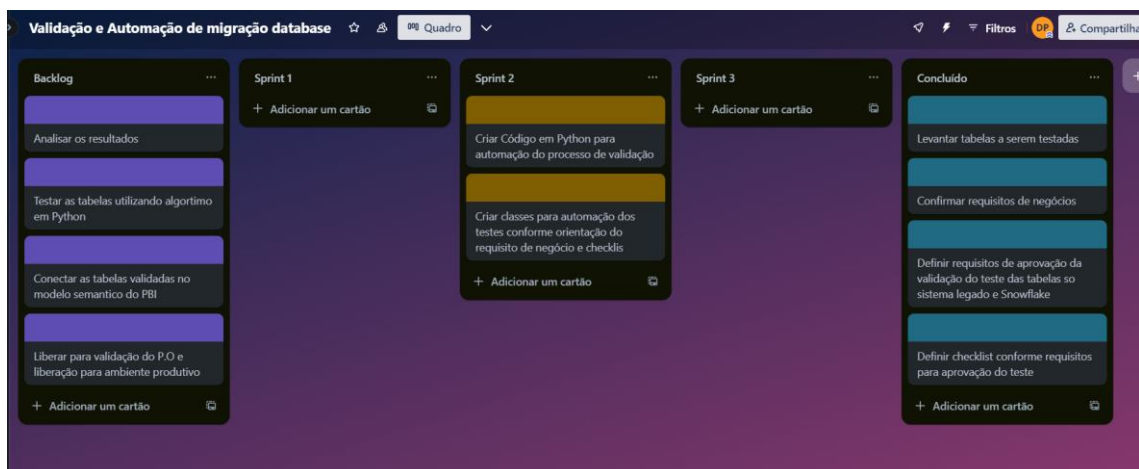
Neste sprint 2, o planejamento das atividades a serem entregues contemplam a parte de desenvolvimento do código que será utilizado na execução do teste.

- Criar Código em Python para automação do processo de validação.
- Criar classe para automação dos testes conforme orientação do requisito do negócio e checklist

A figura 9 evidencia o planejamento realizado no Trello para este sprint 2.

Figura 9 - Captura de tela do backlog no Trello - Sprint 2

(<https://trello.com/b/EGqBH0Eq/valida%C3%A7%C3%A3o-e-automa%C3%A7%C3%A3o-de-migra%C3%A7%C3%A3o-database>)



- Evidência da execução de cada requisito:

Nesta etapa de execução do código e realização do teste, foi utilizado os dados carregados nas tabelas Legado, conforme descrito na Sprint 1, porém para emular a realidade da companhia, foi criado algumas VIEWS com tabelas não normalizadas.

Os dados foram coletados em formato CSV para utilização no python, conforme exemplos abaixo:

Figura 10: Tela de extração dos dados do Legado

SQLQuery1.sql - PA...OS.master (sa (55))

```
select *
from MotorSportsLTDA.dbo.ev_sedan
```

100 %

Resultados Mensagens

	Brand	Model	Accel	TopSpeed	Range	Efficiency	FastCharge	RapidCharge	PowerTrain	PlugType	BodyStyle	Segment	Seats	PriceEuro
1	Tesla	Model 3 Long Range Dual Motor	4.6 sec	233 km/h	450 km	161 Wh/km	940 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	D	5	55480
2	Lucid	Air	2.8 sec	250 km/h	610 km	180 Wh/km	620 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	F	5	105000
3	Tesla	Model 3 Standard Range Plus	5.6 sec	225 km/h	310 km	153 Wh/km	650 km/h	Rapid charging possible	Rear Wheel Drive	Type 2 CCS	Sedan	D	5	46380
4	BMW	i4	4.0 sec	200 km/h	450 km	178 Wh/km	650 km/h	Rapid charging possible	Rear Wheel Drive	Type 2 CCS	Sedan	D	5	65000
5	Porsche	Taycan Turbo S	2.8 sec	260 km/h	375 km	223 Wh/km	780 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	F	4	180781
6	Audi	e-tron GT	3.5 sec	240 km/h	425 km	197 Wh/km	850 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	F	4	125000
7	Tesla	Model 3 Long Range Performance	3.4 sec	261 km/h	435 km	167 Wh/km	910 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	D	5	61480
8	Porsche	Taycan 4S	4.0 sec	250 km/h	365 km	195 Wh/km	730 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	F	4	102945
9	Porsche	Taycan 4S Plus	4.0 sec	250 km/h	425 km	197 Wh/km	890 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	F	4	109302
10	Porsche	Taycan Turbo	3.2 sec	260 km/h	390 km	215 Wh/km	810 km/h	Rapid charging possible	All Wheel Drive	Type 2 CCS	Sedan	F	4	148301

Figura 11: Tela de extração dos dados do Legado

SQLQuery1.sql - PA...OS.master (sa (55))

```
select *
from MotorSportsLTDA.dbo.Electrical_Car_Sales_NA
```

100 %

Resultados Mensagens

	VIN (1-10)	County	city	State	Model	Model Year	Make	Electric Vehicle Type	country	iso2	lat	lng	BodyStyle	Segment	PriceEuro
1	1G0DE14HXV	Snohomish	Mayaville	WA	S-10 PICKUP	1997	CHEVROLET	Battery Electric Vehicle (BEV)	United States	US	48.0809	-122.1561	NULL	NULL	NULL
2	1FTZR0816Y	Thurston	Olympia	WA	RANGER	2000	FORD	Battery Electric Vehicle (BEV)	United States	US	47.0417	-122.8959	NULL	NULL	NULL
3	1FTZR0817Y	King	Seattle	WA	RANGER	2000	FORD	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	NULL	NULL	NULL
4	1FTZR0810Y	King	Seattle	WA	RANGER	2000	FORD	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	NULL	NULL	NULL
5	5YJRE11B68	King	Seattle	WA	ROADSTER	2008	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	Cabrio	S	215000
6	5YJRE11B68	King	Seattle	WA	ROADSTER	2008	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	Cabrio	S	215000
7	5YJRE11B88	King	Seattle	WA	ROADSTER	2008	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	Cabrio	S	215000
8	5YJRE11B18	King	Seattle	WA	ROADSTER	2008	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	Cabrio	S	215000
9	5YJRE11B38	King	Seattle	WA	ROADSTER	2008	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	Cabrio	S	215000
10	5YJRE11B08	King	Seattle	WA	ROADSTER	2008	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	Cabrio	S	215000
11	5YJRE11B08	King	Seattle	WA	ROADSTER	2008	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	Cabrio	S	215000
12	5YJRE11B28	Spokane	Spokane	WA	ROADSTER	2008	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6671	-117.4330	Cabrio	S	215000
13	5YJRE11B88	Pierce	Lakewood	WA	ROADSTER	2008	TESLA	Battery Electric Vehicle (BEV)	United States	US	40.0763	-74.2031	Cabrio	S	215000
14	5YJRE1A11A	King	Bellevue	WA	ROADSTER	2010	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.5951	-122.1535	Cabrio	S	215000
15	5YJRE1A38A	King	Seattle	WA	ROADSTER	2010	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	Cabrio	S	215000
16	5YJRE1A1XA	King	Bellevue	WA	ROADSTER	2010	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.5951	-122.1535	Cabrio	S	215000
17	5YJRE1A16A	King	Seattle	WA	ROADSTER	2010	TESLA	Battery Electric Vehicle (BEV)	United States	US	47.6211	-122.3244	Cabrio	S	215000

Consulta executada com êxito. PARANHOS (16.0 RTM) sa (55) master 00:00:02 61.419 linhas

Figura 12 CSV - View - Electrical_car_sales_mercosul - base LEGADO

```

1  VIN |city|state|Model|Sales|Year|Make|Electric Vehicle Type|country|iso2|lat|lng|BodyStyle|Segment|PriceEuro
2  1G1RA6E41C|Lewis;Toledo;|WA;|VOLT;2012;CHEVROLET;Plug-in Hybrid Electric Vehicle (PHEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
3  5Y3SA1E4S6|Lewis;Toledo;|WA;|MODEL 3;2016;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
4  1FADP5F3H3|Lewis;Toledo;|WA;E-MAX;2017;FORD;Plug-in Hybrid Electric Vehicle (PHEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
5  1M4BZCP8H3|Lewis;Toledo;|WA;LEAF;2019;NISSAN;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;Hatchback;29234
6  1G1RB6S59J|Lewis;Toledo;|WA;VOLT;2018;CHEVROLET;Plug-in Hybrid Electric Vehicle (PHEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
7  5Y3J31EB9J|Lewis;Toledo;|WA;|MODEL 3;2018;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
8  5Y3SA1E24J|Lewis;Toledo;|WA;|MODEL 3;2018;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
9  5Y3J31EB3J|Santa Cruz;Santa Cruz;|CZ;|MODEL 3;2018;TESLA;Battery Electric Vehicle (BEV);Bolivia;BO;-17.7892;63.1975;NULL;NULL;NULL
10  1M4AZ1CP3J|Lewis;Toledo;|WA;LEAF;2018;NISSAN;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;Hatchback;29234
11  1G1FX6S0S3|Lewis;Toledo;|WA;BOLT EV;2018;CHEVROLET;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
12  5Y3J31EA0K|Lewis;Toledo;|WA;|MODEL 3;2019;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
13  1M4VABGE4K|Lewis;Toledo;|WA;E-TRON;2019;AUDI;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
14  1JDKARFP5K|Lewis;Toledo;|WA;PRIUS PRIME;2019;TOYOTA;Plug-in Hybrid Electric Vehicle (PHEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
15  1M4BZ1CP3K|Lewis;Toledo;|WA;LEAF;2019;NISSAN;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;Hatchback;29234
16  1M4BZ1CP8K|Lewis;Toledo;|WA;LEAF;2019;NISSAN;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;Hatchback;29234
17  5Y3J31EBXL|Lewis;Toledo;|WA;|MODEL 3;2020;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
18  5Y3J31EB6L|Lewis;Toledo;|WA;|MODEL 3;2020;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
19  1G1FY5S06L|Lewis;Toledo;|WA;BOLT EV;2020;CHEVROLET;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
20  5Y3J31EASL|Lewis;Toledo;|WA;|MODEL 3;2020;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
21  2C4RC1N79L|Lewis;Toledo;|WA;PACIFICA;2020;CHRYSLER;Plug-in Hybrid Electric Vehicle (PHEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
22  5Y3J31EB0M|Lewis;Toledo;|WA;|MODEL 3;2021;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
23  1G1FY6S08M|Las Animas;Trinidad;|CO;BOLT EV;2021;CHEVROLET;Battery Electric Vehicle (BEV);Bolivia;BO;-14.8292;64.9014;NULL;NULL;NULL
24  KM0C3D1C0M|Lewis;Toledo;|WA;EVE;2022;KIA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
25  7SAYGD0E6N|Lewis;Toledo;|WA;|MODEL Y;2022;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
26  5Y3SA1E55N|Lewis;Toledo;|WA;|MODEL 3;2022;TESLA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
27  1G1FZ6S08M|Lewis;Toledo;|WA;BOLT EV;2022;CHEVROLET;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
28  1C4J3KP69P|Lewis;Toledo;|WA;WRANGLER;2023;JEEP;Plug-in Hybrid Electric Vehicle (PHEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
29  3FMFK3SSBP1|Lewis;Toledo;|WA;MUSTANG MACH-E;2023;FORD;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
30  KM0C3L1BP1|Lewis;Toledo;|WA;NIRO;2023;KIA;Battery Electric Vehicle (BEV);Brazil;BR;-24.7139;-53.7428;NULL;NULL;NULL
31

```

Figura 13: CSV - View - Electrical_car_sales_na - base LEGADO

	electrical_car_sales_nacsv	data
1	WIN (1-10);	city;State;Model;Year;Make;Electric Vehicle Type;Country;ISO2;lat;lng;BodyStyle;Segment;PriceEuro
2	IGDE140V;	Nohomish;Mariesville;NA;S-10 PICKUP;1997;CHEVROLET;Battery Electric Vehicle (BEV);United States;US;48.0889;-122.1561;NULL;NULL;NULL
3	1FZT08019;	Thurston;Olympia;NA;RANGER;2000;FORD;Battery Electric Vehicle (BEV);United States;US;47.0417;-122.8959;NULL;NULL;NULL
4	1FZT08019;	Spokane;Spokane;WA;RANGER;2000;FORD;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
5	1FZT08019;	King;Seattle;Col-4;State;2000;FORD;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;NULL;NULL;NULL
6	5Y0RE11868;	King;Seattle;WA;ROADSTER;2008;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
7	5Y0RE11868;	King;Seattle;WA;ROADSTER;2008;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
8	5Y0RE11868;	King;Seattle;WA;ROADSTER;2008;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
9	5Y0RE11818;	King;Seattle;WA;ROADSTER;2008;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
10	5Y0RE11838;	King;Seattle;WA;ROADSTER;2008;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
11	5Y0RE11808;	King;Seattle;WA;ROADSTER;2008;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
12	5Y0RE11808;	King;Bellevue;WA;ROADSTER;2008;TESLA;Battery Electric Vehicle (BEV);United States;US;47.5951;-122.1535;Cabrio;215000
13	5Y0RE11828;	Spokane;Spokane;WA;RANGER;2000;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6671;-117.4330;Cabrio;215000
14	5Y0RE11808;	Pierce;Lakewood;WA;ROADSTER;2008;TESLA;Battery Electric Vehicle (BEV);United States;US;48.0763;-74.2031;Cabrio;215000
15	5Y0RE1A1A;	King;Bellevue;WA;ROADSTER;2010;TESLA;Battery Electric Vehicle (BEV);United States;US;47.5951;-122.1535;Cabrio;215000
16	5Y0RE1A38A;	King;Seattle;WA;ROADSTER;2010;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
17	5Y0RE1A1A;	King;Bellevue;WA;ROADSTER;2010;TESLA;Battery Electric Vehicle (BEV);United States;US;47.5951;-122.1535;Cabrio;215000
18	5Y0RE1A1A;	King;Seattle;WA;ROADSTER;2010;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
19	5Y0RE1A0A;	King;Seattle;WA;ROADSTER;2010;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
20	1M6ML2186A;	Thurston;Olympia;WA;MHEGO;2010;MHEGO ELECTRIC CARS;Plug-in Hybrid Electric Vehicle (PHEV);United States;US;47.0417;-122.8959;NULL;NULL;NULL
21	5Y0RE1A1A;	King;Seattle;WA;ROADSTER;2010;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
22	1M6ML218A;	King;Seattle;WA;ROADSTER;2010;TESLA;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Cabrio;215000
23	1M6ML218A;	Pierce;Tacoma;WA;MHEGO;2010;MHEGO ELECTRIC CARS;Plug-in Hybrid Electric Vehicle (PHEV);United States;US;47.2431;-122.4511;NULL;NULL;NULL
24	5Y0RE1A32A;	King;Bellevue;WA;ROADSTER;2010;TESLA;Battery Electric Vehicle (BEV);United States;US;47.5951;-122.1535;Cabrio;215000
25	3JNAZCP84B;	King;Seattle;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Hatchback;29234
26	3JNAZCP88B;	King;Seattle;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Hatchback;29234
27	3JNAZCP85B;	King;Bellevue;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.5951;-122.1535;Hatchback;29234
28	3JNAZCP83B;	King;Seattle;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Hatchback;29234
29	3JNAZCP83B;	King;Seattle;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Hatchback;29234
30	3JNAZCP83B;	King;Seattle;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Hatchback;29234
31	3JNAZCP84B;	Thurston;Olympia;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.0417;-122.8959;Hatchback;29234
32	1G0RDE468;	King;Seattle;WA;VOLVO;2020;CHEVROLET;Plug-in Hybrid Electric Vehicle (PHEV);United States;US;47.6211;-122.3244;NULL;NULL;NULL
33	3JNAZCP83B;	King;Bellevue;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.5951;-122.1535;Hatchback;29234
34	3JNAZCP88B;	King;Seattle;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Hatchback;29234
35	3JNAZCP83B;	King;Seattle;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Hatchback;29234
36	3JNAZCP80B;	Clark;Vancouver;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;45.6366;-122.5967;Hatchback;29234
37	3JNAZCP84B;	King;Seattle;WA;LEAF;2011;NISSAN;Battery Electric Vehicle (BEV);United States;US;47.6211;-122.3244;Hatchback;29234

Figura 14: CSV - View - vehicle_registration - base LEGADO

VIN	Registration Class	City	State	Zip	County	Model Year	Make	Body Type	Fuel Type	Unladen Weight	Maximum Gross Weight	Passenger Capacity
1	"VEH"	"98BDE61254006577"	PAS	"BALDWIN"	"NY;11510;"	"MASSAU"	"2005;VOLKS;ZSDS;"	"GAS"	"2805;;2016-12-20T00:00:00;2019-02-17"			
2	"VEH"	"98BDE61254003887"	PAS	"OLEAN"	"NY;14760;"	"CATTARAUGUS"	"2005;VOLKS;ZSDS;"	"GAS"	"2805;;2018-03-05T00:00:00;2020-03-18"			
3	"VEH"	"98BDE61254003873"	PAS	"WEST HENRIETTA"	"NY;14586;"	"MONROE"	"2005;VOLKS;ZSDS;"	"GAS"	"2934;;2017-10-05T00:00:00;2019-10-04"			
4	"VEH"	"98BDE612440031154"	PAS	"BREWSTER"	"NY;10589;"	"PUTNAM"	"2004;VOLKS;ZSDS;"	"GAS"	"2805;;2017-07-07T00:00:00;2019-06-17"			
5	"VEH"	"98BDE612440029825"	PAS	"SELDEN"	"NY;11784;"	"SUFFOLK"	"2004;VOLKS;ZSDS;"	"GAS"	"2805;;2017-11-15T00:00:00;2019-11-22"			
6	"VEH"	"98BDE612440025970"	PAS	"CORNING"	"NY;14830;"	"STEUBEN"	"2004;VOLKS;ZSDS;"	"GAS"	"2771;;2018-01-07T00:00:00;2019-06-14"			
7	"VEH"	"98BDE612440025161"	PAS	"LANCASTER"	"NY;14086;"	"ERIE"	"2004;VOLKS;ZSDS;"	"GAS"	"2805;;2017-03-27T00:00:00;2018-07-17"			
8	"VEH"	"98BDE612440023068"	PAS	"COPIAGUE"	"NY;11726;"	"SUFFOLK"	"2004;VOLKS;ZSDS;"	"GAS"	"2924;;2017-09-12T00:00:00;2019-09-09"			
9	"VEH"	"98BDE612440016461"	PAS	"CARMEL"	"NY;10512;"	"PUTNAM"	"2004;VOLKS;ZSDS;"	"GAS"	"2924;;2018-04-03T00:00:00;2020-03-11"			
10	"VEH"	"98BDE612440014578"	PAS	"BROOKLYN"	"NY;11238;"	"KINGS"	"2004;VOLKS;ZSDS;"	"GAS"	"2805;;2018-02-28T00:00:00;2020-02-06"			
11	"VEH"	"98BDE612440012894"	PAS	"WEST HURLEY"	"NY;12491;"	"ULSTER"	"2004;VOLKS;ZSDS;"	"GAS"	"2796;;2017-09-26T00:00:00;2019-09-25"			
12	"VEH"	"98BDE61244006383"	PAS	"MONTROUSE FALLS"	"NY;14865;"	"SCHUYLER"	"2004;VOLKS;ZSDS;"	"GAS"	"3100;;2018-04-16T00:00:00;2020-04-15"			
13	"VEH"	"98BDE61244004147"	PAS	"HAVERSTRAW"	"NY;10927;"	"ROCKLAND"	"2004;VOLKS;ZSDS;"	"GAS"	"2805;;2016-07-22T00:00:00;2018-07-21"			
14	"VEH"	"98BDE61244003936"	PAS	"HOPEWELL JUNCT"	"NY;12533;"	"DUTCHESS"	"2004;VOLKS;ZSDS;"	"GAS"	"2805;;2018-01-31T00:00:00;2020-01-01"			
15	"VEH"	"98BDE61244003810"	PAS	"SCHENECTADY"	"NY;12304;"	"SCHENECTADY"	"2004;VOLKS;ZSDS;"	"GAS"	"2805;;2017-08-18T00:00:00;2019-08-17"			
16	"VEH"	"98BDE61244002186"	PAS	"YONKERS"	"NY;10701;"	"WESTCHESTER"	"2004;VOLKS;ZSDS;"	"GAS"	"2805;;2016-12-14T00:00:00;2018-12-13"			
17	"VEH"	"98BDE612234067232"	PAS	"ROCHESTER"	"NY;14607;"	"MONROE"	"2003;VOLKS;ZSDS;"	"GAS"	"2857;;2018-02-10T00:00:00;2020-04-03"			
18	"VEH"	"98BDE612234056361"	PAS	"MONTICELLO"	"NY;12701;"	"SULLIVAN"	"2003;VOLKS;ZSDS;"	"GAS"	"2976;;2018-05-29T00:00:00;2020-05-28"			
19	"VEH"	"98BDE61223405663"	PAS	"ROCHESTER"	"NY;14622;"	"MONROE"	"2003;VOLKS;ZSDS;"	"GAS"	"2700;;2017-10-25T00:00:00;2019-10-24"			
20	"VEH"	"98BDE612234053850"	PAS	"NEW HYDE PARK"	"NY;11040;"	"MASSAU"	"2003;VOLKS;ZSDS;"	"GAS"	"2857;;2016-10-05T00:00:00;2018-11-11"			
21	"VEH"	"98BDE612234051791"	PAS	"FREEPORT"	"NY;11520;"	"MASSAU"	"2003;VOLKS;ZSDS;"	"GAS"	"2857;;2017-07-06T00:00:00;2019-07-05"			
22	"VEH"	"98BDE612234050219"	PAS	"LITTLE FALLS"	"NY;13365;"	"HERKIMER"	"2003;VOLKS;ZSDS;"	"GAS"	"2857;;2017-04-17T00:00:00;2019-04-24"			
23	"VEH"	"98BDE612234049085"	PAS	"WEST HURLEY"	"NY;12491;"	"ULSTER"	"2003;VOLKS;ZSDS;"	"GAS"	"2857;;2017-05-22T00:00:00;2019-05-15"			
24	"VEH"	"98BDE612234048244"	PAS	"TOMKINS COVE"	"NY;10986;"	"ROCKLAND"	"2003;VOLKS;ZSDS;"	"GAS"	"2771;;2016-07-26T00:00:00;2018-07-25"			
25	"VEH"	"98BDE612234038538"	PAS	"FRANKLIN SQUARE"	"NY;11010;"	"MASSAU"	"2003;VOLKS;ZSDS;"	"GAS"	"2976;;2018-05-11T00:00:00;2019-07-30"			
26	"VEH"	"98BDE612234036823"	PAS	"HORSEHEADS"	"NY;14845;"	"CHEMUNG"	"2003;VOLKS;ZSDS;"	"GAS"	"2976;;2016-12-27T00:00:00;2018-12-26"			
27	"VEH"	"98BDE612234036613"	SFR	"BINGHAMTON"	"NY;13903;"	"BROOME"	"2003;VOLKS;ZSDS;"	"GAS"	"2719;;2017-06-30T00:00:00;2019-06-29"			
28	"VEH"	"98BDE612234036501"	PAS	"WALKILL"	"NY;12589;"	"ORANGE"	"2003;VOLKS;ZSDS;"	"GAS"	"2932;;2018-03-19T00:00:00;2020-03-18"			
29	"VEH"	"98BDE612234036224"	PAS	"BUFFALO"	"NY;14215;"	"ERIE"	"2003;VOLKS;ZSDS;"	"GAS"	"2857;;2018-06-11T00:00:00;2020-06-10"			
30	"VEH"	"98BDE612234034800"	PAS	"MIDDLETOWN"	"NY;10941;"	"ORANGE"	"2003;VOLKS;ZSDS;"	"GAS"	"2976;;2018-04-05T00:00:00;2020-04-04"			
31	"VEH"	"98BDE612234029855"	PAS	"BRENTWOOD"	"NY;11717;"	"SUFFOLK"	"2003;VOLKS;ZSDS;"	"GAS"	"2976;;2018-03-16T00:00:00;2019-10-11"			
32	"VEH"	"98BDE612234029810"	PAS	"MIDDLETOWN"	"NY;10940;"	"ORANGE"	"2003;VOLKS;ZSDS;"	"GAS"	"2932;;2016-11-01T00:00:00;2018-11-12"			
33	"VEH"	"98BDE612234029192"	PAS	"CATSKILL"	"NY;12414;"	"GREENE"	"2003;VOLKS;ZSDS;"	"GAS"	"2976;;2017-03-24T00:00:00;2019-03-23"			
34	"VEH"	"98BDE612234028382"	PAS	"ASTORIA"	"NY;11103;"	"QUEENS"	"2003;VOLKS;ZSDS;"	"GAS"	"2976;;2018-01-06T00:00:00;2020-02-07"			
35	"VEH"	"98BDE612234028169"	PAS	"CAMILLUS"	"NY;13031;"	"ONONDAGA"	"2003;VOLKS;ZSDS;"	"GAS"	"2857;;2018-04-25T00:00:00;2020-04-24"			
36	"VEH"	"98BDE612234027314"	PAS	"PRT WASHINGTON"	"NY;11050;"	"MASSAU"	"2003;VOLKS;ZSDS;"	"GAS"	"2857;;2018-06-15T00:00:00;2020-07-16"			

Para os dados do Snowflake, também foram coletados os dados em CSV, conforme exemplo abaixo.

Figura 15: - Tela de worksheet do Snowflake

```

1 select *
2 from Prod.electrical_registration
3 where country != 'NA'

```

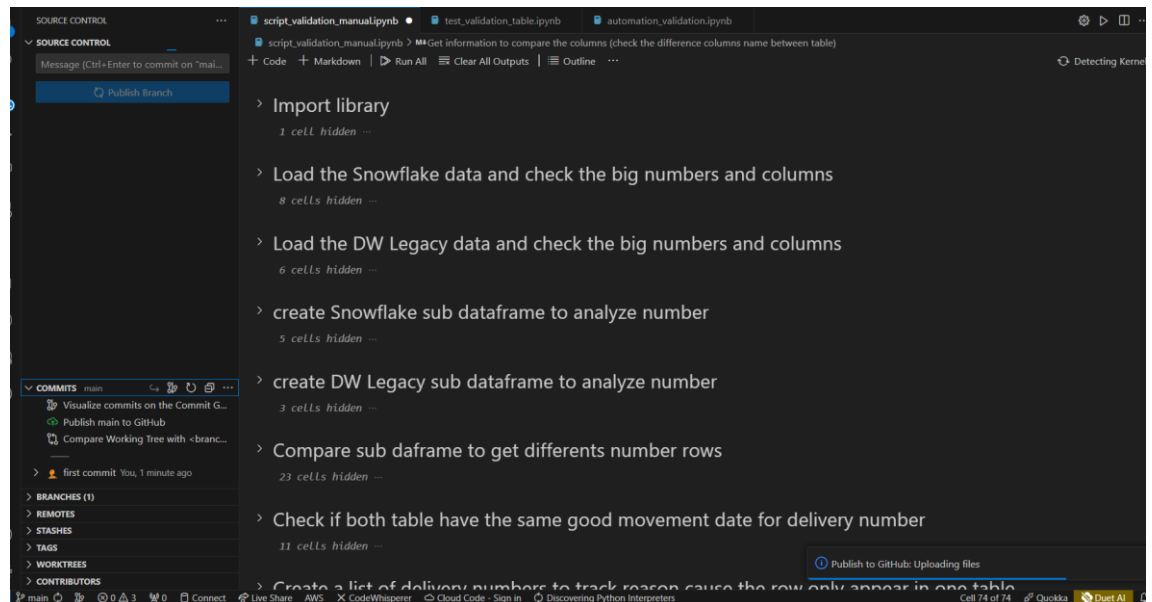
VIN	Country	City	State	Zip	County	Model Year	Make	Body Type	Fuel Type	Unladen Weight	Maximum Gross Weight	Passenger Capacity
1	"US"	"BALDWIN"	"NY"	"11510"	"MASSAU"	"2005"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2016-12-20T00:00:00"	"2019-02-17"
2	"US"	"OLEAN"	"NY"	"14760"	"CATTARAUGUS"	"2005"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2018-03-05T00:00:00"	"2020-03-18"
3	"US"	"WEST HENRIETTA"	"NY"	"14586"	"MONROE"	"2005"	"VOLKS"	"ZSDS"	"GAS"	"2934"	"2017-10-05T00:00:00"	"2019-10-04"
4	"US"	"BREWSTER"	"NY"	"10589"	"PUTNAM"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2017-07-07T00:00:00"	"2019-06-17"
5	"US"	"SELDEN"	"NY"	"11784"	"SUFFOLK"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2017-11-15T00:00:00"	"2019-11-22"
6	"US"	"CORNING"	"NY"	"14830"	"STEUBEN"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2771"	"2018-01-07T00:00:00"	"2019-06-14"
7	"US"	"LANCASTER"	"NY"	"14086"	"ERIE"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2017-03-27T00:00:00"	"2018-07-17"
8	"US"	"COPIAGUE"	"NY"	"11726"	"SUFFOLK"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2924"	"2017-09-12T00:00:00"	"2019-09-09"
9	"US"	"CARMEL"	"NY"	"10512"	"PUTNAM"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2924"	"2018-04-03T00:00:00"	"2020-03-11"
10	"US"	"BROOKLYN"	"NY"	"11238"	"KINGS"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2018-02-28T00:00:00"	"2020-02-06"
11	"US"	"WEST HURLEY"	"NY"	"12491"	"ULSTER"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2796"	"2017-09-26T00:00:00"	"2019-09-25"
12	"US"	"MONTROUSE FALLS"	"NY"	"14865"	"SCHUYLER"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"3100"	"2018-04-16T00:00:00"	"2020-04-15"
13	"US"	"HAVERSTRAW"	"NY"	"10927"	"ROCKLAND"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2016-07-22T00:00:00"	"2018-07-21"
14	"US"	"HOPEWELL JUNCT"	"NY"	"12533"	"DUTCHESS"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2018-01-31T00:00:00"	"2020-01-01"
15	"US"	"SCHENECTADY"	"NY"	"12304"	"SCHENECTADY"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2017-08-18T00:00:00"	"2019-08-17"
16	"US"	"YONKERS"	"NY"	"10701"	"WESTCHESTER"	"2004"	"VOLKS"	"ZSDS"	"GAS"	"2805"	"2016-12-14T00:00:00"	"2018-12-13"
17	"US"	"ROCHESTER"	"NY"	"14607"	"MONROE"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2857"	"2018-02-10T00:00:00"	"2020-04-03"
18	"US"	"MONTICELLO"	"NY"	"12701"	"SULLIVAN"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2976"	"2018-05-29T00:00:00"	"2020-05-28"
19	"US"	"ROCHESTER"	"NY"	"14622"	"MONROE"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2700"	"2017-10-25T00:00:00"	"2019-10-24"
20	"US"	"NEW HYDE PARK"	"NY"	"11040"	"MASSAU"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2857"	"2016-10-05T00:00:00"	"2018-11-11"
21	"US"	"FREEPORT"	"NY"	"11520"	"MASSAU"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2857"	"2017-07-06T00:00:00"	"2019-07-05"
22	"US"	"LITTLE FALLS"	"NY"	"13365"	"HERKIMER"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2857"	"2017-04-17T00:00:00"	"2019-04-24"
23	"US"	"WEST HURLEY"	"NY"	"12491"	"ULSTER"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2857"	"2017-05-22T00:00:00"	"2019-05-15"
24	"US"	"TOMKINS COVE"	"NY"	"10986"	"ROCKLAND"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2771"	"2016-07-26T00:00:00"	"2018-07-25"
25	"US"	"FRANKLIN SQUARE"	"NY"	"11010"	"MASSAU"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2976"	"2018-05-11T00:00:00"	"2019-07-30"
26	"US"	"HORSEHEADS"	"NY"	"14845"	"CHEMUNG"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2976"	"2016-12-27T00:00:00"	"2018-12-26"
27	"US"	"BINGHAMTON"	"NY"	"13903"	"BROOME"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2719"	"2017-06-30T00:00:00"	"2019-06-29"
28	"US"	"WALKILL"	"NY"	"12589"	"ORANGE"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2932"	"2018-03-19T00:00:00"	"2020-03-18"
29	"US"	"BUFFALO"	"NY"	"14215"	"ERIE"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2857"	"2018-06-11T00:00:00"	"2020-06-10"
30	"US"	"MIDDLETOWN"	"NY"	"10941"	"ORANGE"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2976"	"2018-04-05T00:00:00"	"2020-04-04"
31	"US"	"BRENTWOOD"	"NY"	"11717"	"SUFFOLK"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2976"	"2018-03-16T00:00:00"	"2019-10-11"
32	"US"	"MIDDLETOWN"	"NY"	"10940"	"ORANGE"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2932"	"2016-11-01T00:00:00"	"2018-11-12"
33	"US"	"CATSKILL"	"NY"	"12414"	"GREENE"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2976"	"2017-03-24T00:00:00"	"2019-03-23"
34	"US"	"ASTORIA"	"NY"	"11103"	"QUEENS"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2976"	"2018-01-06T00:00:00"	"2020-02-07"
35	"US"	"CAMILLUS"	"NY"	"13031"	"ONONDAGA"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2857"	"2018-04-25T00:00:00"	"2020-04-24"
36	"US"	"PRT WASHINGTON"	"NY"	"11050"	"MASSAU"	"2003"	"VOLKS"	"ZSDS"	"GAS"	"2857"	"2018-06-15T00:00:00"	"2020-07-16"

Figura 16: - Tela dos dados em CSV do electrical_sales_na_snowflake.csv

1	VIN (1-10):	country:	city:	state:	zip:	county:	model year:	make:	electric vehicle type:	country:	iso:	lat:	lon:	body style:	segment:	price:	euro	
13	5V9RE11B28;	spokane;	spokane;	WA;	RODSTER;	2008;	TESLA;	Battery Electric Vehicle (BEV)	United States;	US;	47.6671;	-117.433;	Cabrio;	215000.0;				
14	5V9RE11B88;	pierce;	lakewood;	WA;	RODSTER;	2010;	TESLA;	Battery Electric Vehicle (BEV)	United States;	US;	48.0763;	-74.2031;	Cabrio;	215000.0;				
15	5V9RE1A134;	king;	bellave;	WA;	RODSTER;	2010;	TESLA;	Battery Electric Vehicle (BEV)	United States;	US;	47.5951;	-122.1539;	Cabrio;	215000.0;				
16	5V9RE1A38A;	king;	seattle;	WA;	RODSTER;	2010;	TESLA;	Battery Electric Vehicle (BEV)	United States;	US;	47.6211;	-122.3244;	Cabrio;	215000.0;				
17	1JA1ZAC0P08;	king;	brementon;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.5436;	-122.7121;	Hatchback;	29234.0;				
18	1JA1ZAC0P08;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
19	1JA1ZAC0P08;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
20	1JA1ZAC0P08;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
21	1JA1ZAC0P08;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
22	1JA1ZAC0P08;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
23	1G1R0E647B;	placemat;	bellingham;	WA;	VOLT;	2011;	CHEVROLET;	Plug-in Hybrid Electric Vehicle (PHEV)	United States;	US;	48.7548;	-122.4609;						
24	1JA1ZAC0P1B;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
25	1JA1ZAC0P4B;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSA	Col 7: Male	etric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;			
26	1JA1ZAC0P7B;	placemat;	bellingham;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	48.7548;	-122.4609;	Hatchback;	29234.0;				
27	1G1R0E641B;	placemat;	bellingham;	WA;	VOLT;	2011;	CHEVROLET;	Plug-in Hybrid Electric Vehicle (PHEV)	United States;	US;	47.2431;	-122.4531;						
28	1G1R0E640F;	king;	rent;	WA;	VOLT;	2011;	CHEVROLET;	Plug-in Hybrid Electric Vehicle (PHEV)	United States;	US;	47.3887;	-122.2128;						
29	1G1R0E646B;	king;	seattle;	WA;	VOLT;	2011;	CHEVROLET;	Plug-in Hybrid Electric Vehicle (PHEV)	United States;	US;	47.6211;	-122.3244;						
30	1JA1ZAC0P3B;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
31	1JA1ZAC0P3B;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
32	1JA1ZAC0P3B;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.6211;	-122.3244;	Hatchback;	29234.0;				
33	1JA1ZAC0P3B;	king;	seattle;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.6211;	-122.3244;	Hatchback;	29234.0;				
34	1JA1ZAC0P4B;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
35	1JA1ZAC0PXB;	benton;	kennewick;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	46.1978;	-119.1732;	Hatchback;	29234.0;				
36	1JA1ZAC0P4B;	pierce;	spokane;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.6671;	-117.433;	Hatchback;	29234.0;				
37	1G1R0E641B;	thurston;	olympia;	WA;	VOLT;	2011;	CHEVROLET;	Plug-in Hybrid Electric Vehicle (PHEV)	United States;	US;	47.0417;	-122.8959;						
38	1JA1ZAC0P8B;	placemat;	bellingham;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	48.7548;	-122.4609;	Hatchback;	29234.0;				
39	1JA1ZAC0P2B;	spokane;	spokane;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.6671;	-117.433;	Hatchback;	29234.0;				
40	1JA1ZAC0P6B;	placemat;	bellingham;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	48.7548;	-122.4609;	Hatchback;	29234.0;				
41	1JA1ZAC0PXB;	king;	seattle;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.6211;	-122.3244;	Hatchback;	29234.0;				
42	1JA1ZAC0P3B;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.6211;	-122.3244;	Hatchback;	29234.0;				
43	1JA1ZAC0P3B;	king;	seattle;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.6211;	-122.3244;	Hatchback;	29234.0;				
44	1JA1ZAC0P1B;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
45	1JA1ZAC0P4B;	pierce;	tacoma;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.2431;	-122.4531;	Hatchback;	29234.0;				
46	1JA1ZAC0P8B;	spokane;	spokane;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.6671;	-117.433;	Hatchback;	29234.0;				
47	1JA1ZAC0P5B;	king;	seattle;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	48.0763;	-74.2031;	Hatchback;	29234.0;				
48	1JA1ZAC0P5B;	king;	seattle;	WA;	LEAF;	2011;	NISSAN;	Battery Electric Vehicle (BEV)	United States;	US;	47.6211;	-122.3244;	Hatchback;	29234.0;				

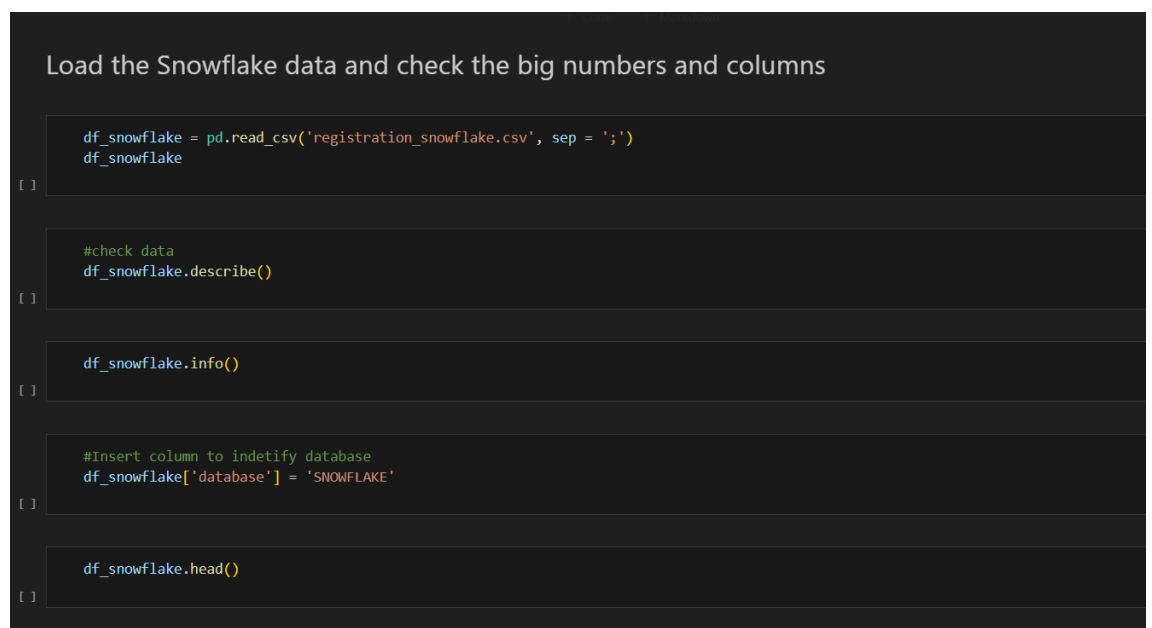
Para a realização dos testes de comparação, inicialmente foi desenvolvido um código no Notebook Jupyter em Python, e esse código é executado linha a linha, realizando as adaptações necessárias para cada necessidade de teste, conforme os requisitos de negócios definido no sprint 1, o código foca na execução da validados dos campos (field validation) do checklist e requisito de negócios.

Figura 17: Snapshot da tela com os tópicos do teste manual.



Na parte de carregamento dos dados, foi criado um código para carregar os arquivos CSV (figura 18).

Figura 18 - carregamento dos dados



Criação de sub dataframe para validação de um conjunto específico de dados utilizados no KPI e modelo semântico (figura 19).

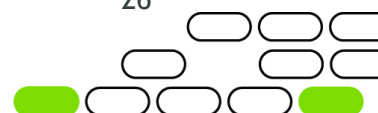


Figura 19 - criação de sub dataframe.

```

v create Snowflake sub dataframe to analyze number

[ ] columns_selected = ['WHOLESALE_DELIVERY_NO', 'WHOLESALE_DELIVERY_ITEM_NO', 'UNIT_WHOLESALE'] Python

[ ] subdf_snowflake = df_snowflake[columns_selected] Python

[ ] subdf_snowflake.head() Python

[ ] subdf_snowflake['UNIT_WHOLESALE'].sum() Python

[ ] subdf_snowflake.count() Python

```

Comparativo das tabelas, figura 20, utilizando merge outer para retornar as linhas existentes entre as tabelas e linhas diferentes.

Figura 20 - comparativo das tabelas

```

... script_validation_manual.ipynb • automation_validation.ipynb M
script_validation_manual.ipynb > Create a list of delivery numbers to track reason cause the row only appear in one table. > #check quantity
+ Code + Markdown + Run All + Restart + Clear All Outputs + Variables + Outline ... Python 3.12.0

[ ] subdf_snowflake.head() Python

[ ] #First test to see that rows / sub dataframe are same
equal = subdf_legacy.equals(subdf_snowflake)
if equal:
    print("dataframe are equal")
else:
    print('dataframe are different') Python

[ ] #merge to get the difference
merge_subdf = pd.merge(subdf_legacy, subdf_snowflake, how='outer', indicator=True).rename(columns={'_merge': 'difference'}) Python

[ ] merge_subdf.head() Python

[ ] #change de values with tables name
merge_subdf['difference'] = merge_subdf['difference'].replace(['left_only', 'right_only'], ['LEGACY', 'SNOWFLAKE']) Python

[ ] merge_subdf['difference'].value_counts() Python

[ ]

```

Todos os dados, códigos e base de dados utilizados estão no link abaixo no GitHub.

<https://github.com/diegofp/XPEcienciadosPA>

- Evidência dos resultados:

O resultado deste sprint foi a criação das classes e definições, com base no código para teste manual, para automação do processo de testes, onde podemos realizar o mesmo padrão de teste para N tabelas do Legado e Snowflake migradas, otimizando o tempo de realização do processo, visto que não será necessário realizar o código por completo, somente será necessário a criação de testes pontuais.

O código criado está descrito em Código 1.

Código 1: automation_vadiation.ipynb

```
class TableComparator:
    def __init__(self, legacy, snowflake):
        self.legacy = legacy
        self.snowflake = snowflake

# modulo para teste de valores nulos nas tabelas

    def null_columns(self):
        resultado_legacy = []
        colunas_nulas_legacy = []
        for coluna in self.legacy.columns:
            resultado_legacy = self.legacy[coluna].isna().sum()
            if resultado_legacy > 0:
                colunas_nulas_legacy.append(coluna)

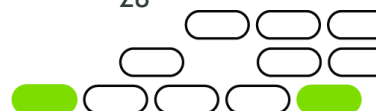
        resultado_snowflake = []
        colunas_nulas_snowflake = []
        for coluna in self.snowflake.columns:
            resultado_snowflake = self.snowflake[coluna].isna().sum()
            if resultado_snowflake > 0:
                colunas_nulas_snowflake.append(coluna)

        print('Colunas com valores nulos no Legado:',
colunas_nulas_legacy)
        print('Colunas com valores nulos no Snowflake:',
colunas_nulas_snowflake)

#teste de verificação se as tabelas possuem colunas com os mesmos nomes

    def columns_name(self):
        legacy_columns_name = []
        snowflake_columns_name = []

        legacy_columns_name = self.legacy.columns
        snowflake_columns_name = self.snowflake.columns
```



```

print('legado:', legacy_columns_name)
print('snowflake:', snowflake_columns_name)

#teste de comparação para verificar quantas linhas estão diferentes ou
iguais entre as tabelas

def compare(self):
    merge_df = pd.merge(self.legacy, self.snowflake, how='outer',
indicator=True).rename(columns={'_merge': "difference"})
    result = merge_df.groupby(merge_df['difference']).count()
    print(result)

#criação de arquivos CSV com os resultado dos teste de comparação

def result_compare(self):
    merge_df = pd.merge(self.legacy, self.snowflake, how='outer',
indicator=True).rename(columns={'_merge': "difference"})
    result = merge_df.groupby(merge_df['difference']).count()
    print(result)
    #criação dos arquivos:
    resultado_iguais = []
    resultados_somente_legacy = []
    resultados_somente_snowflake = []
    resultado_iguais = merge_df[merge_df['difference'] == 'both']
    resultados_somente_legacy = merge_df[merge_df['difference'] ==
'left_only']
    resultados_somente_snowflake = merge_df[merge_df['difference'] ==
'right_only']
    resultado_iguais.to_csv('resultados_iguais.csv', sep = ';',
index=False)
    resultados_somente_legacy.to_csv('resultados_somente_legacy.csv',
sep = ';', index=False)
    resultados_somente_snowflake.to_csv('resultados_somente_snowflake
', sep = ';', index=False)

```

Para rodar esse código é necessário:

- Python 3.12
- Importação da biblioteca Pandas
- Carregamento das tabelas com o seguinte padrão.
 - Legacy = com o endereço do arquivo legado (em CSV ou conexão direta via API)



- Snowflake = com o endereço do arquivo Snowflake (em CSV ou API)

Obs: Para arquivo CSV, é necessário carregar os dados para um dataframe com os nomes descritos acima utilizando a função `pd.read_csv` ('endereço do arquivo'), onde 'pd' é o *alias* para pandas.

Na classe 'TableComparator' temos as seguintes definições de teste:

- Null_columns = Testa se as colunas possuem valores em nulos e retornar em uma lista as colunas de cada tabela que contém os valores nulos.
- Columns_name = Retorna os nomes das colunas de cada tabela para criação de sub dataframe se necessário testar sub conjunto de dados para o KPI.
- Compare = Teste comparativo entre as linhas das tabelas e traz a quantidade de linhas diferentes em cada database
- Result_compare = Cria arquivos CSV com os resultados do teste comparativo.

Este script será importante para o Sprint 3, onde será realizado os testes de validação das tabelas exemplos do Legado e Snowflake.

2.2.2 Lições Aprendidas

Neste Sprint 2 tivemos como lições aprendidas a dificuldade em traduzir todos os requisitos de negócio para apenas um código para teste, para os campos de teste de tabela e teste de KPI, eles precisam serem realizados diretamente no modelo semântico do Power BI que será executado no Sprint 3.

No processo de criação das definições e classes para realizar o teste automático, também foi necessário a adequação de alguns testes, como a comparação de tabelas, para que o processo automatizado retornasse os resultados necessários para validação dos campos.

O processo de teste das tabelas será realizados no sprint 3.



2.3 Sprint 3

2.3.1 Solução

- Evidência do planejamento:
- Evidência da execução de cada requisito:
- Evidência dos resultados:

2.3.2 Lições Aprendidas



3. Considerações Finais

3.1 Resultados

Por meio de um texto detalhado, apresente os principais resultados alcançados pelo seu Projeto Aplicado.

Cite os pontos positivos e negativos, as dificuldades enfrentadas e as experiências vivenciadas durante todo o processo.

3.2 Contribuições

Apresente quais foram as contribuições que o seu Projeto Aplicado trouxe para que o Desafio proposto fosse solucionado.

Cite, por exemplo, as inovações, as vantagens sobre os similares, as melhorias alcançadas, entre outros.

3.3 Próximos passos

Descreva quais são os próximos passos que poderão contribuir com o aprimoramento da solução apresentada pelo seu Projeto Aplicado.

