

**FACULDADE DE INFORMATICA E ADMINISTRAÇÃO PAULISTA**  
**DATA SCIENCE**

RM 98119 – CESAR OLIVEIRA GOES

RM 97885 – FIAMA DOS SANTOS TRAJANO

RM 550759 – GABRIEL SILVA DE NEGREIROS LEAL DA ROCHA

RM 551770 – KARINA MACIEL PALMEIRA

**ENTREGA FINAL – CHALLENGE MNSAIT 2023**

INNOVATORS

São Paulo

Setembro, 2023

# Sumário

<b>INTRODUÇÃO.....</b>	<b>3</b>
<b>1. PRÉ-PROCESSAMENTO DE DADOS KNIME .....</b>	<b>4</b>
1.1 Execução do Knime .....	4
1.2 Criação do Workflow .....	5
<b>2. PRÉ-PROCESSAMENTO DE DADOS GOOGLE COLABORATORY .....</b>	<b>14</b>
2.1 Importando os Datasets .....	14
2.2. Tratamento de dados do dataset de Atibaia.....	15
2.3. Tratamento de dados do dataset do PRONAMP e PRONAF.....	18
2.4. Tratamento de dados do dataset Censo Rural Atibaia - IBGE.....	23
<b>3. INSTALAÇÃO DO MYSQL WORKBENCH.....</b>	<b>35</b>
3.1. Conexão com a base de dados.....	35
3.2. Modelagem de dados.....	36
3.3. Visualização de tabelas no SGBD.....	41
<b>4. ARQUITETURA .....</b>	<b>47</b>
4.1. Conexão com a base de dados.....	47
<b>5. VISUALIZAÇÃO DE DADOS NO GOOGLE LOOKER STUDIO .....</b>	<b>60</b>
5.1. Visão do produtor.....	59
5.2. Visão análise de risco produto .....	60
5.3. Visão de dados de crédito.....	62
5.4. Links de Acesso às Apresentações.....	62
<b>CONCLUSÃO .....</b>	<b>64</b>
<b>REFERÊNCIAS.....</b>	<b>65</b>

## INTRODUÇÃO

- A Minsait nos desafiou a encontrar uma solução para um problema no setor agrícola brasileiro: a assimetria de crédito rural.
- Apesar dos esforços governamentais e de agências de crédito privadas, a disparidade entre o crédito fornecido a grandes produtores e pequenos é enorme.
- A principal razão disso é a falta de dados por parte dos pequenos produtores, dificultando o fornecimento de crédito a eles.
- Em outras palavras, a falta de dados por parte desses pequenos produtores torna a concessão de crédito extremamente difícil e representa um risco muito grande para os credores.
- Isso cria a assimetria de crédito rural, bloqueia investimentos e impede o crescimento e o desenvolvimento dos pequenos agricultores, que são essenciais para a economia brasileira e a segurança alimentar.

### Solução:

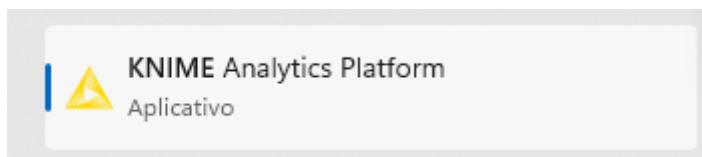
- Para criar o dashboard, acessamos bureau de dados públicos para obter dados significativos de pequenos e médios produtores, incluindo dados de concessão de crédito do Governo Federal (PRONAF e PRONAMP), lista de endereços, tipo de financiamento concedido, dados de produção e perfil dos produtores.
- Realizamos o pré-processamento de dados em um ambiente Python no Google Colaboratory e no Knime, eliminando ruídos e inconsistências.
- Em seguida, realizamos a modelagem dos dados (conceitual, lógica e física) para criar as tabelas relacionais que foram armazenadas no Sistema de Gerenciamento de Banco de Dados (SGBD) MySQL.
- Criamos um ambiente no Oracle Cloud Infrastructure e configuramos uma instância na máquina virtual, garantindo a configuração correta da imagem e das chaves SSH para garantir a segurança do acesso.
- Nessa instância, estabelecemos a conexão com o SGBD MySQL, integrando as tabelas e preparando-as para a geração de gráficos e insights.
- Por fim, desenvolvemos um dashboard no Google Looker Studio com gráficos e insights extremamente valiosos para o credor. São dados de pequenos agricultores que fornecerão aos credores informações essenciais para a concessão de crédito de forma adequada e direcionada desenvolvendo o setor e impulsionando pequenos e médios agricultores.

## 1. PRÉ-PROCESSAMENTO DE DADOS KNIME – TRATANDO MISSING VALUES DA TABELA DE ENDEREÇOS DE ATIBAIA

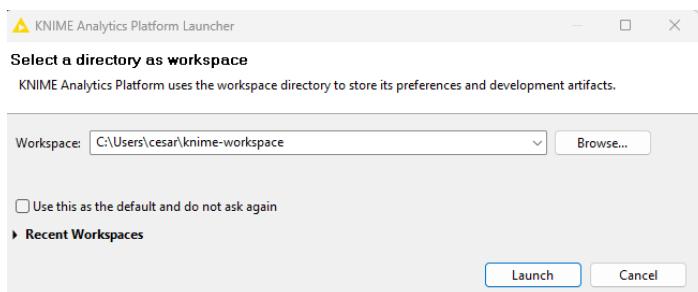
Nesta etapa do projeto, realizamos o pré-processamento da tabela de endereços de produtores rurais do Município de Atibaia na ferramenta Knime. Esta ferramenta permite de forma simples e rápida tratar dados de tabelas relacionais como no caso, missing values, em outras palavras, utilizamos o Knime para remover colunas contendo apenas valores ausentes e realizar o tratamento deles no restante da tabela, deixando-a pronta para utilizá-la em outro ambiente de pré-processamento e tratamento em Python, no caso o Google Colaboratory.

### 1.1 EXECUÇÃO DO KNIME

#### 1.1.1 Abrir o Knime em sua máquina

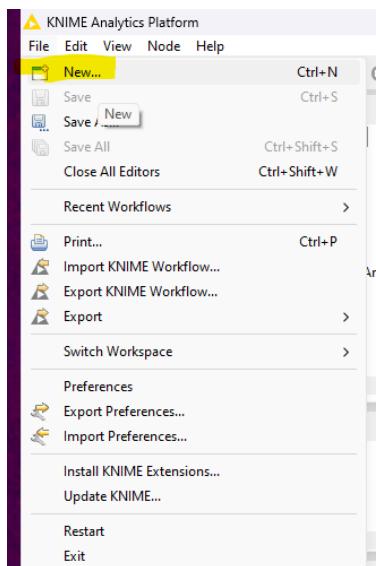


#### 1.1.2. Clicar em Launch



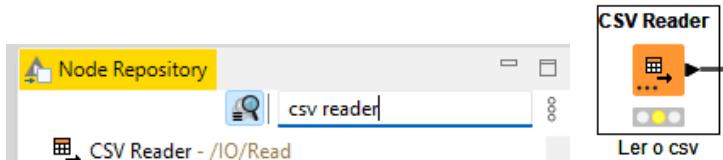
Version 5.1.0  
(Build July 19, 2023)

#### 1.1.3. Gerar um novo Workspace no Knime



## 1.2. CRIAÇÃO DO WORKFLOW

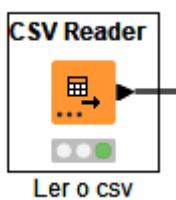
### 1.2.1. No Node Repository escreva CSV Reader e arraste-o até o workflow.



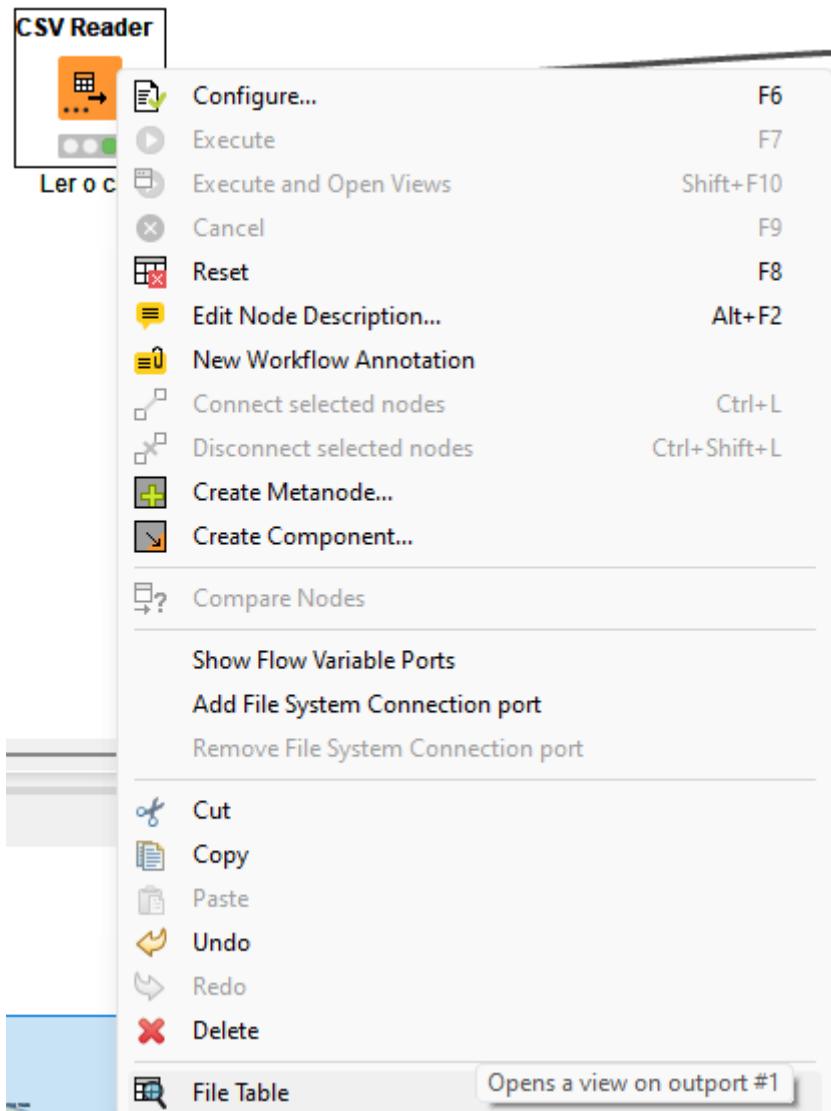
1.2.2. Clique duas vezes no node CSV Reader e configure conforme abaixo. Na aba “Read from”, coloque a opção Local File System, marque a opção “File” abaixo e coloque o caminho onde seu arquivo .csv está salvo na máquina local. Após o arquivo carregar, note que há pontos de interrogação na tabela (?) estes são os Missing Values que vamos tratar, por fim, clique em Apply e Ok.

Row ID	S Column0	S Column1	S Column2	S Column3	S Column4	S Column5	S Column6	S Column7	S Column8	S Column9
Row0	COD_UF	COD_MUNICIPIO	COD_DISTRITO	COD_SUBDISTRITO	SITUACAO	NOM_TIPO_SEGLOGR	NOM_TITULO_SEGLOGR	NOM_SEGLOGR	NUM_ENDERECHO	DSC_MODI
Row1	35	04107	05	00	2	AVENIDA	?	JERONIMO C...	?	SN
Row2	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	?	SN
Row3	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	2193.0	?
Row4	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	?	SN
Row5	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	?	SN
Row6	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	?	SN
Row7	35	04107	05	00	2	ESTRADA	?	DA USINA	220.0	?
Row8	35	04107	05	00	2	ESTRADA	?	DA USINA	?	SN
Row9	35	04107	05	00	2	ESTRADA	?	HISAICHI TA...	?	SN
Row10	35	04107	05	00	2	ESTRADA	?	HISAICHI TA...	425.0	?
Row11	35	04107	05	00	2	ESTRADA	?	HISAICHI TA...	425.0	?
Row12	35	04107	05	00	2	ESTRADA	?	HISAICHI TA...	?	SN
Row13	35	04107	05	00	2	RUA	?	INOSUKE ANDO	?	SN
Row14	35	04107	05	00	2	RUA	?	GIRASSOL	?	SN
Row15	35	04107	05	00	2	ESTRADA	?	FAZENDA SO...	?	SN
Row16	35	04107	05	00	2	ESTRADA	?	AMAZONAS	?	SN
Row17	35	04107	05	00	2	ESTRADA	?	AMAZONAS	?	SN
Row18	35	04107	05	00	2	RUA	?	SABUI	?	SN
Row19	35	04107	05	00	2	ESTRADA	?	MARACANA	?	SN
Row20	35	04107	05	00	2	ESTRADA	?	MARACANA	?	SN
Row21	35	04107	05	nn	?	ESTRADA	?	IMARACANA	?	SN

1.2.3. Pressione F7 para executar o node.



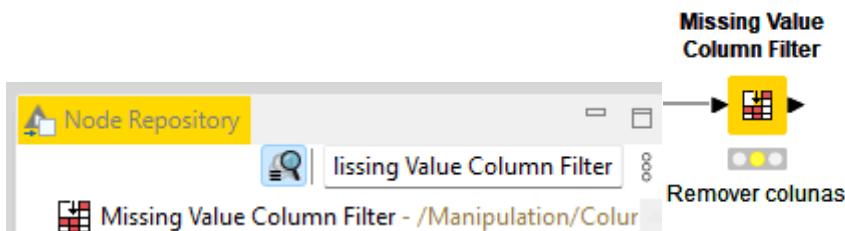
1.2.4. Após executado, clique com o botão direito no node CSV Reader e selecione a opção File Table para visualizar a tabela gerada



1.2.5. Na tabela gerada anote as colunas que possuem TODOS os dados como missing values (?). No caso são as colunas “Column 12” até a “Column 19”, que iremos excluir no próximo node.

File	Edit	Hilite	Navigation	View	(... or col)																							
Table "señal", Rami 602 Spec. Columns: 26 Properties: Row Variables																												
Row ID	S	Column1	S	Column3	S	Column5	S	Column7	S	Column9	S	Column11	S	Column13	S	Column15	S	Column17	S	Column19	S	Column21	S	Column23	S	Column25	S	Column27
Ram001	1	ESTRUCTURA	NOM_TIPO_SEÑAL	NON_TITUL_TIPSEÑAL	NOM_3SEG_OGR	NUM_ID_NIFERO	DSG_MODIFICADOR	NOM_COMP_ELEM	VAL_COMP_ELEM																			
Ram002	1	AHENDA	PERIODICO	PERIODICO	JERONIMO C.	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram003	2	ESTRADA	PERIODICO	PERIODICO	PERIODICO	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram004	2	ESTRADA	PERIODICO	PERIODICO	BENTO SOAMES	2183.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram005	2	ESTRADA	PERIODICO	PERIODICO	BENTO SOAMES	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram006	2	ESTRADA	PERIODICO	PERIODICO	BENTO SOAMES	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram007	2	ESTRADA	PERIODICO	PERIODICO	DA USINA	220.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram008	2	ESTRADA	PERIODICO	PERIODICO	DA USINA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram009	2	ESTRADA	PERIODICO	PERIODICO	HSIAOCHAI TA...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram010	2	ESTRADA	PERIODICO	PERIODICO	HSIAOCHAI TA...	425.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram011	2	ESTRADA	PERIODICO	PERIODICO	HSIAOCHAI TA...	425.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram012	2	ESTRADA	PERIODICO	PERIODICO	HSIAOCHAI TA...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram013	2	RUA	PERIODICO	PERIODICO	HSIAOCHAI TA...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram014	2	ESTRADA	PERIODICO	PERIODICO	HSIAOCHAI TA...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram015	2	ESTRADA	PERIODICO	PERIODICO	PATRIZIA SO...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram016	2	ESTRADA	PERIODICO	PERIODICO	AMAZINAS	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram017	2	ESTRADA	PERIODICO	PERIODICO	AMAZINAS	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram018	2	RUA	PERIODICO	PERIODICO	AMAZINAS	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram019	2	ESTRADA	PERIODICO	PERIODICO	AMAZINAS	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram020	2	ESTRADA	PERIODICO	PERIODICO	AMAZINAS	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram021	2	ESTRADA	PERIODICO	PERIODICO	AMAZINAS	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram022	2	ESTRADA	PERIODICO	PERIODICO	AMAZINAS	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram023	2	ESTRADA	PERIODICO	PERIODICO	STAIPETINGA	1002.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram024	2	ESTRADA	PERIODICO	PERIODICO	STAIPETINGA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram025	2	ESTRADA	PERIODICO	PERIODICO	STAIPETINGA	413.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram026	2	ESTRADA	PERIODICO	PERIODICO	STAIPETINGA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram027	2	ESTRADA	PERIODICO	PERIODICO	STAIPETINGA	413.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram028	2	ESTRADA	PERIODICO	PERIODICO	STAIPETINGA	1000.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram029	2	ESTRADA	PERIODICO	PERIODICO	STAIPETINGA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram030	2	ESTRADA	PERIODICO	PERIODICO	STAIPETINGA	158.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram031	2	AHENDA	PERIODICO	PERIODICO	STAIPETINGA	158.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram032	2	AHENDA	PERIODICO	PERIODICO	AMADOR REC.	915.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram033	2	ESTRADA	PERIODICO	PERIODICO	AMADOR REC.	915.0	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram034	2	ESTRADA	PERIODICO	PERIODICO	SPN SONHO	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram035	2	ESTRADA	PERIODICO	PERIODICO	DOS PEREIRAS	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram036	2	ESTRADA	PERIODICO	PERIODICO	ELVIA DE BR...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram037	2	ESTRADA	PERIODICO	PERIODICO	ELVIA DE BR...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram038	2	ESTRADA	PERIODICO	PERIODICO	ELVIA DE BR...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram039	2	ESTRADA	PERIODICO	PERIODICO	ELVIA DE BR...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram040	2	ESTRADA	PERIODICO	PERIODICO	ELVIA DE BR...	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram041	2	ESTRADA	PERIODICO	PERIODICO	MITSUB	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram042	2	ESTRADA	PERIODICO	PERIODICO	KIUCHI	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram043	2	ESTRADA	PERIODICO	PERIODICO	MINNETTO	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram044	2	ESTRADA	PERIODICO	PERIODICO	YOSHIDA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram045	2	ESTRADA	PERIODICO	PERIODICO	YOSHIDA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram046	2	ESTRADA	PERIODICO	PERIODICO	YOSHIDA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram047	2	ESTRADA	PERIODICO	PERIODICO	YOSHIDA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram048	2	ESTRADA	PERIODICO	PERIODICO	YOSHIDA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram049	2	ESTRADA	PERIODICO	PERIODICO	YOSHIDA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram050	2	ESTRADA	PERIODICO	PERIODICO	YOSHIDA	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram051	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram052	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram053	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram054	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram055	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram056	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram057	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram058	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram059	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ram060	2	ESTRADA	PERIODICO	PERIODICO	TADU HORU	7	SN	7</td																				

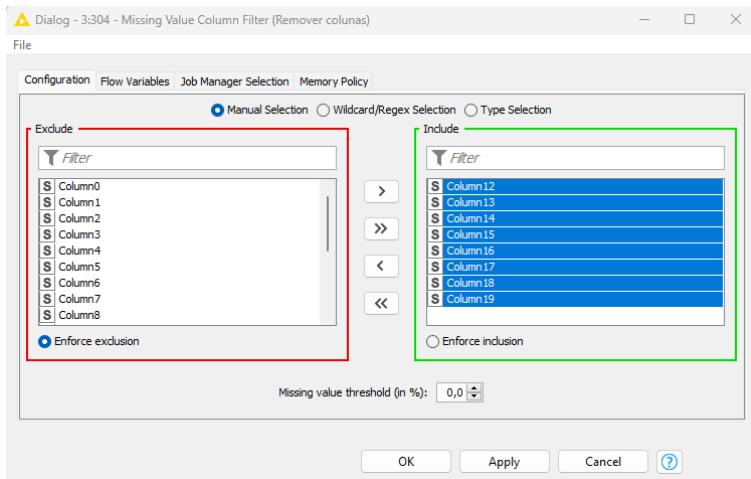
1.2.6. No Node Repository escreva Missing Value Column Filter, e arraste-o até o workflow.



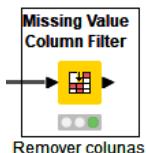
1.2.7. Conecte a saída do node CSV Reader ao node Missing Value Column Filter conforme abaixo.



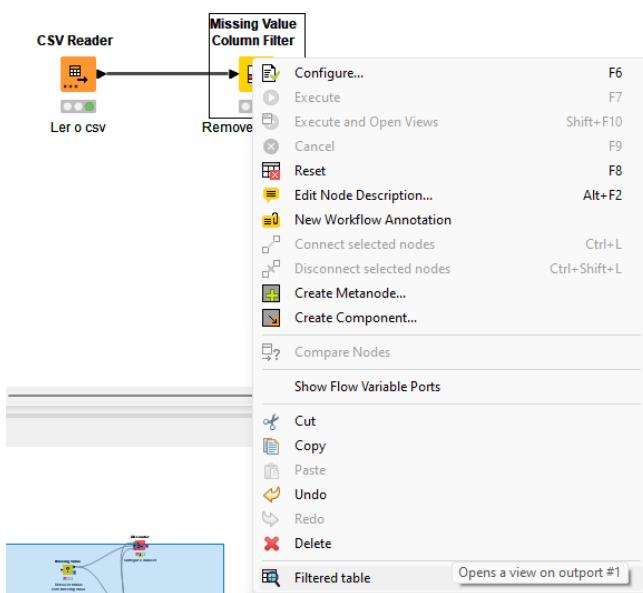
1.2.8. Clique duas vezes no node Missing Value Column Filter e selecione as colunas: Column 12, Column 13, Column 14, Column 15, Column 16, Column 17, Column 18, Column 19. Passe-os para o lado direito da janela de filtro conforme imagem abaixo. Por fim, clique em Apply e Ok.



1.2.9. Pressione F7 para executar o node



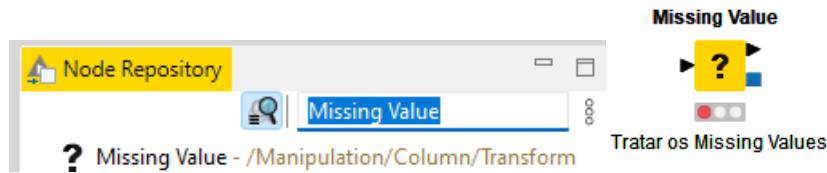
1.2.10. Clique com o botão direito no node Missing Value Column Filter e selecione Filtered Table para visualizar a tabela com as colunas com todos os dados missing values já removidas



### 1.2.11. Na tabela gerada, note que as colunas foram removidas com sucesso

Filtered table - 3:304 - Missing Value Column Filter (Remover colunas)																										
File Edit Help Navigation View																										
Table "default" - Rows: 602 Spec - Columns: 18 Properties Flow Variables																										
Row ID	\$ Column0	\$ Column1	\$ Column2	\$ Column3	\$ Column4	\$ Column5	\$ Column6	\$ Column7	\$ Column8	\$ Column9	\$ Column10	\$ Column11	\$ Column12	\$ Column13	\$ Column14	\$ Column15	\$ Column16	\$ Column17	\$ Column18	\$ Column19	\$ Column20	\$ Column21	\$ Column22	\$ Column23	\$ Column24	\$ Column25
Row0	0	COD_UF	COD_MUNICIPIO	COD_DISTRITO	COD_SUBDISTRITO	SITUACAO	NOM_TITULO_SEGLOGR	NOM_SEGOGR	NUM_ENDERECHO	CSC_MODIFICADOR	NOM_COMP_ELEM1	VAL_COMP_ELEM1	LATITUDE	LONGITUDE	ALTAIDE	DESCRICA	ESPECIE	CP	ALTADE	DESCRICA	ESPECIE	CP	ALTADE	DESCRICA	ESPECIE	CP
Row1	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	?	SN	?	?	-23.071881...,-46.679563...	753.0	BARRIO CAMPO	3	12940970									
Row2	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	2193.0	SN	?	?	-23.081098...,-46.651175...	726.0	BARRIO DA LUSINA	3	12940970									
Row3	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	?	SN	?	?	-23.085945...,-46.669959...	758.0	BARRIO CAMPO	3	12940970									
Row4	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	?	SN	?	?	-23.086161...,-46.669643...	757.0	BARRIO CAMPO	3	12940970									
Row5	35	04107	05	00	2	ESTRADA	?	BENTO SOARES	?	SN	?	?	-23.084793...,-46.671929...	757.0	BARRIO CAMPO	3	12940970									
Row6	35	04107	05	00	2	ESTRADA	?	DA USINA	220.0	SN	?	?	-23.065886...,-46.689426...	773.0	BARRIO DA LUSINA	3	12940970									
Row7	35	04107	05	00	2	ESTRADA	?	DA USINA	?	SN	?	?	-23.060422...,-46.689426...	773.0	BARRIO DA LUSINA	3	12940970									
Row8	35	04107	05	00	2	ESTRADA	?	HISAICHE TA...	?	SN	?	?	-23.064402...,-46.658188...	766.0	BARRIO DA LUSINA	3	12940970									
Row9	35	04107	05	00	2	ESTRADA	?	HISAICHE TA...	425.0	SN	?	?	-23.065436...,-46.669768...	773.0	BARRIO DA LUSINA	3	12940970									
Row10	35	04107	05	00	2	ESTRADA	?	HISAICHE TA...	?	SN	?	?	-23.065436...,-46.669768...	773.0	BARRIO DA LUSINA	3	12940970									

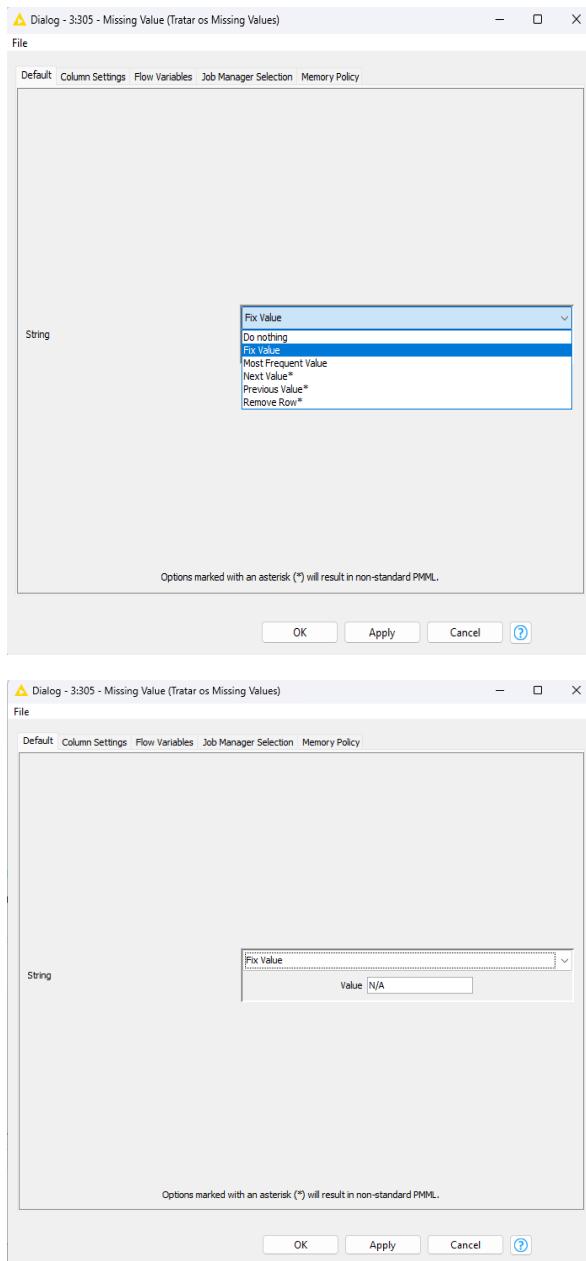
### 1.2.12. Agora é o momento de tratar esses ruídos (missing values), no Node Repository escreva Missing Value, arraste o node até o workflow



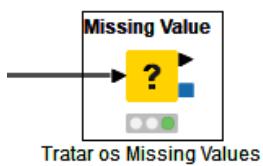
### 1.2.13. Conecte a seta do node Missing Value Column Filter ao node Missing Value conforme abaixo.



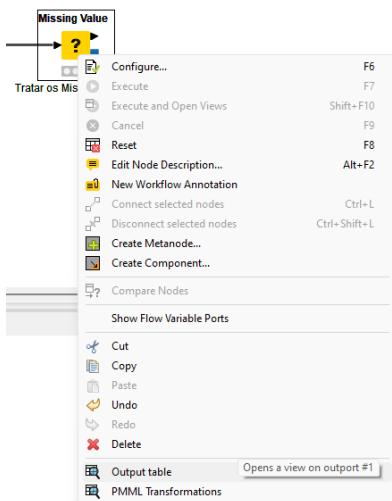
1.2.14. Clique duas vezes no node Missing Value e selecione a opção Fix Value e escreva N/A. Clique em Apply e Ok



1.2.15. Pressione F7 para executar o node



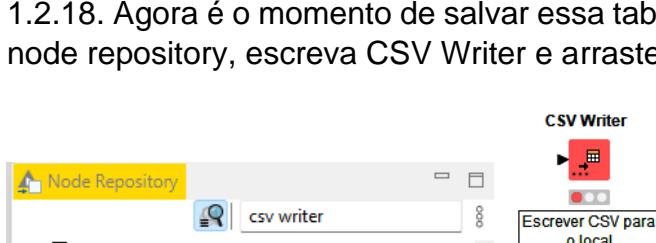
**1.2.16. Clique com o botão direito no node Missing Value e selecione a opção Output Table para visualizar a tabela**



**1.2.17. Ao visualizar a tabela note que os missing values foram transformados em N/A, pois, ao importarmos esse dataset para o Google Looker, conseguimos realizar determinados filtros, concatenações e edições na tabela para remover o N/A dos endereços dos produtores.**

Row ID	S Column1	S Column2	S Column3	S Column4	S Column5	S Column6	S Column7	S Column8	S Column9	S Column10	S Column11	S Column12	S Column13	S Column14	S Column15		
Row1	COD_MUNICIPID	COD_MUNICIPID	DISTRITO	SUBDISTRITO	ESTRADA	NOME_TITULO_SEGURO	NOME_SEGURO	NOME_INFERIOR	DSCE_AHORRADOR	NOM_CNPJ_EBM	VAL_CNPJ_EBM	LATTITUDE	LONGITUDE	ALTIITUDE	EPIC_LOCALIDADE	EPIC_ESPECIE	
Row2	35	0407	05	00	2	AVENIDA	N/A	N/A	N/A	N/A	-23.109265,-	-46.5219	740,0		RESSACA	3	
Row3	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	N/A	N/A	-23.071081,-	-46.679593,-	753,0	BAIRRO CAMPO	3	12940970	
Row4	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	2193,0	N/A	N/A	-23.081099,-	-46.651375	726,0	BAIRRO DA LUSIA	3	12940970
Row5	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	N/A	N/A	-23.081099,-	-46.651375	726,0	BAIRRO DA LUSIA	3	12940970	
Row6	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	N/A	N/A	-23.081099,-	-46.651375	726,0	BAIRRO DA LUSIA	3	12940970	
Row7	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	N/A	N/A	-23.081099,-	-46.651375	726,0	BAIRRO DA LUSIA	3	12940970	
Row8	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	N/A	N/A	-23.081099,-	-46.651375	726,0	BAIRRO DA LUSIA	3	12940970	
Row9	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	N/A	N/A	-23.081099,-	-46.651375	726,0	BAIRRO DA LUSIA	3	12940970	
Row10	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	425,0	N/A	N/A	-23.065436,-	-46.669768,-	773,0	BAIRRO DA LUSIA	3	12940970
Row11	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	425,0	N/A	N/A	-23.064534,-	-46.669699,-	751,0	BAIRRO DA LUSIA	3	12940970
Row12	35	0407	05	00	2	ESTRADA	N/A	BENTO SOARES	425,0	N/A	N/A	-23.064534,-	-46.669699,-	751,0	BAIRRO DA LUSIA	3	12940970
Row13	35	0407	05	00	2	RUA	N/A	INOSUKE ANDO	N/A	N/A	N/A	-23.110154,-	-46.538031,-	784,0	GUANANDUVA	3	12945701
Row14	35	0407	05	00	2	RUA	N/A	GRASSOL	N/A	N/A	N/A	-23.109176,-	-46.505623,-	764,0	GUANANDUVA	3	12940960
Row15	35	0407	05	00	2	ESTRADA	N/A	FAZENDA SO...	N/A	N/A	N/A	-23.117786,-	-46.510238,-	835,0	GUANANDUVA	3	12940910
Row16	35	0407	05	00	2	ESTRADA	N/A	AMAZONAS	N/A	N/A	N/A	-23.166775,-	-46.514578,-	786,0	MARACANA	3	12940950
Row17	35	0407	05	00	2	ESTRADA	N/A	AMAZONAS	N/A	N/A	N/A	-23.166775,-	-46.514578,-	786,0	MARACANA	3	12940950
Row18	35	0407	05	00	2	RUA	N/A	SABU	N/A	N/A	N/A	-23.165316,-	-46.12773,-	766,0	BAIRRO CAIÇO	3	12940960
Row19	35	0407	05	00	2	ESTRADA	N/A	MARACANA	N/A	N/A	N/A	-23.167756,-	-46.151533,-	779,0	MARACANA	3	12940960
Row20	35	0407	05	00	2	ESTRADA	N/A	MARACANA	N/A	N/A	N/A	-23.167756,-	-46.144861,-	780,0	MARACANA	3	12940960
Row21	35	0407	05	00	2	ESTRADA	N/A	MARACANA	N/A	N/A	N/A	-23.165021,-	-46.144861,-	787,0	MARACANA	3	12940960
Row22	35	0407	05	00	2	ESTRADA	N/A	MARACANA	N/A	N/A	N/A	-23.164595,-	-46.1487	782,0	MARACANA	3	12940960
Row23	35	0407	05	00	2	ESTRADA	N/A	ITAPETINGA	1020,0	N/A	N/A	-23.159281,-	-46.575974,-	829,0	BAIRRO ITAPETI	3	12940960
Row24	35	0407	05	00	2	ESTRADA	N/A	ITAPETINGA	N/A	N/A	N/A	-23.17299,-	-46.575974,-	761,0	BAIRRO ITAPETI	3	12940960
Row25	35	0407	05	00	2	ESTRADA	N/A	ITAPETINGA	1020,0	N/A	N/A	-23.17099,-	-46.575974,-	761,0	BAIRRO ITAPETI	3	12940960
Row26	35	0407	05	00	2	ESTRADA	N/A	ITAPETINGA	N/A	N/A	N/A	-23.194041,-	-46.556988,-	933,0	BAIRRO ITAPETI	3	12940960
Row27	35	0407	05	00	2	ESTRADA	N/A	ITAPETINGA	415,0	N/A	N/A	-23.170985,-	-46.570178,-	815,0	BAIRRO ITAPETI	3	12940960
Row28	35	0407	05	00	2	ESTRADA	N/A	ITAPETINGA	1000,0	N/A	N/A	-23.170941,-	-46.555503,-	815,0	BAIRRO ITAPETI	3	12940960
Row29	35	0407	05	00	2	ESTRADA	N/A	ITAPETINGA	N/A	N/A	N/A	-23.171095,-	-46.555503,-	760,0	BAIRRO ITAPETI	3	12940960
Row30	35	0407	05	00	2	ESTRADA	N/A	ITAPETINGA	510,0	N/A	N/A	-23.171958,-	-46.570131,-	827,0	BAIRRO ITAPETI	3	12940960
Row31	35	0407	05	00	2	AVENIDA	N/A	AMADOR REC...	B15,0	N/A	N/A	-23.106061,-	-46.530028,-	748,0	MORIMBI	3	12945000
Row32	35	0407	05	00	2	AVENIDA	N/A	AMADOR REC...	195,0	N/A	N/A	-23.105409,-	-46.527554,-	752,0	MORIMBI	3	12945000
Row33	35	0407	05	00	2	ESTRADA	N/A	AMADOR REC...	195,0	N/A	N/A	-23.105061,-	-46.527554,-	752,0	MORIMBI	3	12945000
Row34	35	0407	05	00	2	ESTRADA	N/A	SEM DENOME...	N/A	N/A	N/A	-23.075779,-	-46.58813,-	907,0	ATIBAIÁ	3	12940970
Row35	35	0407	05	00	2	ESTRADA	N/A	DOS PERNEIS	N/A	N/A	N/A	-23.085475,-	-46.549853,-	779,0	BOA VISTA	3	12940970
Row36	35	0407	05	00	2	ESTRADA	N/A	VELHA DE BR...	1354,0	N/A	N/A	-23.095038,-	-46.557128,-	791,0	BOA VISTA	3	12940933
Row37	35	0407	05	00	2	ESTRADA	N/A	VELHA DE BR...	N/A	N/A	N/A	-23.080151,-	-46.557128,-	791,0	BOA VISTA	3	12940933
Row38	35	0407	05	00	2	ESTRADA	N/A	VELHA DE BR...	N/A	N/A	N/A	-23.075883,-	-46.557128,-	883,0	BOA VISTA	3	N/A
Row39	35	0407	05	00	2	SITTO	N/A	TAKANE	N/A	N/A	N/A	-23.041591,-	-46.577558,-	824,0	BAIRRO DO TANL	3	12940960
Row40	35	0407	05	00	2	SITTO	N/A	OLIVEIRA	N/A	N/A	N/A	-23.037811,-	-46.577558,-	836,0	BAIRRO DO TANL	3	12941110
Row41	35	0407	05	00	2	SITTO	N/A	MITSUICHI	N/A	N/A	N/A	-23.032053,-	-46.577558,-	810,0	BAIRRO DO TANL	3	12940960
Row42	35	0407	05	00	2	SITTO	N/A	KIRUCHI	N/A	N/A	N/A	-23.035471,-	-46.58244,-	793,0	BAIRRO DO TANL	3	12940960
Row43	35	0407	05	00	2	SITTO	N/A	MEAMOTO	N/A	N/A	N/A	-23.037919,-	-46.580558,-	814,0	BAIRRO DO TANL	3	12940960
Row44	35	0407	05	00	2	SITTO	N/A	YOSHIDA	N/A	N/A	N/A	-23.036845,-	-46.577296,-	830,0	BAIRRO DO TANL	3	12940960
Row45	35	0407	05	00	2	SITTO	N/A	YOSHIDA	N/A	N/A	N/A	-23.042805,-	-46.577296,-	830,0	BAIRRO DO TANL	3	12940960
Row46	35	0407	05	00	2	SITTO	N/A	TAKANE 2	N/A	N/A	N/A	-23.041114,-	-46.575845,-	816,0	BAIRRO DO TANL	3	12940970
Row47	35	0407	05	00	2	SITTO	N/A	HAYAKAWA	N/A	N/A	N/A	-23.042040,-	-46.572185,-	833,0	BAIRRO DO TANL	3	12940960
Row48	35	0407	05	00	2	SITTO	N/A	HAYAKAWA	N/A	N/A	N/A	-23.045011,-	-46.574743,-	850,0	BAIRRO DO TANL	3	12940960
Row49	35	0407	05	00	2	SITTO	N/A	HAYAKAWA	N/A	N/A	N/A	-23.042408,-	-46.574743,-	850,0	BAIRRO DO TANL	3	12940960
Row50	35	0407	05	00	2	SITTO	N/A	HAYAKAWA	N/A	N/A	N/A	-23.042518,-	-46.57333,-	828,0	BAIRRO DO TANL	3	12940960
Row51	35	0407	05	00	2	SITTO	N/A	TAIDI KOUJI	N/A	N/A	N/A	-23.024081,-	-46.585600,-	811,0	BAIRRO DO TANL	3	12941220
Row52	35	0407	05	00	2	SITTO	N/A	TAIDI KOUJI 2	N/A	N/A	N/A	-23.025628,-	-46.583634,-	792,0	BAIRRO DO TANL	3	12940960
Row53	35	0407	05	00	2	SITTO	N/A	SUZUKI	N/A	N/A	N/A	-23.024081,-	-46.583634,-	792,0	BAIRRO DO TANL	3	12940960
Row54	35	0407	05	00	2	SITTO	N/A	TSUKA	N/A	N/A	N/A	-23.031833,-	-46.579903,-	825,0	BAIRRO DO TANL	3	12940960
Row55	35	0407	05	00	2	SITTO	N/A	TANABE	N/A	N/A	N/A	-23.025805,-	-46.59923,-	776,0	BAIRRO DO TANL	3	12940960
Row56	35	0407	05	00	2	SITTO	N/A	KOBAYASHI	N/A	N/A	N/A	-23.024805,-	-46.586659,-	797,0	BAIRRO DO TANL	3	12940960
Row57	35	0407	05	00	2	SITTO	N/A	MIYAMOTO	N/A	N/A	N/A	-23.032913,-	-46.581306,-	857,0	BAIRRO DO TANL	3	12940960

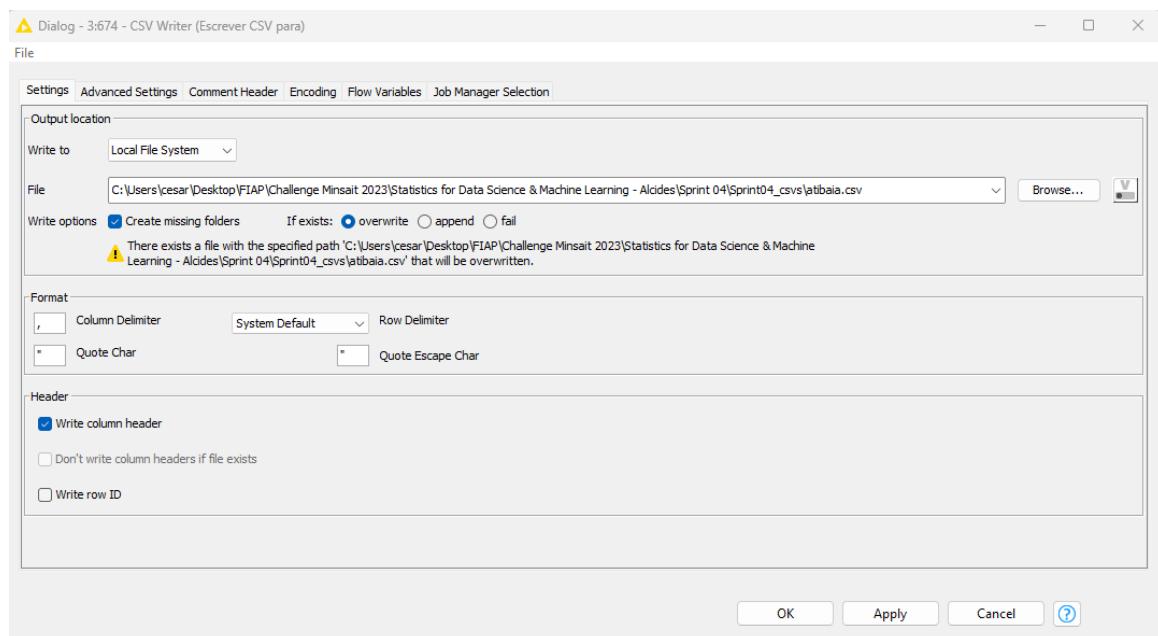
**1.2.18. Agora é o momento de salvar essa tabela tratada no dispositivo local. No node repository, escreva CSV Writer e arraste o node até o workflow.**



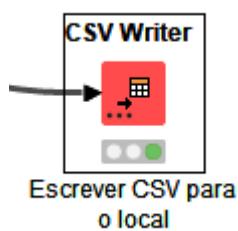
1.2.19. Conecte a saída do node Missing Value Column Filter ao node Missing Value conforme abaixo.



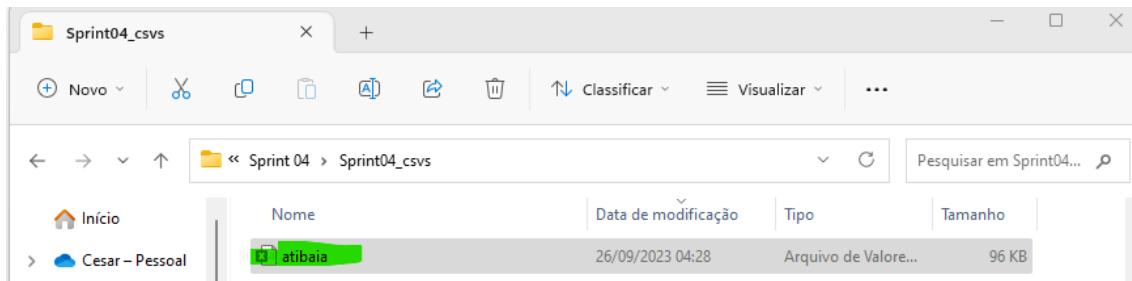
1.2.20. Clique duas vezes no node CSV Writer, no campo Write To selecione Local File System, no campo File coloque o caminho do diretório que deseja salvar o .csv, no campo Write Options selecione Create Missing Folders e Overwrite. Clique em Apply e Ok.



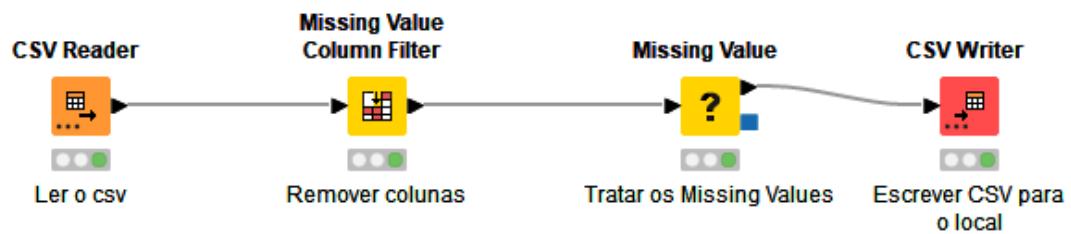
1.2.21. Pressione F7 para executar o node



1.2.22. Vá no diretório local escolhido e verifique se o .csv foi gerado com sucesso



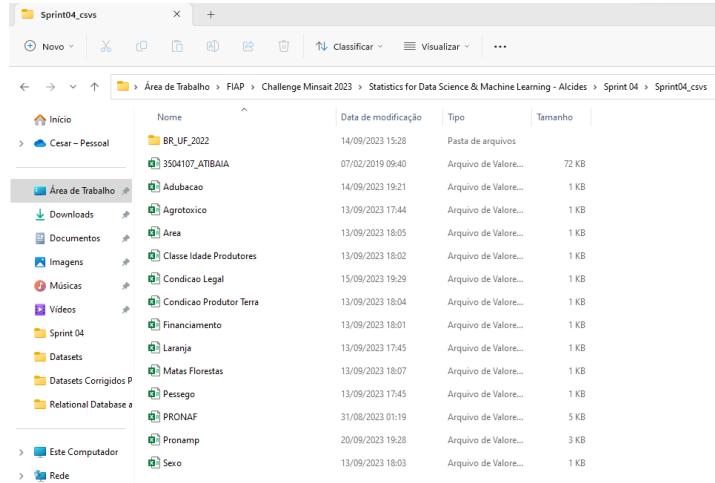
1.2.23. Ao final de tudo o workflow deverá ficar como a imagem abaixo



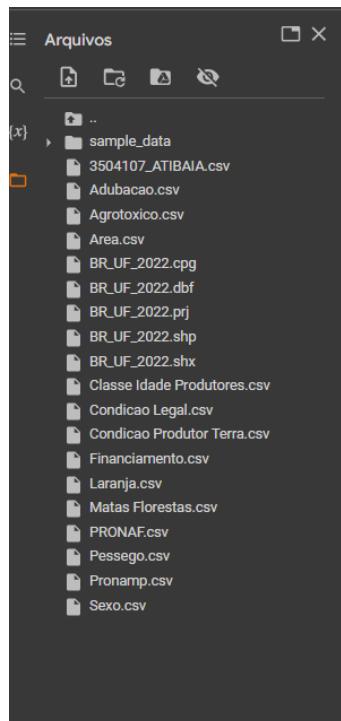
## 2. PRÉ-PROCESSAMENTO DE DADOS GOOGLE COLABORATORY

### 2.1 IMPORTANDO OS DATASETS PARA O AMBIENTE DE ANÁLISE PYTHON

2.1.1. Selezionamos os arquivos csv salvos em um dispositivo local ou outro repositório de preferência. Note que todos esses csv foram baixados de fontes de dados públicas que serão referenciadas ao final desta documentação.



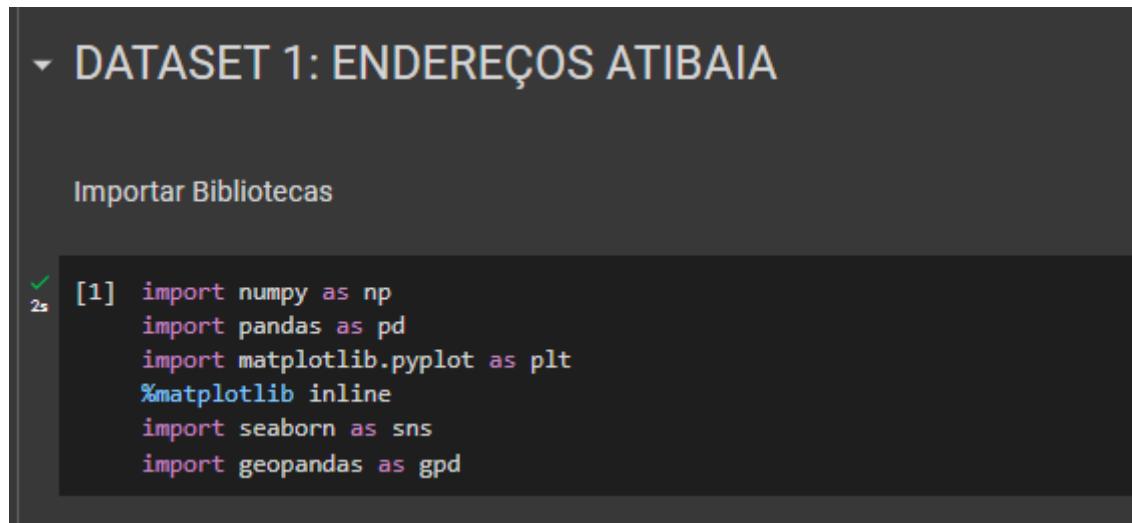
2.1.2. Após isso, carregue os datasets para o Google Colab conforme abaixo:



## 2.2. TRATAMENTO DE DADOS DO DATASET LISTA DE ENDEREÇOS DE ATIBAIA

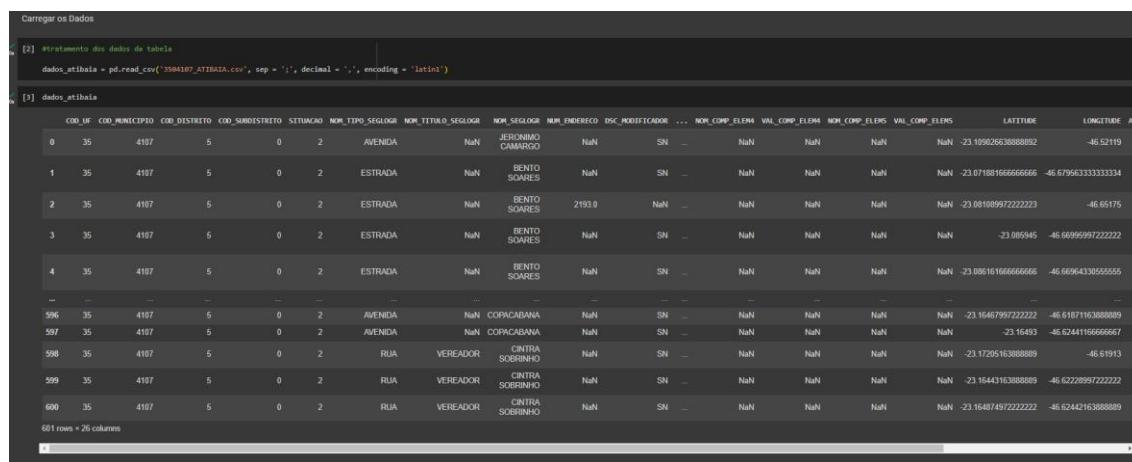
Nesta etapa, realizamos o pré-processamento dos dados utilizando o Google Colab para simular um ambiente Python de Data Science.

2.2.1. Importamos as bibliotecas abaixo para podermos trabalhar com a manipulação dos dados e geração de gráficos.



```
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import geopandas as gpd
```

2.2.2. Fizemos a leitura da tabela de endereços de Atibaia chamada “3504107\_ATIBAIA.csv”.



```
[2] #tratamento dos dados da tabela
dados_atibaia = pd.read_csv('3504107_ATIBAIA.csv', sep = ';', decimal = ',', encoding = 'latin1')

[3] dados_atibaia
```

COD_DF	COD_MUNICIPIO	COD_DISTRITO	COD_SUBDISTRITO	SITUAÇÃO	NOM_TIPO_SELOGR	NOM_TITULO_SELOGR	NOM_SELOGR	NUM_ENDERECHO	DESC_MODIFICADOR	...	NOM_COMP_ELEM1	VAL_COMP_ELEM1	NOM_COMP_ELEM2	VAL_COMP_ELEM2	LATITUDE	LONGITUDE	A1	
0	35	4107	5	0	2	AVENIDA	NaN	JERONIMO CAMARGO	NaN	SN	—	NaN	NaN	NaN	NaN	-23.10902663888892	-46.52119	
1	35	4107	5	0	2	ESTRADA	NaN	BENTO SOARES	NaN	SN	—	NaN	NaN	NaN	NaN	-23.071881666666666	-46.67956333333334	
2	35	4107	5	0	2	ESTRADA	NaN	BENTO SOARES	2193.0	NaN	—	NaN	NaN	NaN	NaN	-23.08108997222223	-46.69175	
3	35	4107	5	0	2	ESTRADA	NaN	BENTO SOARES	NaN	SN	—	NaN	NaN	NaN	NaN	-23.085945	-46.6699599722222	
4	35	4107	5	0	2	ESTRADA	NaN	BENTO SOARES	NaN	SN	—	NaN	NaN	NaN	NaN	-23.086161666666666	-46.66964330555555	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
596	35	4107	5	0	2	AVENIDA	NaN	COPACABANA	NaN	SN	—	NaN	NaN	NaN	NaN	-23.16467997222222	-46.61671163888895	
597	35	4107	5	0	2	AVENIDA	NaN	COPACABANA	NaN	SN	—	NaN	NaN	NaN	NaN	-23.16493	-46.62441166666667	
598	35	4107	5	0	2	RUA	VEREADOR	CINTRA SOBRINHO	NaN	SN	—	NaN	NaN	NaN	NaN	-23.1726163888889	-46.61913	
599	35	4107	5	0	2	RUA	VEREADOR	CINTRA SOBRINHO	NaN	SN	—	NaN	NaN	NaN	NaN	-23.16441638888895	-46.6222899722222	
600	35	4107	5	0	2	RUA	VEREADOR	CINTRA SOBRINHO	NaN	SN	—	NaN	NaN	NaN	NaN	-23.16487497222222	-46.62441638888895	

601 rows × 26 columns

## 2.2.3. Eliminamos as colunas vazias do dataset.

```
Tratar os dados para análises

[4] # eliminar as colunas vazias
dados_atibaia = dados_atibaia.dropna(axis=1, how='all')

dados_atibaia
```

	COD_UF	COD_MUNICIPIO	COD_BAIRRO	COD_SUBDISTRITO	SITUAÇÃO	NOM_TIPO_SEGLOR	NOM_TITULO_SEGLOR	NOM_SEGLOR	NUM_ENDERECHO	DESC_MODIFICADOR	NOM_CNPJ_ELETR	VAL_CNPJ_ELETR	LATITUDE	LONGITUDE	ALTITUDE	DESC_LOCALIDADE	COD_ESPECIE
0	35	4107	5	0	2	AVENIDA		JERONIMO CAMARGO	NaN	NaN	SN	NaN	-23.10902663888892	-46.52119	740.0	RESSACA	3
1	35	4107	5	0	2	ESTRADA		BENTO SOARES	NaN	SN	NaN	NaN	-23.071881666666666	-46.67956333333334	753.0	BAIRRO CAMPO DOS ALEXIOS	3
2	35	4107	5	0	2	ESTRADA		BENTO SOARES	2193.0	NaN	NaN	NaN	-23.0810897222223	-46.65175	726.0	BAIRRO DA USINA	3
3	35	4107	5	0	2	ESTRADA		BENTO SOARES	NaN	SN	NaN	NaN	-23.085945	-46.669959722222	758.0	BAIRRO CAMPO DOS ALEXIOS	3
4	35	4107	5	0	2	ESTRADA		BENTO SOARES	NaN	SN	NaN	NaN	-23.086161666666666	-46.66964330555555	757.0	BAIRRO CAMPO DOS ALEXIOS	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
596	35	4107	5	0	2	AVENIDA		COPACABANA	NaN	SN	NaN	NaN	-23.1646799722222	-46.6167116388889	796.0	MARACANA	3
597	35	4107	5	0	2	AVENIDA		COPACABANA	NaN	SN	NaN	NaN	-23.16493	-46.62441166666667	NaN	MARACANA	3
598	35	4107	5	0	2	RUA	VEREADOR	CINTRA SOBRINHO	NaN	SN	NaN	NaN	-23.1720516388889	-46.61913	795.0	MARACANA	3
599	35	4107	5	0	2	RUA	VEREADOR	CINTRA SOBRINHO	NaN	SN	NaN	NaN	-23.1644316388889	-46.6222899722222	767.0	MARACANA	3
600	35	4107	5	0	2	RUA	VEREADOR	CINTRA SOBRINHO	NaN	SN	SITIO	SONHO MEU	-23.1648749722222	-46.6244216388889	767.0	MARACANA	3

601 rows × 18 columns

## 2.2.4. Filtramos as colunas com informações relevantes para a nossa análise.

```
[8] # selecionar as colunas com as informações relevantes para nossas análises
dados_atibaia_final = dados_atibaia[['COD_UF', 'COD_MUNICIPIO', 'NOME_TIPO', 'NOME_LOGRADOURO', 'NUM_ENDERECHO', 'NOME_COMPLEMENTO', 'LATITUDE', 'LONGITUDE', 'ALTITUDE', 'NOME_BAIRRO', 'CEP']]

dados_atibaia_final
```

	COD_UF	COD_MUNICIPIO	NOME_TIPO	NOME_LOGRADOURO	NUM_ENDERECHO	NOME_COMPLEMENTO	LATITUDE	LONGITUDE	ALTITUDE	NOME_BAIRRO	CEP
0	35	4107	AVENIDA	JERONIMO CAMARGO	NaN	NaN	-23.10902663888892	-46.52119	740.0	RESSACA	12940060.0
1	35	4107	ESTRADA	BENTO SOARES	NaN	NaN	-23.071881666666666	-46.67956333333334	753.0	BAIRRO CAMPO DOS ALEXIOS	12940970.0
2	35	4107	ESTRADA	BENTO SOARES	2193.0	NaN	-23.0810897222223	-46.65175	726.0	BAIRRO DA USINA	12940970.0
3	35	4107	ESTRADA	BENTO SOARES	NaN	NaN	-23.085945	-46.669959722222	758.0	BAIRRO CAMPO DOS ALEXIOS	12940970.0
4	35	4107	ESTRADA	BENTO SOARES	NaN	NaN	-23.086161666666666	-46.66964330555555	757.0	BAIRRO CAMPO DOS ALEXIOS	12940970.0
...	...	...	...	...	...	...	...	...	...	...	...
596	35	4107	AVENIDA	COPACABANA	NaN	NaN	-23.1646799722222	-46.6167116388889	796.0	MARACANA	12940060.0
597	35	4107	AVENIDA	COPACABANA	NaN	NaN	-23.16493	-46.62441166666667	NaN	MARACANA	12940060.0
598	35	4107	RUA	CINTRA SOBRINHO	NaN	NaN	-23.1720516388889	-46.61913	795.0	MARACANA	12940060.0
599	35	4107	RUA	CINTRA SOBRINHO	NaN	NaN	-23.1644316388889	-46.6222899722222	767.0	MARACANA	12940060.0
600	35	4107	RUA	CINTRA SOBRINHO	NaN	SITIO	-23.1648749722222	-46.6244216388889	767.0	MARACANA	12940060.0

601 rows × 11 columns

## 2.2.5. Próximo passo foi desenvolver algumas análises, como por exemplo descobrir qual o bairro e CEP que mais se repete (moda) neste dataset, demonstrando uma aglomeração de agricultores nestes locais.

```
[10] # uma das análises válidas é descobrir a MODA do CEP e NOME_BAIRRO, NOME_LOGRADOURO e NOME_TIPO, desta forma, descobriremos quais os Bairros, CEP, Logradouro e Tipo que mais se repetem para os produtores registrados

moda_cep = dados_atibaia_final['CEP'].mode()[0]
moda_bairro = dados_atibaia_final['NOME_BAIRRO'].mode()[0]
moda_logradouro = dados_atibaia_final['NOME_LOGRADOURO'].mode()[0]
moda_tipo = dados_atibaia_final['NOME_TIPO'].mode()[0]

[11] moda_cep
12940060.0

[12] moda_bairro
'BAIRRO DO TANQUE'

[13] moda_logradouro
'NOVA UNIAO'

[14] moda_tipo
'ESTRADA'
```

2.2.6. Um dos insights interessantes para realizar neste dataset são os dados geográficos como altitude, latitude e longitude, que podem ser extremamente úteis para definir o tipo de cultura, avaliação do terreno, e claro, definir a localização exata da propriedade.

```
# Outra análise útil é encontrar mínimo, máximo e média da altitude das propriedades dos produtores cadastrados, pois, são informações úteis para determinar o que será cultivado em determinado local
# uma vez que determinados tipos de produtos não crescem em qualquer altitude
dados_atibaia_final_altitude = dados_atibaia_final.loc[dados_atibaia_final['ALTITUDE'] != 'NaN']

[16] dados_atibaia_final_altitude
```

COD_UF	COD_MUNICIPIO	NOME_TIPO	NOME_LOGRADOURO	NUM_ENDERECHO	NOME_COMPLEMENTO	LATITUDE	LONGITUDE	ALTITUDE	NOME_BAIRRO	CEP
0	35	4107	AVENIDA JERONIMO CAMARGO	NaN	NaN	-23.109026638888892	-46.52119	740.0	RESSACA	12940060.0
1	35	4107	ESTRADA BENTO SOARES	NaN	NaN	-23.071881666666666	-46.67956333333334	753.0	BAIRRO CAMPO DOS ALEXIOS	12940970.0
2	35	4107	ESTRADA BENTO SOARES	2193.0	NaN	-23.08108997222223	-46.65175	726.0	BAIRRO DA USINA	12940970.0
3	35	4107	ESTRADA BENTO SOARES	NaN	NaN	-23.085945	-46.6699599722222	758.0	BAIRRO CAMPO DOS ALEXIOS	12940970.0
4	35	4107	ESTRADA BENTO SOARES	NaN	NaN	-23.086161666666666	-46.6696433055555	757.0	BAIRRO CAMPO DOS ALEXIOS	12940970.0
...	...	...	...	...	...	...	...	...	...	...
596	35	4107	AVENIDA COPACABANA	NaN	NaN	-23.1646799722222	-46.61871163888889	796.0	MARACANA	12940060.0
597	35	4107	AVENIDA COPACABANA	NaN	NaN	-23.16493	-46.62441166666667	NaN	MARACANA	12940060.0
598	35	4107	RUA CINTRA SOBRINHO	NaN	NaN	-23.17205163888889	-46.61913	795.0	MARACANA	12940060.0
599	35	4107	RUA CINTRA SOBRINHO	NaN	NaN	-23.16443163888889	-46.6222899722222	767.0	MARACANA	12940060.0
600	35	4107	RUA CINTRA SOBRINHO	NaN	SITIO	-23.1648749722222	-46.62442163888889	767.0	MARACANA	12940060.0

601 rows × 11 columns

```
[17] dados_atibaia_final_altitude['ALTITUDE'] = dados_atibaia_final_altitude['ALTITUDE'].astype(float)

[18] dados_atibaia_final_altitude.groupby('COD_MUNICIPIO') \
    .agg(altitude_max = pd.NamedAgg('ALTITUDE','max'),
         altitude_min = pd.NamedAgg('ALTITUDE','min'),
         altitude_media = pd.NamedAgg('ALTITUDE','mean')) \
    .reset_index()
```

COD_MUNICIPIO	altitude_max	altitude_min	altitude_media
0	984.0	708.0	785.035473

2.2.7. Por fim, fizemos o download deste dataset para usá-lo na criação dos dashboards com as informações e insights gerados.

```
from google.colab import files
dados_agrotoxico_final.to_csv('dados_atibaia_final.csv')
files.download('dados_atibaia_final.csv')
```

## 2.3. TRATAMENTO DE DADOS DO DATASET PRONAMP E PRONAF DO BANCO CENTRAL

Nesta etapa, realizamos o pré-processamento dos dados utilizando o Google Colab para simular um ambiente Python de Data Science.

### 2.3.1. Fizemos a leitura dos arquivos extraídos das fontes de dados públicas do Banco Central.

```
[21] #Leitura e tratamento das tabelas, separadas por programa para cobrir cada um mais minuciosamente
dados_bcb_pronaf = pd.read_csv('PRONAF.csv', sep = ';', decimal = ',', thousands='.', encoding = 'utf-8-sig')
dados_bcb_pronamp = pd.read_csv('Pronamp.csv', sep = ';', decimal = ',', thousands='.', encoding = 'utf-8-sig')
```

### 2.3.2. Criamos listas com os Estados/UF para melhor manipulação e aplicação de funções.

```
#lista separada por regiões
centro_oeste = ["DF", "GO", "MT", "MS"]
nordeste = ["AL", "BA", "CE", "MA", "PB", "PE", "PI", "RN", "SE"]
norte = ["AC", "AP", "AM", "PA", "RO", "RR", "TO"]
sudeste = ["SP", "MG", "RJ", "ES"]
sul = ["RS", "SC", "PR"]
nacional = ["DF", "GO", "MT", "MS", "AL", "BA", "CE", "MA", "PB", "PE", "PI", "RN", "SE", "AC", "AP", "AM", "PA", "RO", "RR", "TO", "SP", "MG", "RJ", "ES", "RS", "SC", "PR"]

dados_centro_oeste_pronamp = dados_bcb_pronamp[dados_bcb_pronamp['UF'].isin(centro_oeste)]
dados_nordeste_pronamp = dados_bcb_pronamp[dados_bcb_pronamp['UF'].isin(nordeste)]
dados_norte_pronamp = dados_bcb_pronamp[dados_bcb_pronamp['UF'].isin(norte)]
dados_sudeste_pronamp = dados_bcb_pronamp[dados_bcb_pronamp['UF'].isin(sudeste)]
dados_sul_pronamp = dados_bcb_pronamp[dados_bcb_pronamp['UF'].isin(sul)]
dados_nacional_pronamp = dados_bcb_pronamp[dados_bcb_pronamp['UF'].isin(nacional)]
```

```
[27] dados_centro_oeste_pronaf = dados_bcb_pronaf[dados_bcb_pronaf['UF'].isin(centro_oeste)]
dados_nordeste_pronaf = dados_bcb_pronaf[dados_bcb_pronaf['UF'].isin(nordeste)]
dados_norte_pronaf = dados_bcb_pronaf[dados_bcb_pronaf['UF'].isin(norte)]
dados_sudeste_pronaf = dados_bcb_pronaf[dados_bcb_pronaf['UF'].isin(sudeste)]
dados_sul_pronaf = dados_bcb_pronaf[dados_bcb_pronaf['UF'].isin(sul)]
dados_nacional = dados_bcb_pronaf[dados_bcb_pronaf['UF'].isin(nacional)]
```

2.3.3. Podemos aplicar algumas análises por programa como média, mediana e soma para gerar conhecimento e insights.

- Funções Aplicáveis
- Amostra regional

[28] #Esse valor representa a quantidade de contratos ofertados pelo PRONAMP na região sul do país  
soma\_contratos\_sul = dados\_sul\_pronamp['adesao\_qtdContratos'].sum().round()  
soma\_contratos\_sul

76014

[29] #Esse valor representa a média de contratos ofertados pelo PRONAMP na região sul do país  
media\_contratos\_sul = dados\_sul\_pronamp['adesao\_qtdContratos'].mean().round()  
media\_contratos\_sul

25338.0

[30] #Esse valor representa a mediana do valor amparado pelo PRONAMP na região sul do país  
mediana\_vlrAmparado\_sul = dados\_sul\_pronamp['adesao\_vlrAmparado'].median().round()  
mediana\_vlrAmparado\_sul

4033449557.0

- Amostra Nacional

[32] #Esse valor representa a quantidade de contratos fornecidos no país inteiro pelo PRONAMP  
soma\_qtdContratos\_nacional = dados\_nacional\_pronamp['adesao\_qtdContratos'].sum().round()  
soma\_qtdContratos\_nacional

82492

[33] #Esse valor representa a soma da área amparada pelo PRONAMP no país inteiro  
soma\_areaAmparada\_nacional = dados\_nacional\_pronamp['adesao\_areaAmparada\_ha'].sum().round()  
soma\_areaAmparada\_nacional

3034896.0

[34] #Esse valor representa a média de contratos por estado  
media\_contratos\_nacional = dados\_nacional\_pronamp['adesao\_qtdContratos'].mean().round()  
media\_contratos\_nacional

3437.0

[35] #Esse valor representa a média de área amparada por estado pelo PRONAMP no país inteiro  
media\_areaaAmparada\_nacional = dados\_nacional\_pronamp['adesao\_areaAmparada\_ha'].mean().round()  
media\_areaaAmparada\_nacional

126459.0

[36] #Esse valor representa a soma do valor amparado por estado pelo PRONAMP no país inteiro  
soma\_vlrAmparado\_nacional = dados\_nacional\_pronamp['adesao\_vlrAmparado'].sum().round()  
soma\_vlrAmparado\_nacional

10708999331.0

[37] #Esse valor representa a média do valor amparado por estado pelo PRONAMP tendo em vista a quantidade ofertada no país inteiro  
media\_vlrAmparado\_nacional = dados\_nacional\_pronamp['adesao\_vlrAmparado'].mean().round()  
media\_vlrAmparado\_nacional

445873305.0

2.3.4. Renomeamos as colunas para melhor manipulação dos dados deixando-as padronizadas.

```

Inferências PRONAF e PRONAMP

# Alterar o nome das colunas para facilitar a análise
dados_bcb_pronaf = dados_bcb_pronaf.rename(columns = {"UF": "SIGLA_UF", "Adesão_Area_Aparada_HA": "ADESAO_AREA_AMPARADA_HA", "Adesão_Qtd_Contratos": "ADESAO_QTDE_CONTRATOS", "Adesão_Vlr_Aparado": "ADESAO_VALOR_APPARADO_RS", "Adesão_Vlr_Adicional": "ADESAO_VALOR_ADICIONAL_RS", "com_perdas_Area_Aparada_HA": "COMUNICACAO_PERDA_APPARADA_HA", "com_perdas_Qtd_Contratos": "COMUNICACAO_PERDA_QTD_CONTRATOS", "com_perdas_Vlr_Aparado": "COMUNICACAO_PERDA_VALOR_APPARADO_RS", "com_perdas_Vlr_Adicional": "COMUNICACAO_PERDA_VALOR_APPARADO_RS", "cobert_def_Vlr_Aparado": "COBERTURA_DEF_VALOR_APPARADO_RS", "cobert_def_Vlr_Adicional": "COBERTURA_DEF_VALOR_APPARADO_RS", "cobert_def_qtd": "COBERTURA_DEF_QTD_CONTRATOS", "cobert_def_Vlr_Aparado": "COBERTURA_DEF_VALOR_APPARADO_RS", "cobert_def_Vlr_Adicional": "COBERTURA_DEF_VALOR_APPARADO_RS", "cobert_anal_Qtd_Contratos": "COBERTURA_ANALISE_QTD_CONTRATOS", "cobert_anal_Vlr_Aparado": "COBERTURA_ANALISE_VALOR_APPARADO_RS"}, inplace = True)

dados_bcb_pronaf.head()

# SIGLA_UF ADESAO_AREA_AMPARADA_HA ADESAO_QTDE CONTRATOS ADESAO_VALOR_APPARADO_RS ADESAO_VALOR_APPARADO_RS COMUNICACAO_PERDA_APPARADA_HA COMUNICACAO_PERDA_QTD_CONTRATOS COBERTURA_DEF_VALOR_APPARADO_RS COBERTURA_DEF_AREA_HA COBERTURA_DEF_QTD_CONTRATOS
0 RS 4793804.4 2.199707e+10 1.143756e+09 224847.6 17150 1.169223e+10 2178374.1 17150
1 PR 2767353.2 196541 1.256893e+10 7.239661e+08 1228149.6 78825 5.225561e+09 1108648.4 71
2 SC 625706.1 90991 5.509451e+09 2.381568e+08 180473.6 30346 1.995320e+09 164125.4 26
3 MG 174309.9 50383 3.152658e+09 1.066256e+08 5391.8 1005 9.159065e+07 3656.6
4 ES 81649.6 22830 1.614514e+09 4.703692e+07 2457.4 526 6.929033e+07 1747.6

# Alterar o nome das colunas para facilitar a análise
dados_bcb_pronamp = dados_bcb_pronaf.rename(columns = {"UF": "SIGLA_UF", "Adesão_Area_Aparada_HA": "ADESAO_AREA_AMPARADA_HA", "ADESAO_QTDE_CONTRATOS": "ADESAO_QTDE_CONTRATOS", "Adesão_Vlr_Aparado": "ADESAO_VALOR_APPARADO_RS", "Adesão_Vlr_Adicional": "ADESAO_VALOR_ADICIONAL_RS", "com_perdas_Area_Aparada_HA": "COMUNICACAO_PERDA_APPARADA_HA", "com_perdas_Qtd_Contratos": "COMUNICACAO_PERDA_QTD_CONTRATOS", "com_perdas_Vlr_Aparado": "COMUNICACAO_PERDA_VALOR_APPARADO_RS", "com_perdas_Vlr_Adicional": "COMUNICACAO_PERDA_VALOR_APPARADO_RS", "cobert_def_Vlr_Aparado": "COBERTURA_DEF_VALOR_APPARADO_RS", "cobert_def_Vlr_Adicional": "COBERTURA_DEF_VALOR_APPARADO_RS", "cobert_def_qtd": "COBERTURA_DEF_QTD_CONTRATOS", "cobert_def_Vlr_Aparado": "COBERTURA_DEF_VALOR_APPARADO_RS", "cobert_def_Vlr_Adicional": "COBERTURA_DEF_VALOR_APPARADO_RS", "cobert_anal_Qtd_Contratos": "COBERTURA_ANALISE_QTD_CONTRATOS", "cobert_anal_Vlr_Aparado": "COBERTURA_ANALISE_VALOR_APPARADO_RS"}, inplace = True)

dados_bcb_pronamp.head()

# SIGLA_UF ADESAO_AREA_AMPARADA_HA ADESAO_QTDE CONTRATOS ADESAO_VALOR_APPARADO_RS ADESAO_VALOR_APPARADO_RS COMUNICACAO_PERDA_APPARADA_HA COMUNICACAO_PERDA_QTD_CONTRATOS COBERTURA_DEF_VALOR_APPARADO_RS COBERTURA_DEF_AREA_HA COBERTURA_DEF_QTD_CONTRATOS
0 RS 4793804.4 38894 2.199707e+10 1.143756e+09 224847.6 17150 1.169223e+10 2178374.1 17150
1 PR 2767353.2 196541 1.256893e+10 7.239661e+08 1228149.6 78825 5.225561e+09 1108648.4 71
2 SC 625706.1 90991 5.509451e+09 2.381568e+08 180473.6 30346 1.995320e+09 164125.4 26
3 MG 174309.9 50383 3.152658e+09 1.066256e+08 5391.8 1005 9.159065e+07 3656.6
4 ES 81649.6 22830 1.614514e+09 4.703692e+07 2457.4 526 6.929033e+07 1747.6

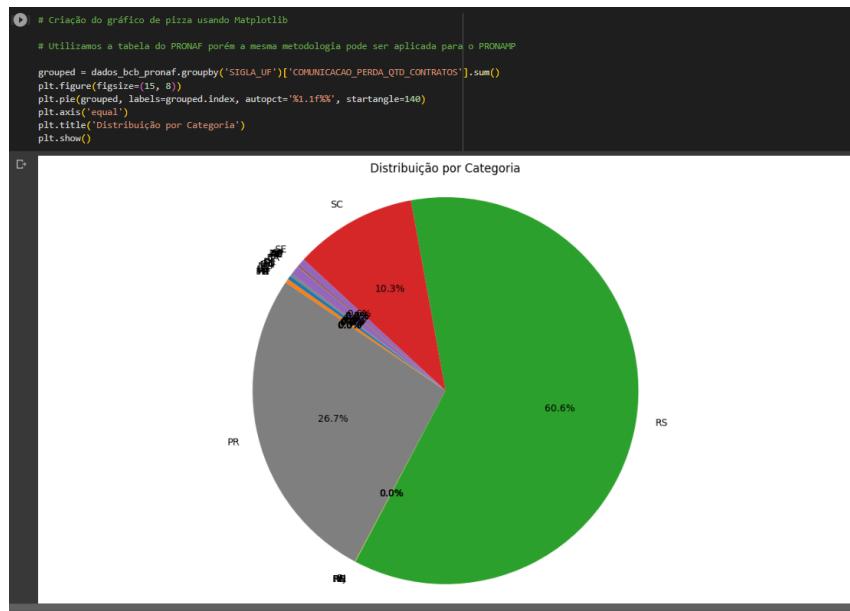
```

2.3.5. Feito a renomeação, podemos gerar uma série de gráficos para integrá-los as análises finais nos dashboards no Google Looker. Vale notar que a capacidade gráfica do Google Colab é limitada, porém, a lógica e resultados das análises serão reaproveitadas para um suporte gráfico superior do Google Looker.

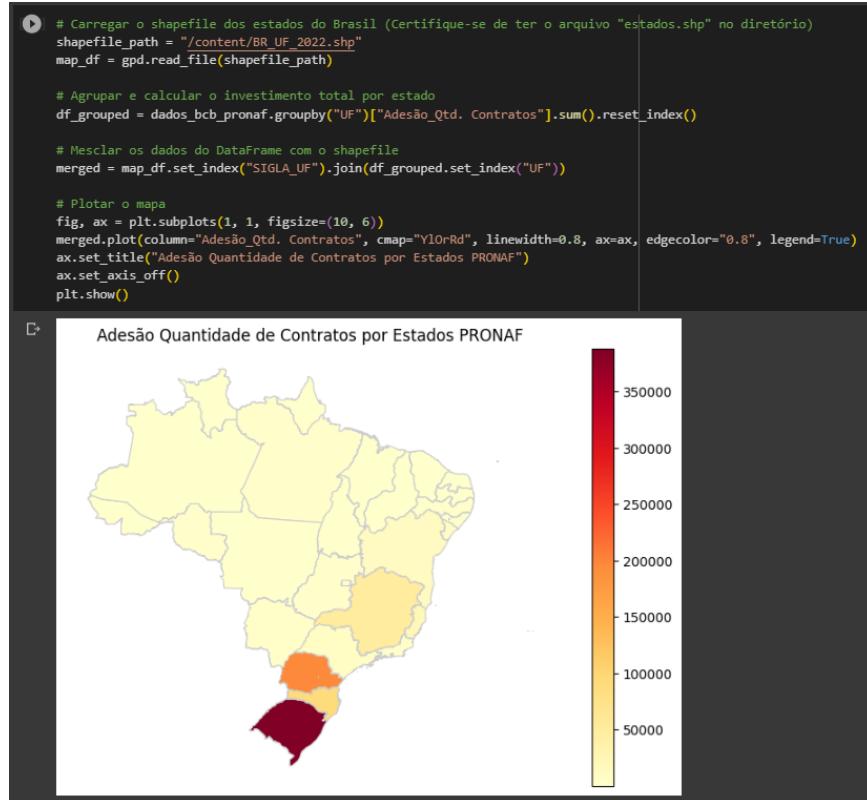
### 2.3.6. Gráfico de Barras:



### 2.3.7. Gráfico de Pizza:



2.3.8. Gráficos biblioteca Geopandas: Este gráfico permite uma análise comparativa dentro do próprio mapa político do Brasil.



2.3.9. Por fim, podemos fazer um merge das tabelas do PRONAMP e PRONAF e gerar em um único gráfico.

2.3.10. Primeiro concatenando as tabelas:

```
dados_concatenados = pd.concat([dados_bcb_pronaf, dados_bcb_pronaf])
dados_concatenados
```

	SIGLA_UF	ADESAO_AREA_AMPARADA_HA	ADESAO_QTDE_CONTRATOS	ADESAO_VALOR_AMPARADO_R\$	ADESAO_VALOR_ADICIONAL_R\$	COMUNICACAO_PERDA_AMPARADA_HA	COMUNICACAO_PERDA_QTD_CONTRATOS	CORUM
0	RS	4793804.4	388094	2.199707e+10	1.143757e+09	2244847.6	179150	
1	PR	2767353.2	196541	1.256893e+10	7.239661e+08	1228149.6	78825	
2	SC	625706.1	90991	5.509491e+09	2.381566e+08	180473.6	30346	
3	MG	174300.9	50383	3.152658e+09	1.066265e+08	5391.8	1005	
4	ES	81649.6	22830	1.614514e+09	4.703692e+07	2457.4	528	
...	...	...	...	...	...	...	...	...
40	SP	8.3	5	6.195467e+04	1.787710e+03	0.0	0	
41	SC	28.5	2	1.236250e+05	7.338430e+03	0.0	0	
42	PR	0.6	1	7.248000e+04	2.174750e+03	0.6	1	
43	ES	2.0	1	7.732800e+04	3.094080e+03	0.0	0	
44	MG	11.0	1	9.419634e+04	1.884230e+03	0.0	0	

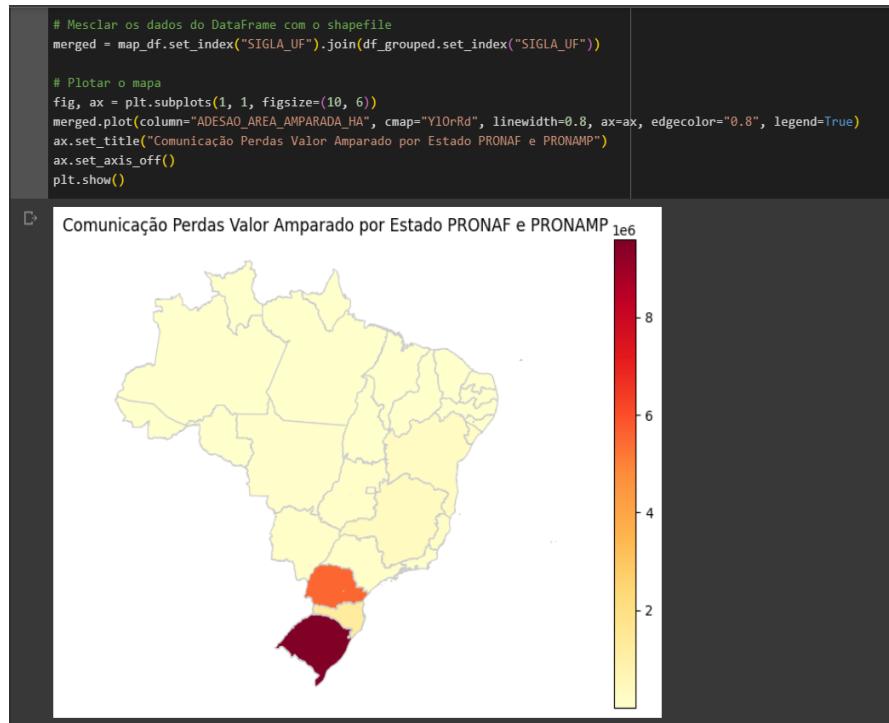
90 rows × 15 columns

### 2.3.11. Depois unificando-as por meio do groupby:

```
# dados_programas = dados_concatenados.groupby('SIGLA_UF').sum().reset_index()
dados_programas
```

	SIGLA_UF	ADESAOAREA_AMPARADA_HA	ADESAO_QTDE_CONTRATOS	ADESAO_VALOR_AMPARADO_R\$	ADESAO_VALOR_ADICIONAL_R\$	COMUNICACAO_PERDA_AMPARADA_HA	COMUNICACAO_PERDA_QTD CONTRATOS	COMUN
0	AC	712.2	242	6.573824e+06	2.012524e+05	0.0	0	
1	AL	34121.0	4774	2.074634e+08	6.773155e+06	3331.4	136	
2	AM	983.8	314	1.153416e+07	3.709826e+05	0.0	0	
3	AP	27.0	6	5.631762e+05	2.254160e+04	0.0	0	
4	BA	294537.2	25064	1.533211e+09	5.451458e+07	84731.0	4554	
5	CE	14550.0	1744	8.031031e+07	2.356336e+06	306.6	46	
6	DF	365.8	280	1.309837e+07	3.470671e+05	4.0	2	
7	ES	163303.2	45662	3.229182e+09	9.408004e+07	4914.8	1056	
8	GO	76608.4	3454	3.368776e+08	1.519405e+07	2942.0	102	
9	MA	17902.4	1928	6.970017e+07	2.474757e+06	0.0	0	
10	MG	348634.0	100774	6.305606e+09	2.132613e+08	10783.6	2010	
11	MS	144363.2	8316	5.535906e+08	2.642043e+07	39846.4	1892	
12	MT	55728.0	1786	1.973221e+08	9.186116e+06	605.0	14	
13	PA	25856.8	3262	2.213621e+08	7.825772e+06	2.0	2	
14	PB	2878.6	898	4.116703e+07	1.192794e+06	17.4	8	
15	PE	17883.2	3618	2.296472e+08	6.726598e+06	260.2	18	

### 2.3.12. E geramos o gráfico com ambos os dados do Pronamp e Pronaf.



## 2.4. TRATAMENTO DE DADOS DO DATASET CENSO RURAL ATIBAIA – IBGE

2.4.1 Nesta etapa, realizamos o pré-processamento dos dados utilizando o Google Colab para simular um ambiente Python de Data Science.

```

• Adubação

[41] dados_adubacao = pd.read_csv('Adubacao.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_adubacao

adubacao_quimica adubacao_organica adubacao_quimica_organica adubacao_nao_realizada
0               136                 77                   308                  78

[42] # converter dados para int
dados_adubacao['adubacao_quimica'] = dados_adubacao['adubacao_quimica'].astype(int)
dados_adubacao['adubacao_organica'] = dados_adubacao['adubacao_organica'].astype(int)
dados_adubacao['adubacao_quimica_organica'] = dados_adubacao['adubacao_quimica_organica'].astype(int)
dados_adubacao['adubacao_nao_realizada'] = dados_adubacao['adubacao_nao_realizada'].astype(int)
dados_adubacao

adubacao_quimica adubacao_organica adubacao_quimica_organica adubacao_nao_realizada
0               136                 77                   308                  78

[43] # adicionar a coluna com a unidade de medida dos dados
dados_adubacao['unidade'] = 'estabelecimentos'
dados_adubacao_final = dados_adubacao
dados_adubacao_final

adubacao_quimica adubacao_organica adubacao_quimica_organica adubacao_nao_realizada         unidade
0               136                 77                   308                  78 estabelecimentos

[44] dados_adubacao_final.dtypes

adubacao_quimica      int64
adubacao_organica     int64
adubacao_quimica_organica  int64
adubacao_nao_realizada int64
unidade                object
dtype: object

[45] from google.colab import files
dados_adubacao_final.to_csv('dados_adubacao_final.csv')
files.download('dados_adubacao_final.csv')

```

2.4.2. Na primeira linha de código, selecionamos o dataset Adubacao.csv extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, convertemos todos os dados da tabela para o tipo int, para assim podermos utilizá-los e analisá-los corretamente sem discrepâncias.

No terceiro passo, adicionamos a coluna unidade, que dirá qual é a unidade de medida desses dados contidos nesta tabela, vale ressaltar que nesta tabela todos os dados correspondem a mesma unidade de medida, no caso, estabelecimentos.

No passo seguinte checamos quais os tipos dos dados inseridos na tabela para verificarmos se não estão discrepantes.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```

• Agrotóxicos

[46] dados_agrotoxico = pd.read_csv('Agrotoxico.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_agrotoxico

  Utilizou  Não utilizou
0        436         163

dados_agrotoxico_final = dados_agrotoxico.rename(columns = {'Utilizou':'utilizado','Não utilizou':'nao_utilizado'})
dados_agrotoxico_final

  utilizado  nao_utilizado
0        436         163

[48] # converter dados para int
dados_agrotoxico_final['utilizado'] = dados_agrotoxico_final['utilizado'].astype(int)
dados_agrotoxico_final['nao_utilizado'] = dados_agrotoxico_final['nao_utilizado'].astype(int)
dados_agrotoxico_final

  utilizado  nao_utilizado
0        436         163

[49] # adicionar a coluna com a unidade de medida dos dados
dados_agrotoxico_final['unidade'] = 'estabelecimentos'
dados_agrotoxico_final

  utilizado  nao_utilizado      unidade
0        436         163  estabelecimentos

[50] dados_agrotoxico_final.dtypes

  utilizado    int64
  nao_utilizado   int64
  unidade        object
  dtype: object

  from google.colab import files
dados_agrotoxico_final.to_csv('dados_agrotoxico_final.csv')
files.download('dados_agrotoxico_final.csv')

```

2.4.3. Na primeira linha de código, selecionamos o dataset Agrotoxico.csv extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, alteramos os nomes das colunas para ficarem de acordo com os padrões e normas de governança de dados.

No terceiro passo, convertemos todos os dados da tabela para o tipo int, para assim podermos utilizá-los e analisá-los corretamente sem discrepâncias.

Em seguida, adicionamos a coluna unidade, que dirá qual é a unidade de medida desses dados contidos nesta tabela, vale ressaltar que nesta tabela todos os dados correspondem a mesma unidade de medida, no caso, estabelecimentos.

No passo seguinte checamos quais os tipos dos dados inseridos na tabela para verificarmos se não estão disparentes.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```

[52] dados_area = pd.read_csv('Area.csv', sep=';', decimal=',', encoding='UTF-8')
dados_area

Área dos estabelecimentos agropecuários
0 12043 hectares

[53] # renomear coluna
dados_area_final = dados_area.rename(columns = {'Área dos estabelecimentos agropecuários':'area_estabelecimentos'})
dados_area_final

area_estabelecimentos
0 12043 hectares

[54] # deixar somente o valor numérico na coluna para análise
dados_area_final['area_estabelecimentos'] = dados_area_final['area_estabelecimentos'].str.replace('12043 hectares', '12043')
dados_area_final

area_estabelecimentos
0 12043

[55] # converter dados para int
dados_area_final['area_estabelecimentos'] = dados_area_final['area_estabelecimentos'].astype(int)
dados_area_final

area_estabelecimentos
0 12043

[56] # adicionar a coluna com a unidade de medida dos dados
dados_area_final['unidade'] = 'hectares'
dados_area_final

area_estabelecimentos  unidade
0 12043  hectares

dados_area_final.dtypes

area_estabelecimentos    int64
unidade                  object
dtype: object

[58] from google.colab import files
dados_area_final.to_csv('dados_area_final.csv')
files.download('dados_area_final.csv')

```

2.4.4. Na primeira linha de código, selecionamos o dataset Area.csv extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, alteramos os nomes das colunas para ficarem de acordo com os padrões e normas de governança de dados.

No terceiro passo, tratamos os dados que possuíam valores numéricos e textuais na mesma coluna, deixamos apenas os numéricos para análise.

No quarto passo, convertemos todos os dados da tabela para o tipo int, para assim podermos utilizá-los e analisá-los corretamente sem discrepâncias.

Em seguida, adicionamos a coluna unidade, que dirá qual é a unidade de medida desses dados contidos nesta tabela, vale ressaltar que nesta tabela todos os dados correspondem a mesma unidade de medida, no caso, hectares.

No passo seguinte, checamos quais os tipos dos dados inseridos na tabela para verificarmos se não estão disparentes.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```

[52] dados_area = pd.read_csv('Area.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_area

Área dos estabelecimentos agropecuários ┌───┐
0          12043 hectares

[53] # renomear coluna
dados_area_final = dados_area.rename(columns = {'Área dos estabelecimentos agropecuários': 'area_estabelecimentos'})
dados_area_final

area_estabelecimentos ┌───┐
0          12043 hectares

[54] # deixar somente o valor numérico na coluna para análise
dados_area_final['area_estabelecimentos'] = dados_area_final['area_estabelecimentos'].str.replace('12043 hectares', '12043')
dados_area_final

area_estabelecimentos ┌───┐
0          12043

[55] # converter dados para int
dados_area_final['area_estabelecimentos'] = dados_area_final['area_estabelecimentos'].astype(int)
dados_area_final

area_estabelecimentos ┌───┐
0          12043

[56] # adicionar a coluna com a unidade de medida dos dados
dados_area_final['unidade'] = 'hectares'
dados_area_final

area_estabelecimentos  unidade ┌───┐
0          12043  hectares

dados_area_final.dtypes

area_estabelecimentos    int64
unidade                  object
dtype: object

[58] from google.colab import files
dados_area_final.to_csv('dados_area_final.csv')
files.download('dados_area_final.csv')

```

2.4.5. Na primeira linha de código, selecionamos o dataset “Classe Idade Produtores.csv” extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, alteramos os nomes das colunas para ficarem de acordo com os padrões e normas de governança de dados.

No terceiro passo, convertemos todos os dados da tabela para o tipo int, para assim podermos utilizá-los e analisá-los corretamente sem discrepâncias.

Em seguida, adicionamos a coluna unidade, que dirá qual é a unidade de medida desses dados contidos nesta tabela, vale ressaltar que nesta tabela todos os dados correspondem a mesma unidade de medida, no caso, estabelecimentos.

No passo seguinte, checamos quais os tipos dos dados inseridos na tabela para verificar se não estão descrepantes.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```

0  dados_classe_idade = pd.read_csv('Classe Idade Produtores.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_classe_idade
Menos de 25 anos 25 a menos de 35 anos 35 a menos de 45 anos 45 a menos de 55 anos 55 a menos de 65 anos 65 a menos de 75 anos 75 anos e mais
0           12            33          119          134          107           78           32

1 # renomear colunas
dados_classe_idade_final = dados_classe_idade.rename(columns = {'Menos de 25 anos':'menor_25_anos', '25 a menos de 35 anos':'maior_25_menor_35','35 a menos de
dados_classe_idade_final
menor_25_anos  maior_25_menor_35  maior_35_menor_45  maior_45_menor_55  maior_55_menor_65  maior_65_menor_75  maior_75_anos
0           12            33          119          134          107           78           32

[61] # converter dados para int
dados_classe_idade_final['menor_25_anos'] = dados_classe_idade_final['menor_25_anos'].astype(int)
dados_classe_idade_final['maior_25_menor_35'] = dados_classe_idade_final['maior_25_menor_35'].astype(int)
dados_classe_idade_final['maior_35_menor_45'] = dados_classe_idade_final['maior_35_menor_45'].astype(int)
dados_classe_idade_final['maior_45_menor_55'] = dados_classe_idade_final['maior_45_menor_55'].astype(int)
dados_classe_idade_final['maior_55_menor_65'] = dados_classe_idade_final['maior_55_menor_65'].astype(int)
dados_classe_idade_final['maior_65_menor_75'] = dados_classe_idade_final['maior_65_menor_75'].astype(int)
dados_classe_idade_final['maior_75_anos'] = dados_classe_idade_final['maior_75_anos'].astype(int)
dados_classe_idade_final
menor_25_anos  maior_25_menor_35  maior_35_menor_45  maior_45_menor_55  maior_55_menor_65  maior_65_menor_75  maior_75_anos
0           12            33          119          134          107           78           32

[62] # adicionar a coluna com a unidade de medida dos dados
dados_classe_idade_final['unidade'] = 'estabelecimentos'
dados_classe_idade_final
menor_25_anos  maior_25_menor_35  maior_35_menor_45  maior_45_menor_55  maior_55_menor_65  maior_65_menor_75  maior_75_anos      unidade
0           12            33          119          134          107           78           32  estabelecimentos

[63] dados_classe_idade_final.dtypes
menor_25_anos        int64
maior_25_menor_35    int64
maior_35_menor_45    int64
maior_45_menor_55    int64
maior_55_menor_65    int64
maior_65_menor_75    int64
maior_75_anos        int64
unidade             object
dtype: object

[64] from google.colab import files
dados_classe_idade_final.to_csv('dados_classe_idade_final.csv')
files.download('dados_classe_idade_final.csv')

```

2.4.6. Na primeira linha de código, selecionamos o dataset “Condicao Legal.csv” extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, alteramos os nomes das colunas para ficarem de acordo com os padrões e normas de governança de dados.

No terceiro passo, tratamos os dados que possuíam valores numéricos e textuais na mesma coluna, deixamos apenas os numéricos para análise.

No quarto passo, convertemos todos os dados da tabela para o tipo int, para assim podermos utilizá-los e analisá-los corretamente sem discrepâncias.

Em seguida, adicionamos a coluna unidade, que dirá qual é a unidade de medida desses dados contidos nesta tabela, vale ressaltar que nesta tabela todos os dados correspondem a mesma unidade de medida, no caso, hectares.

No passo seguinte, checamos quais os tipos dos dados inseridos na tabela para verificarmos se não estão disparentes.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```

• Condicao Legal
+ Código

[1]: dados_condicao_legal = pd.read_csv('Condicao Legal.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_condicao_legal

[2]: união Produtor individual
      0 1333 hectares 10608 hectares

[66]: # renomear colunas
dados_condicao_legal_final = dados_condicao_legal.rename(columns = {'união':'cdm_consorcio_uniao','Produtor individual':'individual'})
dados_condicao_legal_final

      cdm_consorcio_uniao    individual
      0          1333     10608 hectares

[67]: # deixar somente o valor numérico na coluna para análise
dados_condicao_legal_final['cdm_consorcio_uniao'] = dados_condicao_legal_final['cdm_consorcio_uniao'].str.replace('1333 hectares', '1333')
dados_condicao_legal_final['individual'] = dados_condicao_legal_final['individual'].str.replace('10608 hectares', '10608')
dados_condicao_legal_final

      cdm_consorcio_uniao    individual
      0              1333        10608

[68]: # converter dados para int
dados_condicao_legal_final['cdm_consorcio_uniao'] = dados_condicao_legal_final['cdm_consorcio_uniao'].astype(int)
dados_condicao_legal_final['individual'] = dados_condicao_legal_final['individual'].astype(int)
dados_condicao_legal_final

      cdm_consorcio_uniao    individual
      0              1333        10608

[69]: # adicionar a coluna com a unidade de medida dos dados
dados_condicao_legal_final['unidade'] = 'hectares'
dados_condicao_legal_final

      cdm_consorcio_uniao    individual    unidade
      0              1333        10608    hectares

[70]: dados_condicao_legal_final.dtypes
      cdm_consorcio_uniao    int64
      individual            int64
      unidade               object
      dtype: object

[71]: from google.colab import files
dados_condicao_legal_final.to_csv('dados_condicao_legal_final.csv')
files.download('dados_condicao_legal_final.csv')

```

2.4.7. Na primeira linha de código, selecionamos o dataset “Condicao Produtor Terra.csv” extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, alteramos os nomes das colunas para ficarem de acordo com os padrões e normas de governança de dados.

No terceiro passo, tratamos os dados que possuíam valores numéricos e textuais na mesma coluna, deixamos apenas os numéricos para análise.

No quarto passo, convertemos todos os dados da tabela para o tipo int, para assim podermos utilizá-los e analisá-los corretamente sem discrepâncias.

Em seguida, adicionamos a coluna unidade, que dirá qual é a unidade de medida desses dados contidos nesta tabela, vale ressaltar que nesta tabela todos os dados correspondem a mesma unidade de medida, no caso, hectares.

No passo seguinte, checamos quais os tipos dos dados inseridos na tabela para verificarmos se não estão disparentes.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```

• Condicao Produtor Terra

[6] dados_condicao_produtor_terra = pd.read_csv('Condicao Produtor Terra.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_condicao_produtor_terra

PROPRIETÁRIO(A) INCLUSIVE OS(AS) COPROPRIETÁRIOS(AS) DE TERRAS TITULADAS COLETIVAMENTE Arrendatário(a)
0 9625 hectares 1619 hectares

[73] # renomear colunas
dados_condicao_produtor_terra_final = dados_condicao_produtor_terra.rename(columns = {'PROPRIETÁRIO(A) INCLUSIVE OS(AS) COPROPRIETÁRIOS(AS) DE TERRAS TITULADAS COLETIVAMENTE':'proprietarios',
                                                                 'Arrendatário(a)':'arrendatario'})
dados_condicao_produtor_terra_final

proprietarios arrendatario
0 9625 hectares 1619 hectares

[74] # deixar somente o valor numérico na coluna para análise
dados_condicao_produtor_terra_final['proprietarios'] = dados_condicao_produtor_terra_final['proprietarios'].str.replace('9625 hectares', '9625')
dados_condicao_produtor_terra_final['arrendatario'] = dados_condicao_produtor_terra_final['arrendatario'].str.replace('1619 hectares', '1619')
dados_condicao_produtor_terra_final

proprietarios arrendatario
0 9625 1619

[75] # converter dados para int
dados_condicao_produtor_terra_final['proprietarios'] = dados_condicao_produtor_terra_final['proprietarios'].astype(int)
dados_condicao_produtor_terra_final['arrendatario'] = dados_condicao_produtor_terra_final['arrendatario'].astype(int)
dados_condicao_produtor_terra_final

proprietarios arrendatario
0 9625 1619

[76] # adicionar a coluna com a unidade de medida dos dados
dados_condicao_produtor_terra_final['unidade'] = 'hectares'
dados_condicao_produtor_terra_final

proprietarios arrendatario unidade
0 9625 1619 hectares

[77] dados_condicao_produtor_terra_final.dtypes
proprietarios    int64
arrendatario    int64
unidade        object
dtype: object

[78] from google.colab import files
dados_condicao_produtor_terra_final.to_csv('dados_condicao_produtor_terra_final.csv')
files.download('dados_condicao_produtor_terra_final.csv')

```

2.4.8. Na primeira linha de código, selecionamos o dataset “Financiamento.csv” extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, renomeamos as colunas de acordo com os padrões e normas de governança de dados.

No terceiro passo, tratamos os dados que possuíam valores numéricos e textuais na mesma coluna, deixamos apenas os numéricos para análise.

Em seguida, adicionamos a coluna unidade, que dirá qual é a unidade de medida desses dados contidos nesta tabela, vale ressaltar que nesta tabela todos os dados correspondem a mesma unidade de medida, no caso, estabelecimentos.

No passo seguinte, checamos quais os tipos dos dados inseridos na tabela para verificarmos se não estão discrepantes. Note que não houve necessidade de alterar os dados para int uma vez que já estavam de acordo.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```

• Financiamento

[79] dados_financiamento = pd.read_csv('Financiamento.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_financiamento
      Obteve  Investimento  Custoio  Manutenção  Programas governamentais de crédito  Pronaf  Pronamp  Outro programa (federal, estadual ou municipal)  Outras fontes  Não obteve
      0       81        24        54          18                  27       15        5
      11        54        519

[80] # renomear colunas
dados_financiamento_final = dados_financiamento.rename(columns = {'Obteve':'obteve', 'Investimento':'investimento', 'custoio':'custeio', 'Manutenção':'manutencao',
'Programas governamentais de crédito': 'programas_gov_de_credito', 'Pronaf': 'pronaf',
'pronamp': 'pronamp', 'Outro programa (federal, estadual ou municipal)': 'outro_programa',
'Outras fontes':'outras_fontes', 'Não obteve' : 'nao_obteve' })
dados_financiamento_final
      obteve  investimento  custeio  manutencao  programas_gov_de_credito  pronaf  pronamp  outro_programa  outras_fontes  nao_obteve
      0       81        24        54          18                  27       15        5       11        54        519

[81] # adicionar a coluna com a unidade de medida dos dados
dados_financiamento_final['unidade'] = 'estabelecimento'
dados_financiamento_final
      obteve  investimento  custeio  manutencao  programas_gov_de_credito  pronaf  pronamp  outro_programa  outras_fontes  nao_obteve  unidade
      0       81        24        54          18                  27       15        5       11        54        519  estabelecimento

[82] dados_financiamento_final.dtypes
obteve           int64
investimento    int64
custeio          int64
manutencao       int64
programas_gov_de_credito  int64
pronaf           int64
pronamp          int64
outro_programa   int64
outras_fontes    int64
nao_obteve       int64
unidade          object
dtype: object

[83] from google.colab import files
dados_financiamento_final.to_csv('dados_financiamento_final.csv')
files.download('dados_financiamento_final.csv')

```

2.4.9. Na primeira linha de código, selecionamos o dataset “Laranja.csv” extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, renomeamos as colunas de acordo com os padrões e normas de governança de dados. Note que para este dataset não colocamos uma coluna unidade, pois, há diversos tipos de dados nesta tabela, então criar uma tabela de unidade para todos os dados não seria correto. Dito isso, colocamos a unidade ao final do próprio nome da tabela.

No terceiro passo, tratamos os dados das colunas que possuíam valores em reais, no caso, eliminamos o símbolo “R\$” deixando apenas o numérico.

No passo seguinte, convertemos os valores que representam reais para float.

Em seguida, checamos quais os tipos dos dados inseridos na tabela para verificar se não estão discrepantes.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```

• Laranja

[84] dados_laranja = pd.read_csv('Laranja.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_laranja

      Nro de estabelecimentos com 50 pés ou mais  Quantidade produzida  Qtde de pés  Área dos estabelecimentos  Área colhida  Valor da produção
0                           7                      185          6                  22           21        R$ 112.200,00

[85] # renomear colunas
dados_laranja_final = dados_laranja.rename(columns = {'Nro de estabelecimentos com 50 pés ou mais': 'nro_estabelecimentos', 'Quantidade produzida': 'quantidade_produzida(t)', 'Qtde de pés': 'qt_de_pes(unidades)', 'Área dos estabelecimentos': 'area_estabelecimentos(ha)', 'Área colhida': 'area_colhida(ha)', 'Valor da produção': 'valor_producao(R$)'})
dados_laranja_final

      nro_estabelecimentos  quantidade_produzida(t)  qt_de_pes(unidades)  area_estabelecimentos(ha)  area_colhida(ha)  valor_producao(R$)
0                           7                      185                      6                  22           21        R$ 112.200,00

[86] # transformar dados para valor correto
dados_laranja_final['valor_producao(R$)'] = dados_laranja_final['valor_producao(R$)'].str.replace('R$ 112.200,00', '112.200.000', regex=False)
dados_laranja_final.loc[0, 'qt_de_pes(unidades)'] = 6000

[87] # transformar dados para valor correto
dados_laranja_final.loc[0, 'valor_producao(R$)'] = 112200000
dados_laranja_final

      nro_estabelecimentos  quantidade_produzida(t)  qt_de_pes(unidades)  area_estabelecimentos(ha)  area_colhida(ha)  valor_producao(R$)
0                           7                      185                      6000                 22           21        112200000

[88] # converter dados para float
dados_laranja_final['quantidade_produzida(t)'] = dados_laranja_final['quantidade_produzida(t)'].astype(float)
dados_laranja_final['valor_producao(R$)'] = dados_laranja_final['valor_producao(R$)'].astype(float)
dados_laranja_final

      nro_estabelecimentos  quantidade_produzida(t)  qt_de_pes(unidades)  area_estabelecimentos(ha)  area_colhida(ha)  valor_producao(R$)
0                           7                      185.0                     6000                 22           21        112200000.0

[89] dados_laranja_final.dtypes

nro_estabelecimentos    int64
quantidade_produzida(t)  float64
qt_de_pes(unidades)     int64
area_estabelecimentos(ha) int64
area_colhida(ha)         int64
valor_producao(R$)      float64
dtype: object

[90] from google.colab import files
dados_laranja_final.to_csv('dados_laranja_final.csv')
files.download('dados_laranja_final.csv')

```

2.4.10. Na primeira linha de código, selecionamos o dataset “Pessego.csv” extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, renomeamos as colunas de acordo com os padrões e normas de governança de dados. Note que para este dataset não colocamos uma coluna unidade, pois, há diversos tipos de dados nesta tabela, então criar uma tabela de unidade para todos os dados não seria correto. Dito isso, colocamos a unidade ao final do próprio nome da tabela.

No terceiro passo, tratamos os dados das colunas que possuíam valores em reais, no caso, eliminamos o símbolo “R\$” deixando apenas o numérico.

No quarto passo, convertemos os valores que representam reais para float.

Em seguida, checamos quais os tipos dos dados inseridos na tabela para verificar se não estão disparentes.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```

[1] ➜ dados_pessego = pd.read_csv('Pessego.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_pessego

[2] ➜ Nro de estabelecimentos com 50 pés ou mais Quantidade produzida Qtde de pés Área dos estabelecimentos Área colhida Valor da produção
[3] ➜ 0 21 1080 53 100 84 R$ 2.621.780,00

[4] ➜ [92] # renomear colunas
dados_pessego_final = dados_pessego.rename(columns = {'Nro de estabelecimentos com 50 pés ou mais': 'nro_estabelecimentos', 'Quantidade produzida': 'quantidade_produzida(t)', 'Qtde de pés': 'qt_de_pes(unidades)', 'Área dos estabelecimentos': 'area_estabelecimentos(ha)', 'Área colhida': 'area_colhida(ha)', 'Valor da produção': 'valor_producao(R$)' })
dados_pessego_final

[5] ➜ nro_estabelecimentos quantidade_produzida(t) qt_de_pes(unidades) area_estabelecimentos(ha) area_colhida(ha) valor_producao(R$)
[6] ➜ 0 21 1080 53 100 84 R$ 2.621.780,00

[7] ➜ [93] # transformar dados para valor correto
dados_pessego_final.loc[0, 'qt_de_pes(unidades)'] = 53000
dados_pessego_final

[8] ➜ nro_estabelecimentos quantidade_produzida(t) qt_de_pes(unidades) area_estabelecimentos(ha) area_colhida(ha) valor_producao(R$)
[9] ➜ 0 21 1080 53000 100 84 R$ 2.621.780,00

[10] ➜ [94] dados_pessego_final['valor_producao(R$)'] = dados_pessego_final['valor_producao(R$)'].str.replace('R$ 2.621.780,00', '2.621.780.000', regex=False)
dados_pessego_final

[11] ➜ nro_estabelecimentos quantidade_produzida(t) qt_de_pes(unidades) area_estabelecimentos(ha) area_colhida(ha) valor_producao(R$)
[12] ➜ 0 21 1080 53000 100 84 2.621.780.000

[13] ➜ [95] dados_pessego_final.loc[0, 'valor_producao(R$)'] = 2621780000
dados_pessego_final

[14] ➜ nro_estabelecimentos quantidade_produzida(t) qt_de_pes(unidades) area_estabelecimentos(ha) area_colhida(ha) valor_producao(R$)
[15] ➜ 0 21 1080.0 53000 100 84 2.621780e+09

[16] ➜ [96] dados_pessego_final.dtypes
nro_estabelecimentos      int64
quantidade_produzida(t)    float64
qt_de_pes(unidades)        int64
area_estabelecimentos(ha)  int64
area_colhida(ha)           int64
valor_producao(R$)         float64
dtype: object

[17] ➜ [97] # converter dados para float
dados_pessego_final['valor_producao(R$)'] = dados_pessego_final['valor_producao(R$)'].astype(float)
dados_pessego_final['quantidade_produzida(t)'] = dados_pessego_final['quantidade_produzida(t)'].astype(float)
dados_pessego_final

[18] ➜ nro_estabelecimentos quantidade_produzida(t) qt_de_pes(unidades) area_estabelecimentos(ha) area_colhida(ha) valor_producao(R$)
[19] ➜ 0 21 1080.0 53000 100 84 2.621780e+09

[20] ➜ [98] dados_pessego_final.dtypes
nro_estabelecimentos      int64
quantidade_produzida(t)    float64
qt_de_pes(unidades)        int64
area_estabelecimentos(ha)  int64
area_colhida(ha)           int64
valor_producao(R$)         float64
dtype: object

[21] ➜ [99] from google.colab import files
dados_pessego_final.to_csv('dados_pessego_final.csv')
files.download('dados_pessego_final.csv')

```

2.4.11. Na primeira linha de código, selecionamos o dataset “Matas Florestas.csv” extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Em seguida, renomeamos as colunas para os padrões e normas de governança de dados.

No terceiro passo, tratamos os dados que possuíam valores numéricos e textuais na mesma coluna, deixamos apenas os numéricos para análise.

No quarto passo, convertemos todos os dados da tabela para o tipo int, para assim podermos utilizá-los e analisá-los corretamente sem discrepâncias.

Em seguida, adicionamos a coluna unidade, que dirá qual é a unidade de medida desses dados contidos nesta tabela, vale ressaltar que nesta tabela todos os dados correspondem a mesma unidade de medida, no caso, hectares.

No passo seguinte checamos quais os tipos dos dados inseridos na tabela para verificarmos se não estão disparentes.

E por fim, fizemos o download do dataset tratado e com os dados corretos.

```
[100] dados_matas = pd.read_csv('Matas Florestas.csv', sep=';', decimal=',', encoding='UTF-8')
dados_matas
naturais Naturais destinadas á preservação permanente ou reserva legal florestas plantadas
0 196 há 2216 ha 410 ha

[101] # renomear colunas
dados_matas_final = dados_matas.rename(columns={'Naturais': 'naturais',
                                                 'Naturais destinadas á preservação permanente ou reserva legal': 'naturais_preservacao_reserva',
                                                 'Florestas plantadas': 'florestas_plantadas'})
dados_matas_final
naturais Naturais destinadas á preservação permanente ou reserva legal florestas_plantadas
0 196 há 2216 ha 410 ha

[102] # renomear colunas
dados_matas_final.columns = ['naturais',
                             'naturais_preservação_ou_reserva',
                             'florestas_plantadas']
dados_matas_final
naturais naturais_preservação_ou_reserva florestas_plantadas
0 196 2216 410

[103] # transformar os dados para numéricos apenas
dados_matas_final['naturais'] = dados_matas_final['naturais'].str.replace('196 há', '196')
dados_matas_final['naturais_preservação_ou_reserva'] = dados_matas_final['naturais_preservação_ou_reserva'].str.replace('2216 ha', '2216')
dados_matas_final['florestas_plantadas'] = dados_matas_final['florestas_plantadas'].str.replace('410 ha', '410')
dados_matas_final
naturais naturais_preservação_ou_reserva florestas_plantadas
0 196 2216 410

[104] # adicionar a coluna com a unidade de medida dos dados
dados_matas_final['unidade'] = 'hectares'
dados_matas_final
naturais naturais_preservação_ou_reserva florestas_plantadas unidade
0 196 2216 410 hectares

[105] # converter dados para int
dados_matas_final['naturais'] = dados_matas_final['naturais'].astype(int)
dados_matas_final['naturais_preservação_ou_reserva'] = dados_matas_final['naturais_preservação_ou_reserva'].astype(int)
dados_matas_final['florestas_plantadas'] = dados_matas_final['florestas_plantadas'].astype(int)
dados_matas_final
naturais naturais_preservação_ou_reserva florestas_plantadas unidade
0 196 2216 410 hectares

[106] from google.colab import files
dados_matas_final.to_csv('dados_matas_final.csv')
files.download('dados_matas_final.csv')
```

2.4.12. Na primeira linha de código, selecionamos o dataset “Sexo.csv” extraído do site do IBGE, conforme link acima, e realizamos a leitura e exibição dela.

Este dataset não havia nada para ser tratado, apenas verificamos os dados e fizemos o download.

The screenshot shows a Jupyter Notebook cell with the title "Sexo". The code cell contains the following Python code:

```
[107] dados_sexo = pd.read_csv('Sexo.csv', sep = ';', decimal = ',', encoding = 'UTF-8')
dados_sexo
```

Below the code cell, the output shows a table with two columns: "Masculino" and "Feminino". The row index 0 has values 515 and 82 respectively. There is also a small calendar icon next to the table.

The next cell, indicated by a play button icon, contains the command:

```
dados_sexo.dtypes
```

The output of this command is:

```
Masculino    int64
Feminino    int64
dtype: object
```

The final cell, indicated by a play button icon, contains the code:

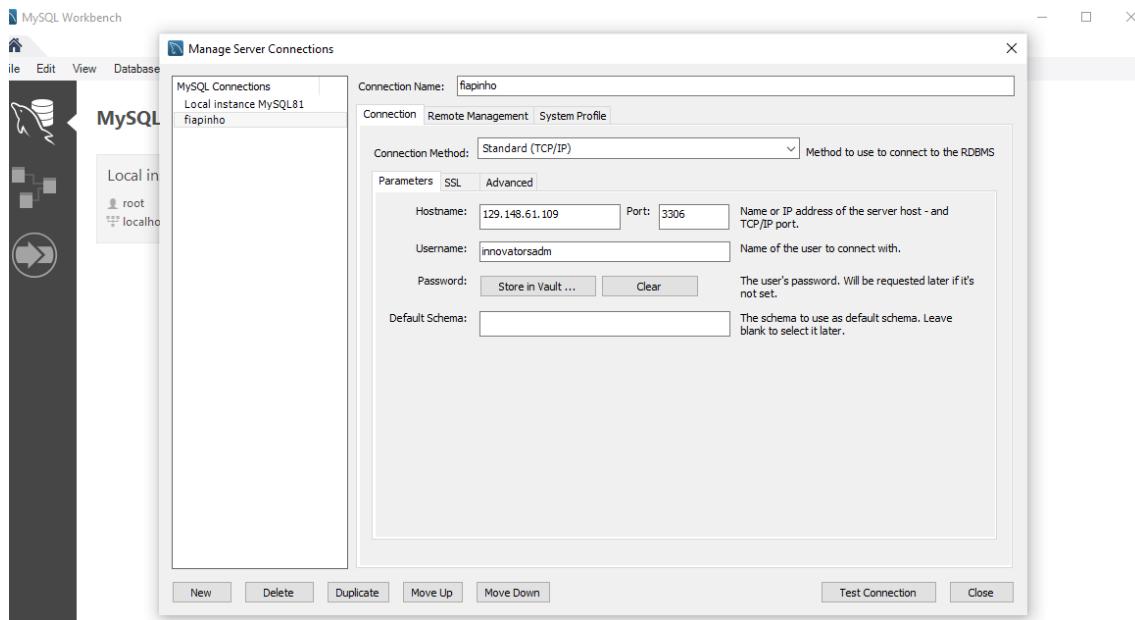
```
from google.colab import files
dados_sexo.to_csv('dados_sexo.csv')
files.download('dados_sexo.csv')
```

### 3. INSTALAÇÃO DO MYSQL WORKBENCH

A documentação inicia a partir do MYSQL INSTALADO.

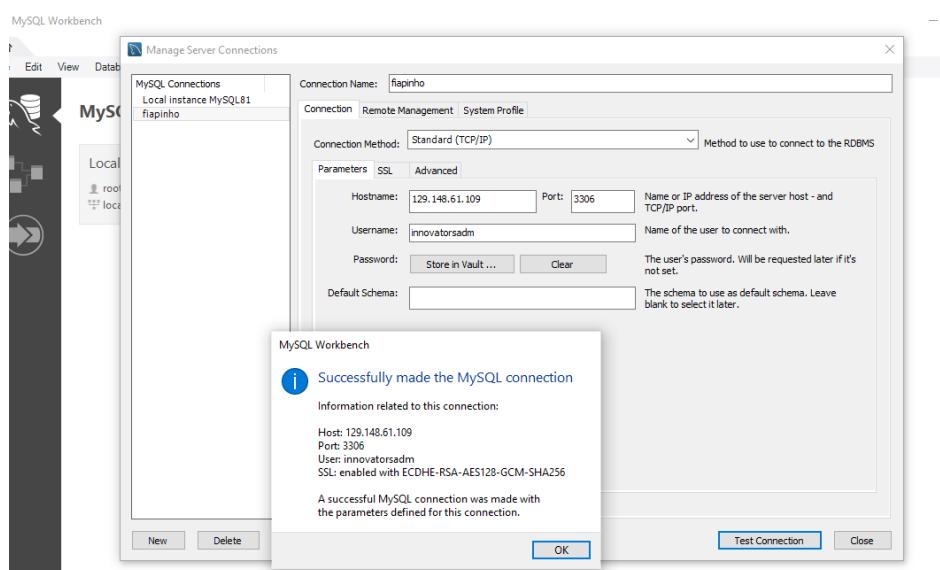
#### 3.1. CONEXÃO COM A BASE DE DADOS

3.1.1 Fazendo a conexão com a base de dados, passando os parâmetros de conexão: Hostname, Porta, username e password, conforme referência abaixo:



Test Connection

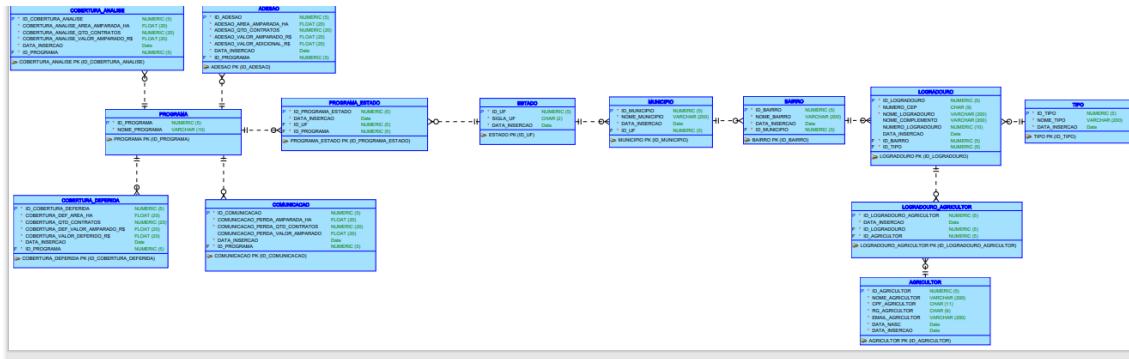
Mensagem de conexão realizada com sucesso.



## 3.2. MODELAGEM DE DADOS

### 3.2.1. Modelo Lógico

Evidência do modelo lógico utilizado para a criação e estruturação do modelo físico acima apresentado.



### 3.2.2. Tabela Estado

Criação da tabela Estado com suas constraints e chave primária.

```
CREATE TABLE ESTADO (
    ID_UF NUMERIC(5) CONSTRAINT PK_ID_UF PRIMARY KEY,
    SIGLA_UF CHAR(2) CONSTRAINT NN_NOME_UF NOT NULL,
    DATA_INSERCAO DATE CONSTRAINT NN_DATA_INSERCAO_ESTADO
NOT NULL
);
```

### 3.2.3. Tabela Município

Criação da tabela Município com suas constraints, chave primária e chave estrangeira.

```
CREATE TABLE MUNICIPIO (
    ID_MUNICIPIO NUMERIC(5) CONSTRAINT PK_ID_MUNICIPIO
PRIMARY KEY,
    ID_UF      NUMERIC(5)   CONSTRAINT FK_ID_UF_MUNICIPIO
REFERENCES ESTADO(ID_UF),
    NOME_MUNICIPIO          VARCHAR2(200)           CONSTRAINT
NN_NOME_MUNICIPIO NOT NULL,
    DATA_INSERCAO            DATE                 CONSTRAINT
NN_DATA_INSERCAO_MUNICIPIO NOT NULL
);
```

### 3.2.4. Tabela Bairro

Criação da tabela Bairro com suas constraints, chave primária e chave estrangeira.

```
CREATE TABLE BAIRRO (
    ID_BAIRRO NUMERIC(5) CONSTRAINT PK_ID_BAIRRO PRIMARY KEY,
    ID_MUNICIPIO          NUMERIC(5)           CONSTRAINT
    FK_ID_MUNICIPIO_BAIRRO REFERENCES MUNICIPIO(ID_MUNICIPIO),
    NOME_BAIRRO VARCHAR2(200) CONSTRAINT NN_NOME_BAIRRO NOT
    NULL,
    DATA_INSERCAO DATE CONSTRAINT NN_DATA_INSERCAO_BAIRRO
    NOT NULL
);
```

### 3.2.5. Tabela Tipo

Criação da tabela Tipo com suas constraints e chave primária.

```
CREATE TABLE TIPO (
    ID_TIPO NUMERIC(5) CONSTRAINT PK_IP_TIPO PRIMARY KEY,
    NOME_TIPO VARCHAR2(200) CONSTRAINT NN_NOME_TIPO NOT
    NULL,
    DATA_INSERCAO DATE CONSTRAINT NN_DATA_INSERCAO_TIPO NOT
    NULL
);
```

### 3.2.6. Tabela Logradouro

Criação da tabela Logradouro com suas constraints, chave primária e chave estrangeira.

```
CREATE TABLE LOGRADOURO (
    ID_LOGRADOURO NUMERIC(5) CONSTRAINT PK_ID_LOGRADOURO
    PRIMARY KEY,
    ID_TIPO NUMERIC(5) CONSTRAINT FK_ID_TIPO_LOGRADOURO
    REFERENCES TIPO(ID_TIPO),
    ID_BAIRRO NUMERIC(5) CONSTRAINT FK_ID_BAIRRO_LOGRADOURO
    REFERENCES BAIRRO(ID_BAIRRO),
    NUMERO_CEP NUMERIC(8) CONSTRAINT NN_NUMERO_CEP NOT NULL,
    NOME_LOGRADOURO VARCHAR2(300)           CONSTRAINT
    NN_NOME_LOGRADOURO NOT NULL,
    NOME_COMPLEMENTO VARCHAR2(200),
    NUMERO_LOGRADOURO NUMBER(10),
    DATA_INSERCAO DATE                     CONSTRAINT
    NN_DATA_INSERCAO_LOGRADOURO NOT NULL
);
```

### 3.2.7. Tabela Agricultor

Criação da tabela Agricultor com suas constraints, chave primária e Chave única.

```
CREATE TABLE AGRICULTOR (
    ID_AGRICULTOR NUMBER(5) CONSTRAINT PK_ID_AGRICULTOR
PRIMARY KEY,
    NOME_AGRICULTOR          VARCHAR2(300)           CONSTRAINT
NN_NOME_AGRICULTOR NOT NULL,
    CPF_AGRICULTOR   CHAR(11)    CONSTRAINT UK_CPF_AGRICULTOR
UNIQUE,
    RG_AGRICULTOR    CHAR(9)    CONSTRAINT UK_RG_AGRICULTOR
UNIQUE,
    EMAIL_AGRICULTOR        VARCHAR2(200)          CONSTRAINT
NN_EMAIL_AGRICULTOR NOT NULL,
    DATA_NASC    DATE      CONSTRAINT NN_DATA_NASC_AGRICULTOR NOT
NULL,
    DATA_INSERCAO           DATE                 CONSTRAINT
NN_DATA_INSERCAO_AGRICULTOR NOT NULL
);
```

### 3.2.8. Tabela Logradouro\_Agricultor

Criação da tabela Logradouro\_Agricultor com suas constraints, chave primária, unique key e chave estrangeira.

```
CREATE TABLE LOGRADOURO_AGRICULTOR(
    ID_LOGRADOURO_AGRICULTOR NUMBER(5)           CONSTRAINT
PK_LOGRADOURO_AGRICULTOR PRIMARY KEY,
    ID_LOGRADOURO             NUMBER(5)            CONSTRAINT
REFERENCES
FK_LOGRADOURO_AGRICULTOR_LOGRADOURO
LOGRADOURO(ID_LOGRADOURO),
    ID_AGRICULTOR             NUMBER(5)           CONSTRAINT
REFERENCES
FK_AGRICULTOR_ID_LOGRADOURO_AGRICULTOR
AGRICULTOR(ID_AGRICULTOR),
    DATA_INSERCAO              DATE                CONSTRAINT
NN_DATA_INSERCAO_LOGRADOURO_AGRICULTOR NOT NULL
);
```

### 3.2.9. Tabela Programa

Criação da tabela Programa com suas constraints, Chave primária.

```
CREATE TABLE PROGRAMA (
    ID_PROGRAMA NUMBER(5) CONSTRAINT PK_ID_PROGRAMA PRIMARY
KEY,
    NOME_PROGRAMA  VARCHAR2(10) CONSTRAINT NN_NOME_PROGRAMA
NOT NULL,
    DATA_INSERCAO DATE      CONSTRAINT NN_DATA_INSERCAO_PROGRAMA
NOT NULL
);
```

### 3.2.10. Tabela Programa\_Estado

Criação da tabela Programa\_Estado com suas constraints, chave primária e chave estrangeira.

```
CREATE TABLE PROGRAMA_ESTADO (
    ID_PROGRAMA_ESTADO          NUMBER(5)      CONSTRAINT
    PK_ID_PROGRAMA_ESTADO PRIMARY KEY,
    ID_UF   NUMBER(5)  CONSTRAINT FK_PROGRAMA_ESTADO_ESTADO
    REFERENCES ESTADO(ID_UF),
    ID_PROGRAMA                  NUMBER(5)      CONSTRAINT
    FK_PROGRAMA_ESTADO_PROGRAMA REFERENCES
    PROGRAMA(ID_PROGRAMA),
    DATA_INSERCAO                DATE         CONSTRAINT
    NN_DATA_INSERCAO_PROGRAMA_ESTADO NOT NULL
);
```

### 3.2.11. Tabela Adesão

Criação da tabela Adesão com suas constraints, chave primária e chave estrangeira.

```
CREATE TABLE ADESAO (
    ID_ADESAO NUMBER(5) CONSTRAINT PK_ID_ADESAO PRIMARY KEY,
    ADESAO_AREA_AMPARADA_HA     FLOAT(20)      CONSTRAINT
    NN_ADESAO_AREA_AMPARADA NOT NULL,
    ADESAO_QTD_CONTRATOS        NUMBER(20)      CONSTRAINT
    NN_ADESAO_QTD_AMPARADA NOT NULL,
    ADESAO_VALOR_AMPARADO_R$     FLOAT(20)      CONSTRAINT
    NN_ADESAO_VALOR_AMPARADO NOT NULL,
    ADESAO_VALOR_ADICIONAL_R$     FLOAT(20)      CONSTRAINT
    NN_ADESAO_VALOR_ADICIONAL NOT NULL,
    ID_PROGRAMA      NUMBER(5) CONSTRAINT FK_ADESAO_PROGRAMA
    REFERENCES PROGRAMA(ID_PROGRAMA),
    DATA_INSERCAO    DATE  CONSTRAINT NN_DATA_INSERCAO_ADESAO
    NOT NULL
);
```

### 3.2.12. Tabela Comunicação

Criação da tabela Comunicação com suas constraints, chave primária e chave estrangeira.

```
CREATE TABLE COMUNICACAO(
    ID_COMUNICACAO NUMBER(5) CONSTRAINT PK_ID_COMUNICACAO
PRIMARY KEY,
    COMUNICACAO_PERDA_AMPARADA_HA      FLOAT(20)      CONSTRAINT
NN_COMUNICACAO_PERDA_AMPARADA NOT NULL,
    COMUNICACAO_PERDA_QTD_CONTRATOS   NUMBER(20)      CONSTRAINT
NN_COMUNICACAO_PERDA_QTD_CONTRATOS NOT NULL,
    COMUNICACAO_PERDA_VALOR_AMPARADO_R$          FLOAT(20)
CONSTRAINT NN_COMUNICACAO_PERDA_VALOR_AMPARADO NOT NULL,
    ID_PROGRAMA NUMBER(5) CONSTRAINT FK_COMUNICACAO_PROGRAMA
REFERENCES PROGRAMA(ID_PROGRAMA),
    DATA_INSERCAO           DATE                  CONSTRAINT
NN_DATA_INSERCAO_COMUNICACAO NOT NULL
);
```

### 3.2.13. Tabela Cobertura\_Análise

Criação da tabela Cobertura\_Análise com suas constraints, chave primária e chave estrangeira.

```
CREATE TABLE COBERTURA_ANALISE(
    ID_COBERTURA_ANALISE           NUMBER(5)      CONSTRAINT
PK_ID_COBERTURA_ANALISE PRIMARY KEY,
    COBERTURA_ANALISE_AREA_AMPARADA_HA FLOAT(20)  CONSTRAINT
NN_COBERTURA_ANALISE_AREA_AMPARADA NOT NULL,
    COBERTURA_ANALISE_QTD_CONTRATOS  NUMBER(20)  CONSTRAINT
NN_COBERTURA_ANALISE_QTD_CONTRATOS NOT NULL,
    COBERTURA_ANALISE_VALOR_AMPARADO_R$          FLOAT(20)
CONSTRAINT NN_COBERTURA_ANALISE_VALOR_AMPARADO NOT NULL,
    ID_PROGRAMA                   NUMBER(5)      CONSTRAINT
FK_COBERTURA_ANALISE_PROGRAMA REFERENCES
PROGRAMA(ID_PROGRAMA),
    DATA_INSERCAO           DATE                  CONSTRAINT
NN_DATA_INSERCAO_COBERTURA_ANALISE NOT NULL
);
```

### 3.2.14. Tabela Cobertura\_Deferida

Criação da tabela Cobertura\_Deferida com suas constraints, chave primária e chave estrangeira.

```
CREATE TABLE COBERTURA_DEFERIDA (
    ID_COBERTURA_DEFERIDA          NUMBER(5)      CONSTRAINT
    PK_ID_COBERTURA_DEFERIDA PRIMARY KEY,
    COBERTURA_DEF_AREA_HA           FLOAT(20)     CONSTRAINT
    NN_COBERTURA_DEF_AREA NOT NULL,
    COBERTURA_QTD CONTRATOS        NUMBER(20)     CONSTRAINT
    NN_COBERTURA_QTD CONTRATOS NOT NULL,
    COBERTURA_DEF_VALOR_AMPARADO_R$ FLOAT(20)     CONSTRAINT
    NN_COBERTURA_DEF_VALOR_AMPARADO NOT NULL,
    COBERTURA_VALOR_DEFERIDO_R$     FLOAT(20)     CONSTRAINT
    NN_COBERTURA_VALOR_DEFERIDO NOT NULL,
    ID_PROGRAMA                      NUMBER(5)      CONSTRAINT
    FK_COBERTURA_DEFERIDA_PROGRAMA REFERENCES
    PROGRAMA(ID_PROGRAMA),
    DATA_INSERCAO                   DATE         CONSTRAINT
    NN_DATA_INSERCAO_COBERTURA_DEFERIDA NOT NULL
);
```

## 3.3. VISUALIZAÇÃO DAS TABELAS NO SGBD

### 3.3.1 Comando "SHOW TABLES;"

The screenshot shows the MySQL Workbench interface with a query editor titled 'Query 1'. The query window contains the following SQL code:

```
1 • use innovators;
2 • Show tables;
```

The results pane displays the following table structure:

Tables_in_innovators
ADESAO
AGRICULTOR
BAIRRO
COBERTURA_ANALISE
COBERTURA_DEFERIDA
COMUNICACAO
ESTADO
LOGRADOURO
LOGRADOURO_AGRICULTOR
MUNICIPIO
PROGRAMA
PROGRAMA_ESTADO
TIPO

### 3.3.2. Visualização dos Dados Importados do CSV.

#### Evidência de Validação dos Dados Importados na Tabela ADESAO

3 • `select * from ADESAO;`

ID_ADESAO	ADESAOAREA_AMPARADA_HA	ADESAO_QTDE_CONTRATOS	ADESAO_VALOR_AMPARADO_R	AD
300	4793804,40	388094	21997065627,43	114:
301	2765753,20	196541	12568931983,07	723:
302	625706,10	90991	5509490593,38	238
303	174380,90	50383	3152658138,89	106:
304	81649,60	22830	1614513681,58	470:
305	70383,00	10255	704620941,36	245:
306	7065,10	3843	209260790,10	561:
307	147268,60	12532	766605308,26	272:
308	126938,20	7677	518022935,38	193:
309	17060,20	2386	103720838,73	338:
310	8917,40	1794	114611675,81	335:
311	8951,20	964	34850082,54	123:
312	7270,90	868	40083430,72	117:
313	1423,50	437	20504801,27	593:
314	4438,40	339	18460014,49	551:
315	2575,90	475	7562423,35	256:
316	72084,40	4115	276182756,21	131:

### 3.3.3. Evidência de Validação dos Dados Importados na Tabela AGRICULTOR

3 • `select * from AGRICULTOR;`

ID_AGRICULTOR	NOME_AGRICULTOR	CPF_AGRICULTOR	RG_AGRICULTOR	EMAIL_AGRICULTOR
1	Rayssa Helena Souza	66288334103	207176413	rayssa_souza@dm
2	Carolina Teresinha Alves	21371706808	416097522	carolina_alves@dm
3	Adriana Manuela Catarina Dias	15355057101	261138017	adriana_dias@tre
4	Jorge Igor Eduardo Cardoso	62942008148	137150891	jorge_igor_cardos
5	Renato Victor Jesus	40921130139	412013800	renato_victor_jes
6	Ian Roberto da Cruz	28647530497	251552184	ian_roberto_da_cru
7	Bento Otávio Fernando Nascimento	27771226523	289197569	bentootavionascim
8	Isabelle Vitória Lima	38589639410	341680564	isabelle_lima79@il
9	Victor Arthur Gael Fogaca	90290938201	506456122	victor_arthur_foga
10	Manuel Leandro Marcos Vinícius Cavalcanti	32060645549	322954186	manuel_leandro_c
11	Daiâne Giovana Pereira	82279385589	498696844	daiane.giovana.pe
12	Aurora Manuela Lera Ramos	72592948414	170017758	aurora.manuela.ra
13	Benedicto Bruno da Costa	95721916400	411936207	benedicto-dacoste
14	Gael Thales Barros	24262373258	162339124	gael-barros74@ro
15	Mariana Matiê Tereza de Silva	35080655769	178451691	marianna_desilva1
16	Otávio Thomas Melo	57505655132	369559769	otavio_melo82@hk
17	Antônia Louise Viana	667516161070	426452082	antonia_viana72@

### 3.3.4 Evidência de Validação dos Dados Importados na Tabela ESTADO

3 • `select * from ESTADO;`

ID_UF	SIGLA_UF	DATA_INSCERCAO
11	RO	2023-09-28
12	AC	2023-09-28
13	AM	2023-09-28
14	RR	2023-09-28
15	PA	2023-09-28
16	AP	2023-09-28
17	TO	2023-09-28
21	MA	2023-09-28
22	PI	2023-09-28
23	CE	2023-09-28
24	RN	2023-09-28
25	PB	2023-09-28
26	PE	2023-09-28
27	AL	2023-09-28
28	SE	2023-09-28
29	BA	2023-09-28
31	MG	2023-09-28
32	ES	2023-09-28

### 3.3.5. Evidência de Validação dos Dados Importados na Tabela MUNICIPIO

3 • select \* from MUNICIPIO;

	ID_MUNICIPIO	ID_UF	NOME_MUNICIPIO	DATA_INSERCAO
1	11	11	Alta Floresta D'Oeste	2023-09-28
2	11	11	Alto Alegre dos Parecis	2023-09-28
3	11	11	Alto Paráso	2023-09-28
4	11	11	Alvorada D'Oeste	2023-09-28
5	11	11	Anquemes	2023-09-28
6	11	11	Buritis	2023-09-28
7	11	11	Cabixi	2023-09-28
8	11	11	Cacaulândia	2023-09-28
9	11	11	Cacoal	2023-09-28
10	11	11	Campo Novo de Rondônia	2023-09-28
11	11	11	Candeias do Jamari	2023-09-28
12	11	11	Castanheiras	2023-09-28
13	11	11	Cerejeiras	2023-09-28
14	11	11	Chupinguaia	2023-09-28
15	11	11	Colorado do Oeste	2023-09-28
16	11	11	Corumbárea	2023-09-28
17	11	11	Costa Marques	2023-09-28
18	11	11	Cujubim	2023-09-28

### 3.3.6. Evidência de Validação dos Dados Importados na Tabela BAIRRO

3 • select \* from BAIRROS;

	ID_BAIRRO	ID_MUNICIPIO	NOME_BAIRRO	DATA_INSERCAO
1	3315	3315	RESSACA	2023-09-28
2	3315	3315	BAIRRO CAMPO DOS ALEIXOS	2023-09-28
3	3315	3315	GUAXINDUVA	2023-09-28
4	3315	3315	MARACANA	2023-09-28
5	3315	3315	BAIRRO ITAPETINGA	2023-09-28
6	3315	3315	MORUMBI	2023-09-28
7	3315	3315	ATIBAIA	2023-09-28
8	3315	3315	BOA VISTA	2023-09-28
9	3315	3315	BAIRRO DO TANQUE	2023-09-28
10	3315	3315	CAETETUBA	2023-09-28
11	3315	3315	PONTE ALTA	2023-09-28
12	3315	3315	CACHOEIRA	2023-09-28
13	3315	3315	RIO ABAXO	2023-09-28
14	3315	3315	RIO ACIMA	2023-09-28
15	3315	3315	CAICARA	2023-09-28
16	3315	3315	SICHA ABC	2023-09-28
17	3315	3315	PEDREIRA	2023-09-28
18	3315	3315	SUL BRASIL	2023-09-28

### 3.3.7. Evidência de Validação dos Dados Importados na Tabela LOGRADOURO

3 • select \* from LOGRADOURO;

	ID_LOGRADOURO	ID_TIPO	ID_BAIRRO	NUMERO_CEP	NOME_LOGRADOURO
1	1	1	1	12940060	AVENIDA JERONIMO CAMARGO
2	9	2	2	12940970	ESTRADA BENTO SOARES
3	9	21	21	12940970	ESTRADA BENTO SOARES
4	9	2	2	12940970	ESTRADA BENTO SOARES
5	9	2	2	12940970	ESTRADA BENTO SOARES
6	9	2	2	12940970	ESTRADA BENTO SOARES
7	9	21	21	12940970	ESTRADA DA USINA
8	9	21	21	12940970	ESTRADA DA USINA
9	9	21	21	12940970	ESTRADA HISAIKI TAKEBYASHI
10	9	21	21	12940970	ESTRADA HISAIKI TAKEBYASHI
11	9	21	21	12940970	ESTRADA HISAIKI TAKEBYASHI
12	9	21	21	12940970	ESTRADA HISAIKI TAKEBYASHI
13	3	3	3	12945751	RUA INOSUKE ANDO
14	3	3	3	12940060	RUA GIRASSOL
15	9	3	3	12940110	ESTRADA FAZENDA SOBERANA
16	9	4	4	12940060	ESTRADA AMAZONAS

### 3.3.8. Evidência de Validação dos Dados Importados na Tabela LOGRADOURO\_AGRICULTOR

3 • `select * from LOGRADOURO_AGRICULTOR;`

ID_LOGRADOURO_AGRICULTOR	ID_LOGRADOURO	ID_AGRICULTOR	DATA_INSCERCAO
1001	76	28	2023-09-28
1002	151	43	2023-09-28
1003	235	26	2023-09-28
1004	318	23	2023-09-28
1005	393	52	2023-09-28
1006	468	29	2023-09-28
1007	543	46	2023-09-28
1008	75	10	2023-09-28
1009	150	52	2023-09-28
1010	234	23	2023-09-28
1011	317	57	2023-09-28
1012	392	37	2023-09-28
1013	467	72	2023-09-28
1014	542	14	2023-09-28
1015	74	23	2023-09-28
1016	149	48	2023-09-28
1017	...	...	.....

Ativ

### 3.3.9. Evidência de Validação dos Dados Importados na Tabela PROGRAMA

Query 1 x

1 • `use innovators;`

2 • `Show tables;`

3 • `select * from PROGRAMA;`

ID_PROGRAMA	NOME_PROGRAMA	DATA_INSCERCAO
101	PRONAF	2023-09-28
102	PRONAMP	2023-09-28
*	NULL	NULL

### 3.3.10. Evidência de Validação dos Dados Importados na Tabela PROGRAMA\_ESTADO

3 • `select * from PROGRAMA_ESTADO;`

ID_PROGRAMA_ESTADO	ID_UF	ID_PROGRAMA	DATA_INSCERCAO
10101	35	101	2023-09-28
10102	41	101	2023-09-28
10103	42	101	2023-09-28
10104	43	101	2023-09-28
10105	50	101	2023-09-28
10106	11	101	2023-09-28
10107	12	101	2023-09-28
10108	13	101	2023-09-28
10109	14	101	2023-09-28
10110	15	101	2023-09-28
10111	16	101	2023-09-28
10112	17	101	2023-09-28
10113	21	101	2023-09-28
10114	24	101	2023-09-28
10115	25	101	2023-09-28
10116	26	101	2023-09-28

### 3.3.11. Evidência de Validação dos Dados Importados na Tabela TIPO

3 • `select * from TIPO;`

ID_TIPO	NOME_TIPO	DATA_INSCAO
1	AVENIDA	2023-09-28
2	ESTRADA	2023-09-28
3	RUA	2023-09-28
4	SITIO	2023-09-28
5	RODOVIA	2023-09-28
6	FAZENDA	2023-09-28
7	CHACARA	2023-09-28
8	ACAMPAMENTO	2023-09-28
9	ENTRADA	2023-09-28
10	SERVIDAO	2023-09-28
HULL	HULL	HULL

### 3.3.12. Evidência de Validação dos Dados Importados na Tabela COMUNICACAO

3 • `select * from COMUNICACAO;`

ID_COMUNICACAO	COMUNICACAO_PERDA_AMPARADA_HA	COMUNICACAO_PERDA_QTD_CONTRATOS	COMU
400	2244847.60	179150	116922
401	1228149.60	78825	522556
402	180473.60	30346	199532
403	5391.80	1005	915896
404	2457.40	528	592093
405	4559.10	436	308005
406	189.20	147	973493
407	42365.50	2277	172087
408	30504.60	1768	139749
409	1665.70	68	834922
410	130.10	9	106965
411	0.00	0	0.00
412	153.30	23	566863
413	8.70	4	172369
414	498.40	11	131928

### 3.3.13. Evidência de Validação dos Dados Importados na Tabela COBERTURA\_ANALISE

3 • `select * from COBERTURA_ANALISE;`

ID_COBERTURA_ANALISE	COBERTURA_ANALISE_AREA_AMPARADA_HA	COBERTURA_ANALISE_QTD_CONTRATOS
200	5.03	622
201	29177.30	1660
202	1064.50	176
203	359.00	101
204	86.00	26
205	119.90	12
206	24.80	20
207	670.00	24
208	783.40	31
209	35.40	3
210	4.00	1
211	0.00	0
212	2.10	1
213	1.00	1
214	50.10	1
215	0.00	0

### 3.3.14. Evidência de Validação dos Dados Importados na Tabela COBERTURA\_DEFERIDA

3 • `select * from COBERTURA_DEFERIDA;`

ID_COBERTURA_DEF	COBERTURA_DEF_AREA_HA	COBERTURA_DEF_QTD CONTRATOS	COBERTURA_DEF_VI
500	2178374.10	173595	11368373539.67
501	1108648.40	71830	4771583742.77
502	164125.40	28102	1840594720.21
503	3656.60	674	62762515.62
504	1747.60	393	45421609.80
505	3282.20	309	21055562.85
506	138.60	106	6993736.49
507	40782.20	2210	165034327.04
508	28985.30	1697	132509618.39
509	1498.10	57	7509157.49
510	38.40	5	505519.35
511	0.00	0	0.00
512	77.40	8	317962.88
513	1.00	1	27192.00
514	329.10	8	879292.25
515	0.00	0	0.00

## 4. ARQUITETURA

A documentação inicia-se partindo do princípio do login já efetuado no OCI (Oracle Cloud):

### 4.1. CRIAÇÃO DA INSTÂNCIA NO ORACLE CLOUD INFRASTRUCTURE

#### Home

**4.1.2.** Efetuação do clique “Create a VM Instance” abaixo, para criação de uma instância e iniciar o processo da instalação da máquina.

#### 4.1.3. No campo ‘name’ inserindo o nome da instância:

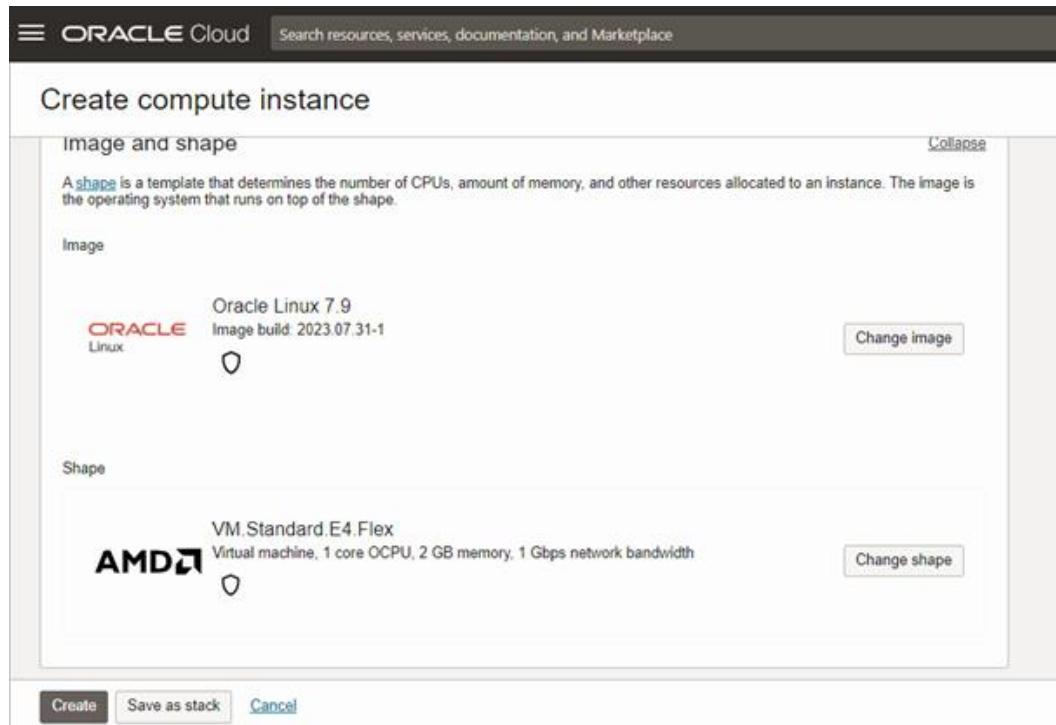
The screenshot shows the 'Create compute instance' wizard. In the 'Name' field, the value 'challenge-innovators-orc' is entered. Other fields like 'Create in compartment' and 'Placement' (Availability domain: AD-1, Fault domain: Let Oracle choose the best fault domain) are also visible.

#### 4.1.4. Configuração da Image e Shape

Nas opções Image and Shape, foram escolhidos as versões Oracle Linux 9.7 e shape AMD com 2GB e 1OCPUs, pois na instação do Oracle o grupo escolheu a versão 21C e também pudemos analisar segundo a documentação da faq da oracle: <https://www.oracle.com/br/database/technologies/appdev/xe/faq.html> precisaríamos de pelo menos 2gb, pois 1gb já não comportaria na versão free.

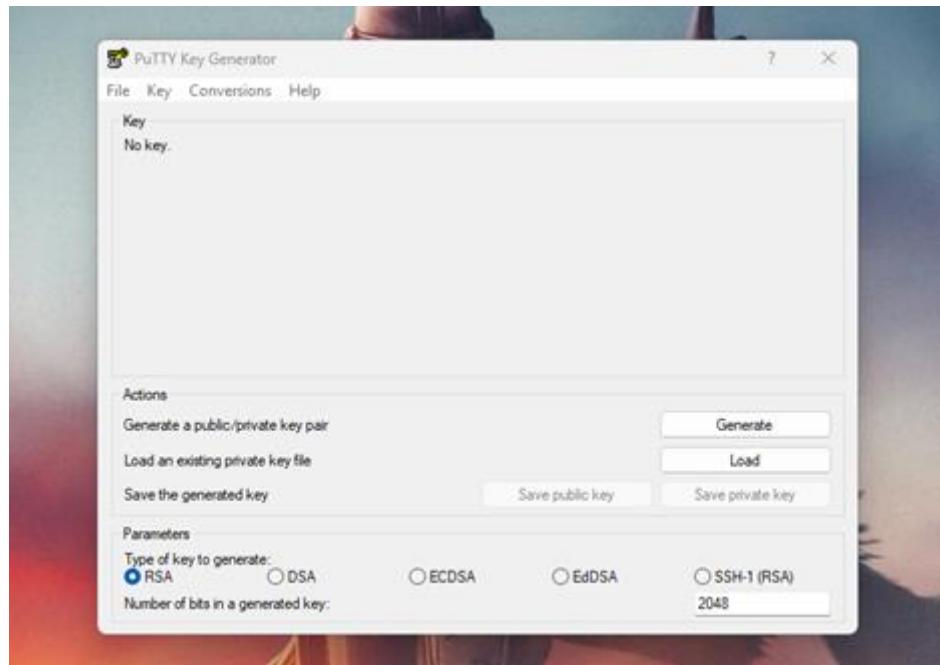
The screenshot shows the 'Create compute instance' wizard. On the left, under 'Image and shape', the 'Image' section is selected, showing 'Oracle Linux 7.9' and 'VM Standard E2.1 Micro'. On the right, the 'Browse all shapes' panel is open, showing configuration options for 'Number of OCPUs' (1 to 114) and 'Amount of memory (GB)' (2 to 1760). A note about 'Burstable' instances is also present.

#### 4.1.5. Image e Shape configurados.

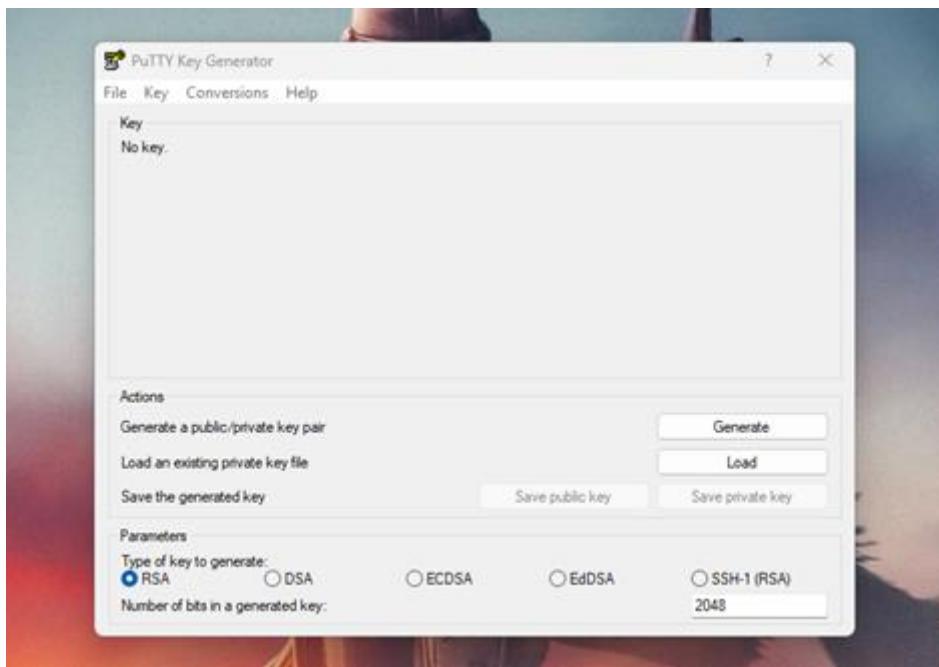


#### 4.1.6. Geração da Chave SSH

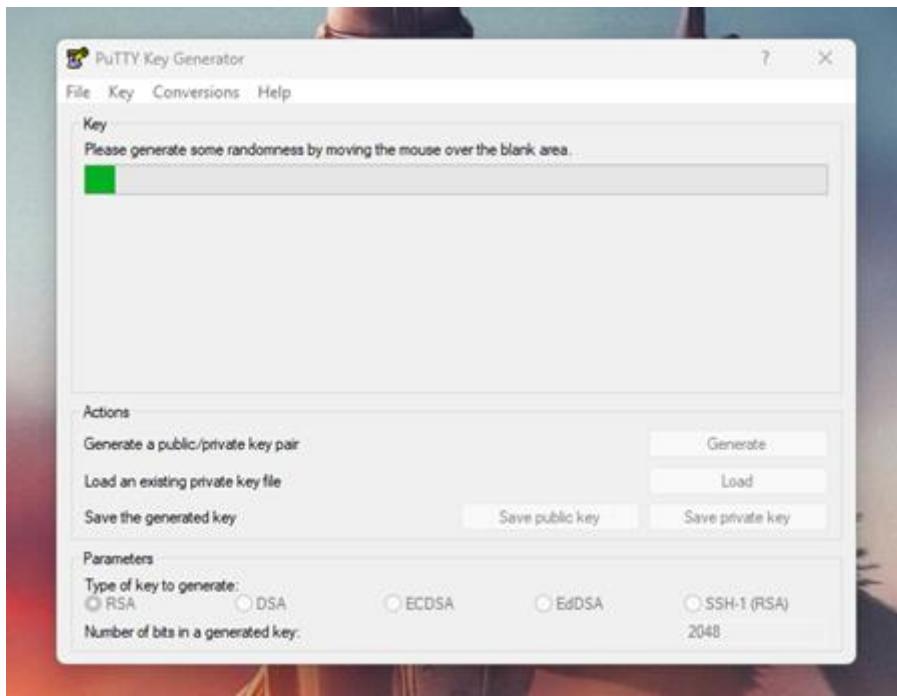
Para gerar a chave SSH, foi utilizado o Putty conforme o material e aula 05 que tivemos no início, onde foi inserido o type da chave e clicado em “generate”.



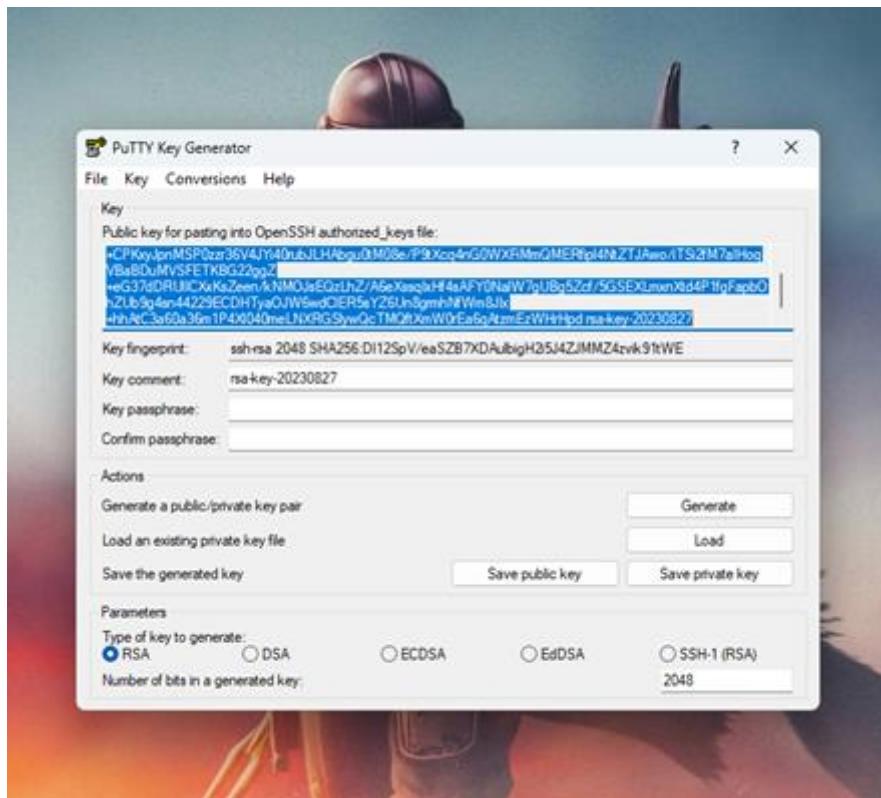
#### 4.1.7. Gerando a chave.



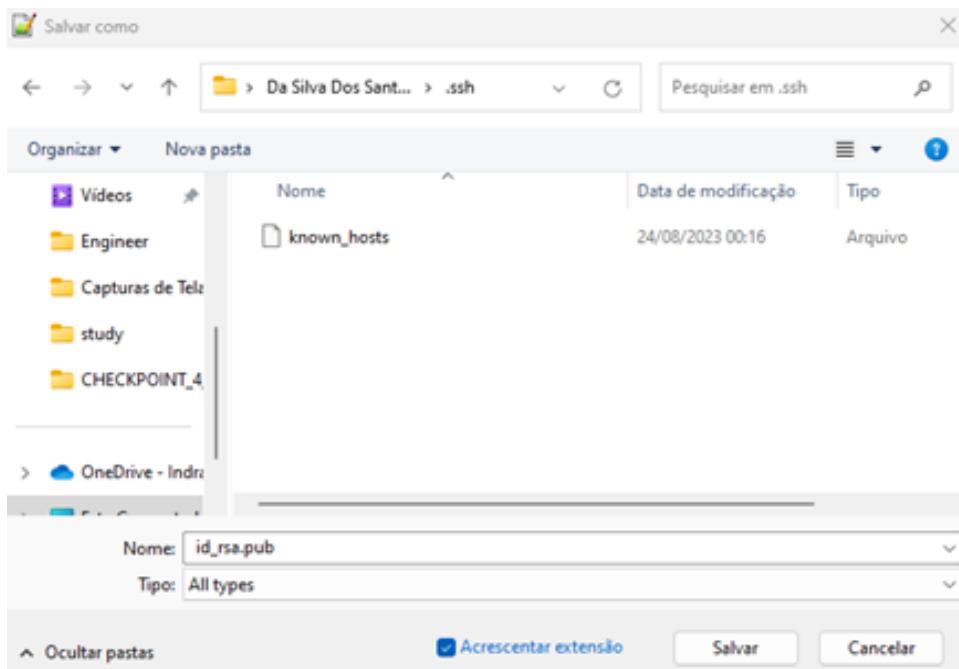
#### 4.1.8. Após a criação da chave, ela foi copiada e direcionada para o notepad, onde será aproveitada mais tarde:



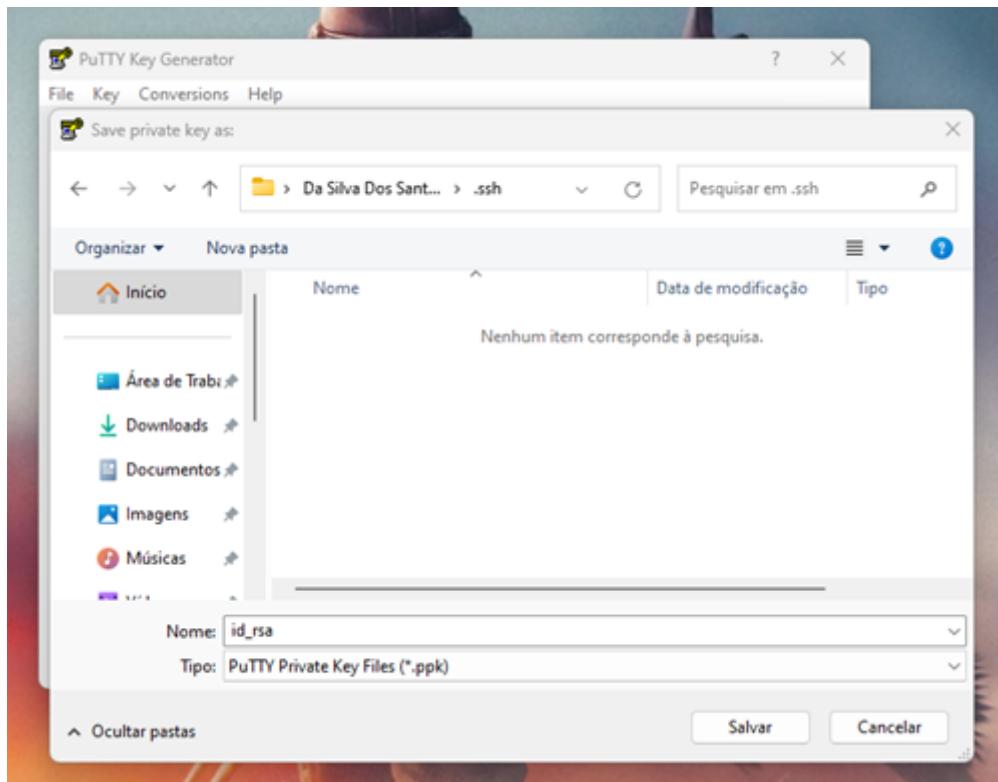
4.1.9. Geramos o primeiro arquivo, pasta .ssh dando o nome de “id\_rsa.pub” para melhor identificar a chave pública.



4.1.10. Depois geramos um outro arquivo da privada no formato .ppk dando o mesmo nome.



4.1.11. A importância desses arquivos estarem na pasta .ssh serve para facilitar a configuração e garantir o gerenciamento das chaves.



4.1.12. Arquivos salvos na .ssh.

Da Silva Dos Santos, Fáima > .ssh				
Nome	Data de modificação	Tipo	Tamanho	
id_rsa	27/08/2023 12:34	PuTTY Private Key File	2 KB	
id_rsa	27/08/2023 12:32	Documento do Microsoft Word	1 KB	
known_hosts	24/08/2023 00:16	Arquivo	1 KB	

#### 4.1.13. Configuração no OCI

Voltando para o OCI, utilizamos a chave copiada e optamos pela opção “paste public key”.

The screenshot shows the Oracle Cloud Compute Instance creation interface. In the top section, under "Add SSH keys", there is a text input field containing a long SSH public key string:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCRjUYVasmizyO6SA0lsxNjBrBtM0eLCQMYUFoHRjNc6XECFKToAHP5T8+P
```

Below the key input, there is a note about generating a key pair or uploading a file, with the "Paste public keys" option selected. A "Save as stack" button is visible at the bottom left of this section.

In the middle section, under "Boot volume", there is a note about what a boot volume is and a checkbox for "Specify a custom boot volume size".

At the bottom, there are three "Create" buttons: "Create", "Save as stack", and "Cancel".

The bottom part of the interface shows advanced options for live migration, with a toggle switch turned on. A link to "Show advanced options" is also present.

4.1.14. Em seguida selecionamos o checkbox na opção “use in-transit encryption”. A opção “live migration” normalmente já fica selecionada, pode deixar pois garantimos que a VM migre para um servidor de forma saudável.

Instância provisionada, ou seja, máquina ok!

#### 4.1.15. Instalação e Configuração do SGBD (Sistema para Banco de Dados)

4.1.16. Utilizamos a plataforma do MobaxTerm para fazer a instalação e configuração do SGBD, pelo usuário root.

Entrando com Moba utilizando os comandos pelo ssh pela instância gerada pelo OCI (Oracle Cloud Infrastructure) e exibimos todas as informações sobre a versão do sistema.

```

• MobaXterm Personal Edition v23.2 •
(X server, SSH client and network tools)

Your computer drives are accessible through the /drives path
Your DISPLAY is set to 192.168.15.82:0.0
When using SSH, your remote DISPLAY is automatically forwarded
Each command status is specified by a special symbol (✓ or ✘)

• Important:
This is MobaXterm Personal Edition. The Professional edition
allows you to customize MobaXterm for your company: you can add
your own logo, your parameters, your welcome message and generate
either an MSI installation package or a portable executable.
We can also modify MobaXterm or develop the plugins you need.
For more information: https://mobaxterm.mobatek.net/download.html

19/09/2023 18:01:37 /mnt/c/Users/fsdos/ssh cd /mnt/c/Users/fsdos/.ssh

19/09/2023 18:01:46 /mnt/c/Users/fsdos/ssh ll
total 3
-rwx----- 1 fsdos UsersGrp 1458 Sep 19 17:25 id_rsa.ppk
-rwx----- 1 fsdos UsersGrp 477 Sep 19 17:24 id_rsa.pub
-rwx----- 1 fsdos UsersGrp 95 Aug 24 00:16 known_hosts

19/09/2023 18:01:48 /mnt/c/Users/fsdos/ssh ssh -i id_rsa.ppk opc@129.148.61.109
Warning: Permanently added '129.148.61.109' (ED25519) to the list of known hosts.
X11 forwarding request failed on channel 0
[opc@innovators-mysql-challenge ~]$ cat /etc/os-release
NAME="Oracle Linux Server"
VERSION="7.9"
ID="ol"
ID_LIKE="fedora"
VARIANT="Server"
VARIANT_ID="server"

```

#### 4.1.17. Iniciando a instalação do Sistema para banco de dados MySQL:

The screenshot displays three terminal windows (labeled 2, 3, and 5) showing the MySQL installation process on a Linux system.

**Terminal 2 (Top):**

```
(11/20): ol7_UEKR6/aarch64/updateinfo
(12/20): ol7_ksplice/aarch64/primary_db
(13/20): ol7_ocl_included/aarch64/primary_db
(14/20): ol7_optional_latest/aarch64/updateinfo
(15/20): ol7_software_collections/aarch64/updateinfo
(16/20): ol7_latest/aarch64/updateinfo
(17/20): ol7_software_collections/aarch64/primary_db
(18/20): ol7_optional_latest/aarch64/primary_db
(19/20): ol7_UEKR6/aarch64/primary_db
(20/20): ol7_latest/aarch64/primary_db
=====
===== N/S matched: mysql =====
mysql-python.aarch64 : An interface to MySQL
akonadi-mysql.aarch64 : Akonadi MySQL backend support
apr-util-mysql.aarch64 : APR utility library MySQL DBD driver
dovecot-mysql.aarch64 : MySQL back end for dovecot
freeradius-mysql.aarch64 : MySQL support for freeradius
libdbi-db2-mysql.aarch64 : MySQL plugin for libdbi
mysql-community-client.aarch64 : MySQL database client applications and tools
mysql-community-client-plugins.aarch64 : Shared plugins for MySQL client applications
mysql-community-common.aarch64 : MySQL database common files for server and client libs
mysql-community-devel.aarch64 : Development header files and libraries for MySQL database client applications
mysql-community-embedded-compat.aarch64 : MySQL embedded compat library
mysql-community-icu-data-files.aarch64 : MySQL packaging of ICU data files
mysql-community-libs.aarch64 : Shared libraries for MySQL database client applications
mysql-community-libs-compat.aarch64 : Shared compat libraries for MySQL 5.6.51 database client applications
mysql-community-server-debug.aarch64 : The debug version of MySQL server
mysql-community-test.aarch64 : Test suite for the MySQL database server
mysql-connector-c++.aarch64 : MySQL database connector for C++
mysql-connector-c++-devel.aarch64 : Development header files and libraries for MySQL C++ client applications
mysql-connector-c++-jdbc.aarch64 : MySQL Driver for C++ which mimics the JDBC 4.0 API
mysql-connector-j.noarch : Standardized MySQL database driver for Java
mysql-connector-java.noarch : Standardized MySQL database driver for Java
mysql-connector-odbc.aarch64 : An ODBC 8.0 driver for MySQL - driver package
mysql-connector-odbc-setup.aarch64 : An ODBC 8.0 driver for MySQL - setup library
mysql-connector-python.aarch64 : Standardized MySQL database driver for Python
mysql-connector-python3.aarch64 : Standardized MySQL database driver for Python 3
mysql-release-el7.aarch64 : MySQL yum repository configuration
mysql-router-community.aarch64 : MySQL Router
```

**Terminal 3 (Middle):**

```
Re-enter new password:
The 'validate_password' component is installed on the server.
The subsequent steps will run with the existing configuration
of the component.
Using existing password for root.

Estimated strength of the password: 100
Change the password for root ? (Press y|Y for Yes, any other key for No) : y

New password:
Re-enter new password:

Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.
```

**Terminal 5 (Bottom):**

```
Package          Arch      Version       Repository   Size
=====
Installing:
  mysql-community-server      aarch64    8.0.33-1.el7      ol7_MySQL80   62 M
Installing for dependencies:
  mysql-community-client      aarch64    8.0.33-1.el7      ol7_MySQL80   16 M
  mysql-community-icu-data-files aarch64  8.0.33-1.el7      ol7_MySQL80   2.1 M
Transaction Summary
Install 1 Package (+2 Dependent packages)

Total download size: 80 M
Installed size: 375 M
Downloading packages:
(1/3): mysql-community-icu-data-files-8.0.33-1.el7.aarch64.rpm | 2.1 MB 00:00:00
(2/3): mysql-community-client-8.0.33-1.el7.aarch64.rpm        | 16 MB 00:00:02
(3/3): mysql-community-server-8.0.33-1.el7.aarch64.rpm        | 62 MB 00:00:06
Total                                         11 MB/s | 80 MB 00:00:07
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mysql-community-icu-data-files-8.0.33-1.el7.aarch64      1/3
  Installing : mysql-community-client-8.0.33-1.el7.aarch64                2/3
  Installing : mysql-community-server-8.0.33-1.el7.aarch64                 3/3
  Verifying  : mysql-community-client-8.0.33-1.el7.aarch64                  1/3
  Verifying  : mysql-community-server-8.0.33-1.el7.aarch64                 2/3
  Verifying  : mysql-community-icu-data-files-8.0.33-1.el7.aarch64            3/3
Installed:
  mysql-community-server.aarch64 0:8.0.33-1.el7
Dependency Installed:
  mysql-community-client.aarch64 0:8.0.33-1.el7
  mysql-community-icu-data-files.aarch64 0:8.0.33-1.el7
```

4.1.18. Instalação feita com sucesso, realizamos a definição de senhas temporárias e definitivas no root e usuário adm para compartilhamento com colegas.

```

Sep 19 21:23:44 innovators-mysql-challenge systemd[1]: Started MySQL Server.
[root@innovators-mysql-challenge ~]# cat /var/log/mysqld.log |grep "A temporary password"
2023-09-19T21:23:39.996212Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: L_q>4x2q0jMd
[root@innovators-mysql-challenge ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.33

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
->;
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
mysql> show databases;
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
mysql> exit
Bye
[root@innovators-mysql-challenge ~]# mysql_secure_installation
-bash: mysql_secure_installation: command not found
[root@innovators-mysql-challenge ~]# mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:

The existing password for the user account root has expired. Please set a new password.

New password:
Re-enter new password:
The 'validate_password' component is installed on the server.


```

#### 4.1.19. Criação do Banco de Dados e Tabelas

Criação do banco de dados e logando no banco de dados para criação das tabelas:

```

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5816
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE innovators2;
Query OK, 1 row affected (0.02 sec)

mysql> use innovators2;
Database changed


```

4.1.20. Criação das tabelas, valendo ressaltar que todas que foram passadas a modelagem do oracle e adaptadas no mysql.

```

mysql> CREATE TABLE ESTADO (
    >     ID_UF NUMERIC(5) NOT NULL,
    >     SIGLA_UF CHAR(2) NOT NULL,
    >     DATA_INSCRICAO_DATE NOT NULL,
    >     CONSTRAINT PK_ID_UF PRIMARY KEY (ID_UF),
    >     CONSTRAINT NN_NAME_UF CHECK (ID_UF IS NOT NULL),
    >     CONSTRAINT NN_SIGLA_UF CHECK (SIGLA_UF IS NOT NULL),
    > );
Query OK, 0 rows affected (0.02 sec)


```

#### 4.1.21. Inserção de Dados nas Tabelas

Inserindo dados nas tabelas:

```
mysql> INSERT INTO ESTADO (ID_UF, SIGLA_UF, DATA_INSCRICAO) VALUES
-- (35, 'SP', '2023-09-28'),
-- (41, 'PR', '2023-09-28'),
-- (42, 'SC', '2023-09-28'),
-- (43, 'RS', '2023-09-28'),
-- (50, 'MS', '2023-09-28'),
-- (10, 'GO', '2023-09-28'),
-- (12, 'AC', '2023-09-28'),
-- (13, 'AM', '2023-09-28'),
-- (14, 'RR', '2023-09-28'),
-- (15, 'PA', '2023-09-28'),
-- (16, 'AP', '2023-09-28'),
-- (17, 'TO', '2023-09-28'),
-- (21, 'MA', '2023-09-28'),
-- (24, 'RN', '2023-09-28'),
-- (25, 'PI', '2023-09-28'),
-- (26, 'CE', '2023-09-28'),
-- (27, 'AL', '2023-09-28'),
-- (28, 'SE', '2023-09-28'),
-- (29, 'BA', '2023-09-28'),
-- (31, 'MG', '2023-09-28'),
-- (33, 'RJ', '2023-09-28'),
-- (51, 'MT', '2023-09-28'),
-- (53, 'GO', '2023-09-28'),
-- (55, 'DF', '2023-09-28'),
-- (22, 'PI', '2023-09-28'),
-- (23, 'CE', '2023-09-28'),
-- (32, 'ES', '2023-09-28');

Query OK, 27 rows affected (0.00 sec)
Records: 27  Duplicates: 0  Warnings: 0
```

4.1.22. Utilizamos o Looker Studio para visualização de dados e fizemos a conexão com as tabelas do usuário adm, para isso entramos na plataforma > adicionar dados > escolhermos a opção do MYSQL:

The screenshot shows the 'Add Data' interface in Looker Studio. At the top, there's a search bar labeled 'Pesquisa'. Below it, a section titled 'Conectar aos dados' (Connect to data) is visible, with tabs for 'Conectar aos dados' and 'Minhas fontes de dados'. A search bar for 'Pesquisa' is also present here. Below the tabs, there are six data source cards arranged in two rows of three:

- Microsoft SQL Server**: By Google. Description: Conecte-se aos bancos de dados do Microsoft SQL Server.
- MySQL**: By Google. Description: Conectar a bancos de dados do MySQL.
- Novo Search Ads 360**: By Google. Description: Conecte sua conta aos dados do novo Search Ads 360.
- PostgreSQL**: By Google. Description: Conectar a bancos de dados do PostgreSQL.
- Search Ads 360**: By Google. Description: Conectar aos relatórios de desempenho do Search Ads 360.
- Search Console**: By Google. Description: Conectar a dados do Search Console.

4.1.23. Inserimos os dados do IP, porta, nome do banco e senha e clicamos e autenticar:

BÁSICO

Autenticação do banco de dados

IP ou nome do host  
129.148.61.109

Porta (opcional)  
3309

Banco de dados  
innovatorsadm

Nome de usuário  
innovators

Senha  
\*\*\*\*\*

Ativar SSL

**AUTENTICAR**

4.1.24. Selecionamos tabela por tabela para fazer as conexões para iniciar as atividades de visualizações de dados.

MySQL

Por Google

Com o conector do MySQL, você pode acessar dados de bancos de dados MySQL no Looker Studio.

**SAIBA MAIS** **INFORMAR UM PROBLEMA**

BÁSICO

Autenticação do banco de dados

IP ou nome do host  
129.148.61.109

Porta (opcional)

Banco de dados  
innovators

Nome de usuário  
Innovatorsadm

Senha

TABELAS

CONSULTA PERSONALIZADA

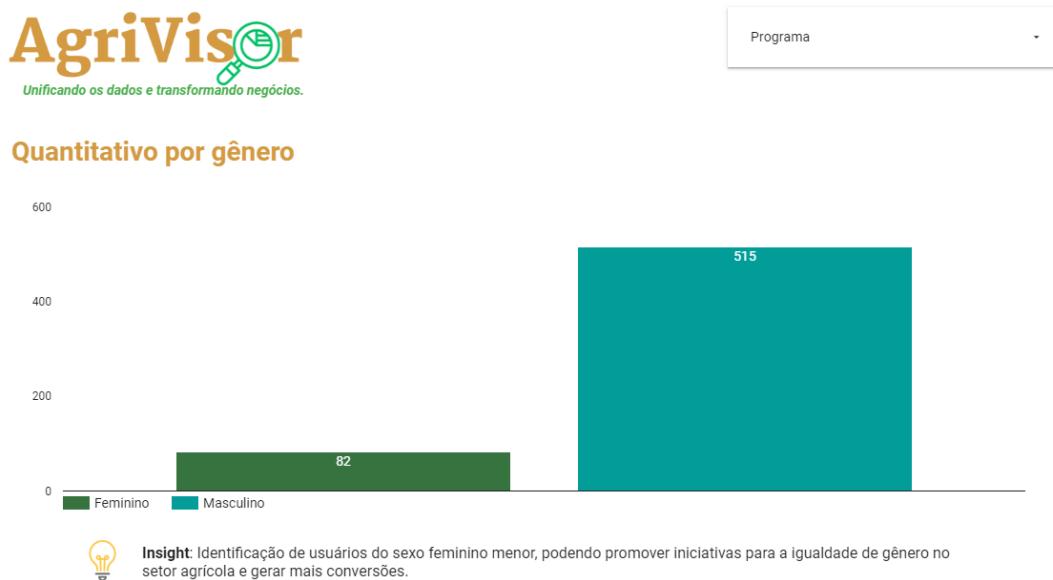
Tabela	Q
ADESAO	
AGRICULTOR	
BAIRRO	
COBERTURA_ANALISE	
COBERTURA_DEFERIDA	
COMUNICACAO	
ESTADO	
LOGRADOURO	
LOGRADOURO_AGRICULTOR	

## 5. VISUALIZAÇÃO DE DADOS NO GOOGLE LOOKER STUDIO

Nessa etapa integramos todos os datasets ao Google Looker Studio e geramos gráficos e insights como forma de demonstrar nossa solução final.

O dashboard foi dividido em três visões: Dados do Produtor, Análise de Risco Produto e Dados de Crédito.

### 5.1. VISÃO DO PRODUTOR



### Ranking de endereços por programa

Endereço	Programa	Qtd Prospects
1. ESTRADA NOVA UNIAO	PRONAF	30
2. ESTRADA NOVA UNIAO	PRONAMP	30
3. ESTRADA BENTO SOARES	PRONAF	21
4. ESTRADA BENTO SOARES	PRONAMP	21
5. ESTRADA LACERDA FRANCO	PRONAMP	20

1 - 100 / 420 < >

**Insight:** Podemos identificar top 5 endereços com agricultores para oportunidades de expansão de negócios nos casos de maior quantidade quando os menor quantidade para abrir mais negócios.

Qtd de Bairros  
**51**

Qtd de Endereços  
**579**

**Insight:** Revelancia para a tomada de decisões relacionadas a meta de cadastros.

Estabelecimentos por Idade	Unidades	< 25	> 35 < 45	> 45 < 55	> 55 < 65	> 75
	estabelecimentos	12	119	134	107	32

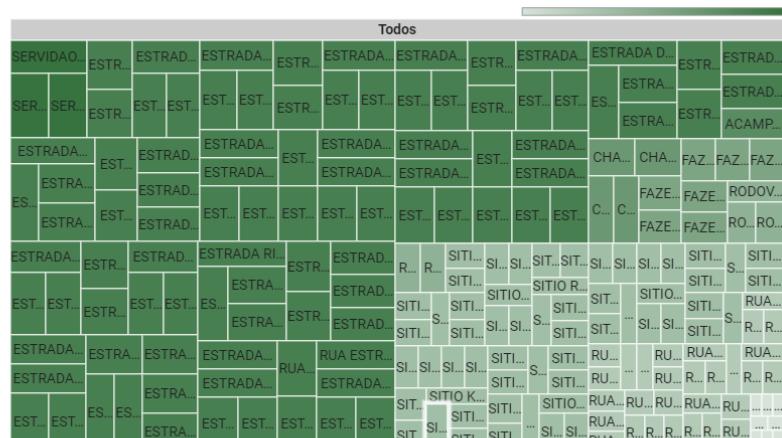


**Insight:** Estratégicamente é indicativo atuar em programas de incentivo para melhorar os negócios da segunda faixa de 35 - 45, uma vez que pode evoluir e até ultrapassar 45-55.

## Média de Endereços



**Insight:** Direcionamento dos recursos para atender os agricultores nos loureadores específicos identificando oportunidades.



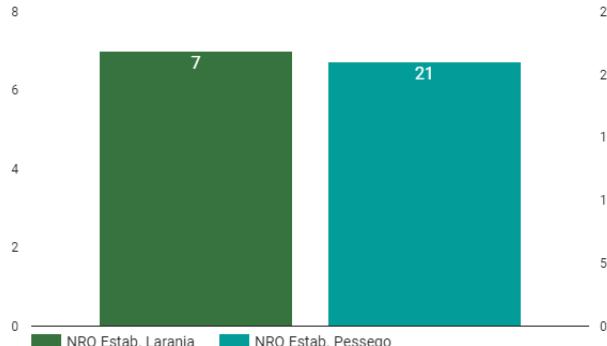
Parceria:  
Desenvolvido por: INNOVATORS minsoit

## 5.2. VISÃO ANÁLISE DE RISCO PRODUTO



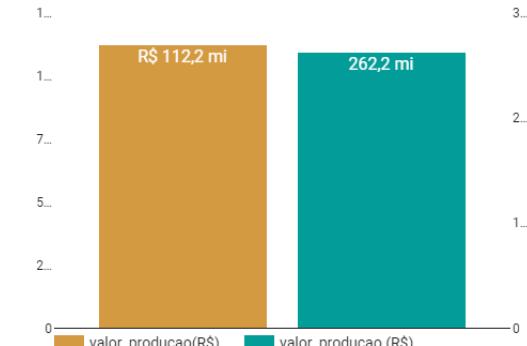
Programa

### Nº de estabelecimentos | Laranja & Pêssego Atibaia / SP

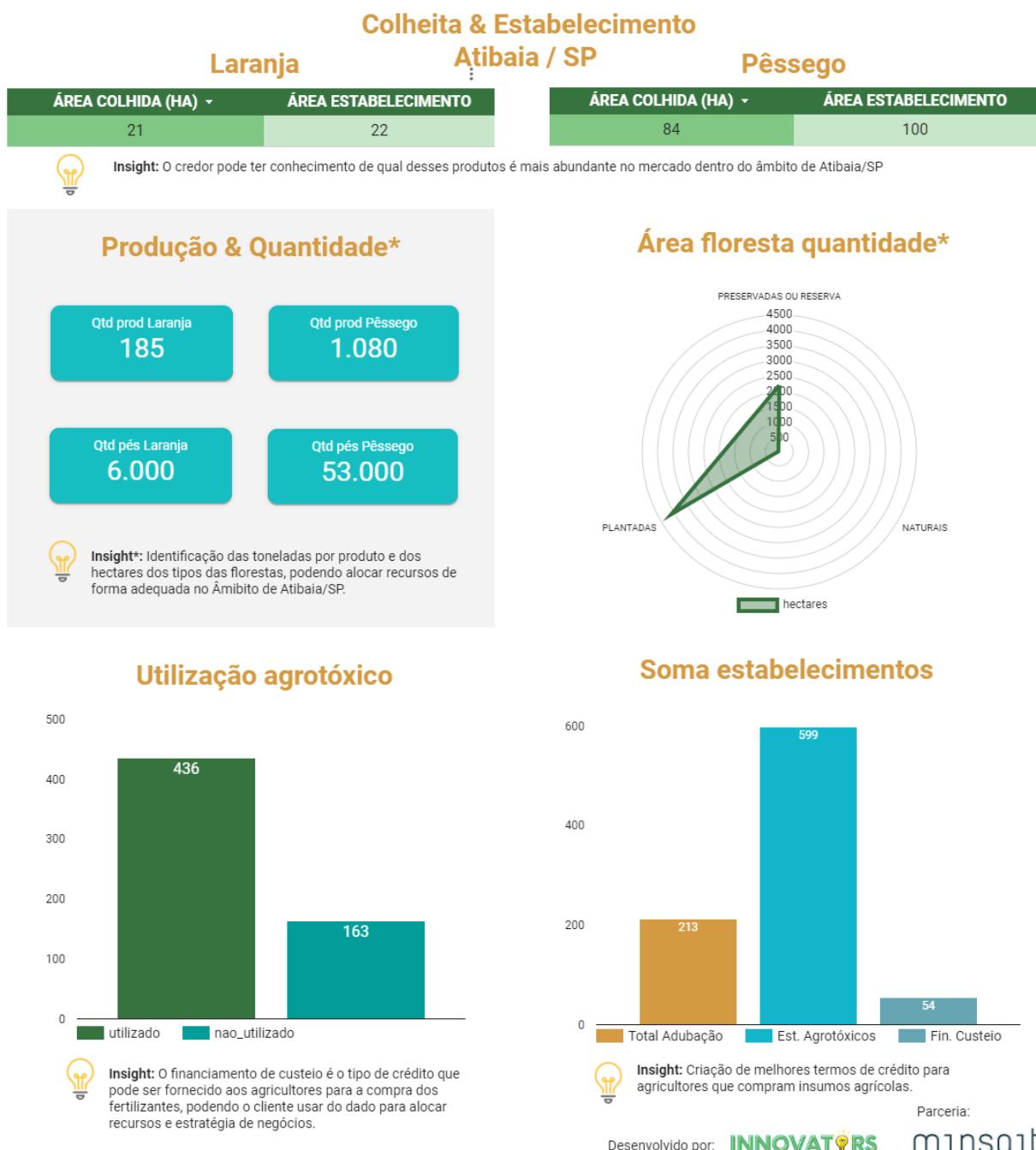


**Insight:** O credor poderá ter conhecimento de quantos estabelecimentos produzem Laranja e Pêssego no Âmbito de Atibaia/SP.

### R\$ de Produção | Laranja & Pêssego Atibaia / SP



**Insight:** Conhecimento de quanto se é gasto na produção, podendo alocar recursos, investimentos e esforços de produção que terá melhor retorno para o produtor.

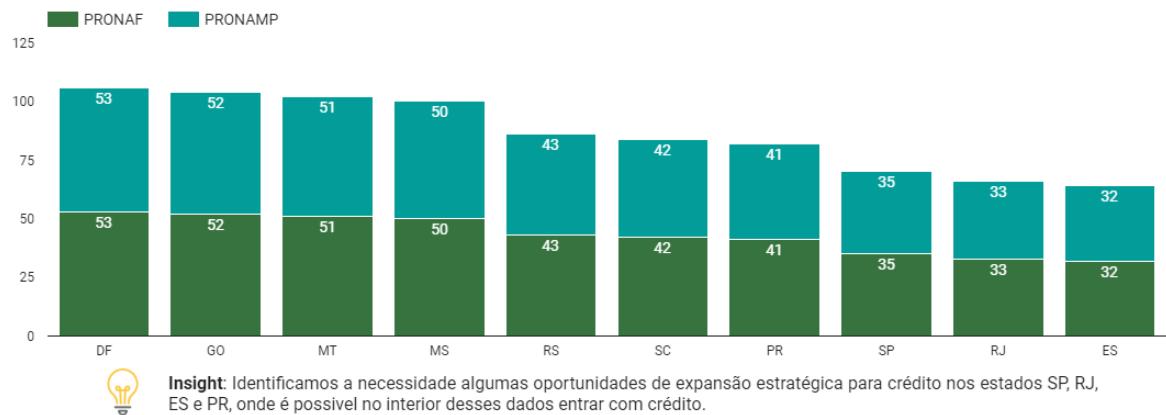


## 5.3 VISÃO DADOS DE CRÉDITO



Programa

### Programa por estado



### Ranking Adesão por estado

ID ADESÃO	ESTADO	ADESÕES ▾
1. 300	AC	388.094
2. 300	AL	388.094
3. 300	AM	388.094
4. 300	AP	388.094
5. 300	BA	388.094

1 - 100 / 1863    <    >

**Insight:** Identificamos o desempenho de diferentes adesões em vários estados para entender quais regiões são mais lucrativos, no caso Acre, Alagoas, Amazonas, Amapá e Bahia.

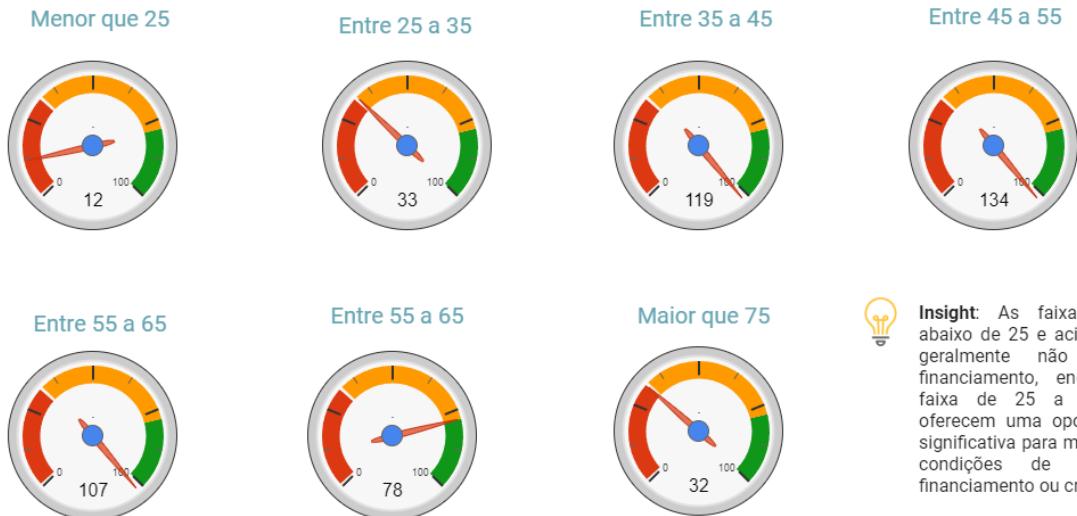
### Dados financiamento x Dados Condição Produtor

UNIDADE	PROPRIETÁRIOS...	ARRENDATARIOS	PRONAF	PRONAMP
hectares	9625	1619	15	5



**Insight:** A quantidade de hectares que o pronaf e pronamp cobrem é muito menor do que as condições de terra, então conclui-se que os produtores não estão pedindo empréstimo o suficiente de acordo com as terras.

## Financiamento por Idade



**Insight:** As faixas etárias abaixo de 25 e acima de 75 geralmente não buscam financiamento, enquanto a faixa de 25 a 35 anos oferecem uma oportunidade significativa para melhorar as condições de seja de financiamento ou crédito.

Parceria:

Desenvolvido por: **INNOVATORS**  **minsait**

## 5.4. LINKS DE ACESSO AS APRESENTAÇÕES

Segue link para acesso ao dashboard no Google Looker Studio:

<https://lookerstudio.google.com/reporting/4f1a3532-1e59-4d0c-ae03-78e20be83cff>

Segue Link de acesso ao nosso vídeo de apresentação da solução no Youtube:

<https://youtu.be/7KGId7FPouk>

## 7. CONCLUSÃO

Acreditamos que com esse projeto conseguiremos alcançar o objetivo de fornecer uma ferramenta extremamente eficaz para ajudar a reduzir a assimetria de crédito de rural no Brasil, e consequentemente fortalecer o setor agrícola e aumentar o nível de investimento nos pequenos e médios agricultores que são tão importantes para nossa economia e segurança alimentar.

Nosso objetivo, é alcançar um despertar de dados tanto nos pequenos e médios produtores rurais e nos credores, uma vez que, fica evidente que ao implementar um processo de coleta, processamento e análise de dados, muito mais oportunidades são abertas para ambos os lados, e como dito anteriormente, fortalecendo o setor e enfrentando o problema da assimetria de crédito rural no país.

Agradecemos a oportunidade de realizar esse projeto em parceria com a Minsait Brasil, uma vez que, este problema afeta a vida e o cotidiano de todos nós brasileiros, e entendê-lo e enfrentá-lo por meio da tecnologia e principalmente por meio de dados, torna o desafio mais tentador.

## 8. REFERÊNCIAS

- <https://www.ibge.gov.br/explica/codigos-dos-municípios.php> <Acessado em 28/08/2023>
- <https://cidades.ibge.gov.br/brasil/sp/atibaia/pesquisa/24/76693> <Acessado em 28/08/2023>
- <https://www.ibge.gov.br/estatísticas/económicas/agricultura-e-pecuária/21814-2017-censo-agropecuário.html?edição=23751&t=downloads> <Acessado em 15/09/2023>
- <https://www.embrapa.br/busca-de-publicações/-/publicação/1026588/ver-1-percepção-do-diagnóstico-ambiental-da-microrregião-de-atibaia/jaraguá-paradocação-da-produção-integrada-de-morango> <Acessado em 02/09/2023>
- <https://www.bcb.gov.br/resultadobusca?termo=DICION%C3%81RIO%20DE%20DADOS&source=> <Acessado em 17/08/2023>
- [https://www.bcb.gov.br/estabilidade-financeira/reportproagro?path=conteúdo%20FMDCR%2FReports%2FP\\_qvcProgramaSubprogramaRegiãoUF.rdl](https://www.bcb.gov.br/estabilidade-financeira/reportproagro?path=conteúdo%20FMDCR%2FReports%2FP_qvcProgramaSubprogramaRegiãoUF.rdl) <Acessado em 02/09/2023>
- <https://www.gov.br/agricultura/pt-br/acesso-a-informação/dados-abertos> <Acessado em 04/09/2023>
- <https://summitagro.estadao.com.br/notícias/do-campo/o-que-e-agricultura-familiar-e-qual-e-a-sua-importância/> <Acessado em 10/09/2023>
- <https://www.ibge.gov.br/estatísticas/económicas/agricultura-e-pecuária/21814-2017-censo-agropecuário.html?edição=23751&t=downloads> <Acessado em 12/09/2023>
- <https://www.atibaia.sp.gov.br/prefeitura/secretarias/agricultura> <Acessado em 29/08/2023>
- [https://municípioagro.agricultura.sp.gov.br/cidadaniano/campo/segurança/seg\\_L>Login/seg\\_Login.php](https://municípioagro.agricultura.sp.gov.br/cidadaniano/campo/segurança/seg_L>Login/seg_Login.php) <Acessado em 03/09/2023>
- <https://portaldatransparência.gov.br/localidades/3504107-atibaia> <Acessado em 04/09/2023>
- <https://transparência.cc/dados/cnpj/44513984000180-SP-cooperativa-agrícola-sul-brasil-de-atibaia/> <Acessado em 08/09/2023>
- <https://www.embrapa.br/manual-de-editoração/padrãoização-e-estilo/tabela> <Acessado em 15/09/2023>