

# MVP Engenharia de Dados

<b>1. Introdução</b>	<b>1</b>
1. Quais as perguntas que este projeto pretende responder?	1
<b>2. Fontes de Dados</b>	<b>2</b>
1. Quais as fontes dos dados?	2
2. Privacidade dos Dados	2
<b>3. Extração dos Dados</b>	<b>3</b>
1. Ferramentas Utilizadas	3
2. Coleta dos dados	3
<b>4. Modelagem dos Dados</b>	<b>4</b>
1. Descrição	4
2. Diagrama Modelo Snowflake	4
<b>5. Catálogo de Dados</b>	<b>5</b>
1. Tabelas de Portais de Imóveis (fonte de dados)	5
2. Tabelas de Cidades e Consolidação de Imóveis	6
3. Tabelas de CEP (Dados Geográficos)	7
4. Tabelas GeoBase	7
<b>6. ETL Databricks</b>	<b>8</b>
<b>7. Análise dos Dados</b>	<b>10</b>
<b>8. Respostas às perguntas</b>	<b>14</b>
<b>9. Autoavaliação</b>	<b>15</b>
<b>10. Conclusão</b>	<b>16</b>

# 1. Introdução

O desafio proposto para este projeto é elaborar um pipeline de dados que engloba a extração dos dados, o carregamento destes dados em uma nuvem pública, o tratamento e limpeza e a posterior análise utilizando os dados já preparados para responder às perguntas deste projeto.

Neste trabalho, decidi fazer uma comparação entre os valores de aluguéis de imóveis nas cidades do Rio de Janeiro e São Paulo, mais especificamente nos bairros mais valorizados e conhecidos de cada cidade: Zona Sul e Zona Oeste, respectivamente.

A escolha desta amostra é significativa para compreender algumas das principais diferenças entre estes dois mercados, porém não é suficiente para mapear todas as sutis diferenças que podem haver e que influem na valorização ou não de um imóvel.

Quando pensamos em moradia, em habitar um lar, muitas variáveis são avaliadas e cada pessoa possui suas próprias preferências e motivos que as levam a decidir por um lugar ou outro, por um tipo de imóvel em particular ou por uma região ou bairro da cidade.

Distância de pontos de interesse, predominância de comércio, proximidade de estações de metrô, oferta de transporte coletivo, áreas verdes, parques, proximidade do trabalho, densidade populacional, silêncio, bares e restaurantes na vizinhança. Entre muitos fatores, o preço, o tamanho, a quantidade de quartos, banheiros e garagens são fatores diretamente quantificáveis e presentes em qualquer anúncio de imóvel na internet. Quando pensamos em localização e bairro, obviamente entramos em um campo mais subjetivo, mais propenso às preferências pessoais.

## 1. Quais as perguntas que este projeto pretende responder?

Qual a média de preço de aluguel em cada cidade e porque esta métrica não é boa para compreendermos a dinâmica do mercado imobiliário?

Qual a média dos preços em alguns bairros mais importantes de cada cidade?

Qual o valor médio cobrado de aluguel por metro quadrado em cada cidade e por bairros?

Quais os tipos de imóveis mais ofertados? Estúdios? Apartamentos de 2 ou 3 dormitórios?

Quais os intervalos de valor dos apartamentos em São Paulo e Rio? Quais os valores máximos e mínimos?

Quais as distribuições dos preços estratificados por cidade, bairros e tipologias?

## 2. Fontes de Dados

### 1. Quais as fontes dos dados?

Os sites de busca de anúncios imobiliários no Brasil agregam milhares de anúncios e anunciantes em diversas plataformas. Descobri recentemente que dois principais grupos possuem a maior parte das plataformas atualmente, em um efeito de concentração de mercado que pode passar despercebido.

Ao analisar os códigos HTML de alguns destes portais, foi possível perceber estruturas de organização dos dados muito semelhantes, que refletem a propriedade deles. Por motivos de tempo para coleta de dados e porque decidi trabalhar com imóveis apenas das duas regiões mais importantes das cidades de São Paulo e Rio de Janeiro, optei por extrair os dados dos portais OLX, Viva Real e Zap Imóveis, que trabalham com esta classificação em Zona Sul do Rio de Janeiro e Zona Oeste de São Paulo. Estes três sites possuem uma estrutura bem semelhante e provavelmente pertencem ao mesmo grupo.

### 2. Privacidade dos Dados

Para extrair os dados de centenas ou milhares de anúncios, utilizei várias ferramentas de programação e bibliotecas Python. Em alguns casos utilizei apenas requests/response e BeautifulSoup para parsear e salvar os dados do código fonte HTML da página. Em outros casos foi preciso utilizar ferramentas mais complexas de web scraping, como Selenium e Scrapy.

Uma coisa em comum foi a necessidade de utilizar soluções de proxies para evitar bloqueio dos websites. O web scraping é uma atividade que os programadores podem realizar que, infelizmente, parece estar fora da lei, mas não está. Não acredito que recolher dados de um site com papel e caneta ou copiando e colando linhas em uma planilha seja tão diferente de recolher dados de forma programática. É apenas uma questão de escala, importante, sem dúvida, mas no fundo todos os dados utilizados já estão públicos nos sites e scrapear para fins de pesquisa (como o caso deste projeto) não fere nenhuma política de privacidade de dados.

Após muita pesquisa e várias tentativas, consegui acessar via API os dados do site VivaReal e dali consegui via linha de comando em Bash, recolher todos os dados já bem estruturados em formato json. Este método, apesar de mais complicado, a princípio, depois de aplicado resulta em uma coleta de dados bem mais eficiente e rápida, se a necessidade de escolher seletores xpath ou css para extrair dos sites apenas aquelas informações que estão presentes na página. Esse método via API recolhe muito mais informações e diretamente do servidor, muito além do que está disponível no site.

### 3. Extração dos Dados

#### 1. Ferramentas Utilizadas

Requests / Response - Biblioteca Python utilizada para fazer uma requisição http ou https a um site e obter a resposta. Em muitos casos a resposta pode ser um código de erro, como 403, 404, etc, o que indica que o site pode estar usando recursos de proteção contra excesso de requisições, uma das estratégias comuns de ataque DDOS para tirar um site do ar, estes sim criminosos.

Beautiful Soup - Uma ferramenta simples e eficaz para decodificar o HTML de um site e procurar o conteúdo desejado através de suas tags ou classes CSS. Para scrapear qualquer site, é necessário inspecionar e compreender a codificação HTML e em muitos casos também é preciso conhecer a linguagem JavaScript.

Selenium - Ferramenta capaz de automatizar várias operações feitas em web browsers, como por exemplo, movimentar o cursor, clicar em um elemento da página, preencher formulários, etc. Esta ferramenta é muito importante quando lidamos com sites que possuem conteúdos dinâmicos baseados em JavaScript, porém é uma ferramenta mais complexa, lenta e frequentemente associada a bots, portanto os sites possuem uma alta taxa de rejeição de requisições quando “percebem” que ela vem de um script Selenium.

Scrapy - É uma ferramenta bem rápida porém exige um pouco mais de habilidade de programação e pode não ser tão útil em sites muito baseados em JavaScript. Pode ser usada em conjunto com Selenium e com Beautiful Soup ou sozinha. Scrapy funciona com diversos componentes interconectados que enviam requisições, recebem as respostas, parseiam o código HTML ou JSON, extrai os dados utilizando seletores xpath ou css e salva os dados de forma estruturada já em json, csv ou outros formatos.

```
def start_requests(self):
    for page in range(1, 101):
        # Original URL
        url = f'https://www.zapimoveis.com.br/aluguel/apartamentos/sp+sao-paulo+zona-oeste/?transacao=aluguel&onde=,5%C3%A3o%20Paulo,
        # Proxy URL
        proxy = 'http://brd-customer-h1_72cf1a4b-zone-web_unlocker1:g98odse25ym@brd.superproxy.io:33335'
        yield scrapy.Request(url=url, callback=self.parse, meta={'proxy': proxy, 'page': page, 'dont_retry': False})

def parse(self, response, **kwargs):
    property_data = response.xpath('//div[@class="ListingCard_result-card__Pumtx"]')
    for p in property_data:
        yield {
            'link': p.xpath('//a[@itemprop="url"]/@href').get() or None,
            'preco': (p.xpath('//div[@data-cy="rp-cardProperty-price-txt"]/p[1]/text()').get().strip()
            if p.xpath('//div[@data-cy="rp-cardProperty-price-txt"]/p[1]/text()').get() else None),
            'endereco': p.xpath('//p[@data-cy="rp-cardProperty-street-txt"]/text()').get() or None,
            'title': p.xpath('//p[@data-cy="rp-cardProperty-location-txt"]/text()').get() or None,
            'area': p.xpath('normalize-space(//li[@data-cy="rp-cardProperty-propertyArea-txt"])').get() or None,
            'iptu': (p.xpath('//div[@data-cy="rp-cardProperty-price-txt"]/p[@class="text-1-75 text-neutral-110 overflow-hidden te
            if p.xpath('//div[@data-cy="rp-cardProperty-price-txt"]/p[@class="text-1-75 text-neutral-110 overflow-hidden text-elli
            'preco_condominio': (p.xpath('//div[@data-cy="rp-cardProperty-price-txt"]/p[@class="text-1-75 text-neutral-110 overflo
            if p.xpath('//div[@data-cy="rp-cardProperty-price-txt"]/p[@class="text-1-75 text-neutral-110 overflow-hidden text-elli
            'quantos': p.xpath('normalize-space(//li[@data-cy="rp-cardProperty-bedroomQuantity-txt"])').get() or None,
            'banheiros': p.xpath('normalize-space(//li[@data-cy="rp-cardProperty-bathroomQuantity-txt"])').get() or None,
            'garagem': p.xpath('normalize-space(//li[@data-cy="rp-cardProperty-parkingSpacesQuantity-txt"])').get() or None
        }

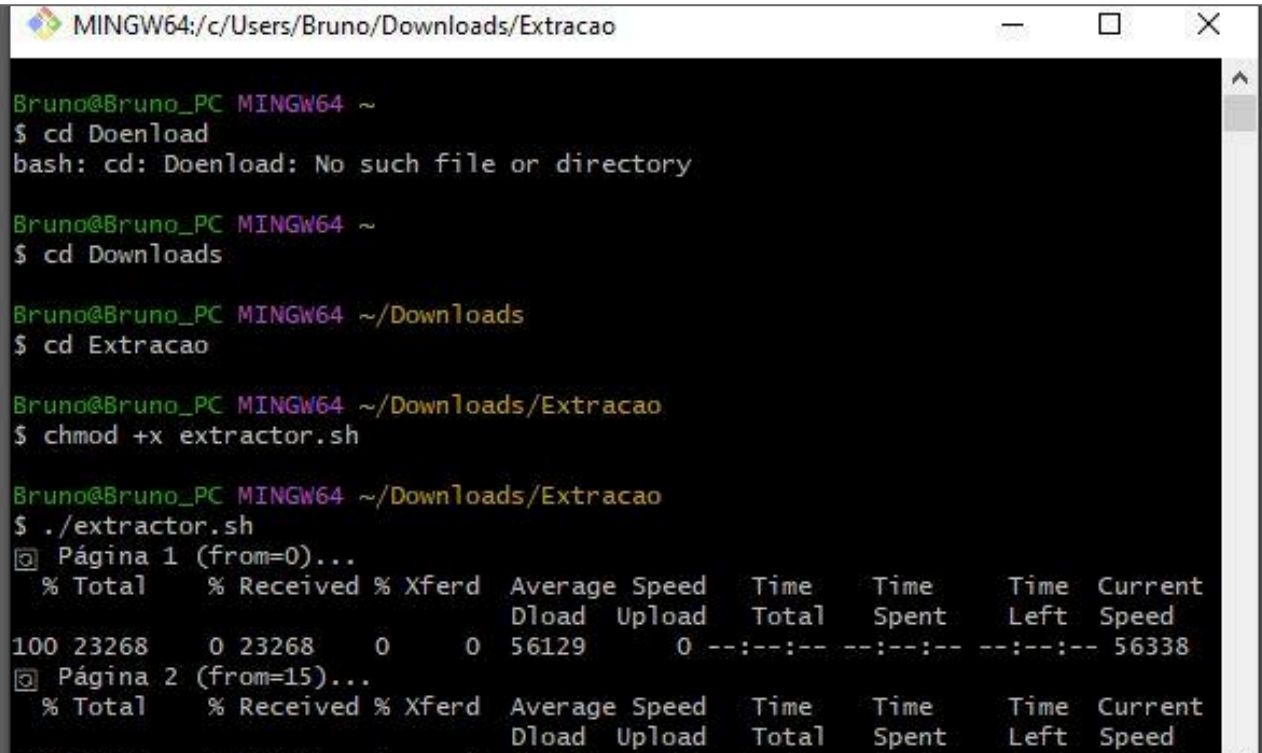
Run the spider using CrawlerProcess instead of CrawlerRunner and reactor
```

## 2. Coleta dos dados

Os sites Viva Real foram scrapeados com request/response e Beautiful Soup, mas foi preciso primeiro extrair apenas os links de cada imóvel e depois entrar em cada link para extrair os dados. Tudo isso feito de forma programática, porém foi um processo mais lento e que exigiu muitas requisições.

Os sites OLX e ZAP foram scrapeados utilizando Selenium e Scrapy, também utilizando serviço de proxy e rotação de IPs para evitar bloqueio. Porém nesses sites foi possível extrair os dados diretamente da resposta em formato JSON, de uma forma mais estruturada e direta. Depois, como descrito anteriormente, também baixei parte dos dados direto de uma API usada pelo site, de forma muito mais completa e eficiente.

Como o objetivo do trabalho é experimentar várias técnicas diferentes, salvei alguns arquivos em formato CSV e outros em formato JSON, simulando uma situação comum da realidade da profissão de engenheiro ou cientista de dados, onde precisamos saber lidar com diferentes tipos de arquivos e ferramentas.



```
MINGW64: c:/Users/Bruno/Downloads/Extracao

Bruno@Bruno_PC MINGW64 ~
$ cd Doenload
bash: cd: Doenload: No such file or directory

Bruno@Bruno_PC MINGW64 ~
$ cd Downloads

Bruno@Bruno_PC MINGW64 ~/Downloads
$ cd Extracao

Bruno@Bruno_PC MINGW64 ~/Downloads/Extracao
$ chmod +x extractor.sh

Bruno@Bruno_PC MINGW64 ~/Downloads/Extracao
$ ./extractor.sh
[+] Página 1 (from=0)...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 23268    0 23268    0     0  56129      0 --:--:-- --:--:-- --:--:-- 56338
[+] Página 2 (from=15)...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
```

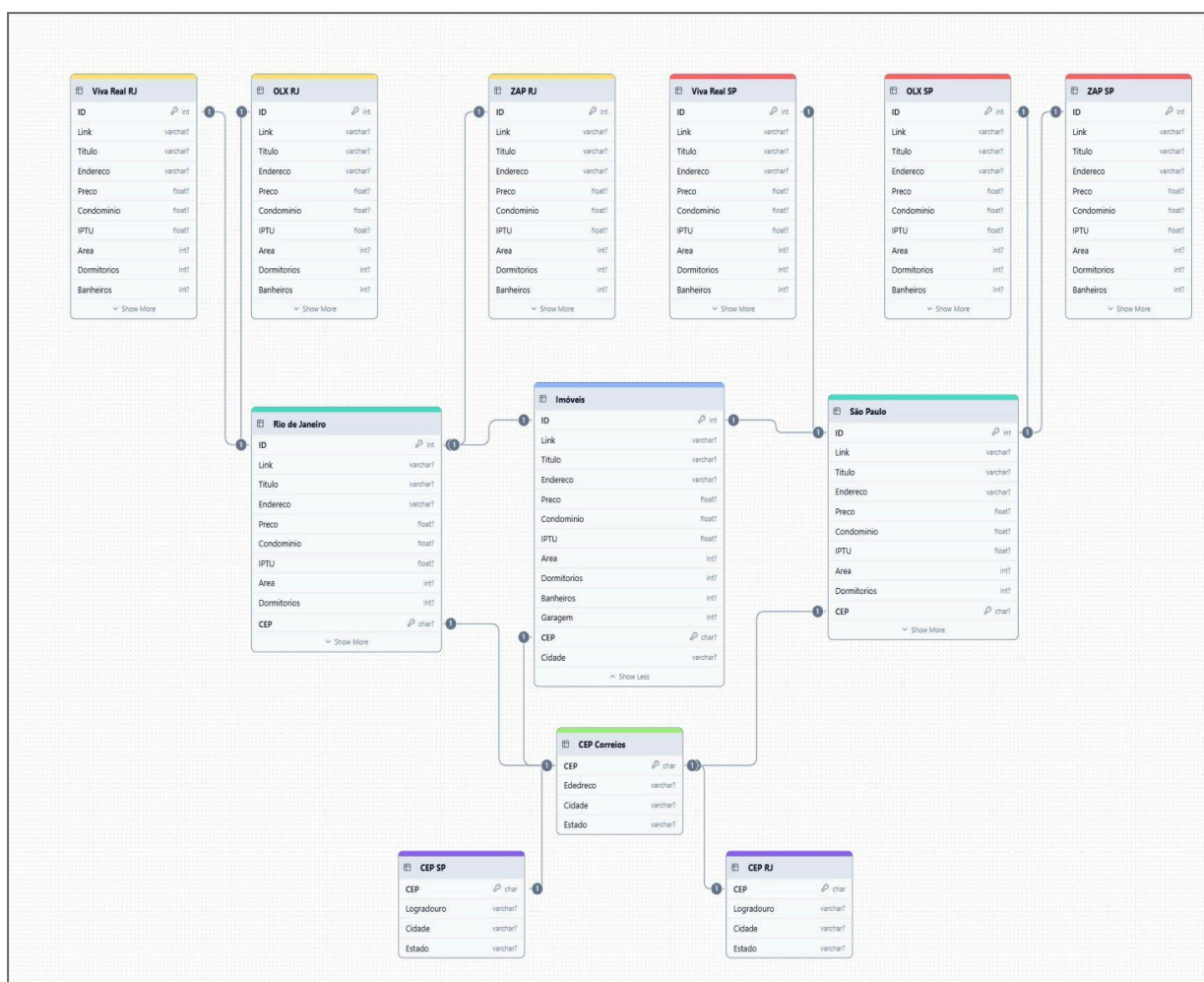
Extração via linha de comando Bash para acessar APIs

## 4. Modelagem dos Dados

### 1. Descrição

Foram extraídas seis bases de dados, três da zona oeste de São Paulo (VivaReal SP, OLX SP e ZAP SP) e três da zona sul do Rio de Janeiro (VivaReal RJ, OLX RJ e ZAP RJ). Após a limpeza e transformação dos dados de cada base, elas foram unificadas em duas grandes tabelas, uma para cada cidade. Após unificar as bases, realizei um processo de enriquecimento das bases, adicionando mais duas tabelas extraídas do site Cep Aberto, uma para cada cidade, contendo cep, endereço, cidade e estado. Estas duas tabelas de cep também foram unificadas em uma tabela contendo todos os ceps das cidades de São Paulo e Rio de Janeiro. O Objetivo de inserir os ceps é poder realizar joins com as tabelas de imóveis e posteriormente usar essas informações para produzir mapas e para consultar outras bases, por exemplo, com informações sócio-econômicas e censitárias.

### 2. Diagrama Modelo Snowflake



## 5. Catálogo de Dados

### 1. Tabelas de Portais de Imóveis (fonte de dados)

Estas tabelas armazenam na camada **BRONZE** os anúncios de imóveis para aluguel nas zonas Sul do Rio de Janeiro e Oeste de São Paulo, listados em diferentes plataformas:

- Viva Real RJ
- Viva Real SP
- OLX RJ
- OLX SP
- ZAP RJ
- ZAP SP

**Metadados:** Formato: CSV ou JSON  
Origem: Web Scraping  
Fonte: Web sites OLX, Viva Real e ZAP Imóveis  
Data: Março 2025 e Abril 2025

#### **Campos Comuns:**

Nome	Descrição	Tipo de Dado	Nullable
ID (PK)	Identificador Único	Varchar	Não
Link	Link para o anúncio	String	Não
Título	Título do anúncio	Varchar	Sim
Endereço	Endereço do imóvel	String	Sim
Preço	Valor do aluguel	Float	Não
Condomínio	Valor do condomínio	Float	Sim
IPTU	Valor do IPTU	Float	Sim
Área	Área do imóvel em M2	Integer	Sim
Dormitórios	Número de dormitórios	Integer	Sim
Banheiros	Número de banheiros	Integer	Sim
Garagem	Número de vagas de garagem	Integer	Sim

## 2. Tabelas de Cidades e Consolidação de Imóveis

As duas primeiras tabelas (Rio de Janeiro e São Paulo) armazenam na camada **SILVER** todos os anúncios coletados por cidade, limpos, tratados e uniformizados para se adequarem a um esquema único.

Já a tabela Imóveis armazena na camada **GOLD** a consolidação de todos os anúncios coletados em ambas as cidades. A partir desta tabela serão produzidas as agregações e análises que deverão responder às perguntas do projeto.

**Metadados:** Formato: Bancos de Dados no Databricks  
Origem: Camada Bronze / Silver  
Fonte: Tratamentos e Limpeza feitos na camada Silver /  
Agregações e análises na camada Gold  
Data: Março 2025 e Abril 2025

### Campos Comuns:

Nome	Descrição	Tipo de Dado	Nullable
ID (PK)	Identificador Único	Varchar	Não
Link	Link para o anúncio	String	Sim
Título	Título do anúncio	Varchar	Sim
Endereço	Endereço do imóvel	String	Sim
Preço	Valor do aluguel	Float	Não
Condomínio	Valor do condomínio	Float	Sim
IPTU	Valor do IPTU	Float	Sim
Área	Área do imóvel em M2	Integer	Sim
Dormitórios	Número de dormitórios	Integer	Sim
Banheiros	Número de banheiros	Integer	Sim
Garagem	Número de vagas de garagem	Integer	Sim
CEP (FK)	Código Postal Correios	Char	Não
Bairro	Nome do Bairro	Varchar	Sim



- 3.
4. Tabelas de CEP (Dados Geográficos)

As tabelas CEP RJ e CEP SP são segmentadas por cidade, contém todos os CEPs de cada cidade e estão armazenadas na camada **BRONZE**.

A Tabela CEP Correios possui os CEPs das duas cidades consolidados, limpos e tratados (preferencialmente listar apenas os CEPs utilizados na tabela imóveis) e está armazenada na camada **SILVER**.

**Metadados:** Formato: CSV  
Origem: Cep Aberto  
Fonte: [www.cepaberto.com](http://www.cepaberto.com)  
Data: Abril 2025

#### **Campos Comuns:**

Nome	Descrição	Tipo de Dado	Nullable
CEP (PK)	Código Postal Correios	Char	Não
Logradouro	Nome do logradouro, endereço	String	Não
Número	Número ou lado da rua	String	Sim
Bairro	Nome do Bairro	Varchar	Sim
Mun_ibge	Código do Município IBGE	Integer	Não
UF_ibge	Código do Estado	Integer	Não

#### 5. Tabelas GeoBase

As tabelas GeoBase são inseridas na camada **BRONZE** e contém os dados de CEP, Latitude, Longitude e o código do setor censitário do IBGE.

Pode ser usada para futuros cruzamentos de dados com fontes de pesquisas específicas de microdados do IBGE, por exemplo, população, renda, escolaridade, etc.

Não cheguei a utilizar esta tabela no projeto final, mas tinha a intenção de utilizar estes microdados do IBGE para produzir mapas enriquecidos na etapa de análise dos dados.

**Metadados:** Formato: CSV  
Origem: IBGE  
Fonte: [www.ibge.gov.br/estatisticas/downloads-estatisticas](http://www.ibge.gov.br/estatisticas/downloads-estatisticas)  
Data: Abril 2025

## Campos Comuns:

Nome	Descrição	Tipo de Dado	Nullable
Postcode (cep) (PK)	Código Postal Correios	Char	Não
Lon	Longitude	String	Não
Lat	Latitude	String	Sim
cd_geocodi	Código setor censitário IBGE	String	Sim

## 6. ETL Databricks

O trabalho de ingestão, tratamento e análise dos dados no Databricks Community foi dividido em etapas, tendo um Notebook em Pyspark para cada etapa. Também foi necessário criar um notebook para inicializar todos os databases quando o cluster se encerrava e era preciso reconectar todas as tabelas delta já criadas e um notebook para exportar arquivos finais.

No notebook “MVP\_Camada\_Bronze”, os arquivos de cada site e das duas cidades pesquisadas foram carregados no DBFS e criadas as databases e tabelas cruas, num total de seis tabelas.

O notebook “MVP\_Camada\_Silver” contém o grosso do trabalho, nesta camada foram feitas as limpezas, transformações e enriquecimento de cada uma das seis tabelas. O processo consiste em abrir as tabelas cruas da camada bronze, realizar limpeza de dados, transformações de nomes de colunas, tipos de dados, extração de informações de uma coluna para outra, separação de strings, uniformização dos esquemas, etc.

Após este tratamento cada tabela passou por um processo de enriquecimento para adicionar informações de CEP. A princípio, basta realizar um join com as tabelas “cep\_rio” ou “cep\_sp” através do campo Endereço ou Logradouro, mas algumas bases estavam com dados faltantes que não conseguiram ser preenchidos neste primeiro join. Em alguns casos foi preciso criar uma segunda estratégia de join (fallback), baseado no campo Bairro. Nestes casos, o cep atribuído ao imóvel foi aproximado para um cep que está no centro geográfico do bairro.








Depois deste processo de limpeza e enriquecimento, as tabelas foram salvas na camada Silver no formato delta tables. No final deste processo todo da camada Silver, realizei uma união entre as três tabelas de cada cidade, com o objetivo de ficar com uma grande tabela com todos os anúncios e salvei estas tabelas também na camada Silver para serem usadas posteriormente na etapa de análise da camada Gold. Todos estes processos na camada silver foram feitos com Pyspark, preferencialmente. Porém como o objetivo é aprender e

experimental, uma parte foi feita em SQL, apesar de ser menos eficiente na hora de modificar esquemas e sobrescrever tabelas já existentes




























Criei mais dois databases chamados “Bronze\_aux” e “Silver\_aux”, para armazenar e tratar estas tabelas auxiliares de cep e informações geográficas. Estas camadas tam,bém seriam úteis para separar e tratar os pipelines de futuras tabelas que pudessem ser usadas para enriquecer a análise, como por exemplo: bases do ibge ou do ipea com dados sócio-econômicos.

Workspace > Users >

**brunolayus@gmail.com**

Name 	Type	Owner
 1_Inicializador Notebook 2025-04-09 08:47:09	Notebook	Bruno Layus
 MVP_Bases Auxiliares 2025-04-10 07:33:04	Notebook	Bruno Layus
<input type="checkbox"/>  MVP_Camada_Bronze_2025-04-03 12:41:47	Notebook	Bruno Layus
 MVP_Camada_Ouro 2025-04-10 13:41:18	Notebook	Bruno Layus
 MVP_Camada_Silver_2025-03-24 19:21:45	Notebook	Bruno Layus
 MVP_Export_Files	Notebook	Bruno Layus

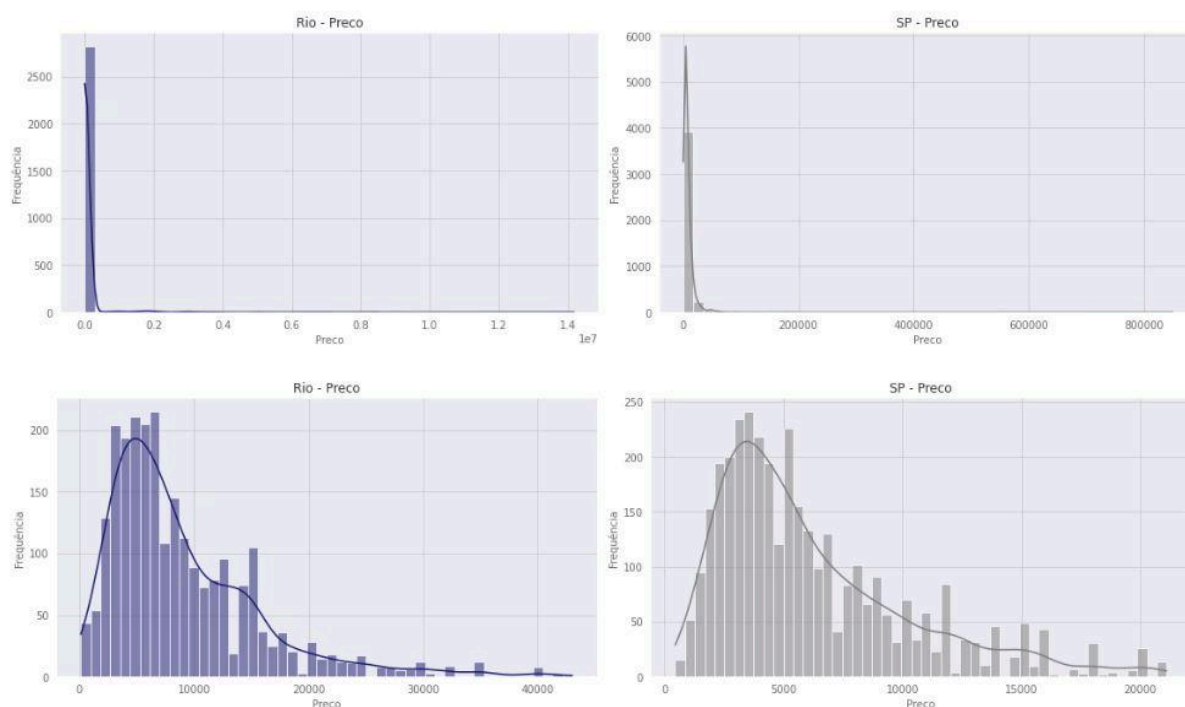
Database Tables DBFS Create Table ✕ 📌

Databases 	Tables
 Filter Databases	 Filter Tables
 bronze_aux	 imoveis_rio 
 bronze_imoveis	 imoveis_sp 
 default	 olx_rio 
 gold_imoveis	 olx_sp 
 silver_aux	 vivareal_rio 
 silver_imoveis	 vivareal_sp 
	 zap_rio 
	 zap_sp 
	 zap_sp_sql 

## 7. Análise dos Dados

No Notebook da Camada Gold desenvolvi as análises estatísticas e gráficos para explorar os dados coletados e responder às perguntas que havia imaginado no início do projeto. Neste momento preferi utilizar pandas e python puro com suas bibliotecas de visualização, principalmente Seaborn e Matplotlib, pois tenho mais familiaridade para executar agregações e operações mais complexas.

Comecei avaliando as distribuições dos dados numéricos como preço, condomínio, lptu e área para identificar padrões e desvios dos dados. Percebi que as distribuições estavam inicialmente muito assimétricas, devido a outliers que atrapalham a análise. Em alguns casos eram imóveis que estavam à venda ou com valores preenchidos errados, nulos, etc. Fiz uma remoção de aproximadamente 5% dos dados mais extremos, conseguindo, desta forma, obter histogramas mais adequados, como no exemplo abaixo:



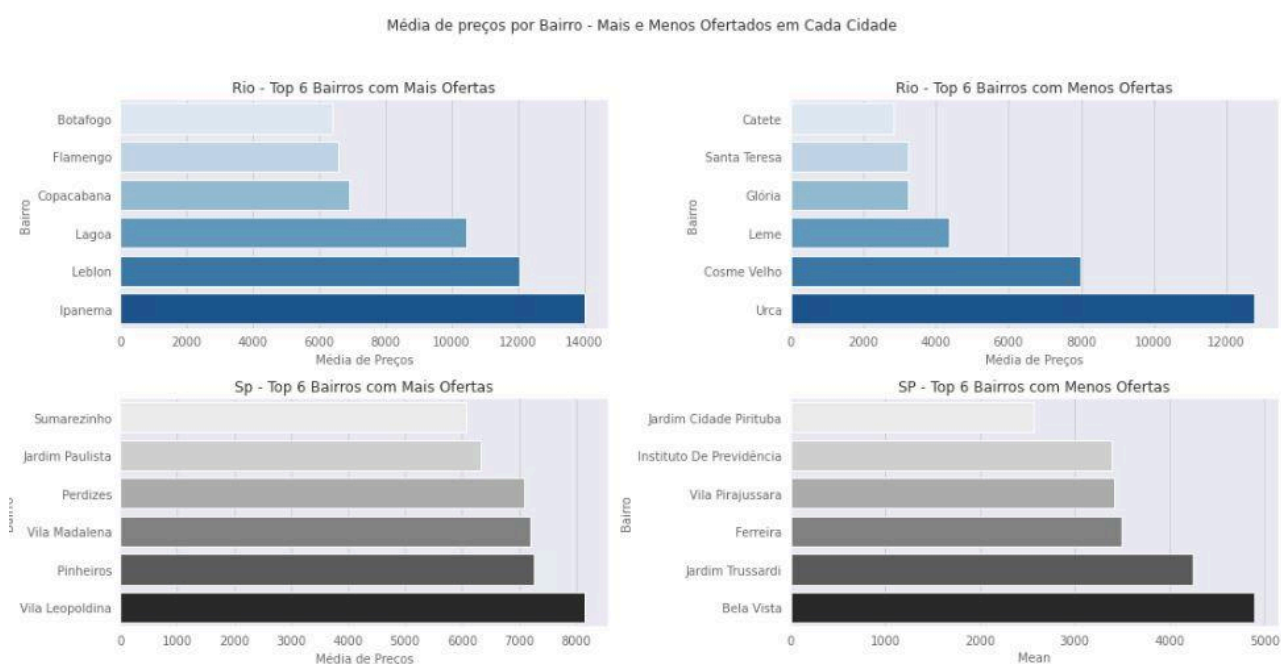
Nos dois gráficos superiores, antes de remover outliers, as distribuições estavam totalmente assimétricas, com uma enorme cauda de poucos dados extremos. Após remover parte dos dados, as distribuições ficaram bem mais reais. Este tipo de distribuição de preços de aluguel é sempre assimétrica, tendo muito mais dados no lado esquerdo da distribuição, onde se concentram as faixas de preço praticados no mercado, com uma pequena parcela de imóveis que realmente são maiores e de alto padrão, custando mais caro do que a maioria.

Depois fiz algumas comparações entre os preços médios de cada cidade, onde fica evidente uma diferença de preço que pode chegar a ser 30% mais caro alugar um imóvel no Rio do que alugar um imóvel similar em São Paulo. Apesar das comparações por média

serem pouco precisas, por experiência e por comprovação dos dados, realmente é mais caro morar no Rio de Janeiro. Outros preços como condomínio e Iptu também são maiores no Rio em média, 31% e 56% maiores, respectivamente

. Apesar disso, uma coisa me intrigou ao analisar os dados: o preço médio por metro quadrado nas duas cidades está quase igual neste conjunto de dados que analisei. Uma possível explicação é que em São Paulo, o mercado de imóveis está saturado de lançamentos residenciais do tipo “Studios”, imóveis pequenos, novos e com preço de mercado acima da média. Esta hipótese precisa ser mais bem elaborada e mais adiante podemos revê-la nos dados.

Depois realizei agrupamentos por bairro, para entender quais lugares são mais valorizados e quais são menos. A dinâmica do mercado imobiliário não obedece à clássica lei da oferta e da demanda. Explico melhor: Bairros com muitas ofertas, tanto em São Paulo, quanto no Rio, como por exemplo Pinheiros e Ipanema, são também os que apresentam valores mais elevados. Além da qualidade dos imóveis nestes bairros serem possivelmente melhores, há uma espécie de efeito manada, um hype que atinge certos bairros da cidade e provoca uma onda de valorização apesar da abundância de oferta.

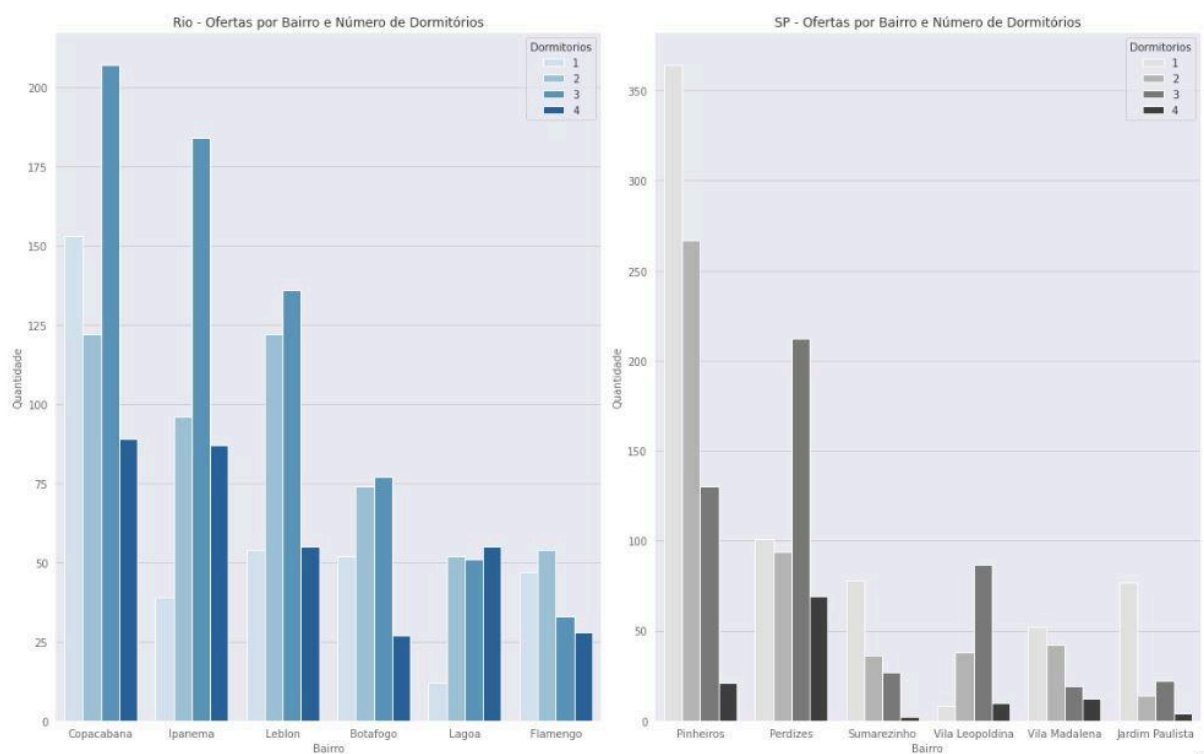
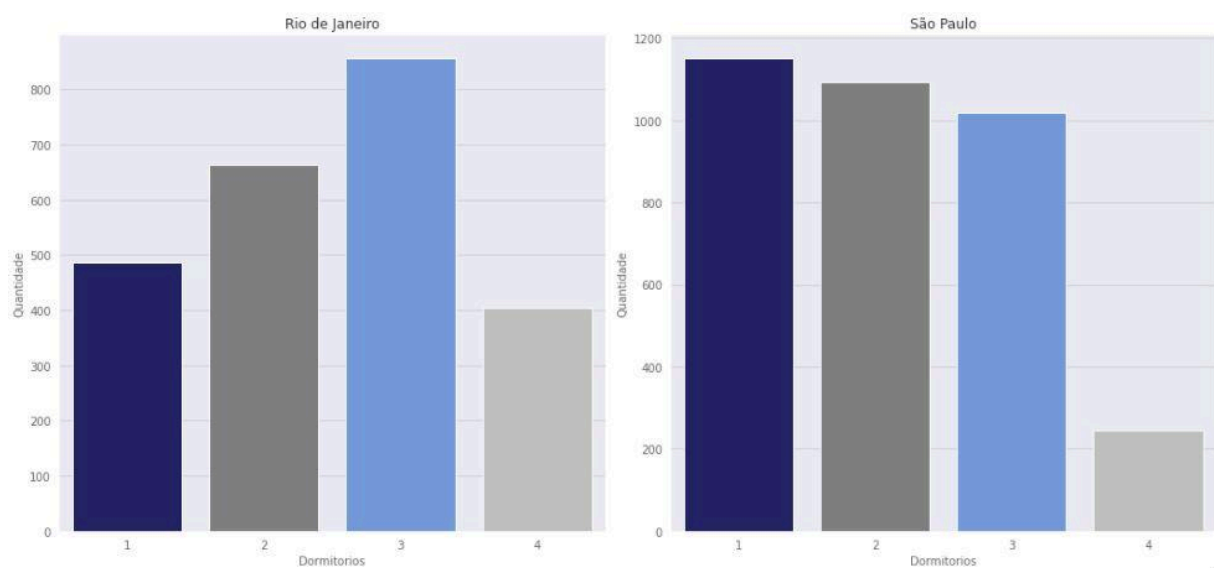


Nos gráficos acima, os bairros foram agrupados por quantidade de ofertas e por média de preços. Os 6 bairros com mais ofertas normalmente coincidem com as maiores médias de preços. Bairros mais periféricos ou menos valorizados também têm poucas ofertas, um efeito de degradação urbana conhecido e talvez creditado à especulação urbana. Ao invés de ocupar os bairros mais degradados e portanto mais baratos, gerando um movimento de valorização e requalificação destas áreas, parece que o poder público e a iniciativa privada preferem uma concentração de riqueza e valor ao invés da distribuição.

Ainda nesta etapa de análise, realizei agregações por tipologia, no caso número de dormitórios, para entender quais tipos de imóvel estão mais disponíveis no mercado e como

o preço pode variar de acordo com esta característica. Faz um pouco parte do senso comum que o tipo de imóvel com mais liquidez, mais saída, é o de 2 quartos ou 3 quartos, imóveis capazes de acomodar uma família em formação ou já formada. Porém neste conjunto de dados São Paulo aparece contra este sentido. A maioria dos imóveis para aluguel em SP são apartamentos de 1 dormitório, mais ou menos empatados com imóveis de 2 e 3 quartos, enquanto no Rio ainda predominam os imóveis de 2 e 3 dormitórios. Isso pode indicar uma virada recente deste (contra) senso comum. Possivelmente há uma relação entre uma mudança do modo de vida urbano com a oferta de imóveis, acontecendo em São Paulo uma mudança mais veloz.

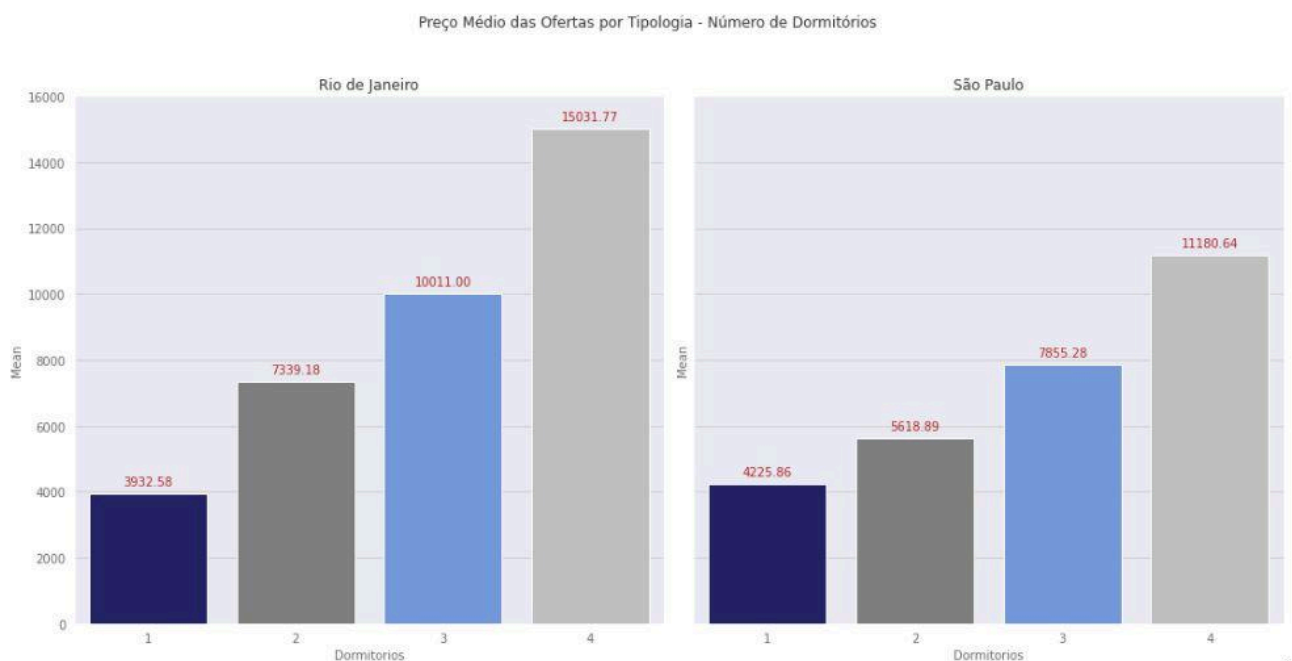
Quantidade de Ofertas por Tipologia - Número de Dormitórios



Neste último gráfico, podemos ver como bairros mais tradicionais, como Copacabana no Rio e Perdizes em São Paulo, ainda possuem a maioria das ofertas em imóveis de 2 e 3 dormitórios. Copacabana apresenta muitas ofertas de apartamentos de 1 dormitório, mas a maioria das ofertas são imóveis de 3 dormitórios, já em Ipanema a maioria dos imóveis são do tipo 3 dormitórios.

O que salta aos olhos é a grande quantidade de apartamentos de 1 dormitório no bairro de Pinheiros, que é quase 3 vezes maior do que a oferta de apartamentos mais “tradicionais” de 3 quartos. Isso ajuda a compreender aquela hipótese do preço médio por m2 ser tão similar nas duas cidades apesar de os preços dos aluguéis serem em média mais altos no Rio. Como em São Paulo há muita oferta de apartamentos menores (com área menor) e estes imóveis são provavelmente mais caros do que a média, isso faz com que o valor por m2 seja quase igual, apesar de os preços médios por tipologia serem quase sempre maiores no Rio.

A única tipologia que está mais cara em São Paulo, como podemos observar no gráfico abaixo, é justamente a de apartamentos de 1 dormitório ou studios.



## 8. Respostas às perguntas

✓ Qual a média de preço de aluguel em cada cidade e porque esta métrica não é boa para compreendermos a dinâmica do mercado imobiliário?

Os preços no Rio podem chegar a ser 30% mais caros do que em São Paulo para apartamentos em regiões e condições similares. A única exceção são os imóveis de um dormitório, que estão ligeiramente mais caros em SP do que no Rio.

✓ Qual a média dos preços em alguns bairros mais importantes de cada cidade?

Bairros mais valorizados no Rio como Ipanema, Leblon e Lagoa possuem média de preços de aluguel bem alta, passando de R\$ 10.000,00. Já em São Paulo, bairros como Pinheiros, Perdizes, Vila Madalena e Jardins possuem quase a mesma média de R\$ 7.000,00. Vila Leopoldina, (bairro de desenvolvimento relativamente recente) está com a média mais alta, em torno de R\$ 8.000,00.

✓ Qual o valor médio cobrado de aluguel por metro quadrado em cada cidade e por bairros?

Esta foi a constatação mais intrigante desta análise de dados: a média de preços por metro quadrado nas duas cidades está quase a mesma, com apenas 5% de diferença a mais na cidade do Rio. Busquei confirmar a hipótese de que há um excesso de apartamentos sobrevalorizados de um dormitório no mercado paulista, com pouca área e em grande quantidade, o que altera esta média de preço, tornando-a uma métrica não muito adequada.

✓ Quais os tipos de imóveis mais ofertados? Estúdios? Apartamentos de 2 ou 3 dormitórios?

Bairros mais tradicionais continuam tendo mais ofertas de apartamentos de 2 e 3 dormitórios, enquanto bairros que estão passando por uma transformação recente possuem mais ofertas de apartamentos de 1 dormitório.

✓ Quais os intervalos de valor dos apartamentos em São Paulo e Rio? Quais os valores máximos e mínimos?

Em São Paulo, em bairros mais periféricos, mas ainda na Zona Oeste, é possível encontrar ofertas a partir de R\$ 2.500,00, já em bairros mais valorizados pode girar em torno de R\$ 7.000,00.

No Rio também é possível encontrar imóveis a partir de R\$ 1.500,00 nos bairros mais perto do Centro, tipo Glória, Catete e Sta. Teresa. Já em Ipanema pode chegar a R\$ 14.000,00 em média.

✓ Quais as distribuições dos preços estratificados por cidade, bairros e tipologias?

Os gráficos apresentados ao longo da sessão anterior respondem a esta pergunta.



## 9. Autoavaliação

O Objetivo deste projeto foi trabalhar com uma pipeline de dados, partindo desde a extração dos dados, a consolidação deles em uma plataforma na nuvem, todo processo de ingestão, tratamento, limpeza, enriquecimento, modelagem, armazenamento e posterior uso destes dados já persistidos para análise.

A plataforma escolhida foi o Databricks Community, uma plataforma muito interessante, capaz de armazenar dados provenientes de origens e formatos diferentes, capaz de executar notebooks online em tempo real com possibilidade de usar diversas linguagens de programação separadamente ou em conjunto, passando por SQL, Python e a poderosa Pyspark, que trabalha com processamento distribuído, muito veloz e adequado para grandes volumes de dados.

Como este projeto é, além de tudo, um espaço para experimentação e aprendizado, acho que realizei em profundidade este objetivo. Fiz a extração de dados de diferentes sites de anúncios de imóveis, utilizando várias ferramentas de web scraping e de acesso por APIS, Depois utilizei a plataforma escolhida para realizar todos os tratamentos e armazenamento destes dados coletados, obedecendo aos princípios de uma arquitetura de dados tipo Medalhão, com camadas Bronze, Silver e Gold, cada uma com suas características.

Em algumas bases optei por tentar utilizar SQL para fazer parte do processo de tratamento, mas acabei preferindo o Pyspark por ser mais eficiente em salvar e sobrescrever dataframes e tabelas delta. Dividi o processo em alguns notebooks, com muitas células de código cada. Depois utilizei estes dados para realizar uma análise exploratória que me intrigou em alguns aspectos, trazendo à tona alguns insights que eu não havia imaginado antes de concluir a análise.

Poderia ter melhorado vários aspectos e isso fica evidente ao olhar para o processo, mas nesta reflexão e resiliência reside o aprendizado consistente. Por exemplo: perdi muitos dias tentando realizar a extração dos dados dos sites por vários caminhos diferentes, na verdade em pouco tempo já havia conseguido extrair parte importante dos dados, mas não estava satisfeito com a forma e com o resultado. Então segui tentando métodos diferentes até encontrar uma forma que julguei eficaz para extrair dados, atividade que considero essencial para um cientista de dados. Poderia ter usado uma base pronta do Kaggle, já organizada e ter realizado um trabalho mais burocrático, que teria me demandado menos tempo, mas isso não reflete a realidade da profissão e nem teria sido tão desafiador.

Certamente há várias passagens do processo de tratamento e de enriquecimento que poderiam ter sido transformadas em funções mais eficientes e menos repetitivas, tornando o processo mais automatizado, mas estava trabalhando com uma variedade de formatos e tabelas e cada uma delas exigiu uma abordagem um pouco diferente das demais.

Na etapa de análise dos dados, gostaria de ter usado bases com geoinformação para realizar alguns mapas e também queria utilizar bases com dados sócio-econômicos para cruzar coisas como nível de renda e escolaridade, demografia, etc. De modo geral fiquei satisfeito com a qualidade da análise, mas há muito espaço para melhoria. O que é ótimo!

## 10. Conclusão

O mercado imobiliário possui uma dinâmica própria que fica mais clara ao olhar para este conjunto de dados das duas principais cidades do Brasil. São cidades que eu conheço bem, então pra mim fica muito mais assertivo analisar esta dinâmica e poder corroborar hipóteses através dos dados. Na verdade, ao olhar para os dados, descobri um monte de coisas que não sabia de forma clara.

Pude experimentar com esta excelente ferramenta que é o Databricks, que, apesar das limitações da versão Community, se mostrou muito eficaz, versátil e rápida. Neste atual estágio de desenvolvimento da área de dados, onde há muita quantidade e velocidade na aquisição e necessidade de uso dos dados de forma organizada e direta, uma plataforma como esta, capaz de rodar códigos com agilidade, online e em diferentes linguagens é essencial para integrar fluxos e pipelines de dados provenientes de várias fontes e em vários formatos. Juntar tudo em uma só ferramenta é condição sine qua non para conseguir processar e usar adequadamente os dados.