DATABRICKS

Grimaldo Oliveira

O que é o DATABRICS

1. Uma poderosa plataforma de colaboração entre os profissionais da área de dados. É uma plataforma fácil de usar para aqueles que desejam executar consultas em seu Data Lake.
2. Podem ser criados vários tipos de visualização para explorar resultados de consultas de diferentes perspectivas, podendo construir e compartilhar dashboards.
3. Você pode ser administrador da plataforma, analista de dados, cientista de dados e engenheiro de dados com diversas funcionalidade.

Grimaldo Oliveira – grimaldo_lopes@hotmail.com

Sites Importantes

DataBricks

https://databricks.com/

Documentação

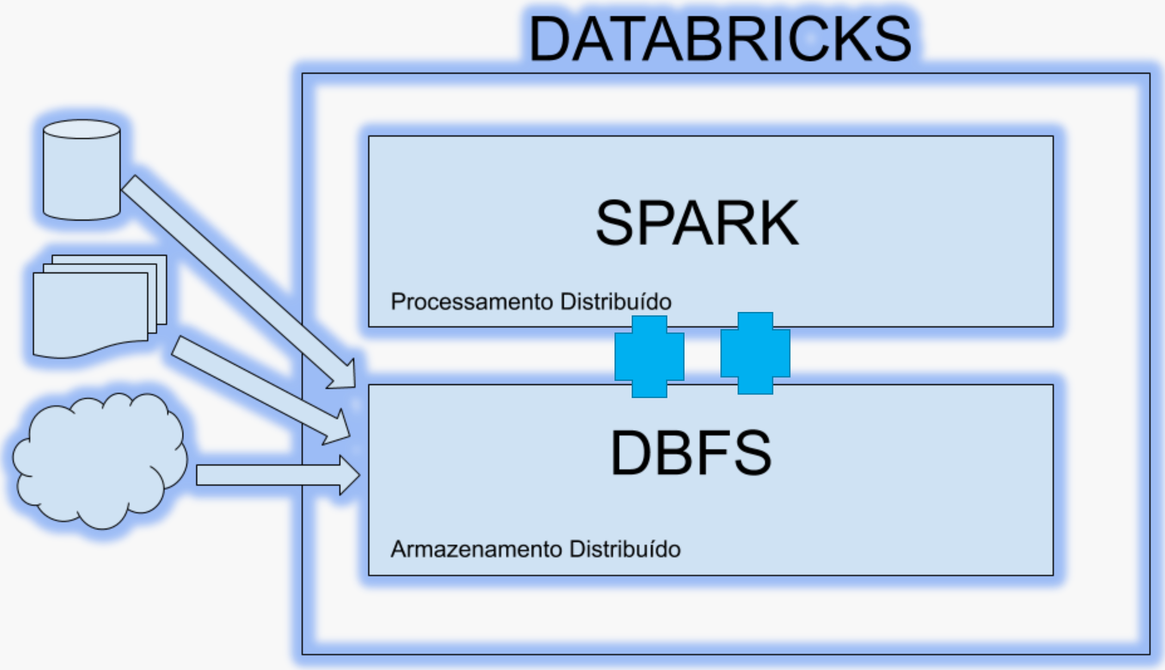
https://docs.databricks.com/

Databricks Community

https://community.cloud.databricks.com/login.html

Ecossistema

Como funciona a leitura dos dados para serem carregados no Databricks Store e gerenciado pelo SPARK.



Databricks

Uma plataforma aberta para armazenar e gerenciar todos os seus dados para que você possa utilizar em seus projetos com dados.

Databricks

Uma plataforma aberta destinada para armazenar e gerenciar todos os seus dados para todas as suas cargas de dados.

É dividida em :

NOTEBOOKS: Permite que os analistas e cientistas de dados construam seus scripts nas linguagens Python, SQL, R, Scala, fazendo consulta aos dados.

ANALYTICS: Use a poderosa interface SQL Analytics para consultar e visualizar dados e preparar dashboards.

DELTA LAKE: Permite que sejam combinados os diversos tipos de dados inseridos no Databricks, pra que ajustes, tratamentos nos dados possam ser executados.

MACHINE LEARNING: É possível desenvolver modelos matemáticos e estatísticos para a geração de informação para seus negócios.

SPARK

Faz todo o gerenciamento e distribuição do processamento executado no Databricks.

Apache Spark

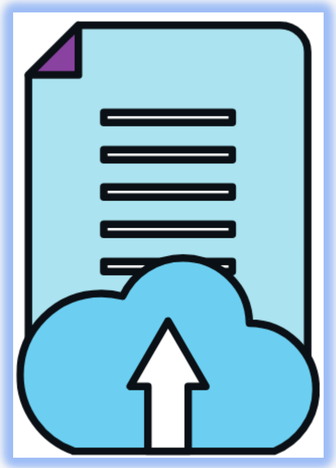
Apache Spark é o principal **mecanismo** de análise unificado para Big Data e aprendizado de máquina que existe no mundo, sendo utilizado pelas grandes corporações. Explorando nas suas execuções o uso de memória e outras otimizações. Anteriormente as empresas utilizavam o Hadoop.

DBFS

Onde os dados são armazenados e gerenciados.

DBFS

Databricks File System (DBFS) é um sistema de arquivos distribuído montado em um espaço de trabalho Databricks e disponível em clusters Databricks.

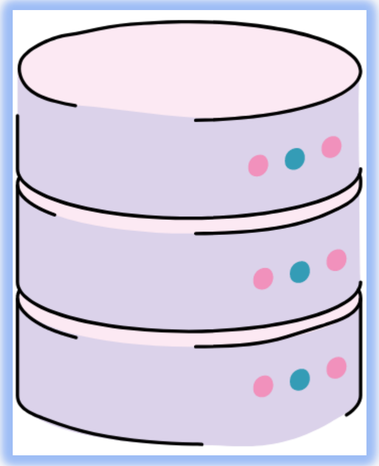


Cluster no Databricks

É onde os recursos para operacionalização são criados. Precisa criar um Cluster para que o Databricks seja operacional.

Cluster

É um conjunto de recursos e configurações em que você cria os seus projetos, dentro dos chamados notebooks, é possível dentro de um cluster executa cargas de trabalho e analisar seus dados.

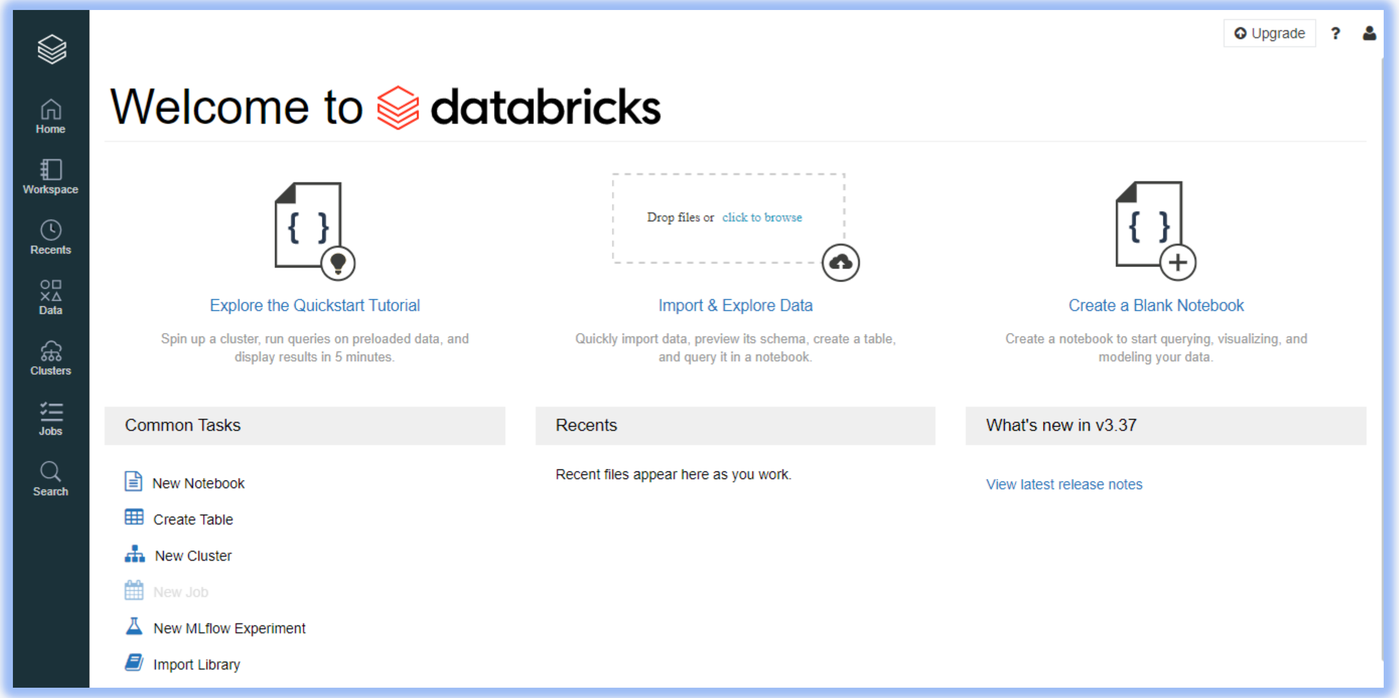


Workspace

É local de trabalho dentro do Databricks.

Workspace

Um espaço de trabalho Databricks é um ambiente para acessar todos os seus ativos Databricks. O espaço de trabalho organiza objetos, e fornece acesso a dados e recursos computacionais, como o clusters.

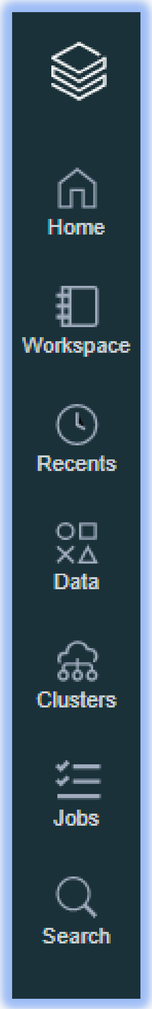


Workspace

É local de trabalho dentro do Databricks.

Workspace

Um espaço de trabalho Databricks é um ambiente para acessar todos os seus ativos Databricks. O espaço de trabalho organiza objetos, e fornece acesso a dados e recursos computacionais, como o clusters.



Retorna a tela de trabalho principal.

Exibe os usuários que estão acessando e permite criar, copiar, deletar, dentre outras tarefas com o notebook.

Exibe os usuários que estão acessando e permite criar, copiar, deletar, dentre outras tarefas com o notebook.

Exibe os notebooks recentes criados.

Permite a criação de tabelas, upload de arquivos, conexões com aws, cassandra, elasticsearch e integrações com ferramentas de dados.

Gerenciamento dos cluster, local para definição e configuração dos clusters.

Local de agendamento dos notebooks.

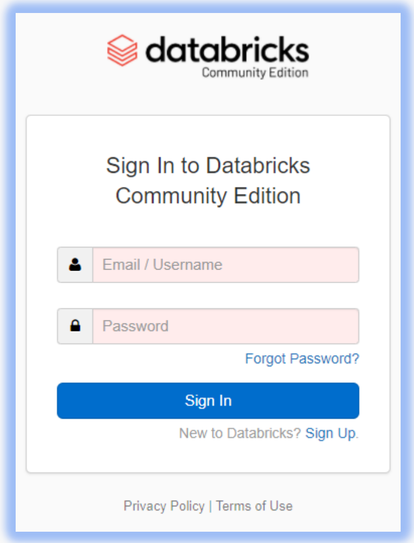
Realiza a busca na área de workspace.

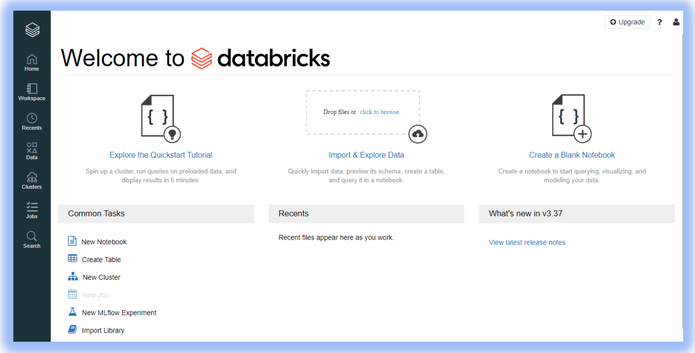
Criar sua conta

Vamos trabalhar na versão Community gratuita.

Criando sua conta na Community

Você deverá criar um acesso gratuito na Community





- Os usuários terão acesso a clusters de 15GB, um gerenciador de clusters e o ambiente de notebook para protótipos de aplicações simples.

- O cluster com os dados fica disponível por 2 horas.

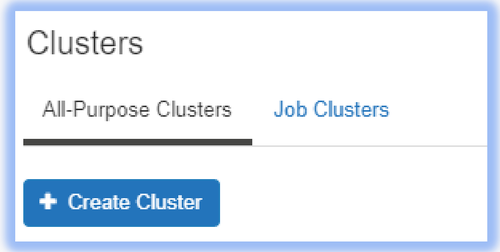
- Limitado a 3 usuários colaborativos.

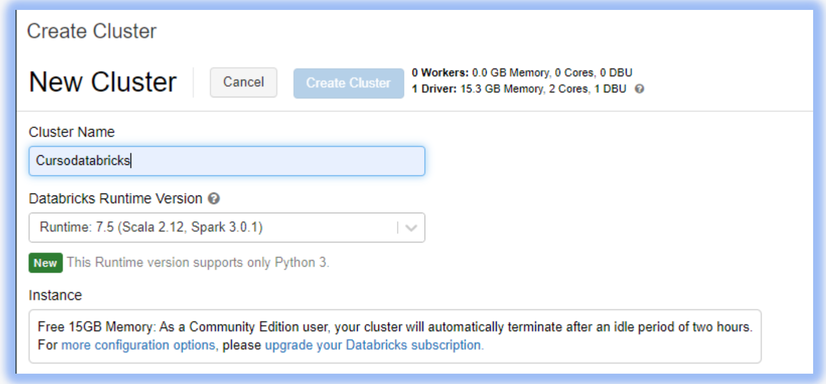
Começando a trabalhar

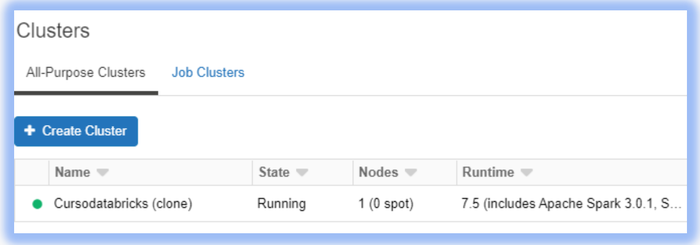
Vamos criar o cluster para iniciarmos nosso trabalho.

Criando o cluster

Primeiro precisaremos criar uma área para carregar os dados e gerar os nossos notebooks para análise.





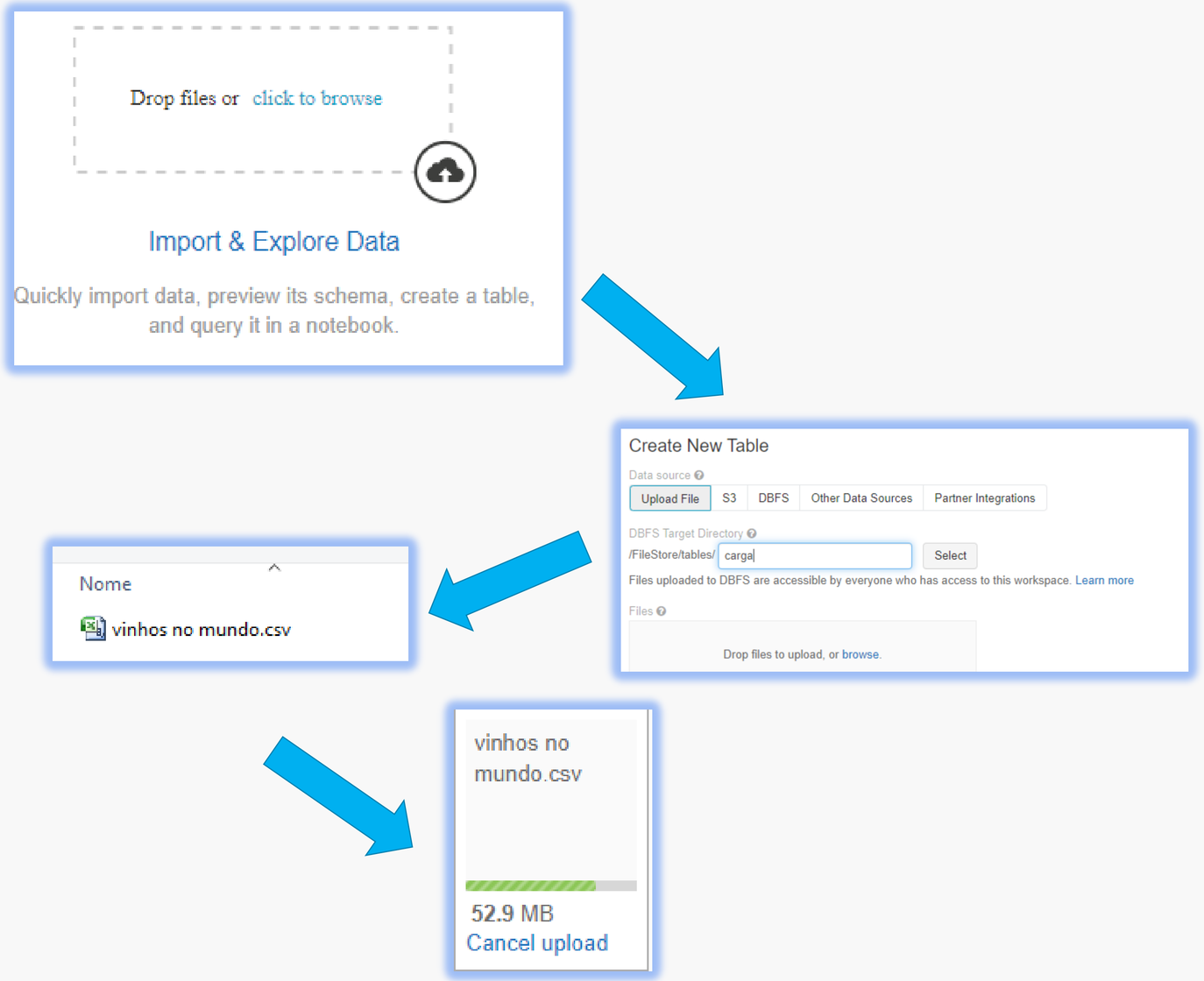


Começando a trabalhar

Vamos carregar os primeiros dados.

Carregando dados

Vamos carregar 129 mil registros em csv do arquivo **vinhos no mundo.csv**

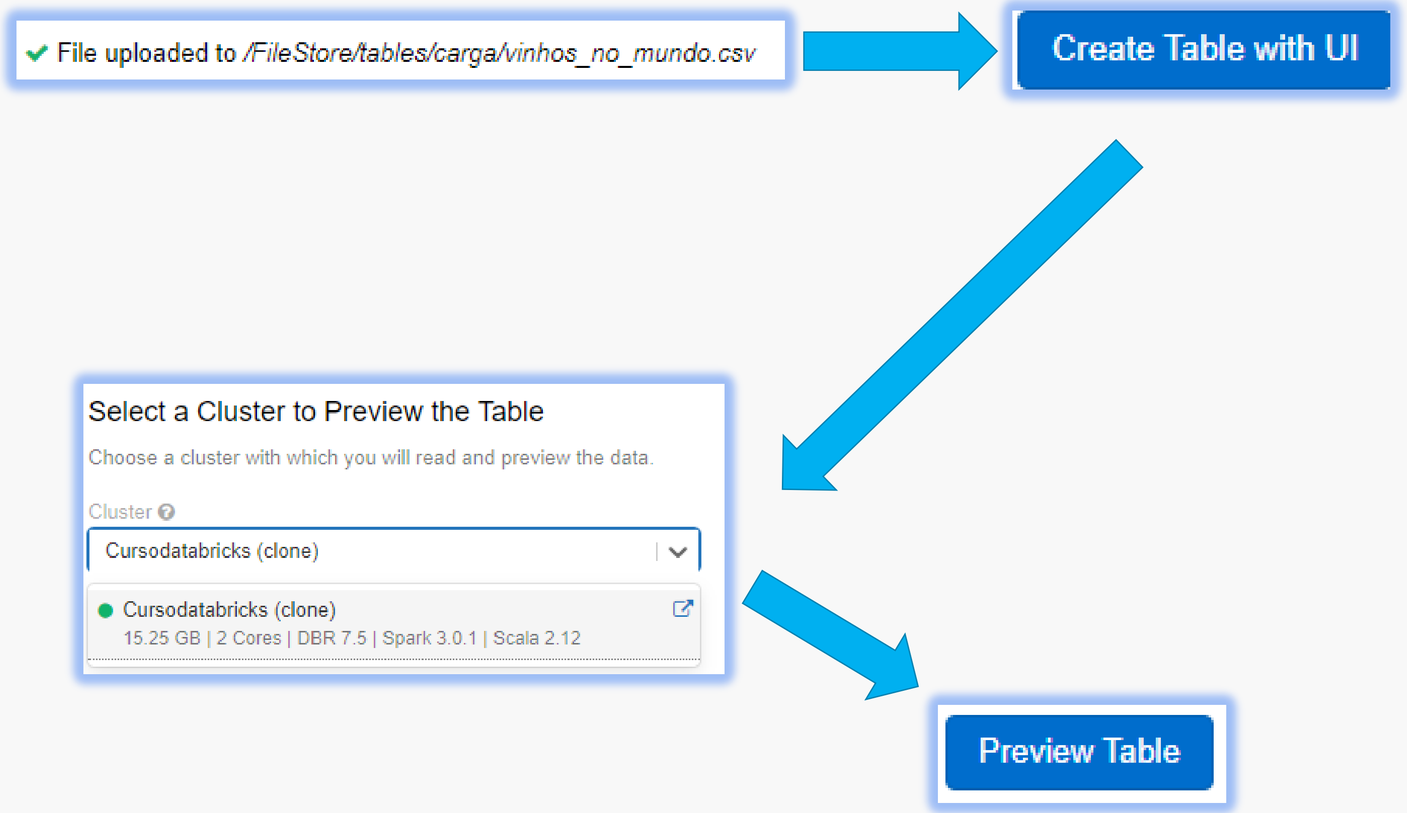


Começando a trabalhar

Vamos carregar os primeiros dados.

Carregando dados

Vamos carregar 129 mil registros em csv do arquivo **vinhos no mundo.csv**

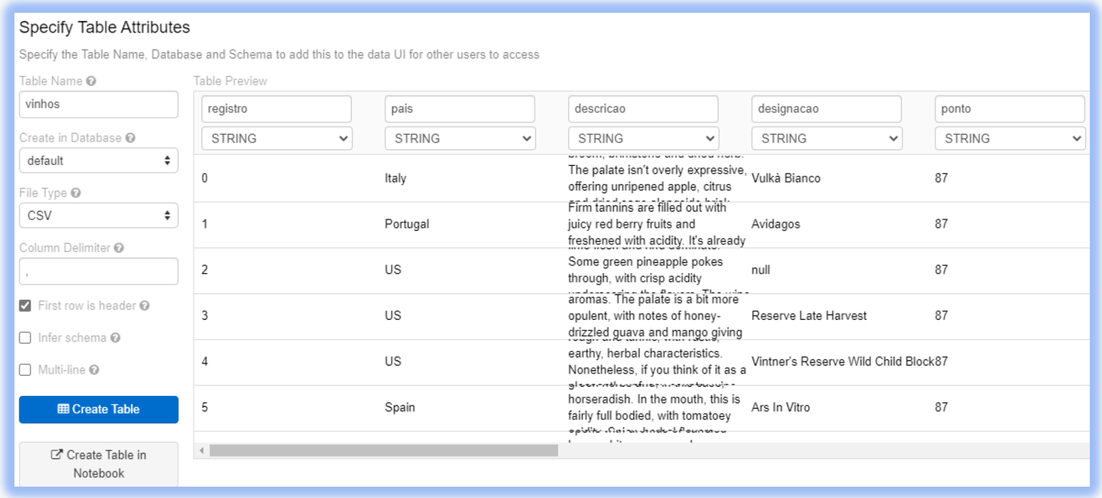


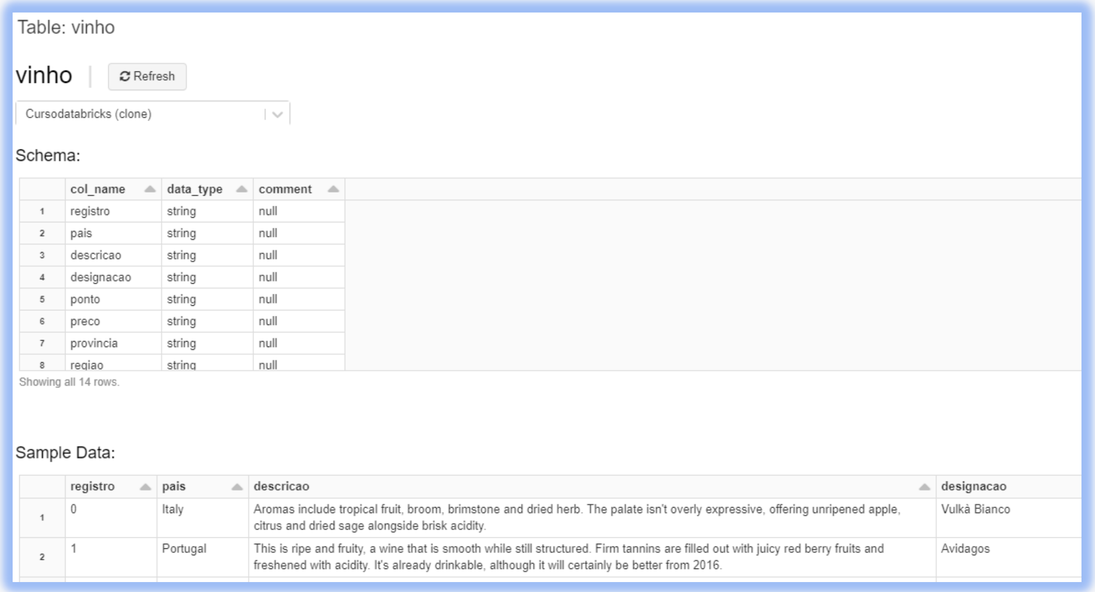
Começando a trabalhar

Vamos carregar os primeiros dados.

Carregando dados

Vamos carregar 129 mil registros em csv do arquivo **vinhos no mundo.csv.** Criando a **tabela vinho.**



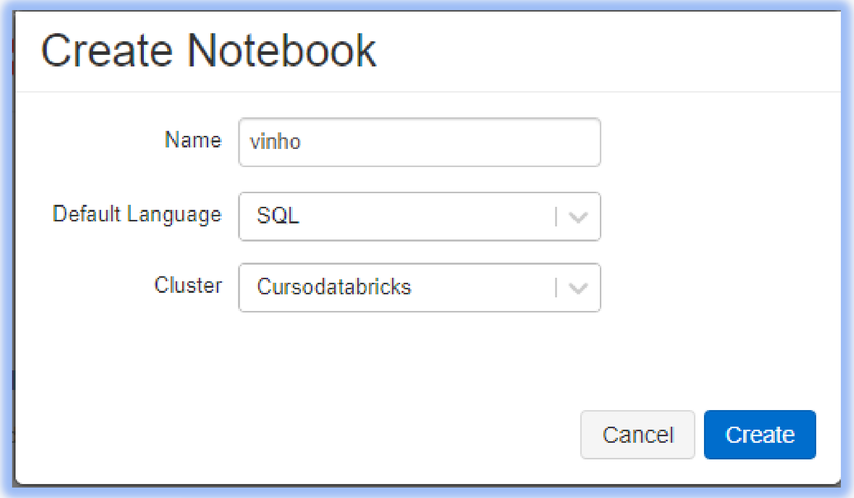


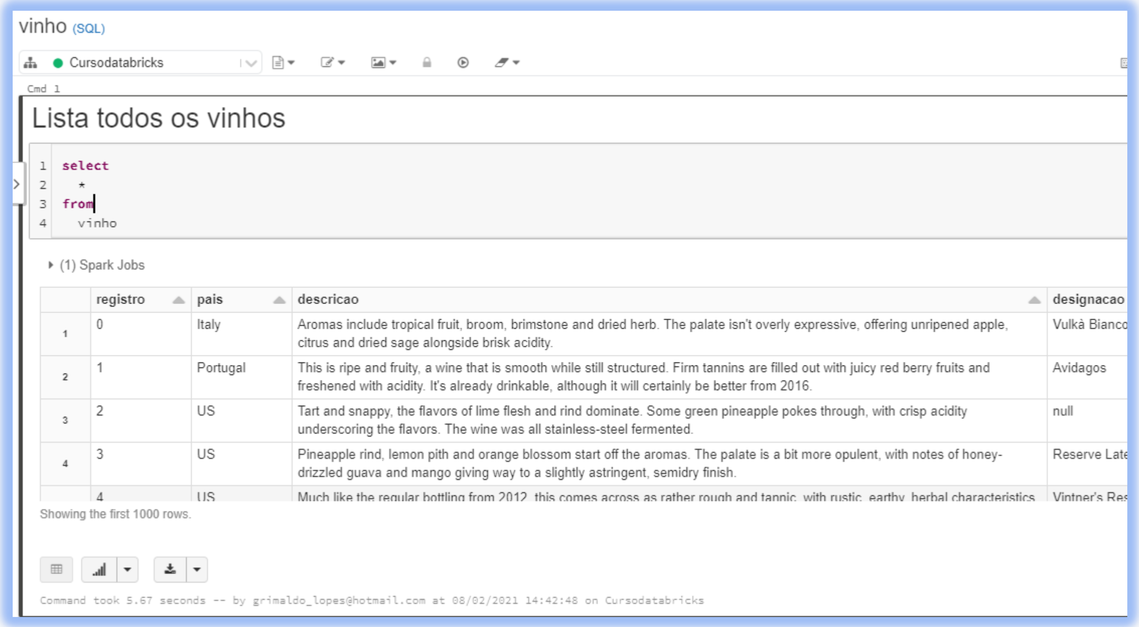
Começando criar um notebook de trabalho

Local onde analisaremos os dados.

Criando um notebook

Local que leremos os dados e faremos análises nos dados, podem serem gerados gráficos. Os notebook podem ser em R, Python,Scala e SQL.



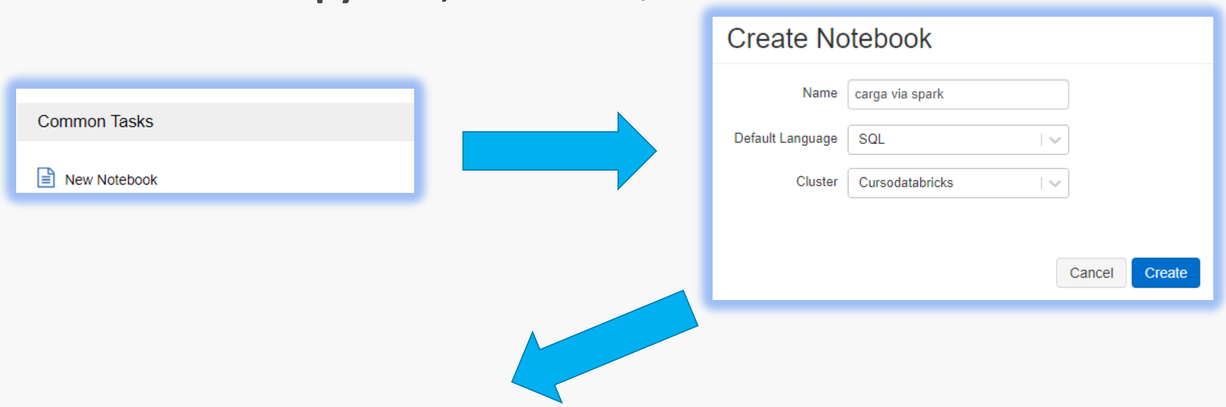


Começando a trabalhar

Vamos carregar os dados de uma forma diferente.

Carregando dados

Vamos carregar + 10 mil registros em csv do arquivo **clientes cartao.csv.** Só que guardando no dataframe do spark**. Diretamente nos notebooks, trabalhando com python, Scala e SQL.**



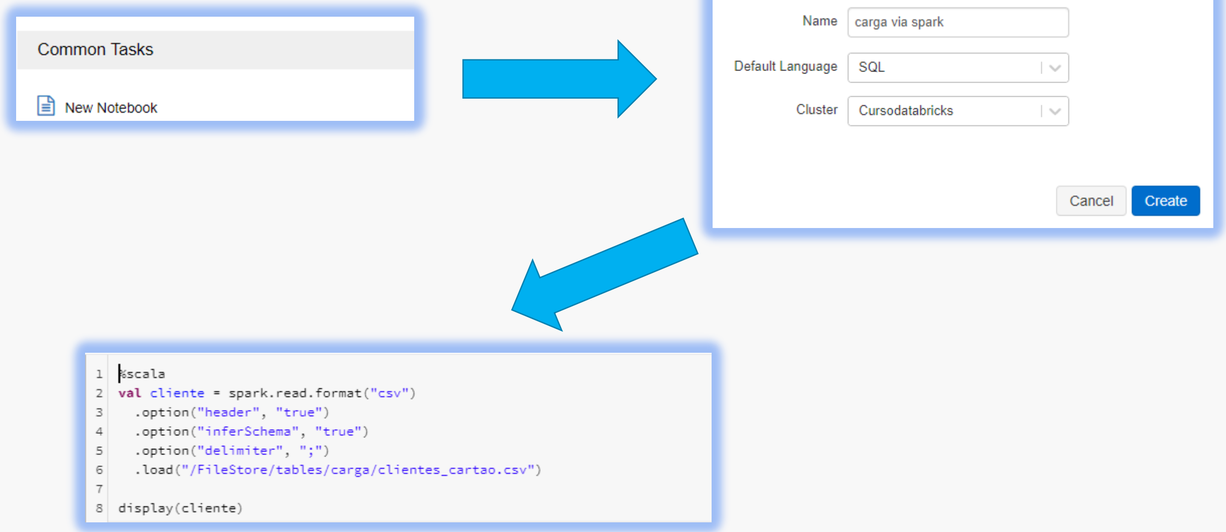

```
%python

clientes = spark.read.format('csv').options(header='true',
inferSchema='true',
delimiter=';').load('/FileStore/tables/carga/clientes_cartao.csv'
)

display(clientes)
```

Começando a trabalhar

Vamos carregar os dados de uma forma diferente.

Carregando dados

Vamos carregar + 10 mil registros em csv do arquivo **clientes cartao.csv.** Só que carregando direto do spark**. Diretamente nos notebooks, trabalhando com python,Scala e SQL.**



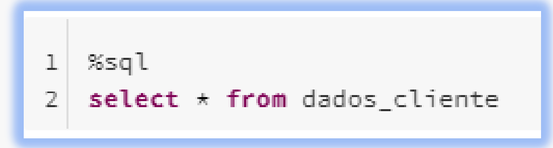

```
%scala
val cliente = spark.read.format("csv")
  .option("header", "true")
  .option("inferSchema", "true")
  .option("delimiter", ";")
  .load("/FileStore/tables/carga/clientes_cartao.csv")

display(cliente)
```

Começando a trabalhar

Vamos carregar os dados de uma forma diferente.

Carregando dados

Vamos carregar + 10 mil registros em csv do arquivo **clientes cartao.csv.** Só que carregando direto do spark**. Diretamente nos notebooks, trabalhando com python, Scala e SQL.**



%scala
cliente.createOrReplaceTempView("dados_cliente“)

%sql
select * from dados_cliente

Começando a trabalhar

Preparando o primeiro gráfico.

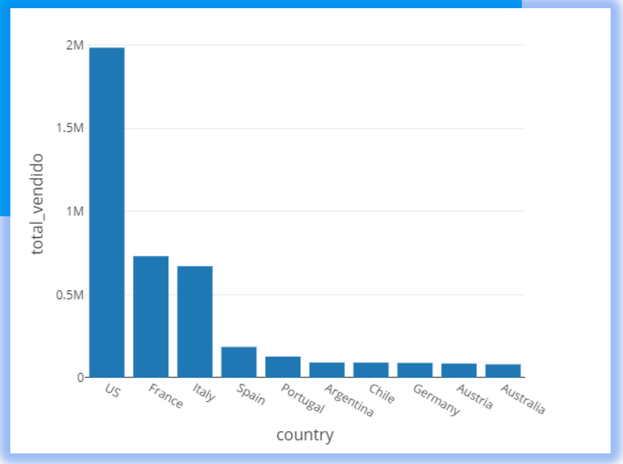
Primeiro gráfico

Vamos executar uma query cujo os resultados vamos representar graficamente.

```
%sql
select country, sum(price) as total_vendido from dados_vinho
where price > 0
group by country
order by total_vendido desc
limit 10
```

	country	total_vendido
1	US	1984106
2	France	731224
3	Italy	670882
4	Spain	185520
5	Portugal	127814
6	Argentina	92060
7	Chile	91793
8	Germany	89586

```
%sql
select pais, sum(preco) as total_vendido from vinho
where preco > 0
group by pais
order by total_vendido desc
limit 10
```



Começando a trabalhar

Preparando o segundo gráfico.

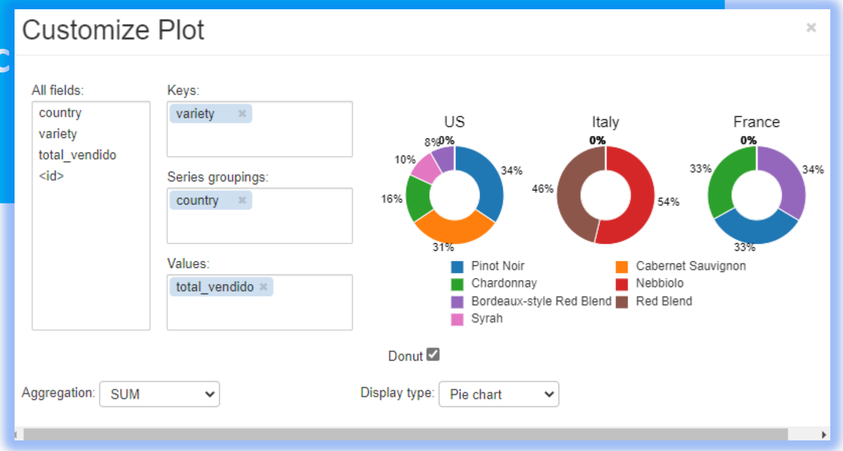
Segundo gráfico

Vamos executar uma query cujo os resultados vamos representar graficamente.

```
%sql
select country, sum(price) as total_vendido from dados_vinho
where price > 0
group by country
order by total_vendido desc
limit 10
```

	country	variety	total_vendido
1	US	Pinot Noir	439530
2	US	Cabernet Sauvignon	398345
3	US	Chardonnay	207482
4	Italy	Nebbiolo	150432
5	France	Bordeaux-style Red Blend	129764
6	Italy	Red Blend	129494
7	France	Pinot Noir	128934
8	France	Chardonnay	127724
9	US	Syrah	125775
10	US	Bordeaux-style Red Blend	105443

```
%sql
select pais, variante, sum(preco) as total_vendido from vinho
where preco > 0
group by pais,variante
order by total_vendido desc
limit 10
```



Começando a trabalhar

Preparando o terceiro gráfico.

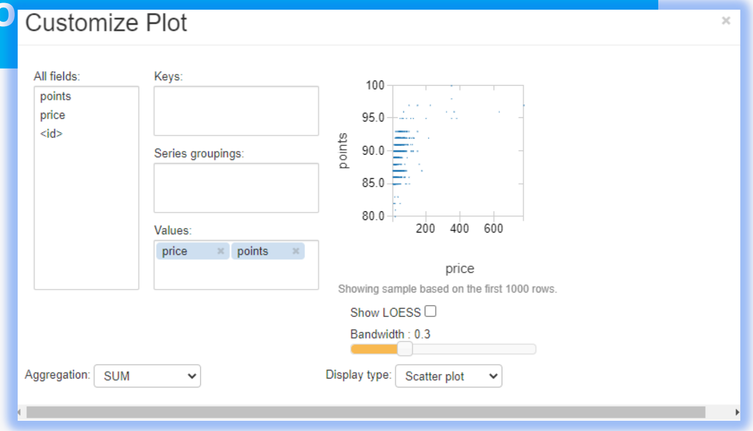
Terceiro gráfico

Vamos executar uma query cujo os resultados vamos representar graficamente.

```
%sql
select points, price from dados_vinho

```

	points	price
1	87	null
2	87	15.0
3	87	14.0
4	87	13.0
5	87	65.0
6	87	15.0
7	87	16.0
8	87	24.0
9	87	12.0
10	87	27.0
11	87	19.0

%sql
select pontos, preco from vinho

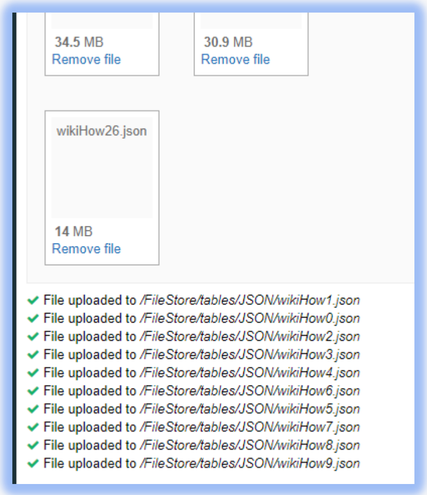


Carregando vários arquivos JSON

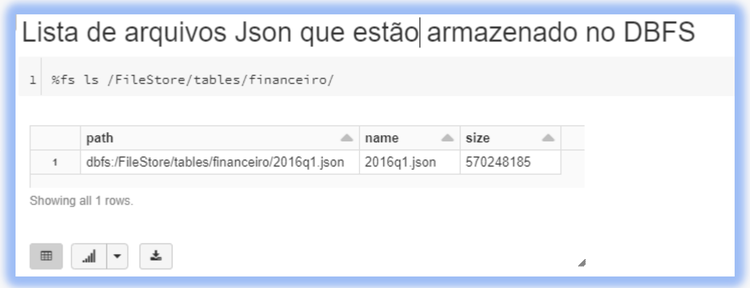
Veja como criar um stream dos arquivos json.

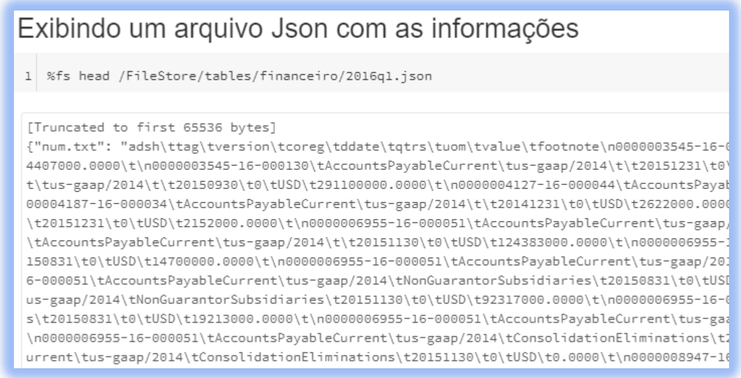
Veremos agora uma demonstração da potencialidade do spark na execução de scripts em Pyspark (python no spark) com a junção de arquivos JSON.

Carregando JSON



Executa Script





VISUALIZAÇÃO

1. É um formato de arquivo de coluna que fornece otimizações para acelerar consultas
2. Tradicionalmente o armazenamento de dados em banco de dados é em linhas, no Parquet armazena os dados de forma colunar
3. Parquet foi criado para suportar compressão
4. O arquivo é dividido em dados e metadados
5. Está presente no Spark, Hive, Impala no ecossistema Hadoop em geral

Pontos Principais Arquivo Parquet

Formato	Espaço utilizado	Tempo de execução	Escaneado
CSV	2 TB	472 seg	2.3 TB
Parquet	260 GB	13,56 seg	5.02 GB

Criando arquivos parquet

Veja como criar um arquivo parquet e realizar sua leitura.

Veremos agora demonstrar a potencialidade do spark na execução de scripts para gravação e leitura de arquivos parquet.

DATAFRAME

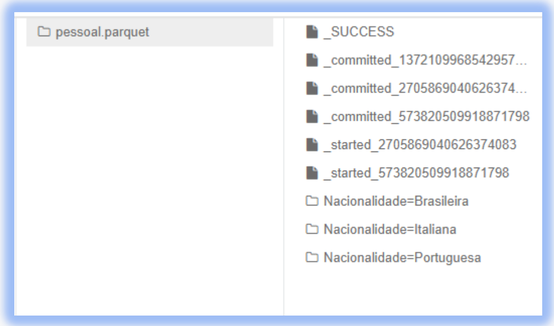
```
#criando um dataframe com dados fixos
dados =[("Grimaldo ","Oliveira","Brasileira","Professor","M",3000),
              ("Ana ","Santos","Portuguesa","Atriz","F",4000),
              ("Roberto","Carlos","Brasileira","Analista","M",4000),
              ("Maria ","Santanna","Italiana","Dentista","F",6000),
              ("Jeane","Andrade","Portuguesa","Medica","F",7000)]
colunas=["Primeiro_Nome","Ultimo_nome","Nacionalidade","Trabalho","Genero","Salario"]
datafparquet=spark.createDataFrame(data,columns)
datafparquet.show()
```

Executa Script

Gravando o arquivo Parquet

```
1 #criando o arquivo parquet
2 datafparquet.write.parquet("/FileStore/tables/parquet/pessoal.parquet")
```

Arquivo gerado



Criando arquivos parquet

Veja como criar um arquivo parquet lendo um arquivo CSV.

Vamos ler um arquivo CSV e vamos guardar no arquivo parquet.

DATAFRAME lendo de um arquivo CSV

```
%python
#Leitura de arquivo CSV
dataframesp= spark.read.format("csv").option("header", "true").load("/FileStore/tables/arquivo/Datafiniti_Hotel_Reviews_Jun19.csv")
dataframesp.show()
```

Executa Script de gravação em parquet

```
#criando o arquivo parquet
dataframesp.write.parquet("/FileStore/tables/parquet/csvparquet3.parquet")
```

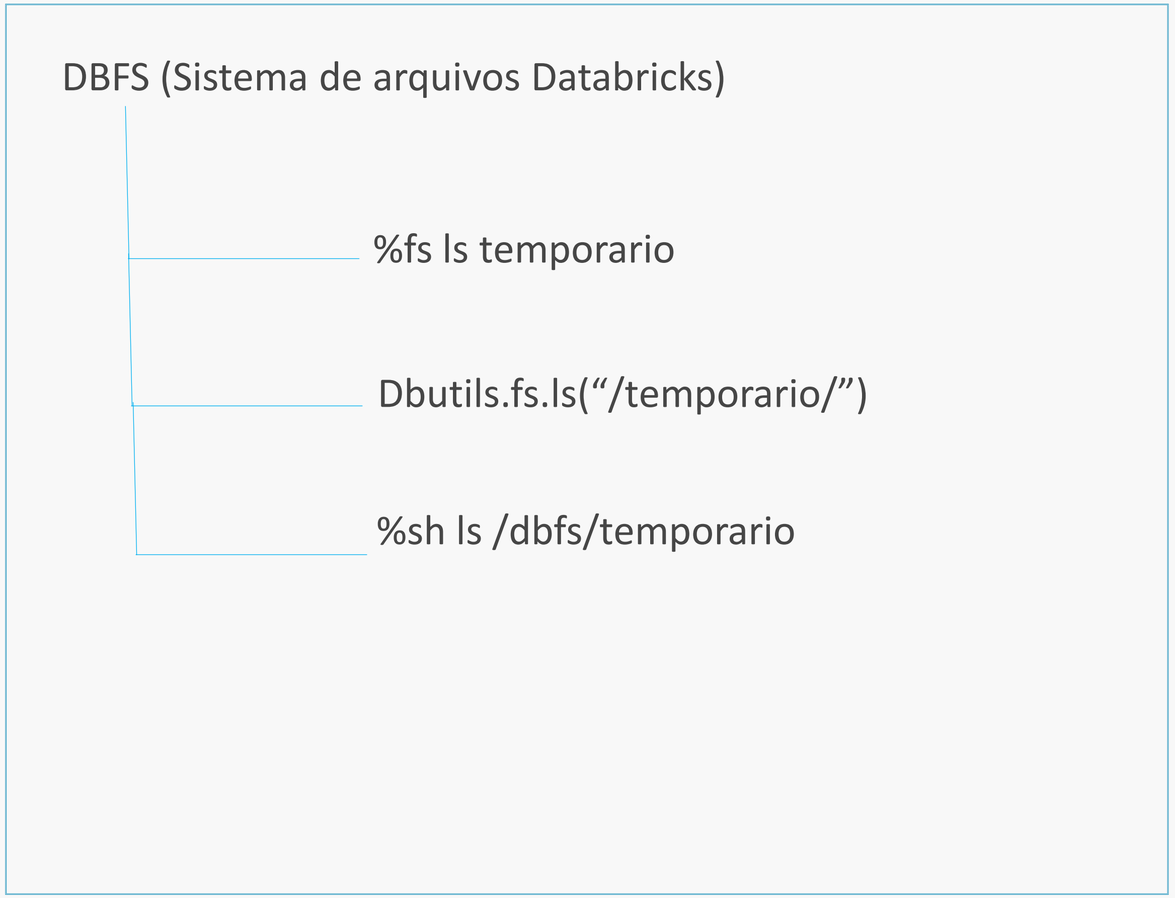
Arquivo gerado

Sistema de Arquivos

Comandos de manipulação.

Comandos

Acessar o sistema de arquivos usando comandos %fs (sistema de **arquivos)** ou **%sh** (shell de comando). Existe a biblioteca **dbutils.fs** Python. São formas diferentes de obter os mesmos resultados.



Construção de dashboard

Visão dos gráficos gerados.

Dashboard

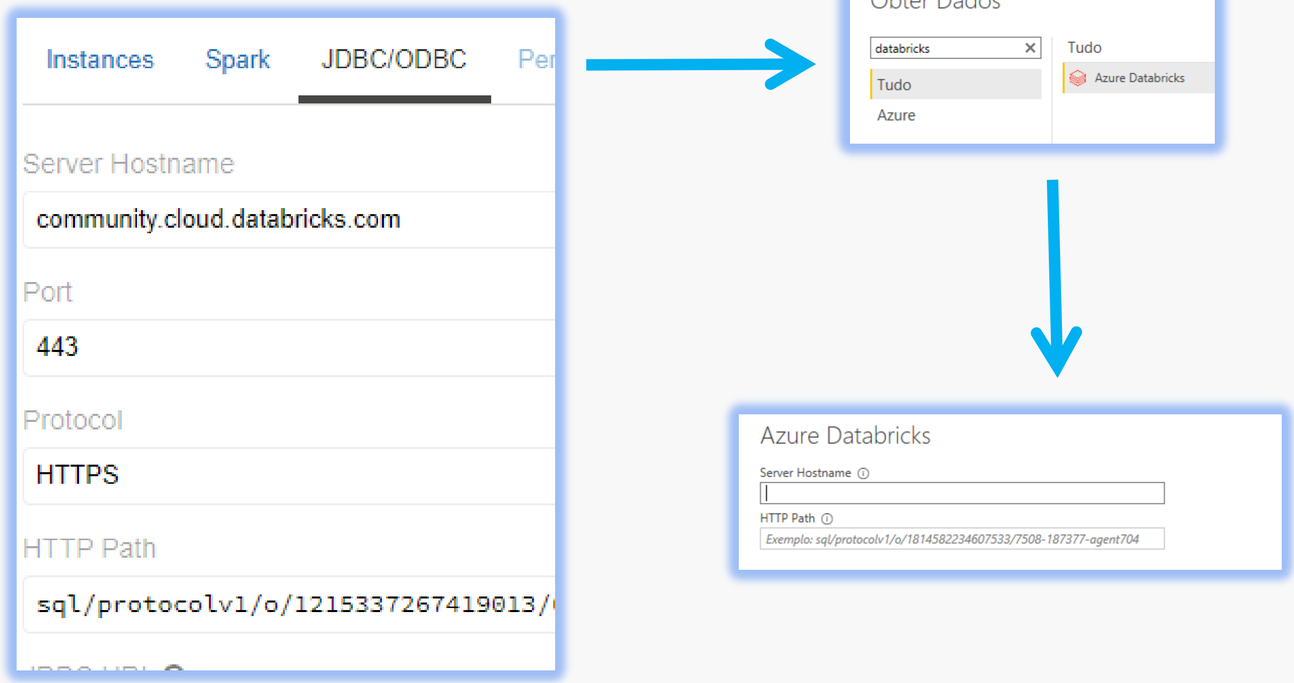
É possível construir um dashboard com gráficos que são gerados. Vamos gerar gráficos em R para aprender sobre esta construção.

Power BI e Databricks

Leia os dados do Databricks no Power BI com uma simples configuração.

Conectando Databricks no Power BI

Vamos conectar o Databrics diretamente no Power BI, ferramenta de visualização de dados.



PRÁTICA

ENVIE AO PROFESSOR

PREPARE UM ESTUDO

Carregando qualquer arquivo do site **kaggle.com** e crie um cluster, scripts, visualizações, dashboards e analise seus dados.

Muito boa sorte e conte comigo!

```
%sql
select country, sum(price) as total_vendido from dados_vinho
where price > 0
group by country
order by total_vendido desc
limit 10
```

	country	variety	total_vendido
1	US	Pinot Noir	439530
2	US	Cabernet Sauvignon	398345
3	US	Chardonnay	207482
4	Italy	Nebbiolo	150432
5	France	Bordeaux-style Red Blend	129764
6	Italy	Red Blend	129494
7	France	Pinot Noir	128934
8	France	Chardonnay	127724
9	US	Syrah	125775
10	US	Bordeaux-style Red Blend	105443