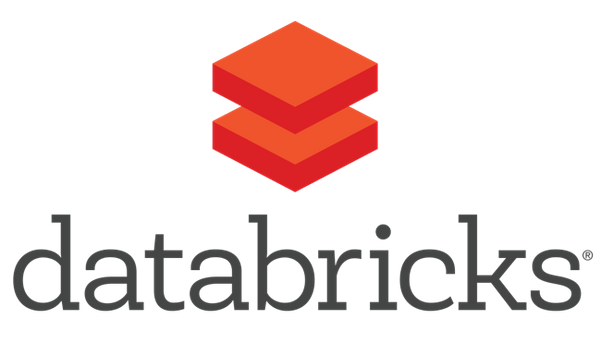
**Tutorial**

**Utilização do ambiente Databricks**

Este tutorial foi desenvolvido pelo **Prof. João Nogueira**, quaisquer dúvida, correção ou sugestão, enviar um email para [joaonogueira.labdata@gmail.com](mailto:joaonogueira.labdata@gmail.com) ou enviar uma mensagem no perfil do linkedin <https://www.linkedin.com/in/joaopaulonogueira/>



**Sumário**

1. Introdução
2. Criando uma conta
3. Criando um cluster
4. Executando código: Hello Databricks!
5. Criando tabelas
6. Carregando as tabelas no código
7. Carregando os notebooks

**Introdução**

O [Databricks](https://databricks.com/) é uma empresa fundada em 2013 pelos criadores do [Spark](https://spark.apache.org/). Construído sob uma moderna arquitetura na cloud, Databricks entrega uma combinação de data warehouses e data lakes para a criação de uma plataforma analítica unificada para processamento de dados e inteligência artificial.

Muitas companhias atualmente utilizam o Databricks, se beneficiando das suas múltiplas capacidades, como engenharia de dados em alta escala, ciência de dados colaborativa, desenvolvimento completo do ciclo analítico de machine learning e business analytics. [Veja alguns exemplos](https://databricks.com/customers).

O Databricks oferece uma conta gratuita, chamada [Databricks Community](https://community.cloud.databricks.com/login.html), onde podemos aprender sobre a plataforma e utilizar muitos de seus recursos para projetos pessoais, como por exemplo desenvolver todo o ciclo de machine learning feito em aula para o problema de classificação.

**Quais os benefícios do Databricks Community?**

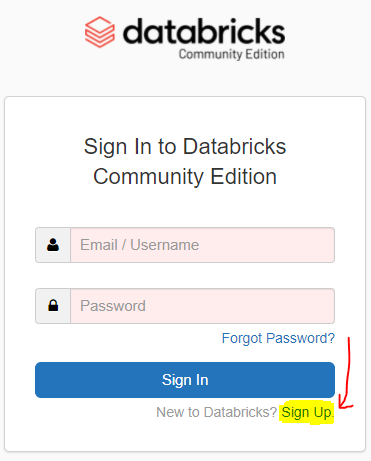
1. Gratuito.
2. Ambiente facilmente gerenciável para criação de clusters e aprendizagem de ferramentas de processamento de big data, como o Spark.
3. Suporte a diferentes APIs do Spark, como [PySpark](https://spark.apache.org/docs/latest/api/python/) e [Spark SQL](https://spark.apache.org/sql/).
4. Construído baseado no Jupyter

**Quais as limitações do Databricks Community?**

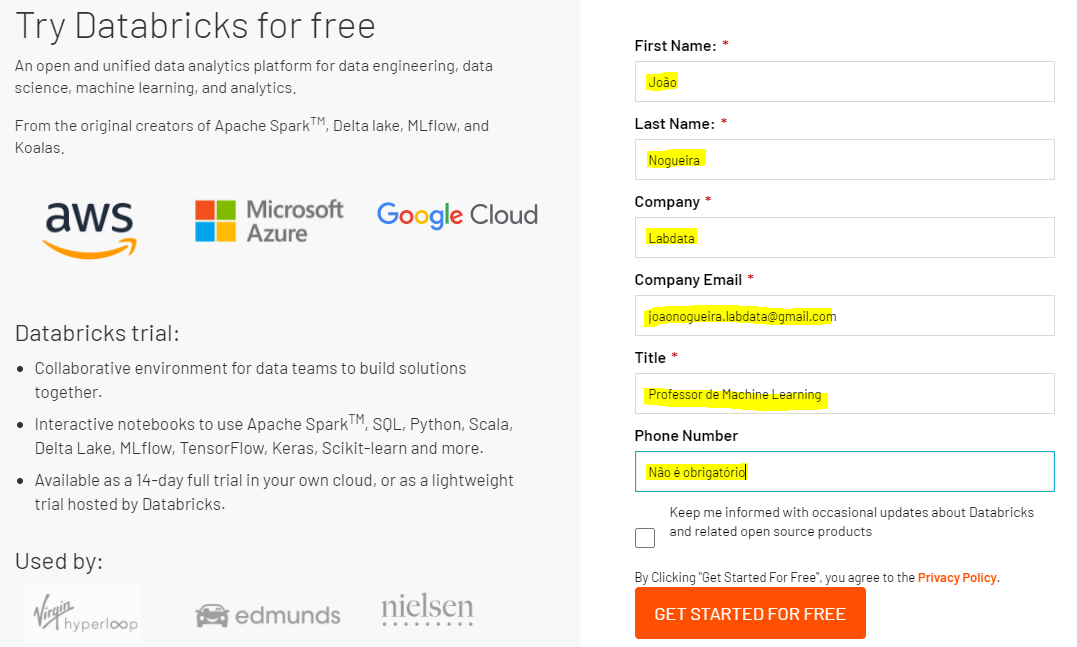
1. Limite de inatividade: qualquer cluster criado será automaticamente desligado depois de um período de 2h de inatividade.
2. Limite de memória e processamento: é disponibilizado um cluster com 15GB de memória ram. Parece pouco, mas é suficiente para nossas atividades em classe.

**Criando uma conta**

Para criar uma conta no Databricks Community, acesse: <https://community.cloud.databricks.com/login.html>

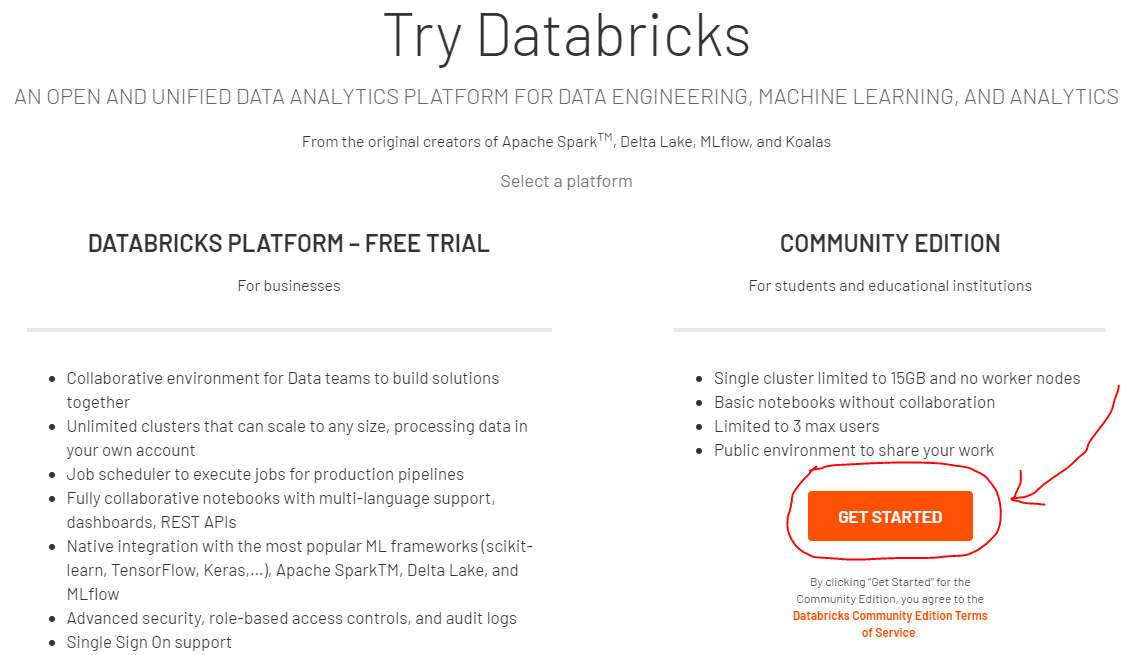


Clique em **Sign Up** e preencha o **formulário** com os seus dados:

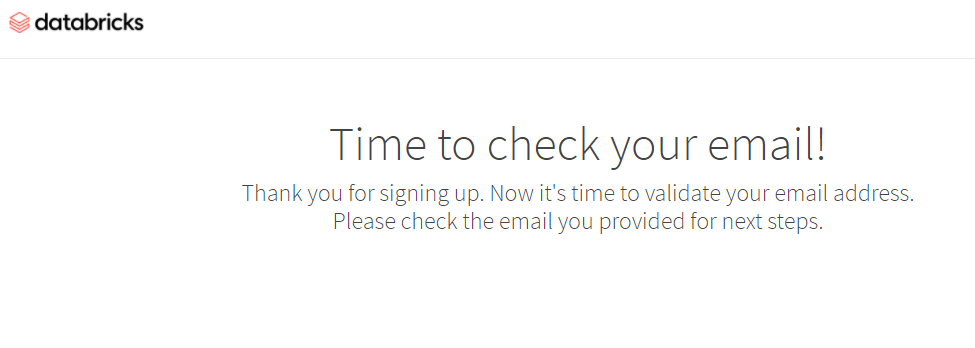


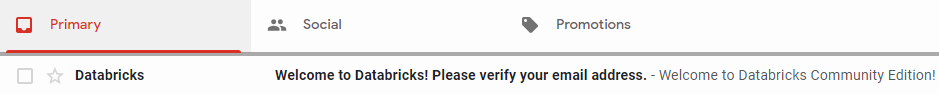
Depois de preencher os dados, clicar no botão **GET STARTED FOR FREE**.

Na próxima página, serão mostradas duas opções: **DATABRICKS PLATFORM - FREE TRIAL** e **COMMUNITY EDITION**. Escolha a opção **COMMUNITY EDITION** clicando no botão **GET STARTED**.

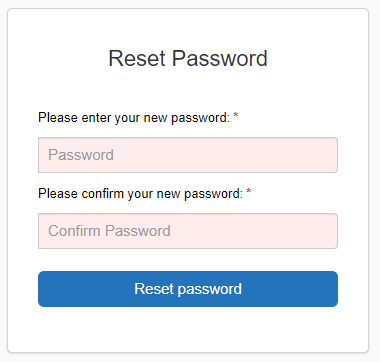


Agora verifique no email fornecido se chegou uma mensagem contendo os próximos passos para a criação da conta. Caso não tenha recebido a mensagem, não esqueça de verificar a **caixa de spam**.

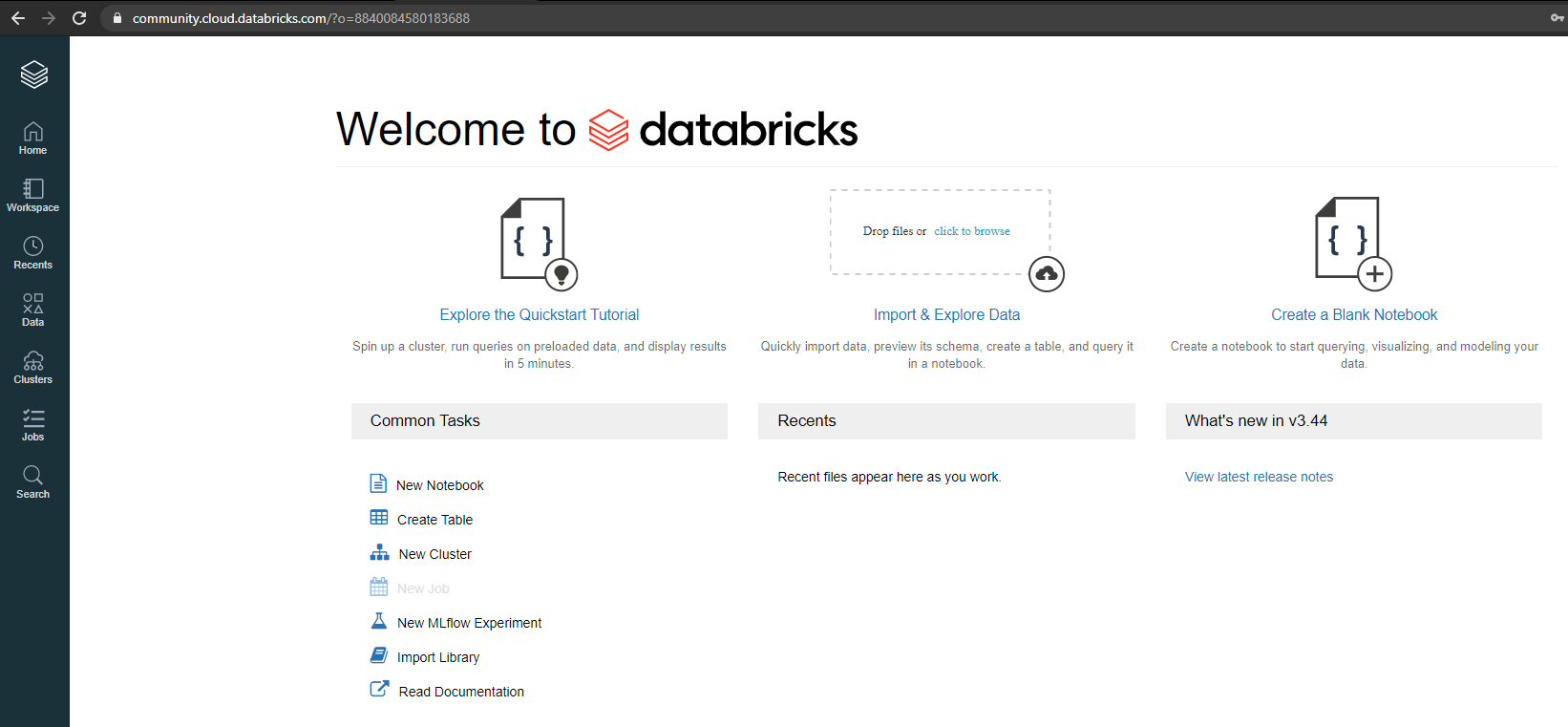




No email, haverá um link para que seja **verificado o seu email** e **criada a senha da sua conta**:

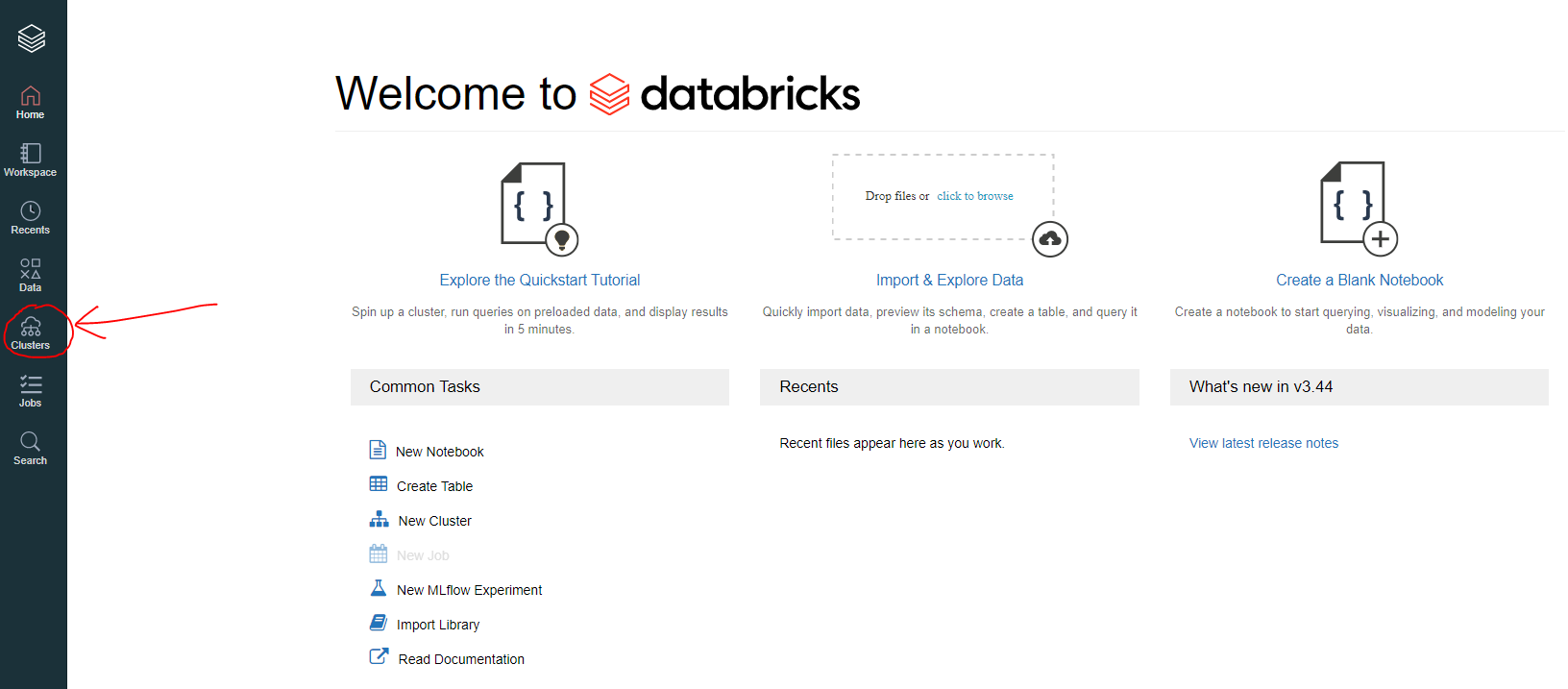


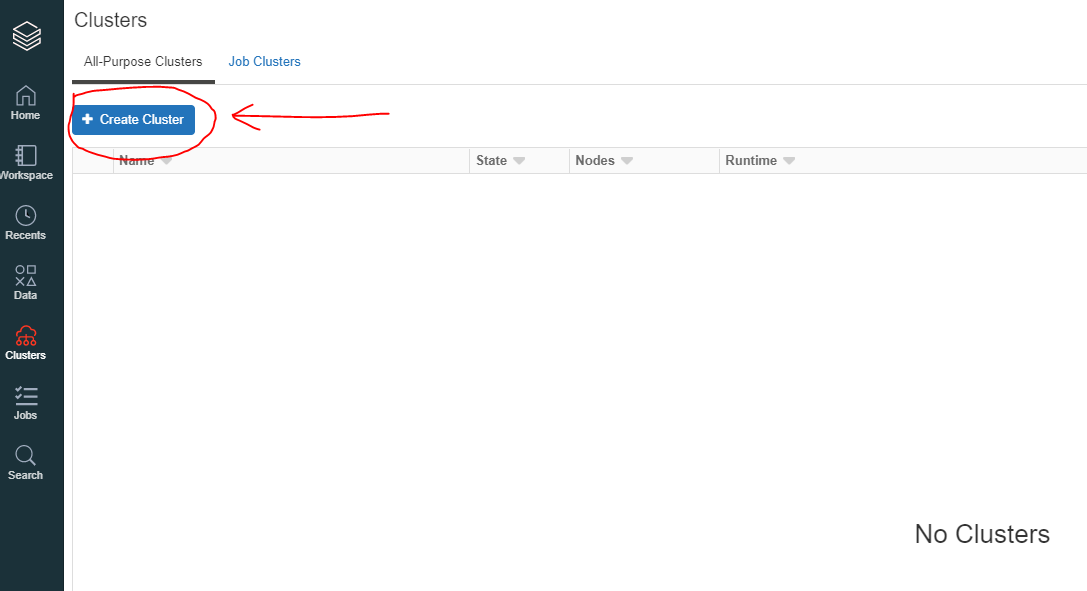
Depois de inserir a nova senha e clicando no botão **Reset password**, você será finalmente direcionado para a página inicial do seu **ambiente do Databricks**.



**Criando um cluster**

Para criar um cluster, clique no botão **Clusters** localizado na barra de opções do lado esquerdo e em seguida clicar no botão **Create Cluster**.





Na página seguinte, preencha o **formulário de criação do cluster** fornecendo um **nome** para o mesmo e clicando em **Create Cluster** (deixe as outras opções com os valores default). Feito isso, o Databricks irá começar a executar a solicitação de criação do cluster:

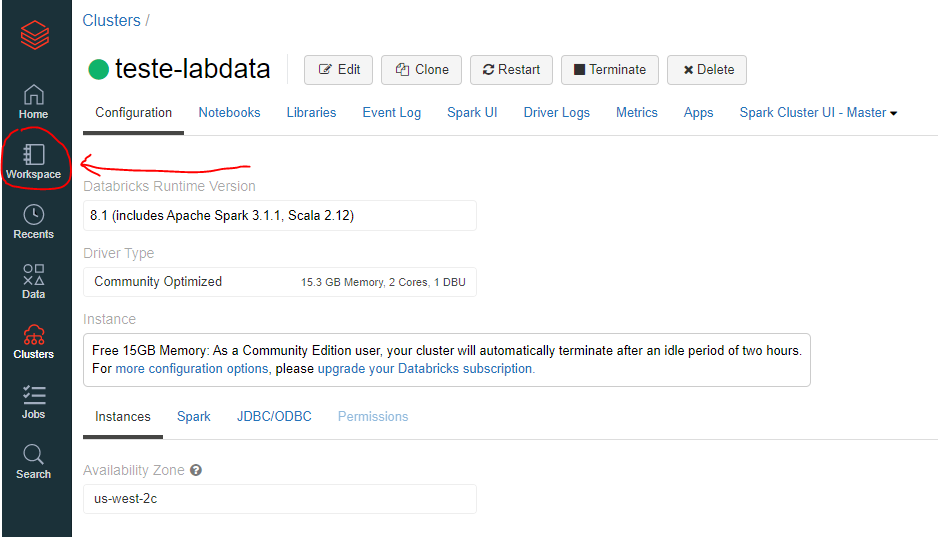


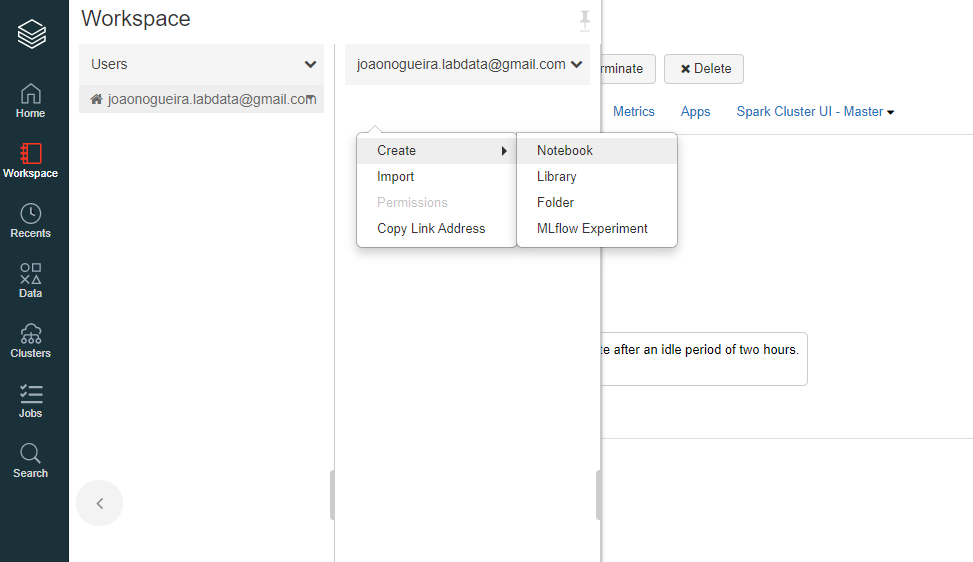
Finalizada a criação do cluster, ele ficará disponível para utilização, com um sinal verde ao lado do nome:



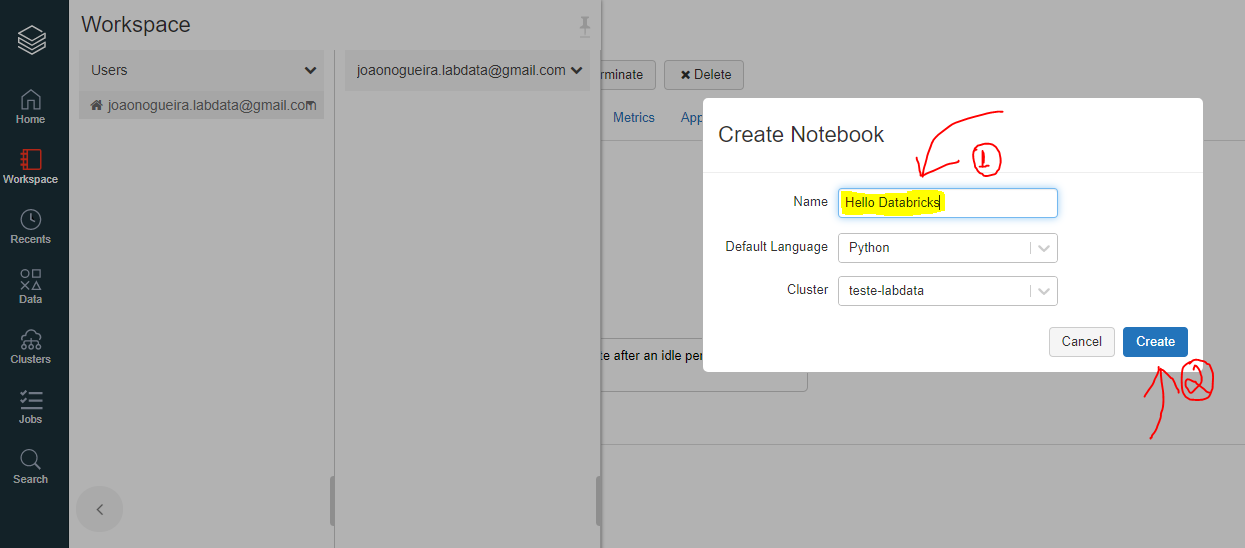
**Executando código: Hello Databricks!**

Para começarmos a executar códigos no Databricks, precisamos primeiro criar um notebook (similar ao notebook jupyter ou no google colab). Para tal, clicamos em **Workspace** > **Botão Direito** > **Create** > **Notebook**.

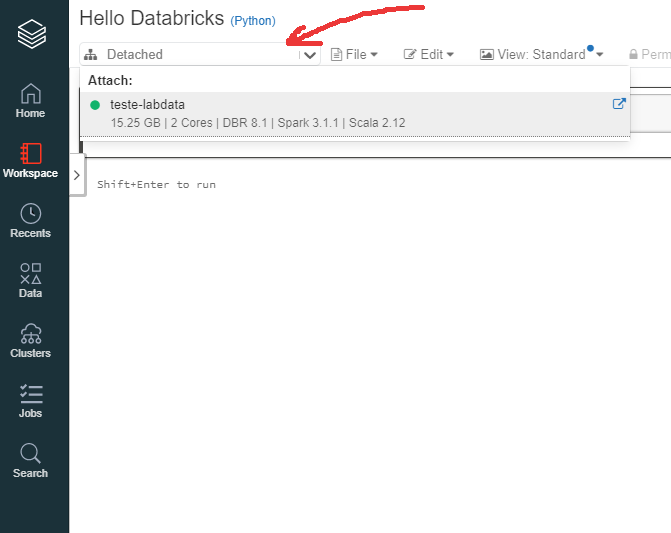




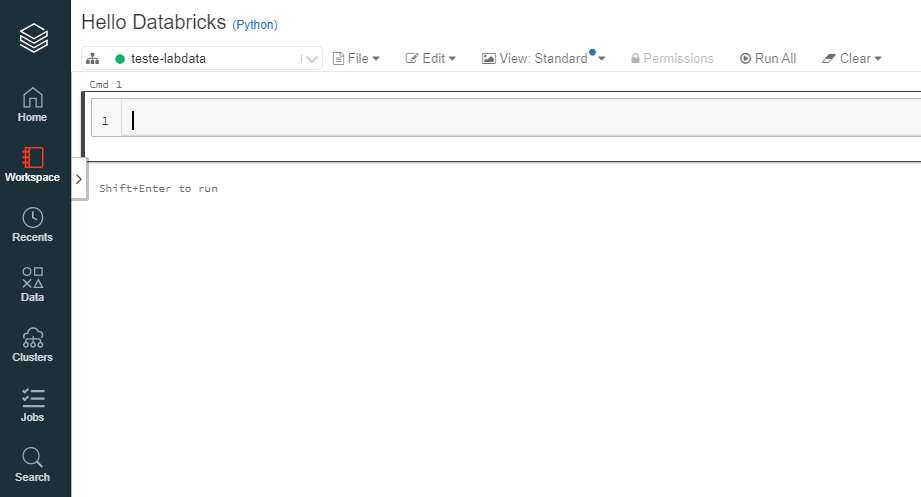
Na janela que aparece, digitar o nome do notebook e clicar em **create**.



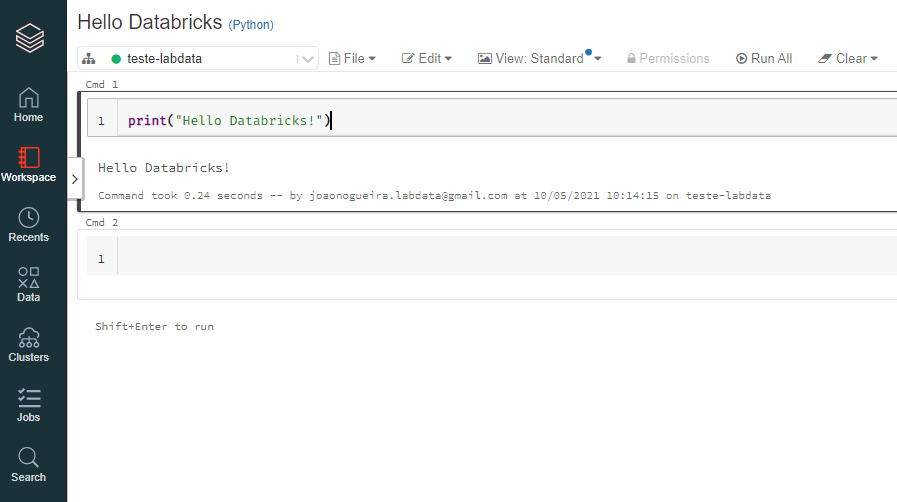
Uma vez criado o notebook, precisamos atribuir um cluster a ele. Clique na caixa **Detached** no canto superior esquerdo e selecione o cluster recém-criado clicando em seu nome e confirmando a operação.



Feito isso, o seu notebook agora está utilizando o cluster recém-criado e o nome do cluster deve estar aparecendo na caixa de cluster do seu notebook.



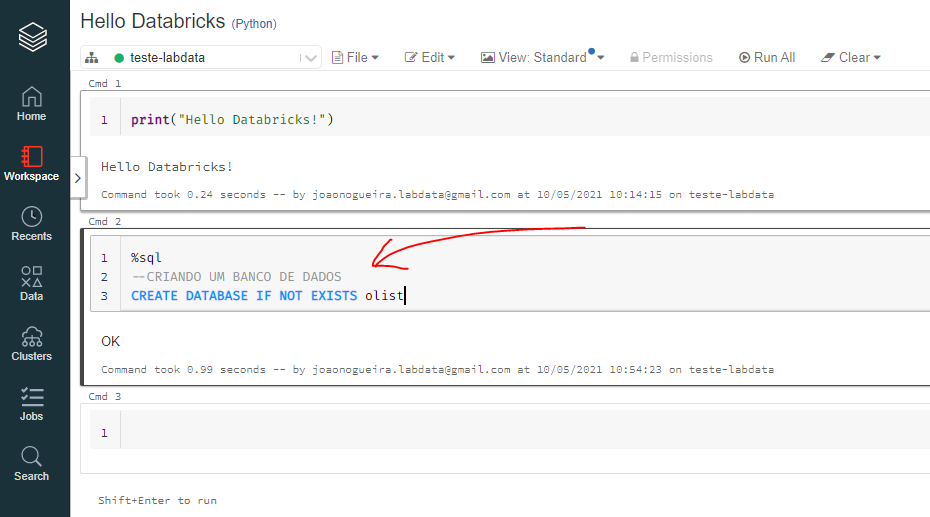
Assim como no Google Colab, o Databricks é composto de células onde podemos executar códigos e textos. Para experimentar, digite o comando como mostrado na próxima figura e pressione Shift+Enter para executar:



**Criando tabelas**

Vamos aprender a criar tabelas utilizando arquivos em csv. Para completar este tutorial, faça o download das bases em formato csv da Olist disponíveis no kaggle no seguinte link: <https://www.kaggle.com/olistbr/brazilian-ecommerce>.

Primeiramente, vamos criar um banco de dados chamado olist, onde iremos armazenar as nossas tabelas. Execute o comando como mostrado na figura abaixo para criar o banco de dados:

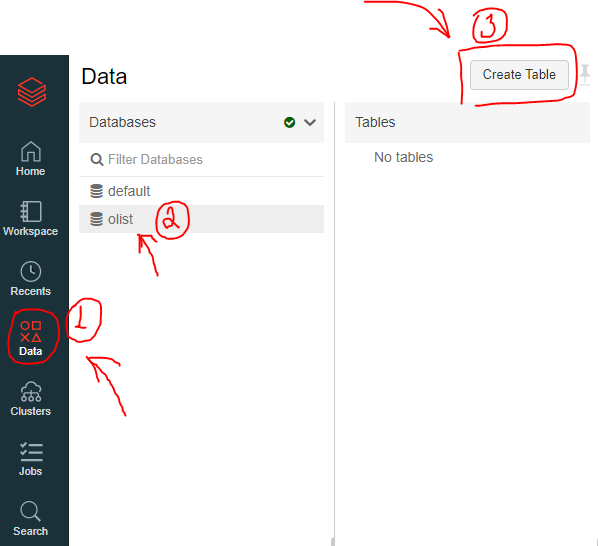


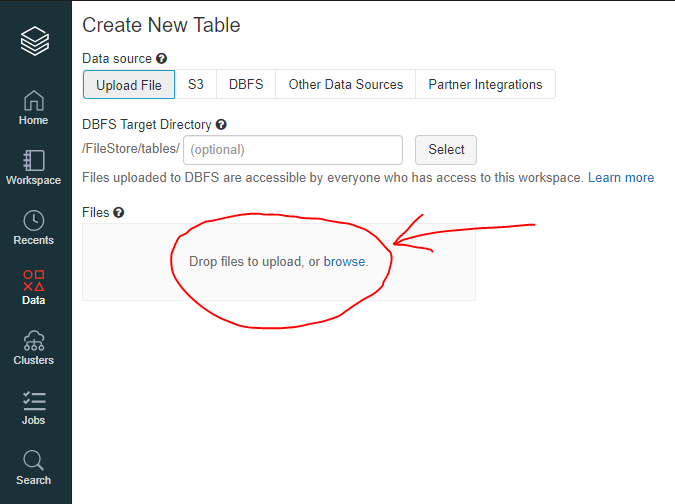
Explicação do código:

1. Na linha 1 temos o comando **%sql**,que indica que queremos executar código SQL na célula atual. Por padrão estamos executando python;
2. Na linha 2 temos um comentário em SQL explicando o que será feito na linha seguinte;
3. Na linha 3 efetivamente criamos um banco de dados chamado **olist**.

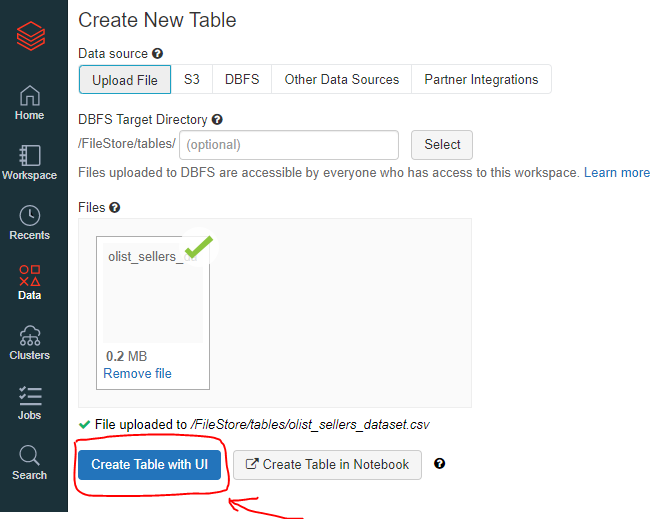
Criado o banco de dados, podemos então prosseguir para criar tabelas dentro do banco de dados **olist**.

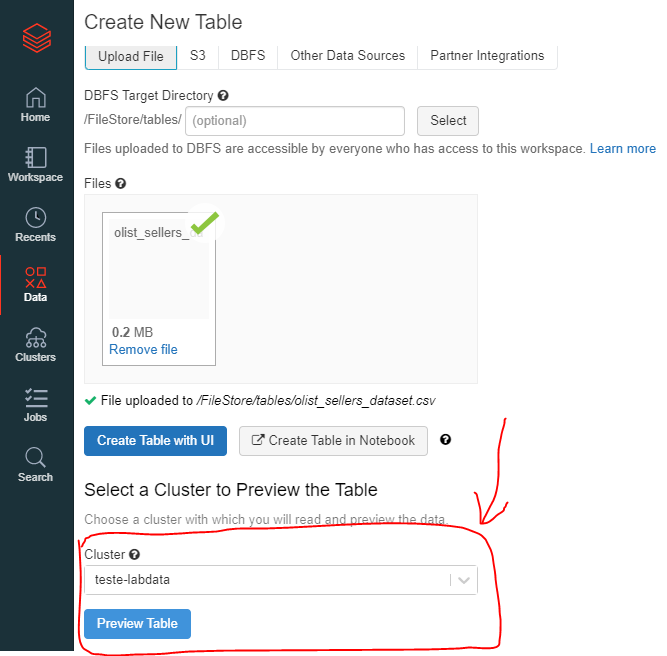
Podemos ver que temos dois bancos de dados: **default**, que é o banco de dados já criado pelo próprio databricks e o **olist**, banco de dados que criamos no passo anterior. Para criar uma tabela no banco de dados **olist**, clique em **Data** > **olist** > **Create Table** e depois selecione o arquivo **olist\_sellers\_dataset.csv**.



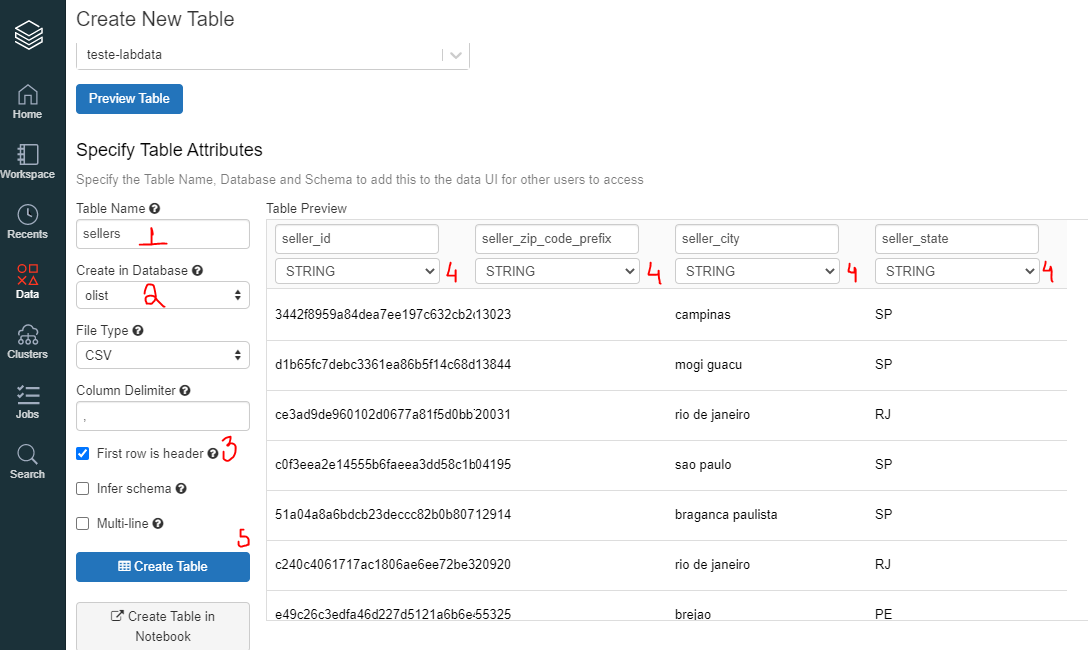


Com o arquivo carregado, selecione a opção **Create Table with UI**.

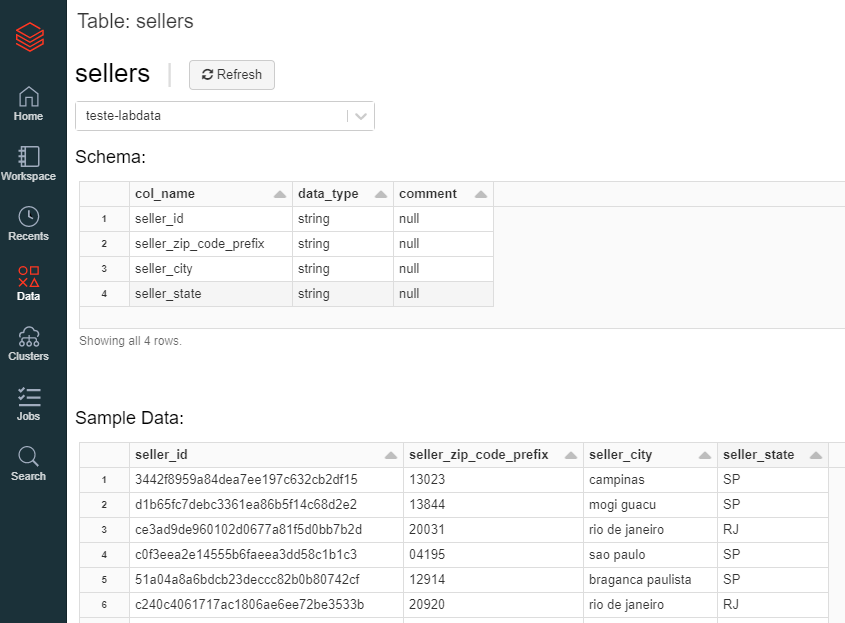


Na próxima janela selecione o cluster criado e clique em **Preview Table**.

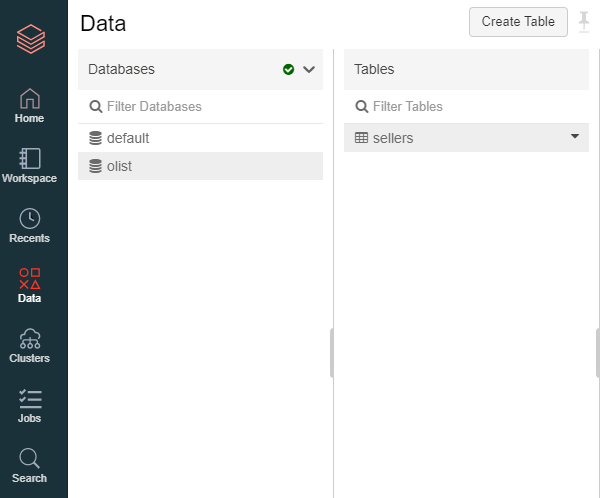
Agora você pode criar a sua tabela. Pode por exemplo, dar um novo nome1 para a sua tabela (apague o nome que aparece e renomeie para **sellers**). Altere2 também o banco de dados, selecionando o banco **olist** e marque a opção **First row is header3.** Por fim, verifique os tipos de dados4 (todos são do tipo **STRING**)e clique em **Create Table5.**



Para confirmar que a tabela foi criada, o Databricks irá nos mostrar uma tela final com o Schema tabela e uma amostra dos dados contidos nela:



Podemos também verificar que a tabela foi criado selecionando a opção **Data > olist**. Na aba da direita aparecerá uma lista de tabelas onde deve ser possível ver a tabela recém-criada, que chamamos de **sellers**.



**Exercício**

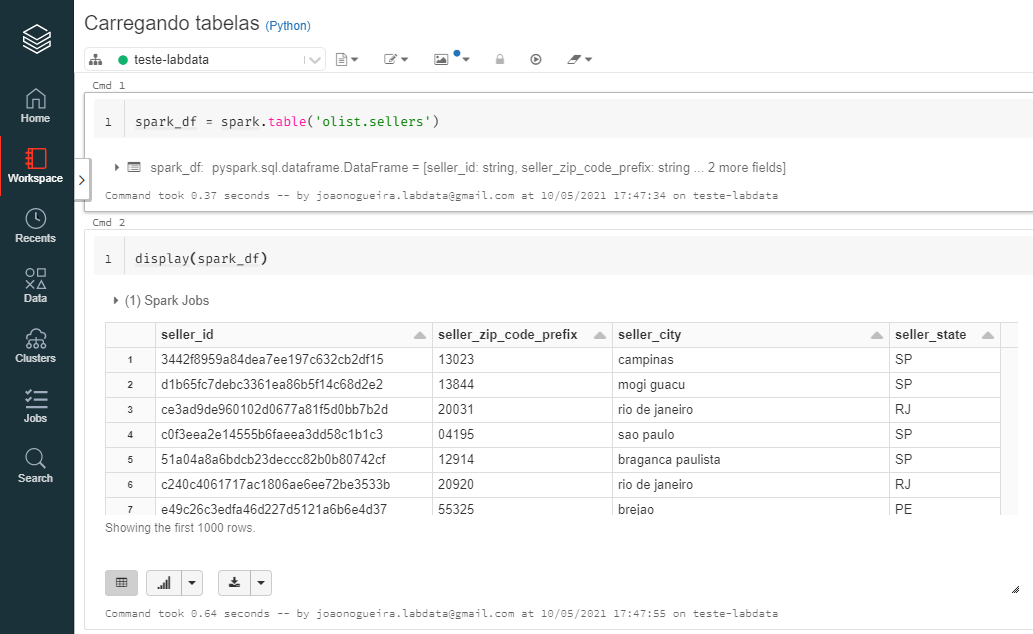
Carregar os arquivos *olist\_order\_items\_dataset.csv* e *olist\_orders\_dataset.csv* e criar, respectivamente, as seguintes tabelas: **order\_items** e **orders**.

**Carregando as tabelas no código**

1. Abra um novo notebook com o nome **Carregando tabelas**.
2. Carregue a tabela utilizando o PySpark com o seguinte comando:

spark\_df = spark.table(‘olist.sellers’)

1. Execute o comando display(spark\_df) para ver os dados carregados.
2. Faça o mesmo para as outras tabelas.



Agora estamos prontos para processarmos os nossos dados no Databricks utilizando o PySpark. O PySpark é uma forma de escrever códigos em Spark utilizando o Python! Logo, tudo o que aprendemos até aqui com python e pandas, irá nos ajudar e acelerar o nosso aprendizado de PySpark. Como o PySpark é uma ferramenta de processamento de big data, o volume de dados que o Pandas não for capaz de processar, certamente o PySpark conseguirá dar conta. Let’s go code!

**Carregando os notebooks**

1. Navegue até o seu Workspace, clique com o botão direito do mouse e selecione a opção **Import**
2. Na janela de diálogo que se abre, arrastar o arquivo **Olist.dbc** e finalizar clicando no botão **Import**.

