

# MVP Engenharia de dados

Profs. Victor Almeida e Silvio Alonso

Aluno: Welton Benedito de Souza

Análise de dados financeiros de empresas petroleiras júniores em 2022

## 1. Descrição

O projeto de aprendizado supervisionado é composto por datasets de demonstração de resultado e balanço patrimonial das empresas registradas na CVM (Comissão de Valores Mobiliários) em 2022 e informações do mercado financeiro como número de ações em circulação e a cotação das ações no IBOVESPA.

As empresas selecionadas para análise, neste projeto, são empresas independentes de óleo e gás natural no Brasil, chamadas empresas petroleiras júniores ou "Junior Oils".

As empresas júniores de petróleo no Brasil listadas na B3 desempenham um papel significativo na exploração e produção de petróleo e gás natural no país. Elas são fundamentais para o setor de energia do Brasil e desempenham um papel vital na busca de recursos energéticos para atender às necessidades do país. Quatro das principais empresas júniores de petróleo na B3 são a PetroRio, a PetroReconcavo, a 3R Petroleum e a Enauta. Cada uma delas tem suas próprias características e áreas de atuação no setor de petróleo e gás.

- **PetroRio:** A PetroRio é uma empresa brasileira de exploração e produção de petróleo e gás natural. Ela se destaca por sua estratégia de crescimento por meio da aquisição de ativos maduros, revitalizando e aumentando sua produção. A empresa tem um histórico de sucesso na revitalização de campos de petróleo, contribuindo para a produção de petróleo nacional.
- **PetroReconcavo:** A PetroReconcavo é outra empresa focada na exploração e produção de petróleo e gás no Brasil. Seu foco principal é em campos terrestres maduros, com operações concentradas principalmente no Nordeste do Brasil. A empresa busca maximizar a recuperação de petróleo em campos já em produção, otimizando sua eficiência.
- **3R Petroleum:** A 3R Petroleum é conhecida por sua atuação no setor de recuperação de campos maduros de petróleo e gás. Ela se concentra em adquirir campos em produção que podem ser revitalizados e operados de forma mais eficiente, prolongando a vida útil desses ativos e aumentando a produção.
- **Enauta:** A Enauta é uma empresa brasileira de exploração e produção de petróleo e gás natural, com ativos tanto em terra quanto offshore. A empresa tem participação em projetos importantes, incluindo o campo de Atlanta, na Bacia de Santos, e outros ativos no Nordeste do Brasil. Ela se dedica a explorar novas oportunidades de produção e desenvolvimento no setor.

As "Junior Oils" operam hoje em mais de 180 campos de exploração de petróleo e gás, tiveram um crescimento considerável impulsionado pela aquisição de ativos da Petrobras por meio do plano de desinvestimento iniciado em 2015, ainda no Governo Dilma Rousseff. A produção em bacias terrestres, que estava em declínio desde 2012, passou a aumentar substancialmente com a entrada dessas empresas, além revitalização de campos maduros de petróleo e gás natural offshore.

As ações das "Junior Oils" são consideradas como "small caps" (são ações de empresas com menor capitalização de mercado em relação às empresas de maior porte listadas em uma bolsa de valores), se enquadram em categorias de menor capitalização de mercado, uma vez que são empresas em estágios iniciais de exploração e produção de petróleo e gás, e frequentemente têm recursos financeiros limitados em comparação com as grandes empresas do setor de petróleo.

Através da análise fundamentalista muito utilizada por investidores e analistas financeiros para avaliar o valor intrínseco de um ativo financeiro, como ações, títulos, moedas e commodities. baseando-se na análise de informações financeiras das "Junior Oils" com o objetivo de determinar se ele está subvalorizado ou supervalorizado no mercado.

A utilização da plataforma do Azure Databricks para desempenhar como um pipeline de dados através do Microsoft Azure, fornecendo capacidades de processamento de dados, desenvolvimento de código, integração com serviços de armazenamento e segurança de dados em escala na nuvem

## 2. Objetivo

É avaliar as alternativas de investimento em ações de empresas através de um estudo da situação financeira e das perspectivas das empresas. A utilização de um dos aspectos da análise fundamentalista para encontrar o valor real de uma empresa para investimento com foco no longo prazo.

### Perguntas elaboradas para atender os objetivos definidos

1 - Margem: Qual foi a margem bruta, margem líquida e a Margem Ebit das empresas Oil junior em 2022?

2 - Capacidade Financeira: Qual é a liquidez das empresas Oil junior em 2022?

3 - Endividamento: Qual é o nível de endividamento da empresa em 2022? Qual a proporção de dívida em relação ao patrimônio líquido?

4 - Rentabilidade dos Ativos e Patrimônio: Qual é a rentabilidade dos ativos e do patrimônio líquido da empresa em 2022?

5 - Marketcap: Qual o valor de mercado das empresas Oil junior em 2022?

6 - P/L: os preços das ações das empresas Oil junior em 2022 estão caros ou baratos?

7 - Dividendos: A empresa distribuiu dividendos aos acionistas em 2022?

8 - Vale a pena a investir nas empresas Oil junior em 2022?

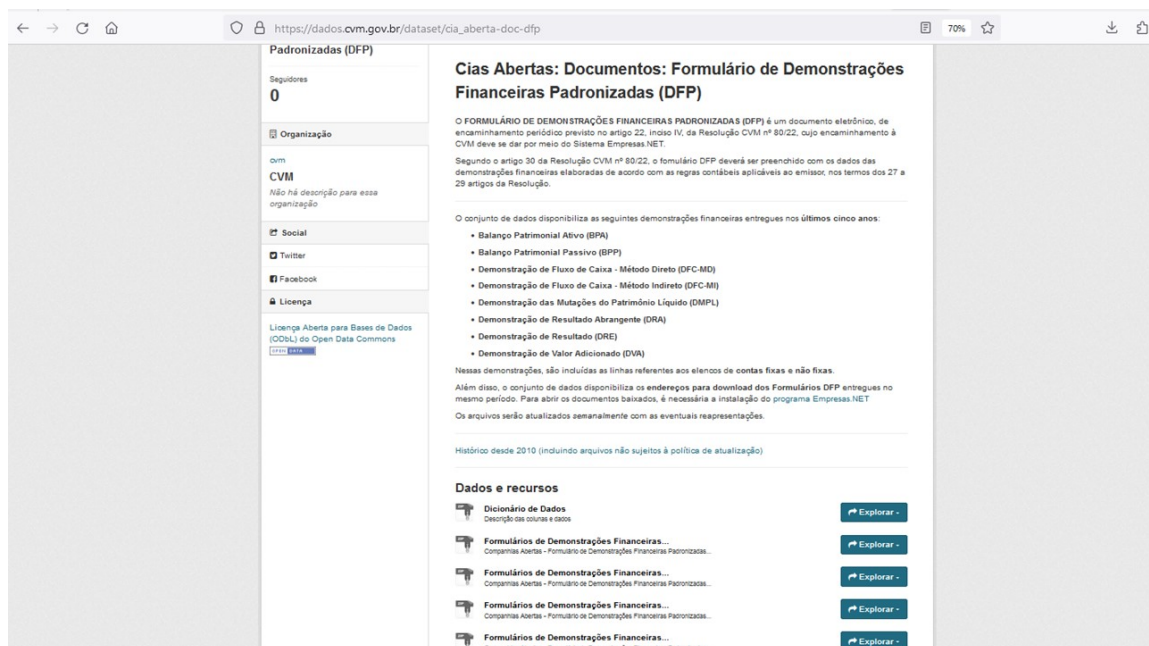
## 3. Detalhamento

### 3.1 Busca pelos dados

[https://dados.cvm.gov.br/dados/CIA\\_ABERTA/DOC/DFP/DADOS/dfp\\_cia\\_aberta\\_2022.zip](https://dados.cvm.gov.br/dados/CIA_ABERTA/DOC/DFP/DADOS/dfp_cia_aberta_2022.zip)

Os arquivos utilizados são:

dfp\_cia\_aberta\_BPA\_con\_2022.csv  
dfp\_cia\_aberta\_BPP\_con\_2022.csv  
dfp\_cia\_aberta\_DRE\_con\_2022.csv



### 3.2 Coleta

Dados baixados para a máquina local e inseridos manualmente no catálogo da databricks da microsoft Azure

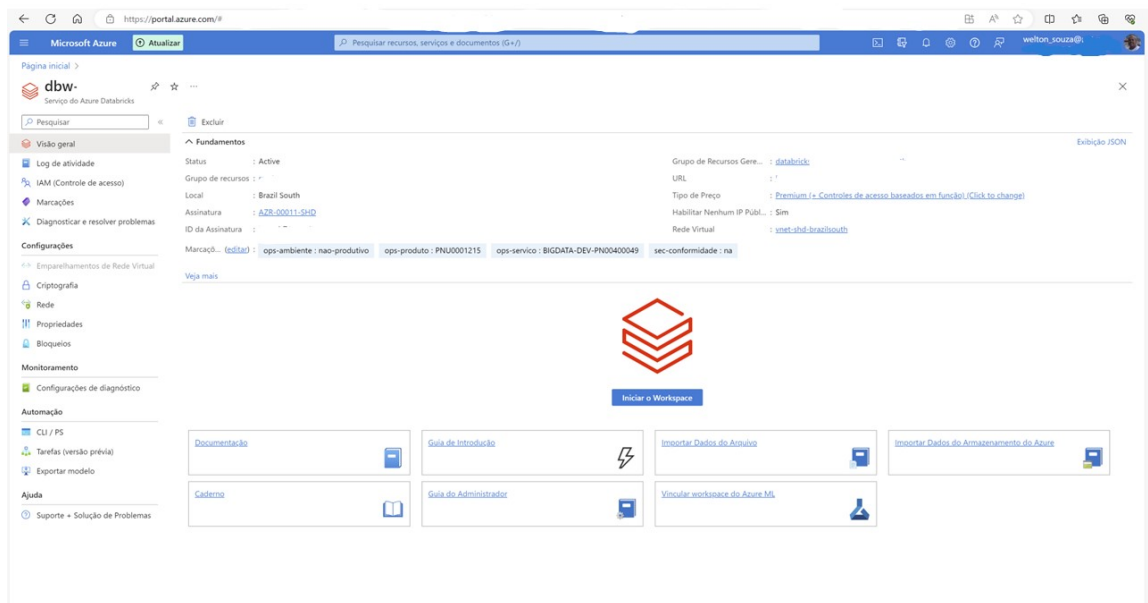
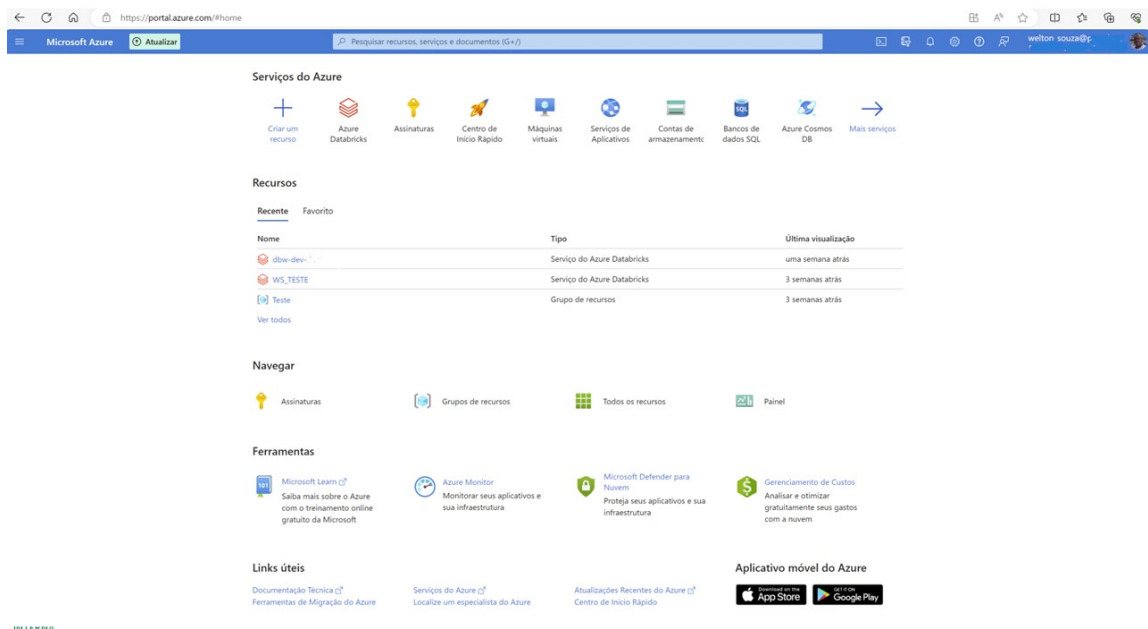
#### 3.2.1 Plataforma de nuvem:

Utilizamos Azure Databricks como plataforma de análise colaborativa baseada no Apache Spark, a estrutura de análise paralela de código aberto mais popular. Foi desenvolvido em colaboração entre Microsoft e Databricks e é profundamente integrado a muitos serviços do Azure.

**Principais Características:**

1. **Ambiente Colaborativo:** Azure Databricks fornece um ambiente de notebook para engenheiros de dados, cientistas de dados e analistas de negócios colaborarem usando o mesmo espaço de trabalho.

2. **Desempenho:** Otimizado para o Microsoft Azure, é projetado para oferecer desempenho aprimorado e uma configuração mais fácil. Por exemplo, com o recurso de autoscaling, os clusters podem aumentar ou diminuir automaticamente com base nas necessidades de processamento.
3. **Segurança:** Está profundamente integrado com o Azure Active Directory e oferece controles de acesso baseados em função. Os dados são protegidos em trânsito e em repouso.
4. **Integração com Serviços Azure:** É integrado com uma série de serviços Azure, como Azure Data Lake Storage, Azure Blob Storage, Azure Synapse Analytics, entre outros. Isso facilita a construção de fluxos de trabalho end-to-end dentro do ecossistema do Azure.
5. **Gerenciamento Simplificado:** O gerenciamento de clusters e trabalho é simplificado no Azure Databricks. Você pode criar, configurar e programar tarefas facilmente por meio de sua interface amigável.

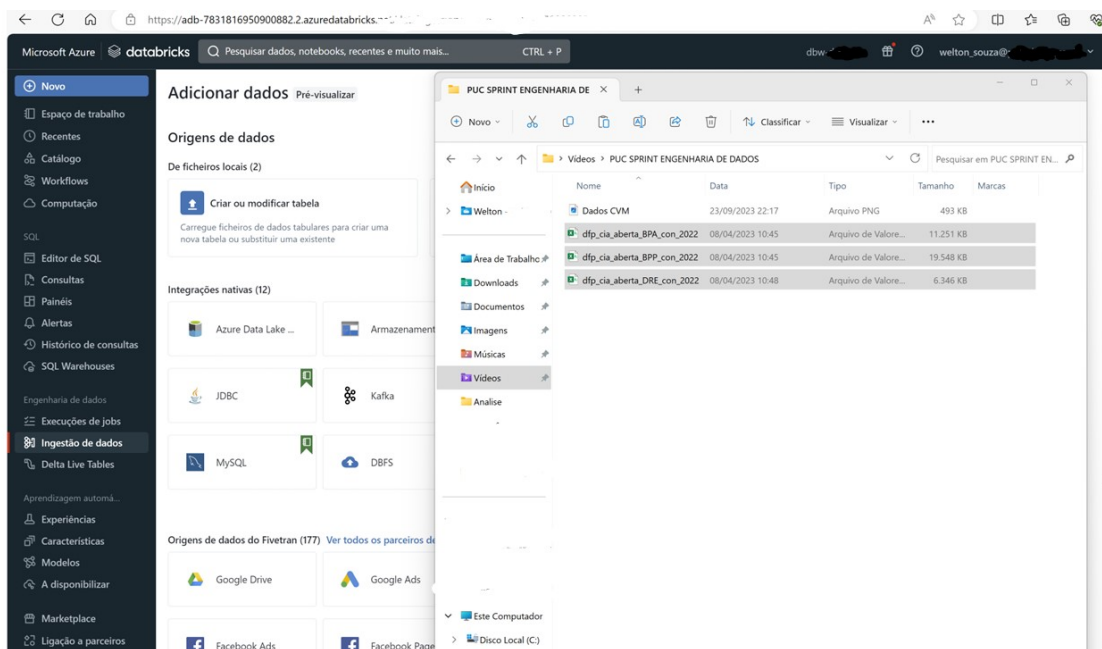


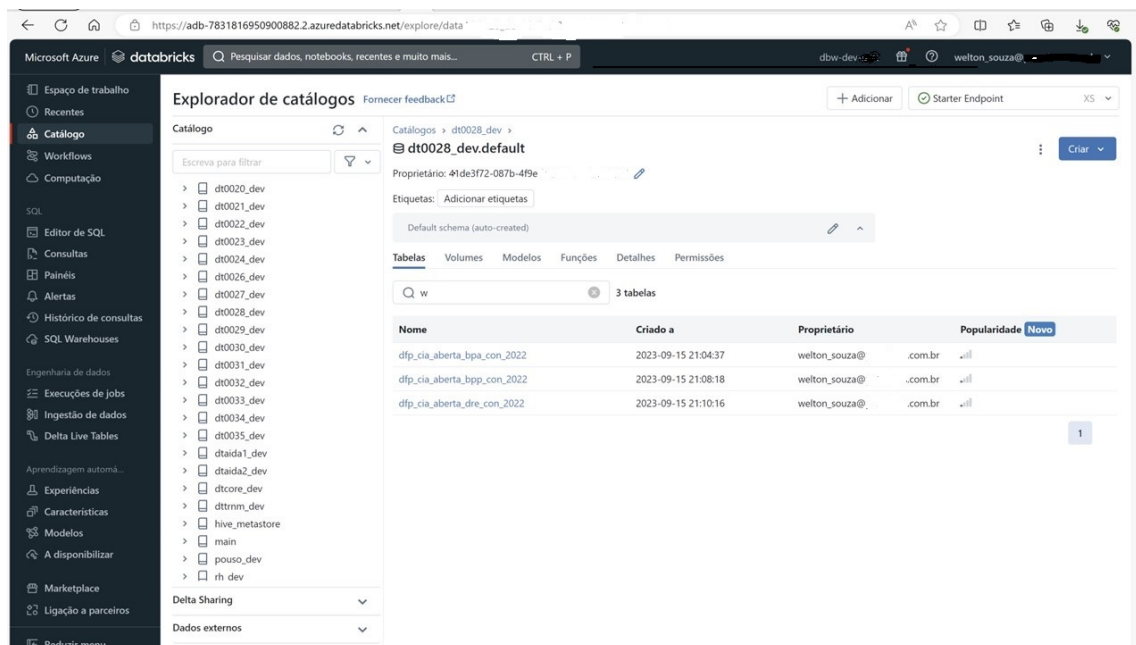
### 3.2.2 Ingestão de dados

O Azure Databricks é uma plataforma poderosa para análise de dados e em relação à ingestão de dados:

- Ele suporta uma variedade de fontes de dados, incluindo bancos de dados, data lakes e streams em tempo real.
- Oferece integração perfeita com serviços Azure, simplificando a importação de dados nativos da Azure.
- Permite o processamento ETL avançado para transformar e limpar dados.
- Escalabilidade para lidar com grandes volumes de dados.
- Automatiza trabalhos de ingestão e fornece segurança, conformidade e monitoramento robustos.

Esses recursos ajudam as organizações a importar, preparar e analisar dados de forma eficiente, possibilitando a tomada de decisões informadas e a construção de soluções de análise de dados avançadas.





## 3.3 Modelagem

### 3.3.1. Catálogo Consolidado do Balanço Patrimonial e a Demonstração Financeira das empresas "Junior Oils" listada na B3(Brasil, Bolsa e Balcão).

#### Introdução

Essas empresas juniores de petróleo desempenham um papel fundamental no desenvolvimento do setor de energia no Brasil, contribuindo para a produção doméstica de petróleo e gás natural, além de promoverem investimentos e inovação no setor. Seus esforços são cruciais para atender à crescente demanda energética do país e para a economia brasileira como um todo.

Este catálogo apresenta uma estrutura detalhada das principais contas associadas as demonstrações das empresas.

#### Contas das demonstrações

Para este projeto temos contas associadas ao ano fiscal de 2022, como se observa:

##### Contas de Ativo:

- Caixa e Equivalentes de Caixa
- Aplicações Financeiras
- Ativo Circulante
- Ativo Total

##### Contas de Passivo:

- Passivo Circulante
- Empréstimos e Financiamentos
- Patrimônio Líquido Consolidado

##### Contas de resultado:

- Receita de Venda de Bens e/ou Serviços
- Custo dos Bens e/ou Serviços Vendidos
- Despesas com Vendas
- Despesas Gerais e Administrativas
- Resultado Bruto
- Lucro/Prejuízo Consolidado do Período
- Participação dos Acionistas Não Controladores
- Atribuído a Sócios da Empresa Controladora

## Estrutura das Demonstrações

Coluna	Tipo	Comentário
CNPJ_CIA	string	CNPJ da Companhia
DT_REFER	date	Data de Referência
DENOM_CIA	string	Denominação da Companhia
CD_CVM	bigint	Código CVM
GRUPO_DFP	string	Grupo de DFP
MOEDA	string	Moeda
ESCALA_MOEDA	string	Escala da Moeda
ORDEM_EXERC	string	Ordem do Exercício
DT_FIM_EXERC	date	Data de Fim do Exercício
CD_CONTA	bigint	Código da Conta
DS_CONTA	string	Descrição da Conta
VL_CONTA	double	Valor da Conta
ST_CONTA_FIXA	string	Status da Conta Fixa

### 3.3.1.1 Catálogo Consolidado do Balanço Patrimonial do Ativo

Coluna	Tipo	Comentário	Etiquetas
CNPJ_CIA	string	CNPJ (Cadastro Nacional da Pessoa Jurídica)	🔍
DT_REFER	date	Data de referência para os dados apresentados	🔍
VERSAO	bigint	Versão dos dados	🔍
DENOM_CIA	string	Denominação ou nome da companhia	🔍
CD_CVM	bigint	Código da CVM (Comissão de Valores Mobiliários) da empresa	🔍
GRUPO_DFP	string	Grupo de demonstrações financeiras ao qual os dados pertencem	🔍
MOEDA	string	Moeda na qual os valores da conta estão denominados	🔍
ESCALA_MOEDA	string	Escala da moeda usada	🔍
ORDEM_EXERC	string	Ordem do exercício financeiro	🔍
DT_FIM_EXERC	date	Ordem do exercício financeiro	🔍
CD_CONTA	string	Código da conta	🔍
DS_CONTA	string	Descrição da conta	🔍
VL_CONTA	double	Valor da conta	🔍
ST_CONTA_FIXA	string	Status da conta fixa	🔍

Microsoft Azure | databricks | Pesquisador dados, notebooks, recentes e muito mais... CTRL + P

dbw-dev-dtc | welton\_souza@p... | XS

### Explorador de catálogos

dt0028\_dev > default > dt0028\_dev.default.dfp\_cia\_aberta\_bpa\_con\_2022

Proprietário: welton\_souza@... Popularidade: 0 Tamanho: 282.6KB, 1 ficheiro Última atualização:

Etiquetas: Adicionar etiquetas

Created by the file upload UI

Colunas Dados de exemplo Detalhes Permissões Histórico Linhagem Informações

Type	MANAGED
Storage Location	abfsc://unity-catalog@sthdtpbdlkc.dfs.core.windows.net/d142e1fb-3fac-44a8-a2dc-08218f92afc5/tables/0451a7ef-87da-4440-b95f-4c32e695a58f
Properties	delta.lastCommitTimestamp=169586020000 delta.lastUpdateVersion=20 delta.minWriterVersion=2 delta.minReaderVersion=1 delta.checkpoint.writeStatsAsJson=true delta.checkpoint.writeStatsAsJson=false
Securable Kind	TABLE_DELTA
Generation	20
Metastore Id	d142e1fb-3fac-44a8-a2dc-08218f92afc5
Created At	15/09/2023, 21:04:37
Created By	welton_souza@...
Updated At	27/09/2023, 21:16:44
Updated By	welton_souza@...
Table Id	0451a7ef-87da-4440-b95f-4c32e695a58f
Delta Runtime Properties Kopairs	(vazio)
Securable Type	TABLE
Provisioning Info	state=ACTIVE

### 3.3.1.2 Catálogo Consolidado do Balanço Patrimonial do Passivo

Microsoft Azure | databricks | Pesquisador dados, notebooks, recentes e muito mais... CTRL + P

dbw-dev-dtc | welton\_souza@p... | XS

### Explorador de catálogos

dt0028\_dev > default > dt0028\_dev.default.dfp\_cia\_aberta\_bpp\_con\_2022

Proprietário: welton\_souza@... Popularidade: 0 Tamanho: 282.6KB, 1 ficheiro Última atualização:

Etiquetas: Adicionar etiquetas

Created by the file upload UI

Colunas Dados de exemplo Detalhes Permissões Histórico Linhagem Informações

Filtrar colunas...

Coluna	Tipo	Comentário	Etiquetas
CNPJ_CIA	string	CNPJ (Cadastro Nacional da Pessoa Jurídica)	🔗
DT_REFER	date	Data de referência para os dados apresentados	🔗
VERSAO	bigint	Versão dos dados	🔗
DENOM_CIA	string	Denominação ou nome da companhia	🔗
CD_CVM	bigint	Código da CVM (Comissão de Valores Mobiliários) da empresa	🔗
GRUPO_DFP	string	Grupo de demonstrações financeiras ao qual os dados pertencem	🔗
MOEDA	string	Moeda na qual os valores da conta estão denominados	🔗
ESCALA_MOEDA	string	Escala da moeda usada	🔗
ORDEN_EXERC	string	Ordem do exercício financeiro	🔗
DT_FIM_EXERC	date	Ordem do exercício financeiro	🔗
CD_CONTA	string	Código da conta	🔗
DS_CONTA	string	Descrição da conta	🔗
VL_CONTA	double	Valor da conta	🔗
ST_CONTA_FIXA	string	Status da conta fixa	🔗

1



Microsoft Azure databricks

Explorador de catálogos

Catálogos > dt0028\_dev > default > dt0028\_dev.default.dfp\_cia\_aberta\_bpp\_con\_2022

Proprietário: welton\_souza@ Popularidade: Tamanho: 399.7KiB, 1 ficheiro Última atualização:

Etiquetas: Adicionar etiquetas

Created by the file upload UI

Colunas Dados de exemplo Detalhes Permissões Histórico Linhagem Informações

Type	MANAGED
Storage Location	abfs://unity-catalog@sthdtpdtkc.dfs.core.windows.net/d142e1fb-3fac-44a8-a2dc-08218f92a1c5/tables/f633679-5ade-43c4-bd79-9f8c5da7404a
Properties	delta.lastCommitTimestamp=1694822895000 delta.lastUpdateVersion=0 delta.minWriterVersion=2 delta.minReaderVersion=1 delta.checkpoint.writeStatusAsStruct=true delta.checkpoint.writeStatusAsIcon=false
Securable Kind	TABLE_DELTA
Metastore Id	d142e1fb-3fac-44a8-a2dc-08218f92a1c5
Created At	15/09/2023, 21:08:18
Created By	welton_souza@
Updated At	15/09/2023, 21:08:18
Updated By	welton_souza@
Table Id	f633679-5ade-43c4-bd79-9f8c5da7404a
Delta Runtime Properties Kyspairs	(vazio)
Securable Type	TABLE
Provisioning Info	state=ACTIVE

### 3.3.1.3 Catálogo Consolidado da Demonstração do Resultado

Explorador de catálogos

Catálogos > dt0028\_dev > default > dt0028\_dev.default.dfp\_cia\_aberta\_dre\_con\_2022

Proprietário: welton\_souza@ Popularidade: Tamanho: 282.6KiB, 1 ficheiro Última atualização:

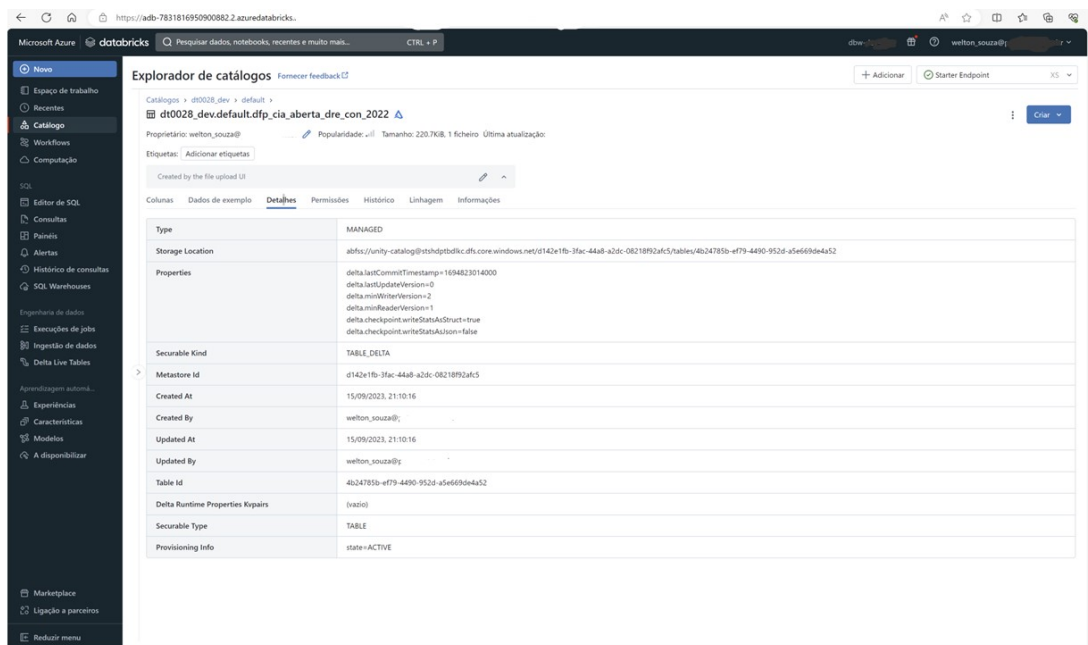
Etiquetas: Adicionar etiquetas

Created by the file upload UI

Colunas Dados de exemplo Detalhes Permissões Histórico Linhagem Informações

Filtrar colunas...

Coluna	Tipo	Comentário	Etiquetas
CNPJ_CIA	string	CNPJ (Cadastro Nacional da Pessoa Jurídica)	
DT_REFER	date	Data de referência para os dados apresentados	
VERSAO	bigint	Versão dos dados	
DENOM_CIA	string	Denominação ou nome da companhia	
CD_CVM	bigint	Código da CVM (Comissão de Valores Mobiliários) da empresa	
GRUPO_DFP	string	Grupo de demonstrações financeiras ao qual os dados pertencem	
MOEDA	string	Moeda na qual os valores da conta estão denominados	
ESCALA_MOEDA	string	Escala da moeda usada	
ORDEN_EXERC	string	Ordem do exercício financeiro	
DT_FIM_EXERC	date	Ordem do exercício financeiro	
CD_CONTA	string	Código da conta	
DS_CONTA	string	Descrição da conta	
VL_CONTA	double	Valor da conta	
ST_CONTA_FIXA	string	Status da conta fixa	



Este catálogo proporciona uma visão minuciosa e estruturada dos dados referentes ao ano de 2022. Essas informações são de vital importância para conduzir análises financeiras e avaliar o desempenho das empresas durante esse período. Com o suporte do ambiente de processamento de big data Spark da Azure Databricks, bem como a utilização de bibliotecas Python, como o Pandas.

Essa combinação de tecnologia avançada e bibliotecas de análise de dados nos permite aproveitar ao máximo os dados disponíveis, transformando-os em informações valiosas que podem orientar estratégias e impulsionar o sucesso das decisões

### 3.4. Carga

#### ETL (Extract, Transform, Load)

Com Azure Databricks, os usuários podem criar fluxos de trabalho de ETL complexos e escaláveis para transformar e mover dados entre diferentes sistemas de armazenamento.

#### Carga da tabela Balanço patrimonial e Demonstração de resultado 2022

Importa a classe SparkSession que é a entrada para a funcionalidade do Spark, usada principalmente para criar DataFrames. Inicializa uma nova sessão Spark ou recupera uma existente. Consulta todos os registros das tabelas no catálogo e armazena o resultado no DataFrame Spark df\_spark.

#### Vantagens:

1. **Escalabilidade:** Baseado em Apache Spark, aproveita a capacidade de processar grandes volumes de dados em paralelo.
2. **Flexibilidade:** Suporta múltiplas linguagens, como Scala, Python, SQL e R.
3. **Integração com BI Tools:** Pode ser facilmente conectado a ferramentas de BI, como Power BI, para visualização de dados.

4. **Conectividade Nativa:** Conecta-se naturalmente a muitos sistemas de armazenamento de dados no Azure, simplificando o acesso e movimentação de dados.

Azure Databricks oferece uma plataforma unificada e colaborativa para processamento e análise de grandes conjuntos de dados. Seja para ETL, aprendizado de máquina ou análise avançada, ele é uma opção robusta e integrada para organizações que desejam aproveitar o poder do Apache Spark dentro do ecossistema Azure.

Fonte: <https://learn.microsoft.com/pt-br/azure/databricks/>

### 3.4.1 Extração

#### 3.4.1.1 Instalação das bibliotecas

A instalação de bibliotecas Python é essencial, uma vez que por meio do Python realizaremos as análises de dados. Essas bibliotecas fornecerão as ferramentas e funcionalidades necessárias para explorar, visualizar e processar os dados de maneira eficaz, permitindo-nos extrair insights valiosos e tomar decisões informadas.

```
In [0]: # Para suprimir as mensagens de saída completamente no ambiente
!pip install pandas pandas-datareader numpy requests==2.28.1 plotly yahoo_fin yfinance

# Para suprimir as mensagens de saída completamente no ambiente Windows
!pip install pandas pandas-datareader numpy requests plotly yahoo_fin yfinance vect

[notice] A new release of pip available: 22.2.2 -> 23.2.1
[notice] To update, run: pip install --upgrade pip

[notice] A new release of pip available: 22.2.2 -> 23.2.1
[notice] To update, run: pip install --upgrade pip
```

#### 3.4.1.2 Importação das bibliotecas

A importação das bibliotecas Python é uma etapa fundamental, pois por meio dela habilitamos as ferramentas e funcionalidades necessárias para conduzir nossas análises de dados. Essas bibliotecas são como as 'ferramentas de trabalho' que nos permitem manipular, processar e explorar os dados de forma eficaz, capacitando-nos a realizar análises detalhadas e tomar decisões bem fundamentadas.

```
In [0]: #Importação de bibliotecas

import io
import pandas as pd
from pandas import set_option
from pandas.plotting import scatter_matrix
from datetime import datetime
import numpy as np
import requests
import time
import plotly
import os
import plotly.graph_objects as go
import fundamentus
import pandas_datareader as pdr
from yahoo_fin import stock_info as si
import yfinance as yf
```

```

import vectorbt as vbt
import matplotlib as mpl
from matplotlib.patches import Patch, Polygon
from matplotlib.lines import Line2D
from IPython.display import display, Markdown
import matplotlib.pyplot as plt
from matplotlib import cm
import seaborn as sns
import missingno as ms # para tratamento de missings
# Configuração para não exibir os warnings
import warnings
warnings.filterwarnings("ignore")
import warnings
from pyspark.sql import SparkSession
from pyspark.sql import SparkSession
from pyspark.sql import SparkSession

```

### 3.4.1.2 Sessão Spark para Pandas

Converte o DataFrame do Spark para um DataFrame Pandas, que é geralmente mais amigável para manipulações não distribuídas.

- **Streaming em Tempo Real:** Suporta processamento de dados em tempo real através do Spark Streaming.

E o uso da biblioteca Pandas do Python no Azure Databricks oferece um ambiente poderoso e flexível para análise de big data e machine learning na nuvem. A combinação dessas tecnologias permite que as organizações aproveitem ao máximo seus dados, realizem análises avançadas e desenvolvam soluções de inteligência artificial de alto desempenho, gerenciado para cientistas de dados, engenheiros de dados e analistas trabalharem com grandes volumes de dados de maneira eficiente.

```

In [0]: # Importe as bibliotecas necessárias
#from pyspark.sql import SparkSession

# Crie uma sessão Spark
spark = SparkSession.builder.getOrCreate()
df_spark = spark.sql("SELECT * FROM `dt0028_dev`.`default`.`dfp_cia_aberta_bpa_con_

df_demonstrativo_BPA = df_spark.toPandas()

df_demonstrativo_BPA.head(10)

```

Out[0]:

	CNPJ_CIA	DT_REFER	VERSAO	DENOM_CIA	CD_CVM	GRUPO_DFP	MOEDA	ESCALA_M
0	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	
1	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	
2	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	
3	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	
4	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	
5	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	
6	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	
7	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	
8	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	
9	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Ativo	REAL	



```
In [0]: # Importe as bibliotecas necessárias
        #from pyspark.sql import SparkSession

        # Crie uma sessão Spark
        spark1 = SparkSession.builder.getOrCreate()
        df_spark1 = spark1.sql("SELECT * FROM `dt0028_dev`.`default`.`dfp_cia_aberta_bpp_cc")

        df_demonstrativo_BPP = df_spark1.toPandas()

        df_demonstrativo_BPP
```

Out[0]:

	CNPJ_CIA	DT_REFER	VERSAO	DENOM_CIA	CD_CVM	GRUPO_DFP	MOEDA	ESCA
<b>0</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Passivo	REAL	
<b>1</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Passivo	REAL	
<b>2</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Passivo	REAL	
<b>3</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Passivo	REAL	
<b>4</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Balanço Patrimonial Passivo	REAL	
...	...	...	...	...	...	...	...	...
<b>100753</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Balanço Patrimonial Passivo	REAL	
<b>100754</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Balanço Patrimonial Passivo	REAL	
<b>100755</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Balanço Patrimonial Passivo	REAL	
<b>100756</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Balanço Patrimonial Passivo	REAL	
<b>100757</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Balanço Patrimonial Passivo	REAL	

100758 rows × 14 columns

```
In [0]: # Importe as bibliotecas necessárias
        #from pyspark.sql import SparkSession

        # Crie uma sessão Spark
        spark2 = SparkSession.builder.getOrCreate()
        df_spark2 = spark2.sql("SELECT * FROM `dt0028_dev`.`default`.`dfp_cia_aberta_dre_cc")

        df_demonstrativo_DRE = df_spark2.toPandas()

        df_demonstrativo_DRE
```



Out[0]:

	CNPJ_CIA	DT_REFER	VERSAO	DENOM_CIA	CD_CVM	GRUPO_DFP	MOEDA	ESCA
<b>0</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Demonstração do Resultado	REAL	
<b>1</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Demonstração do Resultado	REAL	
<b>2</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Demonstração do Resultado	REAL	
<b>3</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Demonstração do Resultado	REAL	
<b>4</b>	00.000.000/0001-91	2022-12-31	1	BCO BRASIL S.A.	1023	DF Consolidado - Demonstração do Resultado	REAL	
...	...	...	...	...	...	...	...	...
<b>31239</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Demonstração do Resultado	REAL	
<b>31240</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Demonstração do Resultado	REAL	
<b>31241</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Demonstração do Resultado	REAL	
<b>31242</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Demonstração do Resultado	REAL	
<b>31243</b>	97.837.181/0001-47	2022-12-31	1	DEXCO S.A.	21091	DF Consolidado - Demonstração do Resultado	REAL	

31244 rows × 15 columns



### 3.4.2 Transformação

A transformação do processo ETL (Extração, Transformação e Carga) no Databricks é uma decisão estratégica valiosa para lidar com grandes volumes de dados. O Databricks é uma plataforma de análise de big data baseada no Apache Spark que oferece várias vantagens

para a transformação de dados em comparação com abordagens tradicionais de ETL, tais como:

**Escalabilidade e Desempenho Aprimorados:** O Databricks utiliza o Apache Spark, que é altamente escalável e pode processar grandes quantidades de dados em paralelo. Isso melhora significativamente o desempenho das transformações ETL, tornando possível lidar com dados em tempo real ou em lote.

**Flexibilidade:** O Databricks suporta várias linguagens de programação, como Python, Scala e SQL, o que dá aos engenheiros de dados a flexibilidade de escolher a linguagem mais adequada para suas transformações ETL.

**Integração com Fontes de Dados:** O Databricks pode se integrar facilmente a uma variedade de fontes de dados, incluindo armazenamentos em nuvem, bancos de dados relacionais e NoSQL. Isso facilita a extração de dados de várias fontes para posterior transformação.

**Bibliotecas e Funções Pré-Construídas:** O Databricks fornece bibliotecas e funções pré-construídas para facilitar muitas tarefas comuns de ETL, como limpeza de dados, agregação e enriquecimento. Isso economiza tempo e esforço no desenvolvimento de transformações personalizadas.

**Automação:** O Databricks suporta a automação de tarefas ETL por meio de agendamento e orquestração. Isso permite que as transformações sejam executadas em horários específicos ou em resposta a eventos, tornando o processo mais eficiente.

**Colaboração e Monitoramento:** A plataforma Databricks oferece recursos de colaboração para equipes de engenharia de dados trabalharem juntas em projetos ETL. Além disso, fornece ferramentas de monitoramento e geração de relatórios para rastrear o desempenho das transformações.

**Segurança e Conformidade:** O Databricks inclui recursos avançados de segurança e conformidade, ajudando as organizações a manter a integridade e a privacidade dos dados durante o processo ETL.

**Custo:** Embora o Databricks seja uma plataforma poderosa, é importante observar que o custo de uso pode ser significativo, especialmente para grandes volumes de dados. As organizações devem avaliar cuidadosamente os custos em relação aos benefícios antes de fazer a migração.

```
In [0]: # Suponhamos que você tenha três DataFrames: df_demonstrativo_BPA, df_demonstrativo_BPP, df_demonstrativo_DRE

# Concatenar os DataFrames em um único DataFrame
dfs = [df_demonstrativo_BPA, df_demonstrativo_BPP, df_demonstrativo_DRE]
df_concatenado = pd.concat(dfs)

# Converter os códigos desejados para strings
codigos_desejados = ['25291', '22365', '22187', '25780']

# Converter a data '2022-01-01' em um objeto de data
data_limite = datetime.strptime('2022-01-01', '%Y-%m-%d').date()

# Filtrar os valores da coluna 'CD_CVM' com os códigos desejados, data '2022-01-01'
df_filtrado = df_concatenado[df_concatenado['CD_CVM'].astype(str).isin(codigos_desejados) & df_concatenado['data'] <= data_limite]
```

```
# Preparar a demonstração do resultado do exercício
dre = df_filtrado[df_filtrado['GRUPO_DFP'] == 'DF Consolidado - Demonstração do Res

# Preparar o balanço patrimonial do ativo
bpa = df_filtrado[df_filtrado['GRUPO_DFP'] == 'DF Consolidado - Balanço Patrimonial

# Preparar o balanço patrimonial do passivo
bpp = df_filtrado[df_filtrado['GRUPO_DFP'] == 'DF Consolidado - Balanço Patrimonial
```

```
In [0]: # Tabela dinâmica dos objetos selecionados da DRE
df_dre = pd.pivot_table(dre, index=['DENOM_CIA', 'DS_CONTA', 'CD_CONTA'], columns=[

# Tabela dinâmica dos objetos selecionados da BPA
df_bpa = pd.pivot_table(bpa, index=['DENOM_CIA', 'DS_CONTA', 'CD_CONTA'], columns=[

# Tabela dinâmica dos objetos selecionados da BPP
df_bpp = pd.pivot_table(bpp, index=['DENOM_CIA', 'DS_CONTA', 'CD_CONTA'], columns=[
```

### 3.4.3 Carga

A etapa de carga (Load) no processo ETL (Extração, Transformação e Carga) desempenha um papel crítico na movimentação dos dados transformados para o destino final:

**Eficiência na Carga:** A eficiência na etapa de carga é fundamental para garantir que os dados sejam movidos de forma rápida e precisa para o destino final. Isso envolve otimizar o processo de carga para minimizar o tempo de inatividade e manter os dados atualizados.

**Integridade dos Dados:** Durante a etapa de carga, é importante garantir a integridade dos dados. Isso inclui a validação dos dados transformados para garantir que eles estejam corretos e atendam aos padrões de qualidade estabelecidos.

**Controle de Versões:** É importante ter um mecanismo de controle de versões para gerenciar as alterações nos esquemas de dados ao longo do tempo. Isso é especialmente relevante quando se lida com grandes volumes de dados e muitas transformações ETL ao longo do tempo.

**Monitoramento e Relatórios:** A etapa de carga deve ser monitorada de perto para identificar quaisquer problemas ou falhas que possam ocorrer. Isso inclui o monitoramento de métricas de desempenho, como tempo de execução, taxa de sucesso e erros.

**Escalabilidade:** À medida que a quantidade de dados cresce, a capacidade de escalabilidade na etapa de carga se torna crucial. Deve ser possível aumentar os recursos de hardware ou a infraestrutura de forma eficiente para lidar com volumes crescentes de dados.

**Segurança e Conformidade:** A etapa de carga deve ser protegida adequadamente para garantir a segurança dos dados sensíveis. Isso inclui criptografia, autenticação e autorização adequadas. Além disso, é importante garantir a conformidade com regulamentos de privacidade e segurança de dados, como o GDPR ou o HIPAA, quando aplicável.

**Agendamento e Orquestração:** A carga de dados geralmente é agendada para ocorrer em momentos específicos, e a orquestração dessas cargas pode ser feita por meio de ferramentas de agendamento ETL ou sistemas de orquestração de fluxo de trabalho.

Recuperação de Falhas: É importante ter um plano de recuperação de falhas para lidar com interrupções inesperadas durante a etapa de carga. Isso pode incluir backups regulares dos dados e a capacidade de retomar a carga a partir do ponto de falha.

Logging e Auditoria: Manter registros detalhados de todas as operações de carga é crucial para a auditoria e a solução de problemas. Os registros podem ajudar a rastrear quem acessou os dados, quando e quais operações foram realizadas.

Em resumo, a etapa de carga no processo ETL é crítica para o sucesso de qualquer projeto de análise de dados. Ela envolve não apenas a transferência de dados, mas também a garantia de que os dados estejam corretos, seguros e disponíveis para análise. A escolha das ferramentas e tecnologias adequadas, juntamente com as melhores práticas de ETL, é essencial para garantir uma carga de dados eficiente e confiável.

#### 3.4.3.1 Carga do balanço patrimonial ativo

```
In [0]: df_bpa
```

			VL_CONTA
			2022-12-31
DENOM_CIA	DS_CONTA	CD_CONTA	
3R PETROLEUM ÓLEO E GÁS S.A.	Adiantamento para cessão de blocos	1.02.02.02.01	593549.0
	Adiantamentos	1.01.08.03.01	193011.0
	Aplicações Financeiras	1.01.02	31353.0
	Aplicações Financeiras Avaliadas a Valor Justo através de Outros Resultados Abrangentes	1.01.02.02	0.0
		1.02.01.02	0.0
...	...	...	...
PETRORECÔNCAVO S.A.	Tributos Diferidos	1.02.01.07	167840.0
	Tributos a Recuperar	1.01.06	99243.0
	Títulos Designados a Valor Justo	1.01.02.01.02	0.0
		1.02.01.01.01	0.0
	Títulos para Negociação	1.01.02.01.01	0.0

269 rows × 4 columns

#### 3.4.3.2 Carga do balanço patrimonial passivo

```
In [0]: df_bpp
```

Out[0]:

			VL_CONTA
			2022-12-31
DENOM_CIA	DS_CONTA	DT_FIM_EXERC	CD_CONTA
3R PETROLEUM ÓLEO E GÁS S.A.	Adiantamento para Futuro Aumento de Capital	2.02.02.02.02	0.0
		2.03.02.06	0.0
	Ajuste acumulado de conversão	2.03.02.07	106383.0
	Ajustes Acumulados de Conversão	2.03.07	0.0
	Ajustes de Avaliação Patrimonial	2.03.06	0.0
...	...	...	...
PETRORECÔNCAVO S.A.	Valor a pagar por Aquisições	2.01.05.02.07	405886.0
	Valores a Pagar de Arrendamentos	2.01.05.02.04	20382.0
		2.02.02.02.03	7112.0
	Valores a pagar por aquisições	2.02.02.02.05	512386.0
	Ágio na Emissão de Ações	2.03.02.01	0.0

475 rows × 1 columns

3.4.3.3 Carga da demonstração do resultado

```
In [0]: df_dre
```

Out[0]:

			VL_CONTA
			DT_INI_EXERC
			2022-01-01
			DT_FIM_EXERC
			2022-12-31
DENOM_CIA	DS_CONTA	CD_CONTA	
3R PETROLEUM ÓLEO E GÁS S.A.	(Perda) / reversão no valor recuperável de ativos	3.04.04.02	-123318.0
		3.04.05.04	0.0
	Atribuído a Sócios Não Controladores	3.11.02	-21853.0
	Atribuído a Sócios da Empresa Controladora	3.11.01	149569.0
	Corrente	3.08.01	-71675.0
...	...	...	...
PETRORECÔNCAVO S.A.	Resultado Líquido das Operações Continuadas	3.09	1153391.0
	Resultado Líquido de Operações Descontinuadas	3.10	0.0
	Resultado de Equivalência Patrimonial	3.04.06	0.0
	Variação cambial ativa	3.06.01.01	0.0
	Variação cambial passiva	3.06.02.01	0.0

149 rows × 4 columns

## 3.5 Análise de dados

### 3.5.1 Qualidade de dados

Para análise da qualidade de dados, utilizaremos o dataframe "df\_filtrado" pois é dataframe com a concatenação de todos os dataset que consta no catálogo já com os devido filtros necessários para análise de dados.

In [0]: `df_filtrado.info()`

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 893 entries, 7741 to 12051
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   CNPJ_CIA              893 non-null   object
1   DT_REFER              893 non-null   object
2   VERSAO                893 non-null   int64
3   DENOM_CIA             893 non-null   object
4   CD_CVM                893 non-null   int64
5   GRUPO_DFP             893 non-null   object
6   MOEDA                 893 non-null   object
7   ESCALA_MOEDA          893 non-null   object
8   ORDEM_EXERC           893 non-null   object
9   DT_FIM_EXERC          893 non-null   object
10  CD_CONTA              893 non-null   object
11  DS_CONTA              893 non-null   object
12  VL_CONTA              893 non-null   float64
13  ST_CONTA_FIXA         893 non-null   object
14  DT_INI_EXERC          149 non-null   object
dtypes: float64(1), int64(2), object(12)
memory usage: 111.6+ KB

```

```

In [0]: # verificando nulls no dataset
df_filtrado.isnull().sum()

```

```

Out[0]: CNPJ_CIA          0
DT_REFER          0
VERSAO            0
DENOM_CIA         0
CD_CVM            0
GRUPO_DFP         0
MOEDA             0
ESCALA_MOEDA      0
ORDEM_EXERC       0
DT_FIM_EXERC      0
CD_CONTA          0
DS_CONTA          0
VL_CONTA          0
ST_CONTA_FIXA     0
DT_INI_EXERC      744
dtype: int64

```

```

In [0]: # recuperando os nomes das colunas
col = list(df_filtrado.columns)

# o novo dataset irá conter todas as colunas com exceção da última (classe)
atributos = df_filtrado[col[0:-1]]

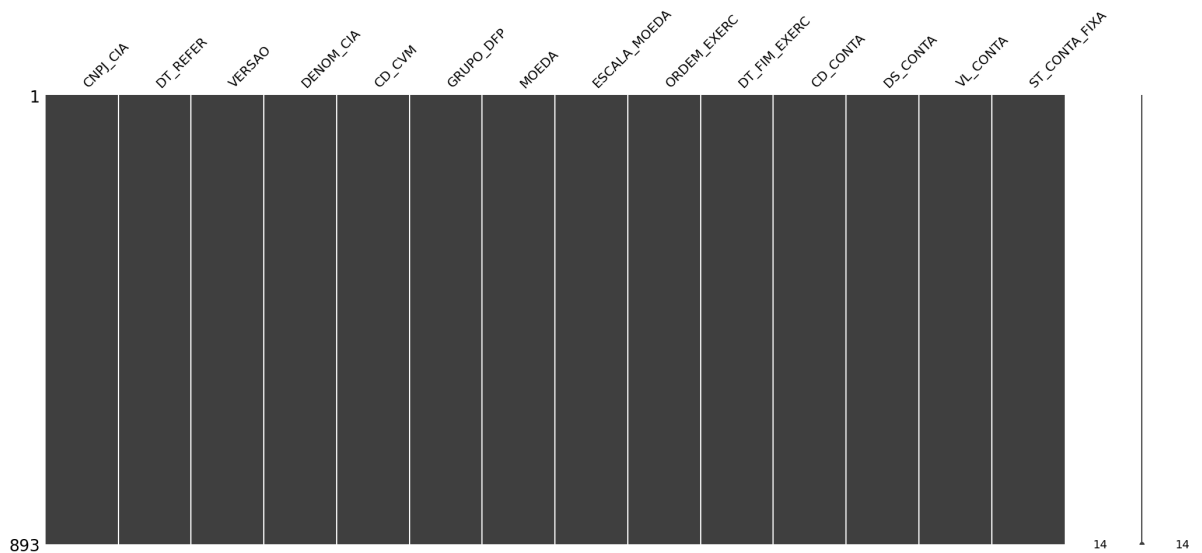
# exibindo visualização matricial da nulidade do dataset
ms.matrix(atributos)

```

```

Out[0]: <AxesSubplot:>

```



Podemos constatar que os dados estão integros e qualificados para análise.

### 3.5.2 Solução do problema

```
In [0]: # Denominação da empresas "Junior Oils" selecionada
lista_de_empresas = dre['DENOM_CIA'].unique()
lista_de_empresas = lista_de_empresas.transpose()
lista_de_empresas = lista_de_empresas.tolist()
lista_de_empresas = sorted(lista_de_empresas)
lista_de_empresas
```

```
Out[0]: ['3R PETROLEUM ÓLEO E GÁS S.A.',
'ENAUTA PARTICIPAÇÕES S.A.',
'PETRO RIO S.A.',
'PETRORECÔNCAVO S.A.']
```

```
In [0]: # 'ticker' é um código formado por 4 letras maiúsculas e 1 número que representam o ativo
# '.SA' representa que o ativo está listado na B3 - A Bolsa de Valores Brasileira
# As letras representam as iniciais do razão social, enquanto que o número "3" representa o setor de petróleo e gás

tickers = ['RRRP3.SA', 'ENAT3.SA', 'PRI03.SA', 'RECV3.SA']
tickers
```

```
Out[0]: ['RRRP3.SA', 'ENAT3.SA', 'PRI03.SA', 'RECV3.SA']
```

```
In [0]: #o número de empresas "Junior Oils"
n_empresas = len(tickers)
n_empresas
```

```
Out[0]: 4
```

```
In [0]: # Define paleta de cores

# # alternativa A: definição manual das cores a partir de lista de cores em hex
coresViridis = ["#440154", "#3b528b", "#21918c", "#5ec962", "#fde725"]

# alternativa B: definição de cores utilizando palette do próprio matplotlib
# a paleta viridis é robusta para pessoas com daltonismo
vcmap = plt.get_cmap("viridis", 6) # um a mais do que o necessário para evitar utilizar a cor branca

# exibe o mapa de cores
display(vcmap)

# converte o mapa de cores em um vetor, para uso de cores individuais
```



```
coresViridis = [mpl.colors.rgb2hex(vcmapi) for i in range(6)] # extrai lista de cores
display(coresViridis)

# define uma cor default
# corDefault = coresViridis[1]

# Carrega uma folha de estilos (opcional - questões estéticas)
plt.style.use("ggplot")

# Modifica mais alguns parâmetros gráficos (opcional - questões estéticas)
plt.rc("axes", facecolor="#fafafa", grid = True)
plt.rc("grid", color = "#f0f0f0")
plt.rc("axes", edgecolor="#cccccc", lw=0.1)
```

viridis



```
[ '#
440
15
4',
'#4
144
8
7',
'#2
a78
8
e',
'#2
2a8
8
4',
'#7
ad1
5
1',
'#f
de7
2
5']
```

### 3.5.2.1 Resposta da 1ª pergunta

#### 3.5.2.1.1 Margem Bruta

Métrica de análise para medir a porcentagem de lucro bruto que a companhia gera com cada serviço prestado ou produto vendido.

*Resultado Bruto / Receita de Vendas de Bens e/ou Serviços*

```
In [0]: margem_bruta = pd.DataFrame()

for i in range(0,n_empresas):
    calculo_margem = pd.Series((df_dre.loc[lista_de_empresas[i],:].loc['Resultado B
    margem_bruta = pd.concat([margem_bruta, calculo_margem], axis=1)

margem_bruta.columns = lista_de_empresas

margem_bruta
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECONCAVO S.A.
0	0.514902	0.493246	0.669001	0.484394

Conforme gráfico abaixo a Petro Rio tem a maior margem bruta entre as 4 "Junior Oil", todavia, a 3R Petroleum que tem a segunda melhor margem bruta, seguido pela Enauta e a 3R.

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

# Plota o gráfico de barras
b1 = ax.barh(y = tickers, width = margem_bruta.iloc[-1], color = coresViridis)

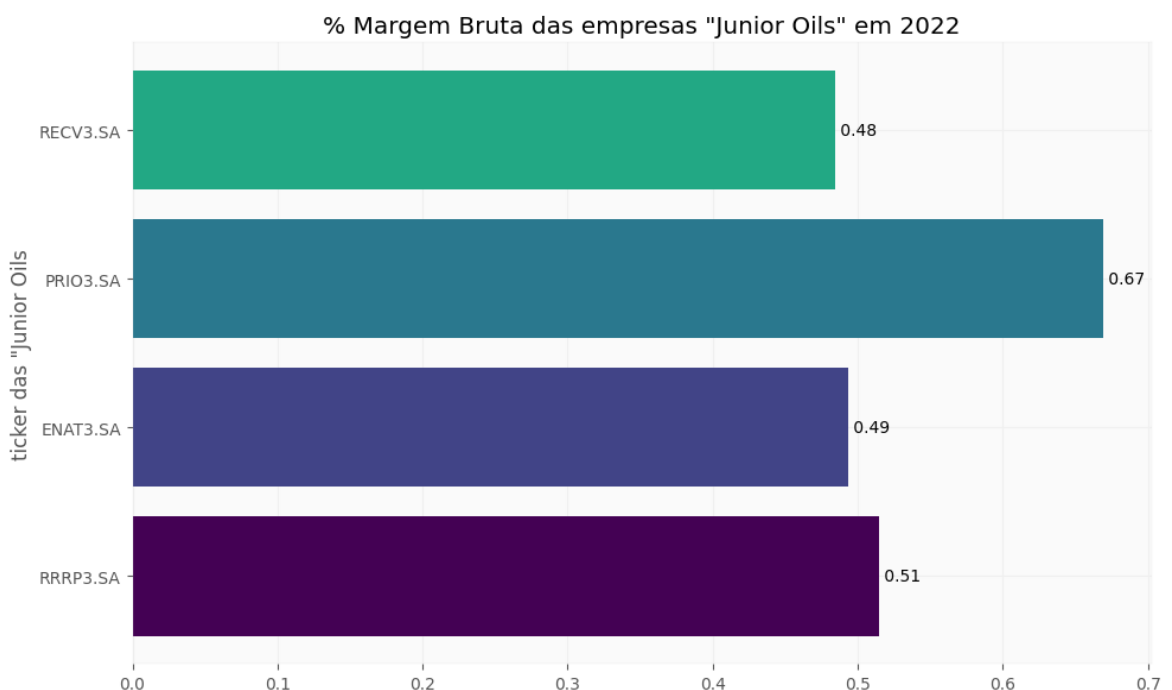
# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
    width = rect.get_width()
    ax.text(
        x = width,
        y = rect.get_y() + rect.get_height() / 2, # posição y
        s = "{:.2f}".format(rect.get_width()), # string com o valor
        ha = "left", # alinhamento na horizontal
        va = "center" # alinhamento na vertical
    )

# Acrescenta um título ao gráfico
plt.title('% Margem Bruta das empresas "Junior Oils" em 2022')

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils"') # é importante esclarecer a unidade e a ord

# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()

# Exibe o gráfico
plt.show()
```



#### 3.5.2.1.2 Margem Líquida

Métrica de análise para medir a porcentagem de lucro líquido alcançado pela organização em relação à sua receita.

```
In [0]: # Criar dataframe chamado margem_liquida
margem_liquida = pd.DataFrame()
#Criar um loop calculando para cada empresa a margem líquida, concatenando todos
for i in range(0,n_empresas):
    calculo_margem_liq = pd.Series((df_dre.loc[lista_de_empresas[i],:].loc['Lucr
    margem_liquida = pd.concat([margem_liquida, calculo_margem_liq], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
margem_liquida.columns = lista_de_empresas
# Exibe o dataframe
margem_liquida
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECONCAVO S.A.
0	0.074149	0.176235	0.538554	0.387572

Conforme gráfico abaixo a Petro Rio tem a maior margem líquida entres as 4 "Junior Oil", consequencia de ter uma margem bruta a maior, porém diferente da PetroReconcavo que tinha a menor margem bruta, demonstra uma boa gestão operacional tendo a segunda maior margem líquida. Todavia a 3R Petroleum que tem a segunda melhor margem bruta, entretanto, tem a pior margem líquida, ou seja, sua gestão operacional impacta negativamente para o resultado da empresa. A Enauta segue a mesma situação da 3R.

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

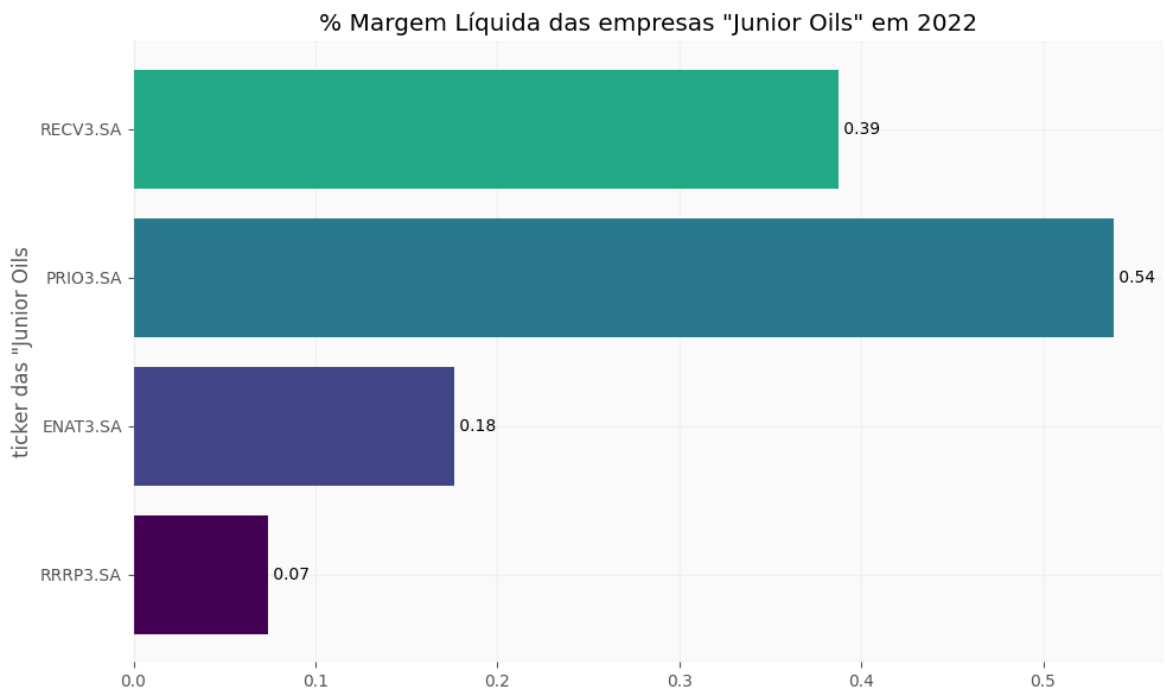
# Plota o gráfico de barras
b1 = ax.barh(y = tickers, width = margem_liquida.iloc[-1], color = coresViridis)

# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
    width = rect.get_width()
    ax.text(
        x = width, # posição x
        y = rect.get_y() + rect.get_height() / 2, # posição y
        s = "{:.2f}".format(rect.get_width()), # string com o valor
        ha = "left", # alinhamento na horizontal
        va = "center" # alinhamento na vertical
    )

# Acrescenta um título ao gráfico
plt.title('% Margem Líquida das empresas "Junior Oils" em 2022')

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils") # é importante esclarecer a unidade e a ord

# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()
```



### 3.5.2.1.3 Margem EBIT

É um indicador que possibilita compreender e avaliar a eficiência operacional de uma empresa. Dessa forma, esse índice é utilizado para realizar comparativos da lucratividade operacionais de companhias de setores análogos

#### *EBIT/Receita Total*

EBIT (Earn Before Interest and Taxes) É um indicador não-contábil que demonstra o lucro de uma empresa antes do reconhecimento das despesas com juros e impostos, também chamado de LAJIR, ou seja, Lucro operacional.

```
In [0]: # Criar dataframe chamado ebit_ajustado
ebit_ajustado = pd.DataFrame()
#Criar um loop calculando para cada empresa o ebit, concatenando todos no mesmo d
for i in range(0, n_empresas):
    ebit_ajustado_ = pd.Series(df_dre.loc[lista_de_empresas[i],:].loc['Receita de V
                                df_dre.loc[lista_de_empresas[i],:].loc['Custo dos Be
                                df_dre.loc[lista_de_empresas[i],:].loc['Despesas com
                                df_dre.loc[lista_de_empresas[i],:].loc['Despesas Ger
    ebit_ajustado = pd.concat([ebit_ajustado, ebit_ajustado_], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
ebit_ajustado.columns=lista_de_empresas
```

```
In [0]: # Criar dataframe chamado margem_ebit
margem_ebit = pd.DataFrame()
#Criar um loop calculando para cada empresa a margem ebit, concatenando todos no
for i in range(0, n_empresas):
    margem_ebit_ = pd.Series((df_dre.loc[lista_de_empresas[i],:].loc['Receita de Ve
                                df_dre.loc[lista_de_empresas[i],:].loc['Custo dos Be
                                df_dre.loc[lista_de_empresas[i],:].loc['Despesas com
                                df_dre.loc[lista_de_empresas[i],:].loc['Despesas Ger
    margem_ebit = pd.concat([margem_ebit, margem_ebit_], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
margem_ebit.columns = lista_de_empresas
# Exibe o dataframe
margem_ebit
```

Out[0]:	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	0.318084	0.410858	0.661658	0.447304

Petro Rio tem a maior capacidade de gerar lucro as 4 "Junior Oil", Ao contrário da 3R Petroleum que tem a menor capacidade de gerar lucro, conforme gráfico abaixo:

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

# Plota o gráfico de barras
b1 = ax.bar(x = tickers, height = margem_ebit.iloc[-1], color = coresViridis)

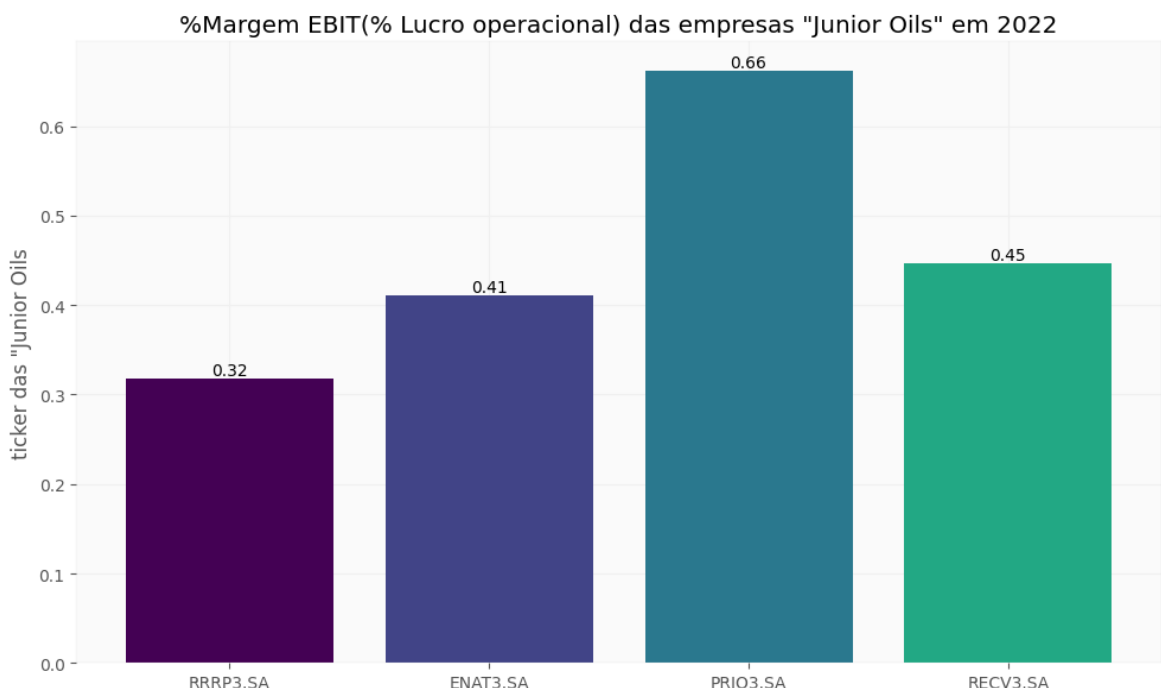
# Acrescenta os valores sobre as barras
ax.bar_label(ax.containers[0], fmt="%0.2f") # formatação do número com uma casa de

# Acrescenta um título ao gráfico
plt.title('%Margem EBIT(% Lucro operacional) das empresas "Junior Oils" em 2022')

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils"') # é importante esclarecer a unidade e a ord

# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()

# Exibe o gráfico
plt.show()
```



## Conclusão

Em termos de margem bruta, a Petro Rio teve a maior margem bruta em 2022, tornando-se uma boa opção de investimento em relação a esse indicador.

Em termos de margem líquida, a Petro Rio também teve a maior margem líquida, o que indica que a empresa teve um bom desempenho na conversão das receitas em lucro

líquido.

Quanto à margem EBIT, a Enauta liderou com a maior margem EBIT, o que sugere uma forte rentabilidade operacional.

Com base nessas métricas, a Petro Rio parece ter tido o melhor desempenho geral em 2022, com margens bruta e líquida fortes. No entanto, a Enauta também apresentou uma margem EBIT muito alta, o que a torna uma opção atraente em termos de rentabilidade operacional.

### 3.5.2.2 Resposta da 2ª pergunta

#### 3.5.2.2.1 Disponibilidades

Representa o saldo de caixa livre de uma empresa,

```
In [0]: # Criar dataframe chamado caixa_ajustado
caixa_ajustado = pd.DataFrame()
#Criar um loop calculando para cada empresa a disponibilidade, concatenando todos
for i in range(0, n_empresas):
    caixa_aj = pd.Series(df_bpa.loc[lista_de_empresas[i],:].loc['Caixa e Equivalent
    caixa_ajustado = pd.concat([caixa_ajustado, caixa_aj], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
caixa_ajustado.columns=lista_de_empresas
# Exibe o dataframe
caixa_ajustado
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	831795.0	2431794.0	9612961.0	1611191.0

Podemos observar que a disponibilidade da Petro Rio reflete uma boa saúde financeira, enquanto a 3R Petroleum tem uma disponibilidade positiva menor entre as empresas.

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com Largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

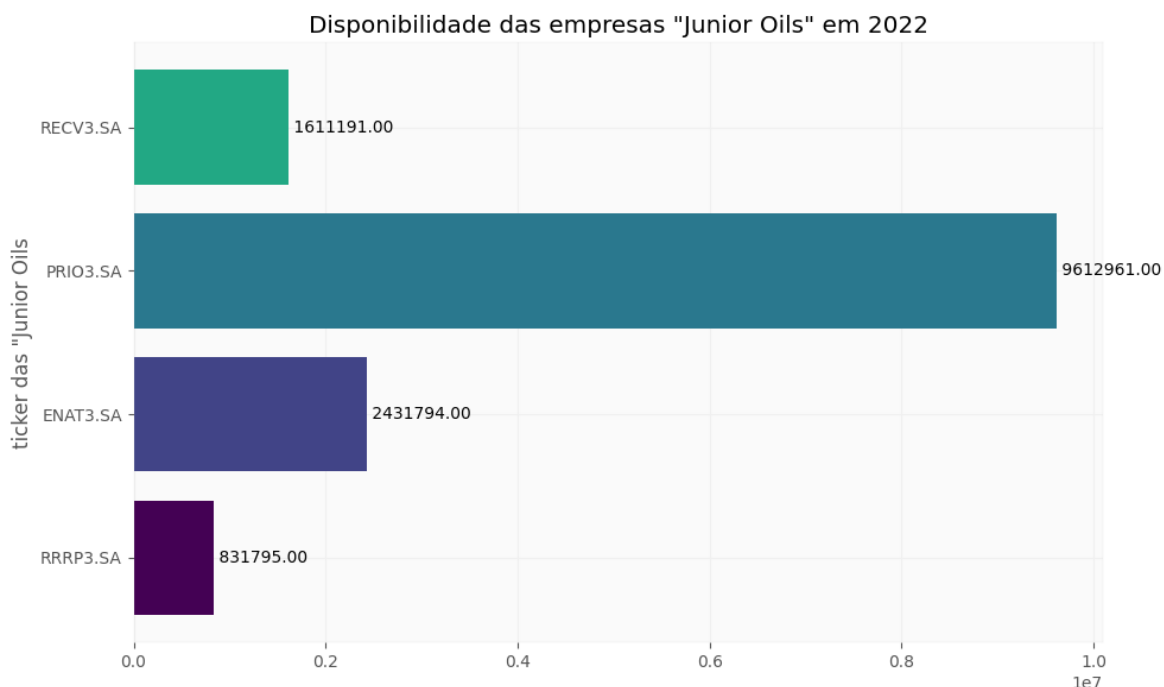
# Plota o gráfico de barras
b1 = ax.barh(y = tickers, width = caixa_ajustado.iloc[-1], color = coresViridis)

# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
    width = rect.get_width()
    ax.text(
        x = width, # posição x
        y = rect.get_y() + rect.get_height() / 2, # posição y
        s = "{:.2f}".format(rect.get_width()), # string com o valor
        ha = "left", # alinhamento na horizontal
        va = "center" # alinhamento na vertical
    )

# Acrescenta um título ao gráfico
plt.title('Disponibilidade das empresas "Junior Oils" em 2022')

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils") # é importante esclarecer a unidade e a ord
```

```
# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()
```



### 3.5.2.2.2 Liquidez Corrente (Ativo Circulante / Passivo Circulante)

Indica a capacidade da companhia em honrar suas obrigações no curto prazo.

```
In [0]: # Criar dataframe chamado liquidez_corrente
liquidez_corrente = pd.DataFrame()
#Criar um loop calculando para cada empresa a liquidez corrente, concatenando tod
for i in range(0, n_empresas):
    liq_corrente = pd.Series((df_bpa.loc[lista_de_empresas[i],:].loc['Ativo Circulan
    liquidez_corrente = pd.concat([liquidez_corrente, liq_corrente], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
liquidez_corrente.columns = lista_de_empresas
# Exibe o dataframe
liquidez_corrente
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	1.761892	1.721665	7.640217	1.53274

A liquidez corrente das empresas demonstram boas capacidades de arcar com suas obrigações e dívidas de curta prazo, ou seja, todas empresas tem um boa saúde financeira e em destaque a Petro Rio que além da ótima saúde financeira, reflete capacidade de fazer maiores investimentos que em conjuntos com outras informações pode confirmar as análises.

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com Largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

# Plota o gráfico de barras
b1 = ax.barh(y = tickers, width = liquidez_corrente.iloc[-1], color = coresViridi

# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
```

```

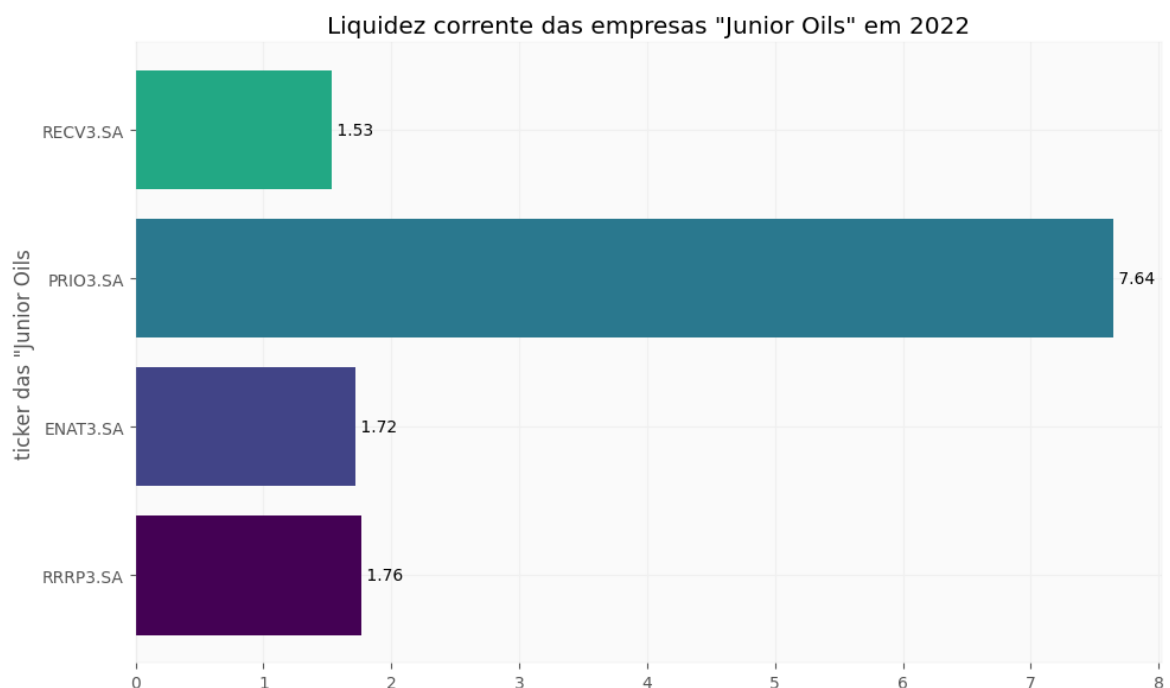
width = rect.get_width()
ax.text(
    x = width,
    y = rect.get_y() + rect.get_height() / 2,
    s = "{:.2f}".format(rect.get_width()),
    ha = "left",
    va = "center"
)

# Acrescenta um título ao gráfico
plt.title('Liquidez corrente das empresas "Junior Oils" em 2022')

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils"') # é importante esclarecer a unidade e a ord

# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()

```



## Conclusão

Em termos de disponibilidade financeira, a Petro Rio possui a maior quantidade de recursos disponíveis em 2022, o que indica uma posição financeira forte em termos de caixa e equivalentes de caixa.

Em relação à liquidez corrente, a Petro Rio também lidera com uma liquidez corrente de 7,64. Isso significa que a empresa possui ativos de curto prazo suficientes para cobrir suas obrigações de curto prazo em 7,64 vezes.

Com base nessas métricas, a Petro Rio parece ser a empresa com a melhor capacidade financeira em 2022 entre as empresas Oil junior, sugerindo uma posição financeira sólida.

### 3.5.2.3 Resposta da 3ª pergunta

#### 3.5.2.3.1 Dívida Bruta/Patrimônio líquido



O indicador de dívida bruta/patrimônio líquido avalia o total de dívidas que o negócio possui sobre o valor real que a empresa tem.

*Dívida Bruta = dívidas de curto prazo + dívidas de longo prazo e o Patrimônio líquido = ativos – passivos*

```
In [0]: # Criar dataframe chamado divida_bruta
divida_bruta = pd.DataFrame()
#Criar um loop calculando para cada empresa a dívida bruta, concatenando todos no
for i in range(0, n_empresas):
    dbpl = pd.Series((df_bpp.loc[lista_de_empresas[i], :].loc['Empréstimos e Financiamentos'],
                     df_bpp.loc[lista_de_empresas[i], :].loc['Empréstimos e Financiamentos - Juros']))
    divida_bruta = pd.concat([divida_bruta, dbpl], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
divida_bruta.columns = lista_de_empresas

# Criar dataframe chamado pl_ajustado
pl_ajustado = pd.DataFrame()
#Criar um loop calculando para cada empresa a patrimônio líquido ajustado, concatenando todos no
for i in range(0, n_empresas):
    pl_ajd = pd.Series(df_bpp.loc[lista_de_empresas[i], :].loc['Patrimônio Líquido C',
                     df_bpp.loc[lista_de_empresas[i], :].loc['Patrimônio Líquido C - Juros']))
    pl_ajustado = pd.concat([pl_ajustado, pl_ajd], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
pl_ajustado.columns = lista_de_empresas

#Criar um dataframe com o cálculo da dívida bruta sobre o patrimônio líquido
divida_bruta_pl_df = divida_bruta/pl_ajustado
# Exibe o dataframe
divida_bruta_pl_df
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECONCAVO S.A.
0	0.245664	0.536172	0.932343	0.172046

Este indicador tem grande importância quanto a alavancagem financeira, pois quanto maior, demonstra a possibilidade do patrimônio líquido este comprometido com dívidas. Podemos verificar que a 3R Petroleum e a Petroreconcavo demonstram que suas dívidas estão bem equilibradas, não comprometendo seus negócios. Entretanto, a Petro Rio, tem suas dívidas bastante comprometida, tendo um alta alavancagem financeira, porém como vimos a sua liquidez corrente, podem verificar que possivelmente apesar do indicador mostrar um alerta que sua dívida compromete o seu patrimônio, há uma grande disponibilidade que tem capacidade de reduzir essas dívidas, porém pode ser estratégico para o crescimento da empresa.

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com largura 10 e altura 6 (em pixels)
fig, ax = plt.subplots(figsize = (10, 6))

# Plota o gráfico de barras
b1 = ax.barh(y = tickers, width = divida_bruta_pl_df.iloc[-1], color = coresVirid)

# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
    width = rect.get_width()
    ax.text(
        x = width,
        y = rect.get_y() + rect.get_height() / 2,
        s = "{:.2f}".format(rect.get_width()),
        # posição x
        # posição y
        # string com o valor
```

```

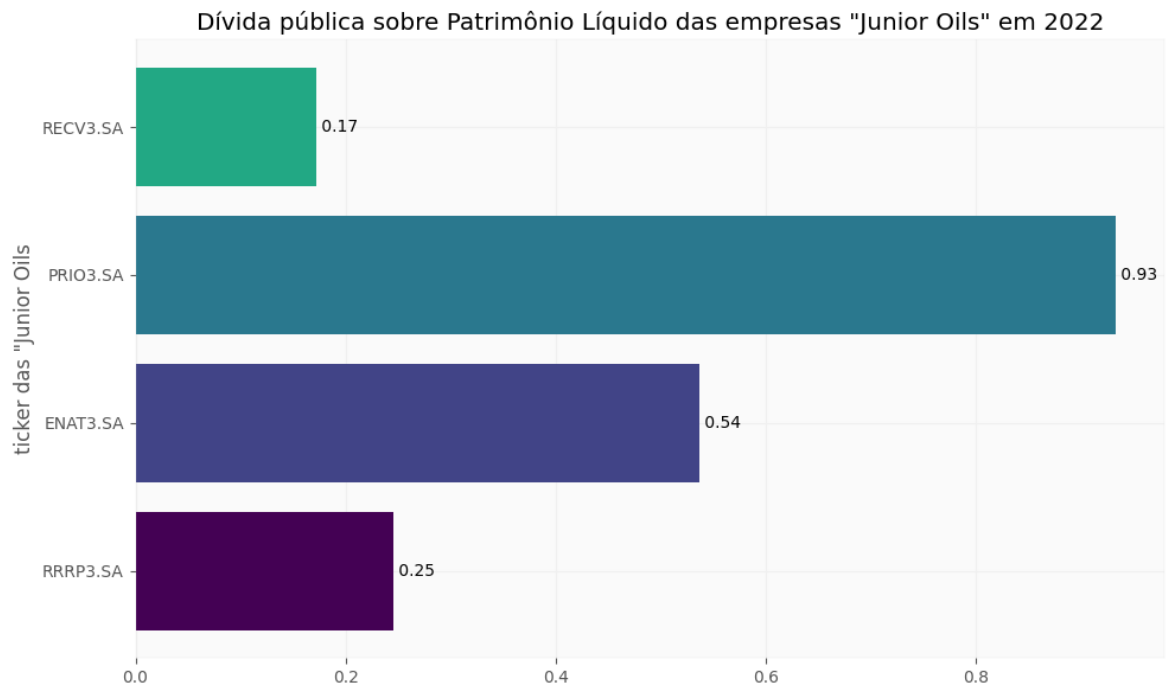
        ha = "left",                                # alinhamento na horizontal
        va = "center"                               # alinhamento na vertical
    )

# Acrescenta um título ao gráfico
plt.title('Dívida pública sobre Patrimônio Líquido das empresas "Junior Oils" em

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils"') # é importante esclarecer a unidade e a ord

# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()

```



### 3.5.2.3.2 Dívida Líquida/Ebit

Analisar o índice de endividamento de uma empresa. Seu resultado demonstra o número de anos que uma empresa levaria para pagar sua dívida líquida no cenário em que o EBIT permanece constante.

*Dívida Líquida = Dívida Bruta - caixa e equivalentes de caixa da empresa*

```

In [0]: # Criar dataframe chamado divida_liquida_ebit
divida_liquida_ebit = pd.DataFrame()
#Criar um loop calculando para cada empresa a dívida líquida, concatenando todos
for i in range(0, n_empresas):
    dbpl = pd.Series((df_bpp.loc[lista_de_empresas[i], :].loc['Empréstimos e Financ
                    df_bpp.loc[lista_de_empresas[i], :].loc['Empréstimos e Financ
    divida_liquida_ebit = pd.concat([divida_liquida_ebit, dbpl], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
divida_liquida_ebit.columns = lista_de_empresas
#Criar um dataframe com o cálculo da dívida líquida sobre o ebit
divida_liquida_ebit = divida_liquida_ebit/ebit_ajustado
#Exibe o dataframe
divida_liquida_ebit

```

Out[0]:	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	0.459229	1.51005	-0.092675	0.221277

Neste indicador expressa que a Enauta levaria um ano e meio para quitar suas dívidas que não é um mal indicador, enquanto que Petro Rio teve o indicador negativo, isso ocorre porque há mais disponibilidade que dívida, visto na liquidez corrente, logo todas as empresas tem um bom indicador.

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

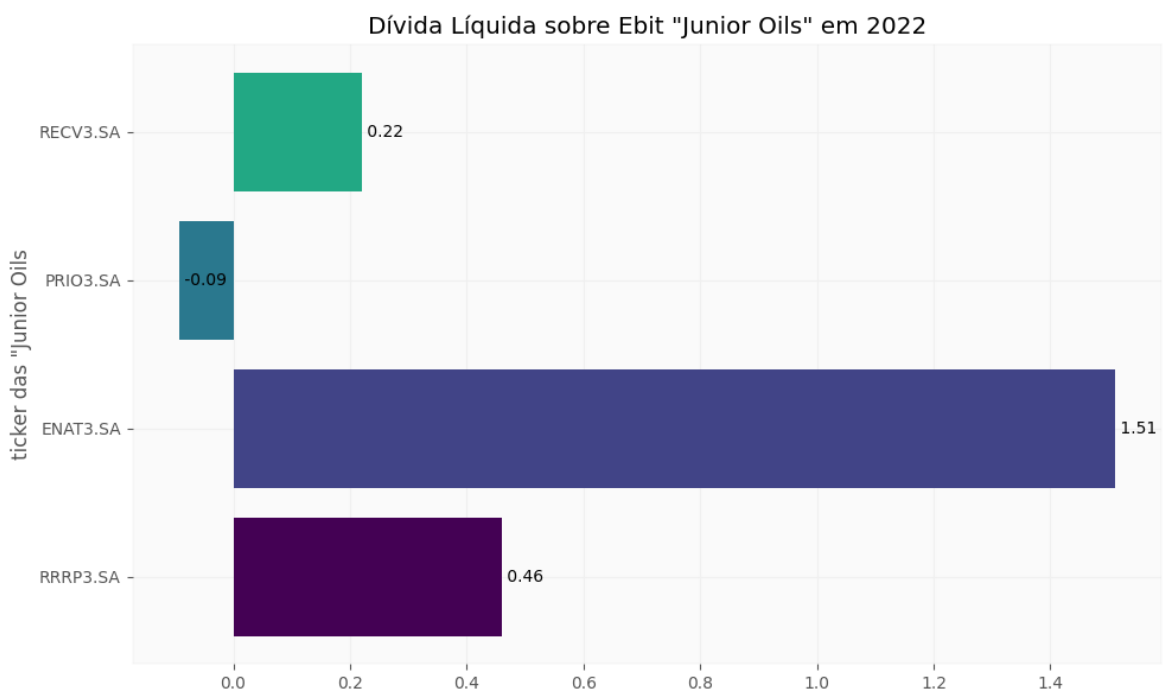
# Plota o gráfico de barras
b1 = ax.barh(y = tickers, width = divida_liquida_ebit.iloc[-1], color = coresViri

# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
    width = rect.get_width()
    ax.text(
        x = width,
        y = rect.get_y() + rect.get_height() / 2,
        s = "{:.2f}".format(rect.get_width()),
        ha = "left",
        va = "center"
    )

# Acrescenta um título ao gráfico
plt.title('Dívida Líquida sobre Ebit "Junior Oils" em 2022')

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils"') # é importante esclarecer a unidade e a ord

# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()
```



**Conclusão**

- A Petroreconcavo apresenta a menor dívida bruta sobre o patrimônio líquido (0,17), o que indica que a empresa tem uma alavancagem financeira mais baixa em relação ao seu patrimônio líquido.
- A Petro Rio apresenta a maior dívida bruta sobre o patrimônio líquido (0,93), sugerindo uma alavancagem financeira significativamente maior em relação ao seu patrimônio líquido.
- No que diz respeito à dívida líquida sobre o EBIT, a Petro Rio tem um valor negativo (-0,09), o que pode ser interpretado como uma posição líquida de caixa em relação ao EBIT. Isso indica uma situação financeira positiva nesse aspecto.
- A Enauta apresenta a maior dívida líquida sobre o EBIT (1,51), indicando uma maior alavancagem em relação à sua lucratividade operacional.

Com base nessas métricas, a Petroreconcavo parece ter um nível de endividamento mais baixo em relação ao seu patrimônio líquido, enquanto a Petro Rio possui a maior alavancagem financeira. No entanto, a Petro Rio tem uma posição de caixa líquido em relação ao EBIT, o que pode ser uma vantagem financeira.

#### 3.5.2.4 Resposta da 4ª pergunta

##### Ebit/Ativo

é a capacidade que uma empresa apresenta de gerar lucro com base nas dimensões do seu patrimônio. Quanto maior for a relação EBIT/Ativo, maior é a geração de lucro por parte da empresa e consequentemente melhores os seus resultados para o investidor.

Este indicador demonstra que tanto a Petro Rio como Petroreconcavo tem boa capacidade de gerar lucro e melhores resultados para o investidor, apesar da 3R Petroleum ter o menor indicador, apresenta capacidade de gerar lucro também.

```
In [0]: # Criar dataframe chamado Ebit_Ativo
Ebit_Ativo = pd.DataFrame()
#Criar um loop calculando para cada empresa o ebit sobre o ativo, concatenando to
for i in range(0,n_empresas):
    ativo_ = pd.Series(df_bpa.loc[lista_de_empresas[i],:].loc['Ativo Total'].iloc[-
    Ebit_Ativo = pd.concat([Ebit_Ativo, ativo_], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
Ebit_Ativo.columns = lista_de_empresas
#Criar um dataframe com o cálculo do ebit sobre o ativo
Ebit_Ativo = ebit_ajustado/Ebit_Ativo
#Exibe o dataframe
Ebit_Ativo
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	0.068732	0.107994	0.207385	0.202146

##### 3.5.2.4.1 ROIC - Return Over Invested Capital

É um indicador que avalia a rentabilidade do investimento aplicado pelos acionistas e credores na empresa.

```
In [0]: #Criar um dataframe ROIC com o cálculo do ebit sobre a dívida bruta
ROIC = ebit_ajustado/divida_bruta
#Exibe o dataframe
ROIC
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	0.520773	0.405599	0.456528	2.030489

O ROIC demonstra a rentabilidade em relação ao valor investido, e neste ponto entre as empresas a Petrorecôncavo se destaca, enquanto as outras apresenta indicadores próximo entre elas.

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

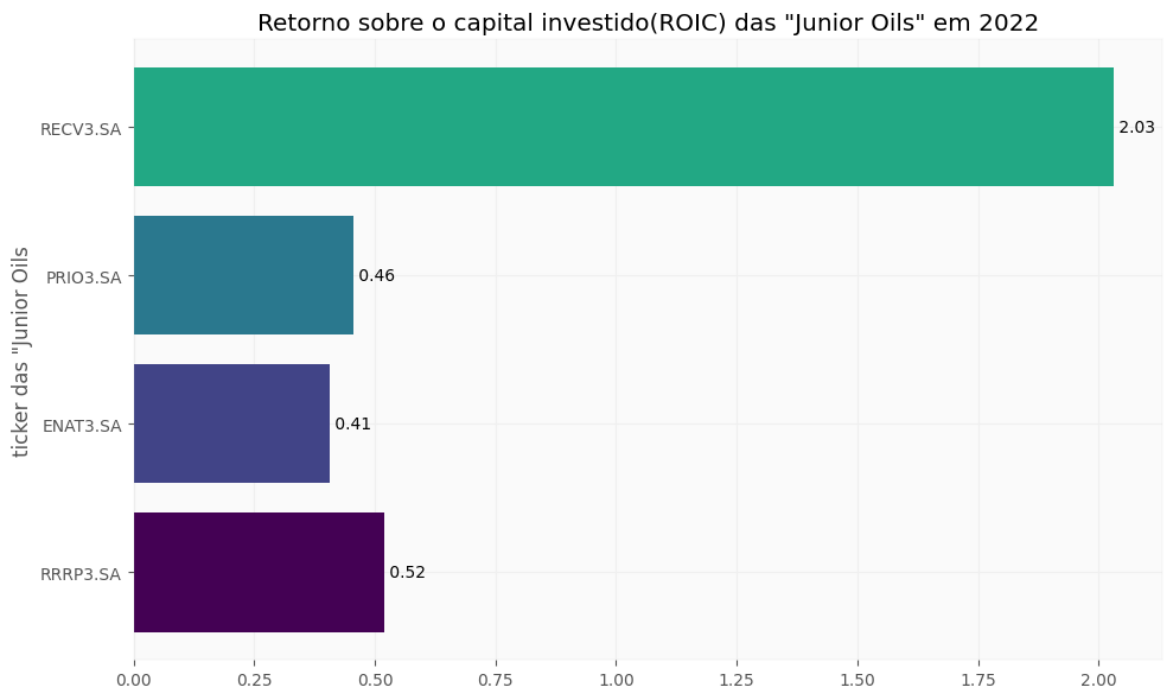
# Plota o gráfico de barras
b1 = ax.barh(y = tickers, width = ROIC.iloc[-1], color = coresViridis)

# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
    width = rect.get_width()
    ax.text(
        x = width,
        y = rect.get_y() + rect.get_height() / 2,
        s = "{:.2f}".format(rect.get_width()),
        ha = "left",
        va = "center"
    )

# Acrescenta um título ao gráfico
plt.title('Retorno sobre o capital investido(ROIC) das "Junior Oils" em 2022')

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils"') # é importante esclarecer a unidade e a ord

# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()
```



#### 3.5.2.4.2 ROE - Return on Equity

É um indicador que avalia em lucro líquido gerado em relação ao patrimônio líquido.

$ROE = \text{lucro líquido} / \text{Patrimônio Líquido}$

```
In [0]: # Criar dataframe chamado ROE
roe = pd.DataFrame()
#Criar um loop calculando para cada empresa o ROE, concatenando todos no mesmo da
for i in range(0,n_empresas):
    calculo_roe_ = pd.Series((df_dre.loc[lista_de_empresas[i],:].loc['Lucro/Prej
    roe = pd.concat([roe, calculo_roe_], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
roe.columns = lista_de_empresas
#Exibe o dataframe
roe
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	0.029823	0.093283	0.346448	0.302687

O ROE indica quanto maior, mais eficiente é a empresa para geração de lucro, logo a Petro Rio e Petrorecôncavo apresenta grandes potenciais de geração de bons retornos ao investidor

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

# Plota o gráfico de barras
b1 = ax.barh(y = tickers, width = roe.iloc[-1], color = coresViridis)

# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
    width = rect.get_width()
    ax.text(
        x = width,
        y = rect.get_y() + rect.get_height() / 2,
        s = "{:.2f}".format(rect.get_width()),
        # posição x
        # posição y
        # string com o valor
```

```

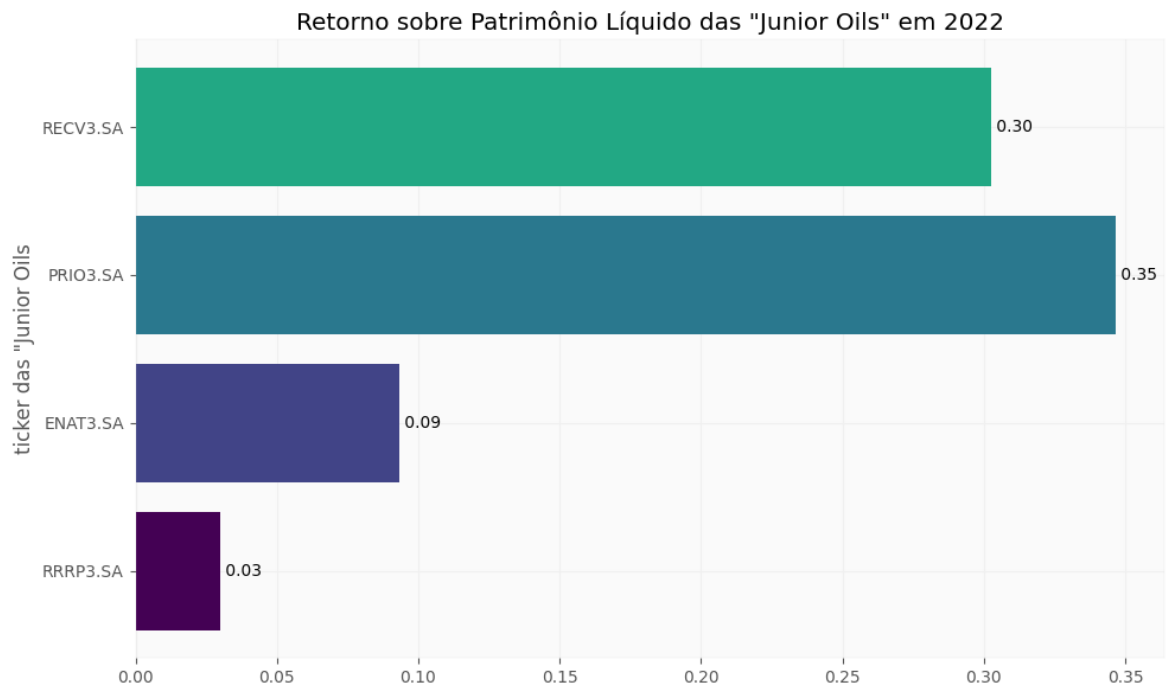
        ha = "left",                                # alinhamento na horizontal
        va = "center"                               # alinhamento na vertical
    )

# Acrescenta um título ao gráfico
plt.title('Retorno sobre Patrimônio Líquido das "Junior Oils" em 2022')

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils"') # é importante esclarecer a unidade e a ord

# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()

```



## Conclusão

- A Petroreconcavo apresenta o maior ROIC (2,03), o que indica uma alta eficiência na geração de retorno em relação ao capital investido. Isso sugere que a empresa está gerando um bom retorno em seus ativos.
- A Petro Rio possui um ROE mais alto (0,35), indicando uma eficiência sólida na geração de retorno em relação ao seu patrimônio líquido. Isso sugere que a empresa está utilizando efetivamente o capital dos acionistas para gerar lucros.
- A Enauta e a 3R Petroleum têm métricas mais baixas tanto para ROIC quanto para ROE, o que sugere uma eficiência menor na utilização de ativos e patrimônio líquido em comparação com as outras empresas.

Com base nessas métricas, a Petroreconcavo se destaca em termos de eficiência na rentabilidade dos ativos, enquanto a Petro Rio lidera em termos de eficiência na geração de retorno em relação ao patrimônio líquido.

### 3.5.2.5 Resposta da 5ª pergunta

Equity e Preço

É o valor pertencente aos proprietários e acionistas dessa determinada empresa. Ou seja, é a soma do valor real de cada ação.

```
In [0]: # Criar um DataFrame vazio para armazenar as informações de todos os tickers
cotacao = pd.DataFrame()
# Iterar sobre a lista de tickers
for ticker in tickers:
    # Obter os preços históricos da ação no período desejado
    data = si.get_data(ticker, start_date='2022-01-01', end_date='2022-12-31')
    # Selecionar somente a coluna de preços de fechamento (Close)
    close_prices = data['close']
    # Adicionar as informações do ticker ao DataFrame criado anteriormente
    cotacao[ticker] = close_prices
# Os nomes das empresas selecionadas como coluna do dataframe
cotacao.columns = lista_de_empresas
# Exibir o DataFrame com as informações de todos os tickers
cotacao
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
2022-01-03	34.150002	13.35	20.290001	17.570000
2022-01-04	34.480000	13.70	20.719999	17.830000
2022-01-05	32.009998	12.80	18.490000	16.400000
2022-01-06	31.809999	12.09	18.959999	16.360001
2022-01-07	34.000000	12.82	19.820000	17.219999
...	...	...	...	...
2022-12-23	35.860001	12.91	35.490002	30.330000
2022-12-26	36.110001	13.23	35.439999	30.730000
2022-12-27	36.000000	13.10	35.580002	30.950001
2022-12-28	37.029999	13.31	37.250000	33.049999
2022-12-29	37.750000	13.50	37.209999	33.130001

250 rows × 4 columns

Podemos observar que em 2022 a Petro Rio e Petrorecôncavo apresenta crescimento de suas ações no mercado, diferente da Enauta que houve queda do preço e a 3R Petroleum bem volátil.

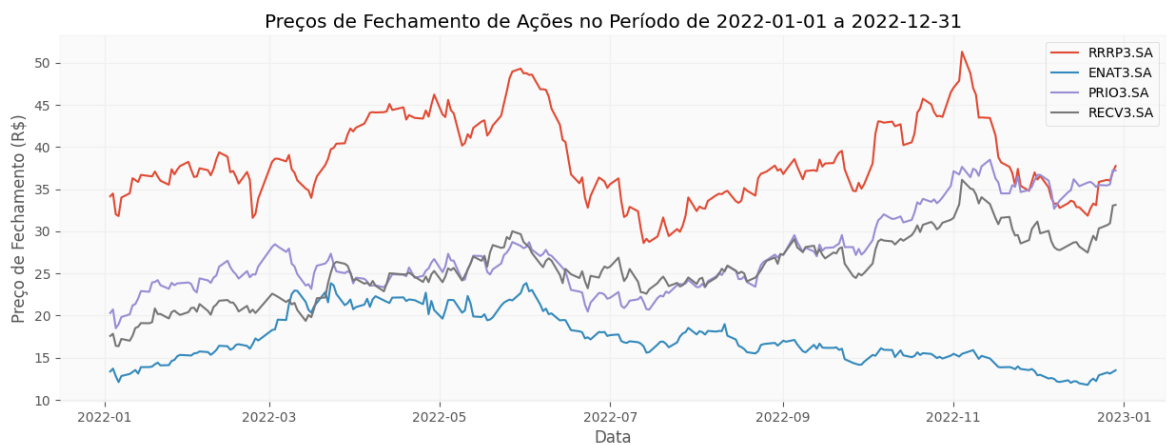
```
In [0]: # Criar dataframe chamado cotacao
cotacao = pd.DataFrame()
# Iterar sobre a lista de tickers
for ticker in tickers:
```



```

# Obter os preços históricos da ação no período desejado
data = si.get_data(ticker, start_date='2022-01-01', end_date='2022-12-31')
# Selecionar somente a coluna de preços de fechamento (Close)
close_prices = data['close']
# Adicionar as informações do ticker ao DataFrame criado anteriormente
cotacao[ticker] = close_prices
# Prepara a figura (fig) e a área do gráfico (ax) com largura 15 e altura 5
fig, ax = plt.subplots(figsize=(15, 5))
# Plota o gráfico de linha para cada ticker
for ticker in tickers:
    ax.plot(cotacao.index, cotacao[ticker], label=ticker)
# Acrescenta um título ao gráfico
ax.set_title('Preços de Fechamento de Ações no Período de 2022-01-01 a 2022-12-31')
# Define os rótulos dos eixos
ax.set_xlabel('Data')
ax.set_ylabel('Preço de Fechamento (R$)')
# Acrescenta a legenda ao gráfico
ax.legend()
# Exibe o gráfico
plt.show()

```



Quantidade das ações das empresas

```

In [0]: valores = [203087632, 265806905, 887229147, 293183802]

dicionario = {'Empresas': lista_de_empresas, 'Valores': valores, 'Tickers': tickers}
df = pd.DataFrame(dicionario)

n_acoes = df.groupby(['Empresas'])['Valores'].sum().reset_index()

colunas = {}
for empresa in n_acoes['Empresas']:
    for index, row in n_acoes.iterrows():
        if row['Empresas'] == empresa:
            colunas[index] = row['Valores']

n_acoes = pd.DataFrame(colunas, index=[0])

# dividir numero de ações por 1000
n_acoes = n_acoes/1000
#Exibe o dataframe
n_acoes

```

```

Out[0]:

```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	203087.632	265806.905	887229.147	293183.802

3.5.2.5.1 Valor de Mercado (Marketcap)

É o quanto uma empresa ou produto está valendo para o mercado, conforme a lei de oferta e demanda.

```
In [0]: # Criar um DataFrame vazio para armazenar as informações de todos os tickers
cotacao = pd.DataFrame()

# Iterar sobre a lista de tickers
for ticker in tickers:
    # Obter os preços históricos da ação no período desejado
    data = si.get_data(ticker, start_date='2022-01-01', end_date='2022-12-31')
    # Selecionar somente a coluna de preços de fechamento (Close)
    close_prices = data['close']
    # Adicionar as informações do ticker ao DataFrame criado anteriormente
    cotacao[ticker] = close_prices
# Os nomes das empresas selecionadas como coluna do dataframe
cotacao.columns = lista_de_empresas
# Exibir o DataFrame com as informações de todos os tickers
cotacao
# Criar um dataframe com o cálculo do marketcap
marketcap = n_acoes * cotacao.iloc[-1]
# Exibe o dataframe
marketcap
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECONCAVO S.A.
0	7666558.108	3.588393e+06	3.301380e+07	9.713180e+06

Este indicador demonstra que a Petro Rio está bem valorado diante do mercado entre as 4 empresas "Junior Oils".

```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

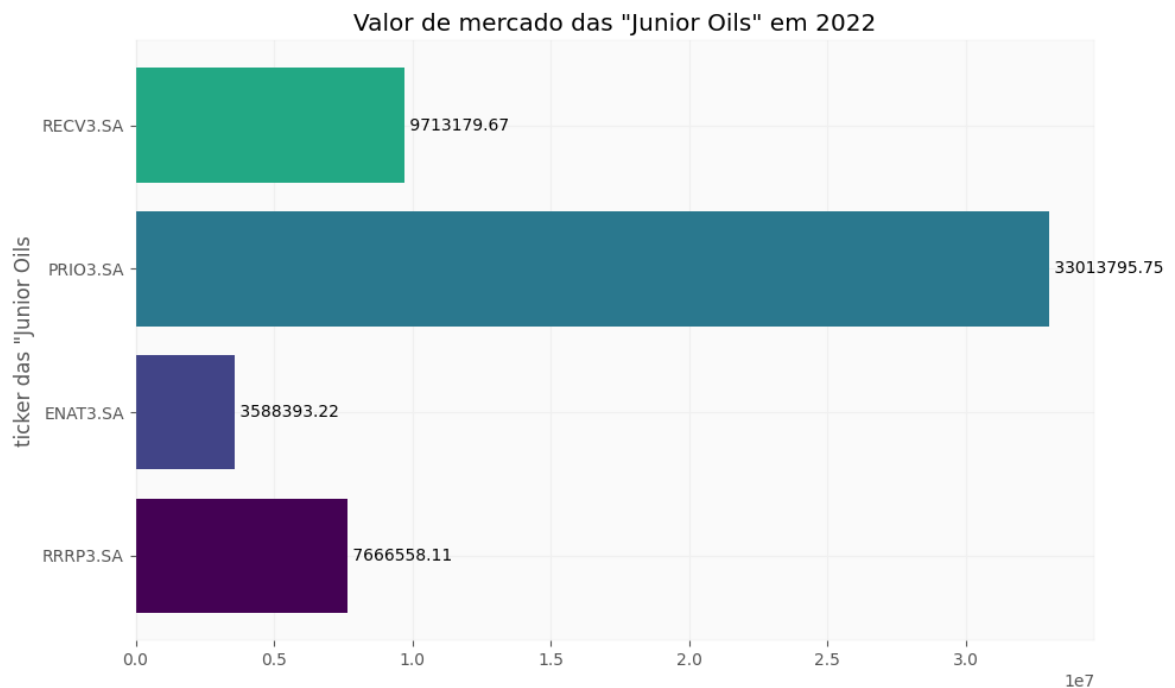
# Plota o gráfico de barras
b1 = ax.barh(y = tickers, width = marketcap.iloc[-1], color = coresViridis)

# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
    width = rect.get_width()
    ax.text(
        x = width,                                     # posição x
        y = rect.get_y() + rect.get_height() / 2,     # posição y
        s = "{:.2f}".format(rect.get_width()),         # string com o valor
        ha = "left",                                   # alinhamento na horizontal
        va = "center",                                 # alinhamento na vertical
    )

# Acrescenta um título ao gráfico
plt.title('Valor de mercado das "Junior Oils" em 2022')

# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils"') # é importante esclarecer a unidade e a ord

# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()
```



### 3.5.2.5.2 Valor da Firma (Enterprise Value)

É um indicador leva em conta a cotação das ações de uma empresa (valor de mercado), juntamente com seus ativos (caixa e patrimônio) e passivos (dívidas), para definir quando a companhia realmente vale.

```
In [0]: #Criar um dataframe com o cálculo do valor da firma
ev = marketcap + (divida_bruta - caixa_ajustado)
#Exibe o dataframe
ev
```

```
Out[0]:
```

	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	7886805.108	3.360168e+06	3.262359e+07	8.757570e+06

Da mesma forma o valor da firma da Petro Rio é maior que seus pares

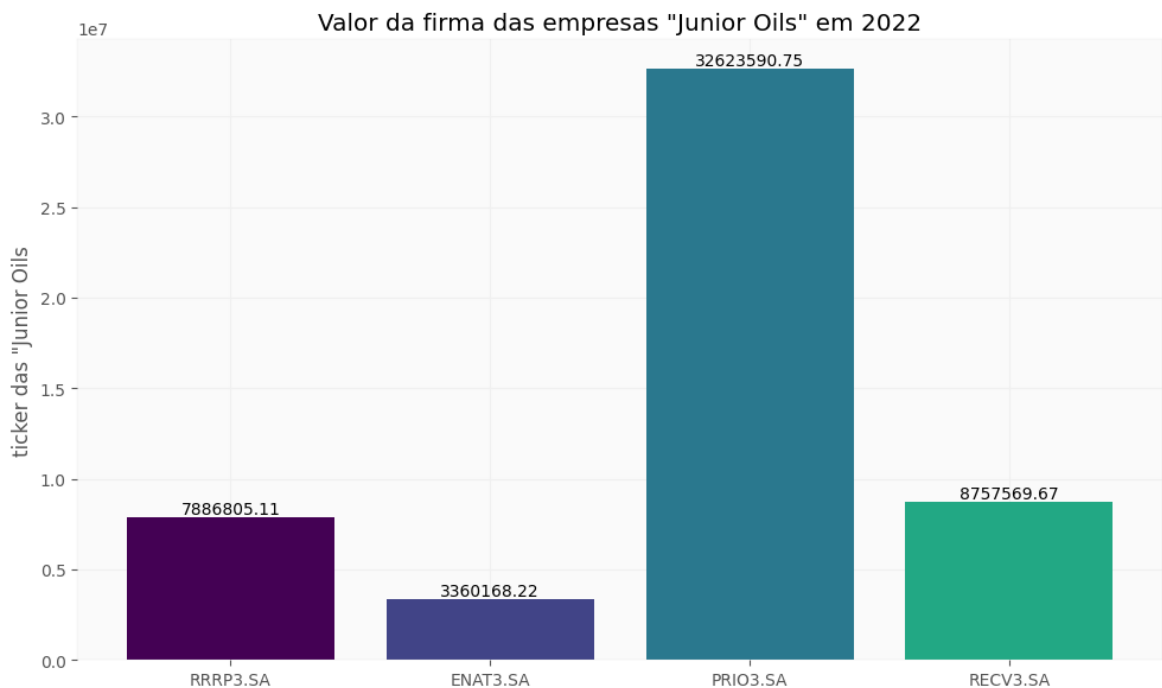
```
In [0]: # Prepara a figura (fig) e a área do gráfico (ax) com largura 10 e altura 6 (em p
fig, ax = plt.subplots(figsize = (10, 6))

# Plotar o gráfico de barras
b1 = ax.bar(x = tickers, height = ev.iloc[-1], color = coresViridis)

# Acrescenta os valores sobre as barras
for i, rect in enumerate(b1.patches):
    height = rect.get_height()
    ax.text(
        x = rect.get_x() + rect.get_width() / 2,      # posição x
        y = height + 5,                                # posição y
        s = "{:.2f}".format(height),                  # string com o valor
        ha = "center",                                # alinhamento na horizontal
        va = "bottom",                                # alinhamento na vertical
    )

# Acrescenta um título ao gráfico
plt.title('Valor da firma das empresas "Junior Oils" em 2022')
# Define os rótulos dos eixos
# plt.xlabel('país') # como já está no título, não é necessário
plt.ylabel('ticker das "Junior Oils") # é importante esclarecer a unidade e a ord
```

```
# Ajusta as margens do gráfico (para eliminar margens desnecessárias)
plt.tight_layout()
# Exibe o gráfico
plt.show()
```



## Conclusão

- A Petro Rio tem o maior valor de mercado (market cap), tornando-se a empresa mais valiosa entre as empresas Oil junior em 2022.
- Em termos de valor da firma (enterprise value), a Petro Rio também lidera, indicando que, quando se considera a dívida e outros passivos financeiros, ela ainda é a empresa mais valiosa.
- A Petroreconcavo vem em segundo lugar em ambas as métricas, com um valor de mercado e valor da firma consideráveis.
- A 3R Petroleum e a Enauta têm valores de mercado e valor da firma menores em comparação com as outras duas empresas.

Com base nessas métricas, a Petro Rio é a empresa mais valiosa em termos de valor de mercado e valor da firma, refletindo sua posição de destaque entre as empresas Oil junior.

### 3.5.2.6 Resposta da 6ª pergunta

#### 3.5.2.6.1 Preço/Lucro

É um índice usado para avaliar se o preço das ações de uma empresa está caro ou barato.

```
In [0]: ## Criar dataframe chamado Lucro por ação(LPA)
LPA = pd.DataFrame()
#Criar um loop calculando para cada empresa o lucro, concatenando todos no mesmo
for i in range(0,n_empresas):
    LPA_ = pd.Series(df_dre.loc[lista_de_empresas[i],:].loc['Atribuído a Sócios da
    LPA = pd.concat([LPA, LPA_], axis=1)
# Os nomes das empresas selecionadas como coluna do dataframe
LPA.columns = lista_de_empresas
```

```
# Calcular o Lucro por ação
LPA = LPA/n_acoes
#Exibe o dataframe
LPA
# Criar o dataframe do Preço/Lucro, calculando o preço da cotação da ação sobre o
PL = cotacao.iloc[-1]/LPA
#Exibe o dataframe
PL
#Exibe o indicadores como linha
LPA['Indicadores']='LPA'
PL['Indicadores']='PL'
#Criar dataframe preço por lucro concatenando os dois indicadores
df_pl = pd.concat([LPA,PL])
#Reindexar as linhas dos indicadores de acordo com as empresas selecionadas
df_pl = df_pl.reindex(columns=['Indicadores'] + list(df_pl.columns[:-1]))
#Exibe o dataframe
df_pl
```

Out[0]:

	Indicadores	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	LPA	0.736475	1.442306	3.862668	3.934020
0	PL	51.257668	9.360008	9.633237	8.421411

### 3.5.2.6.2 Preço/VPA

Indica quanto os acionistas topam pagar, no momento, pelo patrimônio líquido da companhia.

In [0]:

```
# Criar o dataframe do valor patrimonial por ação, calculando o patrimônio líquido
vpa = pl_ajustado/n_acoes
# Criar o dataframe do preço por valor patrimonial por ação, calculando a cotação
pvpa = cotacao.iloc[-1]/vpa
#Exibe o indicadores como linha
vpa['Indicadores']='VPA'
pvpa['Indicadores']='P/VPA'
#Criar dataframe preço por lucro concatenando os dois indicadores
df_pvpa = pd.concat([vpa,pvpa])
#Reindexar as linhas dos indicadores de acordo com as empresas selecionadas
df_pvpa = df_pvpa.reindex(columns=['Indicadores'] + list(df_pvpa.columns[:-1]))
#Exibe o dataframe
df_pvpa
```

Out[0]:

	Indicadores	3R PETROLEUM ÓLEO E GÁS S.A.	ENAUTA PARTICIPAÇÕES S.A.	PETRO RIO S.A.	PETRORECÔNCAVO S.A.
0	VPA	21.086656	15.461679	11.149337	12.996973
0	P/VPA	1.790232	0.873126	3.337418	2.549055

### Indicadores Financeiros

In [0]:

```
#Criar colunas para o dataframe
margem_bruta['Indicadores']='mrgbrut'
margem_liquida['Indicadores']='mrgliq'
ebit_ajustado['Indicadores']='ebit'
margem_ebit['Indicadores']='mrgebit'
caixa_ajustado['Indicadores']='disp'
liquidez_corrente['Indicadores']='liqc'
divida_bruta_pl_df['Indicadores']='divbpatr'
```

```

divida_liquida_ebit['Indicadores']='divliqebit'
Ebit_Ativo['Indicadores']='ebitativ'
ROIC['Indicadores']='roic'
roe['Indicadores']='roe'
ev['Indicadores']='ev'
cotacao['Indicadores']='cotacao'
vpa['Indicadores']='patr/acao'
pvpa['Indicadores']='p/vpa'
LPA['Indicadores']='LPA'
PL['Indicadores']='PL'
#Criar dataframe de indicadores concatenando os indicadores
indicadores = pd.concat([cotacao.iloc[[-1]],LPA,PL,vpa,pvpa,margem_bruta,margem_l
                        liquidez_corrente,divida_bruta_pl_df,divida_liquida_ebit,Ebit_
#Transposição de colunas e linhas do dataframe indicadores
indicadores = indicadores.T
# Define a primeira linha como o nome dos DataFrames
indicadores.columns = indicadores.loc['Indicadores', :]
indicadores = indicadores.drop('Indicadores')
# Define o nome da linhas pelo nomes das empresas
indicadores.index.name = 'Empresas'
#Transformar o objetos dos indicadores na forma numerica
indicadores = indicadores.apply(pd.to_numeric, errors='coerce')
#Arredondar os números dos indicadores
indicadores = indicadores.round(2)
#Exibe o dataframe
indicadores

```

Out[0]:

	Indicadores	cotacao	LPA	PL	patr/acao	p/vpa	mrgbrut	mrgliq	mrgebit	liqc	div
--	-------------	---------	-----	----	-----------	-------	---------	--------	---------	------	-----

Empresas											
3R PETROLEUM ÓLEO E GÁS S.A.	37.75	0.74	51.26	21.09	1.79	0.51	0.07	0.32	1.76		
ENAUTA PARTICIPAÇÕES S.A.	13.50	1.44	9.36	15.46	0.87	0.49	0.18	0.41	1.72		
PETRO RIO S.A.	37.21	3.86	9.63	11.15	3.34	0.67	0.54	0.66	7.64		
PETRORECÔNCAVO S.A.	33.13	3.93	8.42	13.00	2.55	0.48	0.39	0.45	1.53		

## Conclusão

- A Petroreconcavo apresenta o menor P/L (8,42), indicando que suas ações estão sendo negociadas a um preço mais baixo em relação aos seus lucros por ação. Isso pode sugerir que as ações da Petroreconcavo estão relativamente mais baratas em termos de lucro.
- No que diz respeito ao P/VPA, novamente a Petroreconcavo lidera com o valor mais baixo (0,74), sugerindo que as ações da Petroreconcavo são negociadas a um preço significativamente inferior em relação ao seu valor patrimonial por ação.
- A Enauta também possui métricas de P/L (9,36) e P/VPA (1,44) relativamente atraentes em comparação com as outras empresas.

- A 3R Petroleum possui o maior P/L (51,26) e P/VPA (3,93), indicando que suas ações estão sendo negociadas a preços mais elevados em relação aos lucros e ao valor patrimonial por ação.

Com base nessas métricas, a Petroreconcavo parece oferecer ações com preços relativamente mais baixos em relação aos lucros (P/L) e ao valor patrimonial por ação (P/VPA), sugerindo que suas ações podem ser consideradas como potencialmente mais baratas.

### 3.5.2.7 Resposta da 7ª pergunta

#### 3.5.2.7.1 Dividend Yield

```
In [0]: close = yf.download(tickers, start="2022-01-01", end="2022-12-31")['Close']
div = vbt.YFData.download(tickers, start="2022-01-01", end="2022-12-31").get('Div
div.index = pd.to_datetime(div.index.date)
close.index = pd.to_datetime(close.index.date)
div = div.reindex(sorted(div.columns), axis =1)
close = close.reindex(sorted(close.columns), axis =1)
div_12m = div.rolling('365D').sum()
div_12m = (div_12m/close) * 100

[*****100%*****] 4 of 4 completed
```

```
In [0]: div_12m.vbt.plot(width = 600, height = 200).show()
```

```
In [0]: DY = pd.DataFrame(div_12m.iloc[-1])
DY.columns = ["Dividend Yield"]
DY
```

Out[0]: **Dividend Yield**

symbol	
ENAT3.SA	12.654926
PRI03.SA	0.000000
RECV3.SA	2.338467
RRRP3.SA	0.000000

### Conclusão

O Dividend Yield é calculado como o dividendo por ação dividido pelo preço da ação e é uma medida de retorno que os investidores recebem por meio de dividendos em relação ao preço das ações

- A Enauta teve o maior Dividend Yield, com 12,65%, indicando que a empresa distribuiu um retorno relativamente alto aos acionistas em relação ao preço das ações em 2022. Isso pode ser atraente para investidores em busca de renda por meio de dividendos.
- A Petroreconcavo também distribuiu dividendos, com um Dividend Yield de 2,34%, embora seja menor do que o da Enauta, ainda representa uma distribuição de dividendos significativa em relação ao preço das ações.
- A 3R Petroleum e a Petro Rio não distribuíram dividendos em 2022, com um Dividend Yield de 0,00% para ambas as empresas.

Com base nessas métricas, a Enauta se destacou em termos de distribuição de dividendos, oferecendo um Dividend Yield consideravelmente alto em relação ao preço das ações.

#### 3.5.2.8 Resposta da 8ª pergunta

##### **Conclusão**

A decisão de investir nas empresas Oil junior em 2022 deve ser baseada em uma análise abrangente que leve em consideração todas as métricas e fatores discutidos nas sete perguntas anteriores.

- **Margem e Rentabilidade:**
  - Em termos de margens, a Petroreconcavo e a Petro Rio tiveram um bom desempenho, enquanto a Enauta e a 3R Petroleum tiveram margens mais baixas.
  - Em relação à rentabilidade, a Petroreconcavo apresentou o maior ROIC, enquanto a Petro Rio liderou em ROE.
- **Liquidez e Capacidade Financeira:**
  - A Petro Rio tem a maior disponibilidade financeira e uma liquidez corrente sólida.
  - A Enauta também possui uma liquidez corrente decente, mas com menos disponibilidade financeira.
- **Endividamento:**
  - A Petroreconcavo tem uma alavancagem mais baixa em relação ao patrimônio líquido.
  - A Petro Rio possui uma alavancagem mais alta, mas também uma posição de caixa líquido em relação ao EBIT.
- **Valor de Mercado:**



- A Petro Rio tem o maior valor de mercado e valor da firma, tornando-se a empresa mais valiosa.
- **Preços das Ações:**
  - A Petroreconcavo apresenta múltiplos P/L e P/VPA mais baixos, sugerindo que suas ações podem ser consideradas mais baratas em relação aos lucros e ao valor patrimonial.
- **Dividendos:**
  - A Enauta distribuiu o maior Dividend Yield, seguida pela Petroreconcavo. A Petro Rio e a 3R Petroleum não distribuíram dividendos em 2022.

Com base nessas análises, a decisão de investir nas empresas Oil junior em 2022 depende de seus objetivos de investimento e tolerância ao risco. Aqui estão algumas considerações gerais:

Para empresas com forte capacidade financeira e liquidez, a Petro Rio parece ser uma opção sólida, apesar de sua alavancagem mais alta.

Para ações com múltiplos P/L e P/VPA mais baixos, a Petroreconcavo e a Enauta podem ser consideradas, sendo que a Enauta também oferece um Dividend Yield atrativo.

Se a distribuição de dividendos é uma prioridade, a Enauta se destaca com o maior Dividend Yield.

Para uma combinação de margens sólidas e boa rentabilidade, a Petroreconcavo pode ser uma escolha interessante.

No entanto, investir em ações sempre envolve riscos e que a diversificação de investimentos é uma estratégia importante para mitigar esses riscos. Além disso, fazer uma análise mais detalhada das empresas, consultar um consultor financeiro e estar ciente das condições do mercado e das mudanças nas circunstâncias das empresas antes de tomar decisões de investimento.

Portanto, a solução não é uma recomendação de compra e venda de ações, mas através de indicadores das empresas selecionadas considerando o resultado do ano de 2022 potenciais investimentos capazes de possibilitar retorno financeiros aos investidores. Entretanto neste projeto não foram informados os possíveis análise de riscos (incertezas) sobre os investimentos.

## 4. Autoavaliação

Estou extremamente satisfeito com o progresso que fiz ao longo deste projeto no Microsoft Azure Databricks. Foi uma jornada desafiadora, mas incrivelmente gratificante, trabalhando com dados cruciais no setor de petróleo e aprendendo a lidar com uma variedade de tarefas relacionadas à engenharia de dados.

Ao longo deste projeto, enfrentei várias necessidades e dificuldades, incluindo:

**Coleta de Dados na Nuvem:** A coleta de dados na nuvem trouxe uma série de desafios, como garantir a segurança dos dados, a integridade e a eficiência do processo. Aprendi a configurar pipelines de coleta de dados robustos e a lidar com dados em tempo real.

**Modelagem de Dados:** A modelagem de dados exigiu a seleção das estruturas de dados adequadas para armazenar as informações financeiras da empresa júnior de petróleo. Foi crucial definir os esquemas de dados corretos para permitir análises eficazes.

**Carga de Dados:** Carregar grandes volumes de dados com eficiência foi um desafio importante. Aprendi a otimizar o processo de carga para garantir que os dados estivessem prontos para análise em tempo hábil.

**Análise de Dados:** A análise de dados envolveu a aplicação de várias técnicas estatísticas e de aprendizado de máquina para extrair insights valiosos das demonstrações financeiras. Adquiri conhecimentos avançados em ciência de dados para realizar análises significativas.

Além disso, durante a Sprint de Engenharia de Dados, adquiri conhecimentos valiosos que me permitiram aprofundar meu entendimento sobre a gestão de dados na nuvem, processamento de big data e análise avançada. Essas habilidades são extremamente relevantes e podem ser aplicadas em projetos futuros.

Olhando para o futuro, meu objetivo é continuar aprimorando minhas habilidades em ciência de dados e engenharia de dados, buscando informações financeiras adicionais e produzindo insights valiosos para otimizar a gestão de recursos. Este projeto me motivou a continuar explorando novas oportunidades e contribuir para a eficiência e eficácia em minha área de atuação. Estou animado com as perspectivas que a ciência de dados oferece e estou ansioso para continuar aprimorando meu conhecimento e habilidades nesse campo.