Dependências do projeto

```
import requests
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Objeto para armazenar resultados vindos do github

```
class Result(object):
    encryptTime = 0
    decryptTime = 0
    size = 0

    def __init__(self, encryptTime, decryptTime, size):
        self.encryptTime = encryptTime
        self.decryptTime = decryptTime
        self.size = size

def make_result(encryptTime, decryptTime, size):
    return Result(encryptTime, decryptTime, size)
```

Definindo url de cada arquivo de resultado do teste

```
NUM_LANGUAGES = 5
BASE_URL = 'https://raw.githubusercontent.com/gprando55/ecies-benchmarking/main/pk

node = [BASE_URL+'/nodejs/node-1kb.txt', BASE_URL+'/nodejs/node-10kb.txt',BASE_URL

python = [BASE_URL+'/python/python-1kb.txt', BASE_URL+'/python/python-10kb.txt', B

java = [BASE_URL+'/java/ecies/java-1kb.txt', BASE_URL+'/java/ecies/java-10kb.txt',

golang = [BASE_URL+'/golang/go-1kb.txt', BASE_URL+'/golang/go-10kb.txt', BASE_URL+

rust = [BASE_URL+'/rust/ecies/rust-1kb.txt', BASE_URL+'/rust/ecies/rust-10kb.txt',
```

Coletando resultados para Nodejs

```
nodeResults = []
for url in node:
  page = requests.get(url)
  encrypt,decrypt,_ = page.text.split('\n')
  result = make_result(encrypt.split(' ')[2], decrypt.split(' ')[2], decrypt.split
  nodeResults.append(result)
print(nodeResults)
```

```
[<__main__.Result object at 0x7f4015a9ec90>, <__main__.Result object at 0x7f4
```

Coletando resultados para Golang

```
golangResults = []
for url in golang:
  page = requests.get(url)
  encrypt,decrypt,_ = page.text.split('\n')
  result = make_result(encrypt.split(' ')[2], decrypt.split(' ')[2], decrypt.split
  golangResults.append(result)
print(golangResults)
```

```
[<__main__.Result object at 0x7f4015a2c990>, <__main__.Result object at 0x7f4
```

Coletando resultados para Python

```
pythonResults = []
for url in python:
  page = requests.get(url)
  encrypt,decrypt,_ = page.text.split('\n')
  result = make_result(encrypt.split(' ')[3], decrypt.split(' ')[3], decrypt.split
  pythonResults.append(result)
print(pythonResults)
```

```
[<__main__.Result object at 0x7f4015a2c290>, <__main__.Result object at 0x7f4
```

Coletando resultados para Java

```
javaResults = []
for url in java:
  page = requests.get(url)
  encrypt,decrypt = page.text.split('\n')
  result = make_result(encrypt.split(' ')[2], decrypt.split(' ')[2], decrypt.split
  javaResults.append(result)
print(javaResults)
```

```
[<__main__.Result object at 0x7f4015a816d0>, <__main__.Result object at 0x7f4
```

Coletando resultados para Rust

```
rustResults = []
for url in rust:
  page = requests.get(url)
  encrypt,decrypt = page.text.split('\n')
  result = make_result(encrypt.split(' ')[2], decrypt.split(' ')[2], decrypt.split
  rustResults.append(result)
print(rustResults)
```

```
[<__main__.Result object at 0x7f4015a2c390>, <__main__.Result object at 0x7f4
```

Montando vetores de tempos por linguagens para plotar

```
X = []

golangEncrypt = []
golangDecrypt = []
pythonEncrypt = []
pythonDecrypt = []
nodeEncrypt = []
nodeDecrypt = []
javaEncrypt = []
javaDecrypt = []
rustEncrypt = []
rustDecrypt = []


for i in range(5):
  X.append(nodeResults[i].size.upper())
  golangEncrypt.append(float(golangResults[i].encryptTime)/1000000)
  golangDecrypt.append(float(golangResults[i].decryptTime)/1000000)
  pythonEncrypt.append(float(pythonResults[i].encryptTime))
  pythonDecrypt.append(float(pythonResults[i].decryptTime))
  nodeEncrypt.append(float(nodeResults[i].encryptTime))
  nodeDecrypt.append(float(nodeResults[i].decryptTime))
  javaEncrypt.append(float(javaResults[i].encryptTime)/1000000)
  javaDecrypt.append(float(javaResults[i].decryptTime)/1000000)
  rustEncrypt.append(float(rustResults[i].encryptTime.replace("ms", "")))
  rustDecrypt.append(float(rustResults[i].decryptTime.replace("ms", "")))
```

Definindo vars gráfico

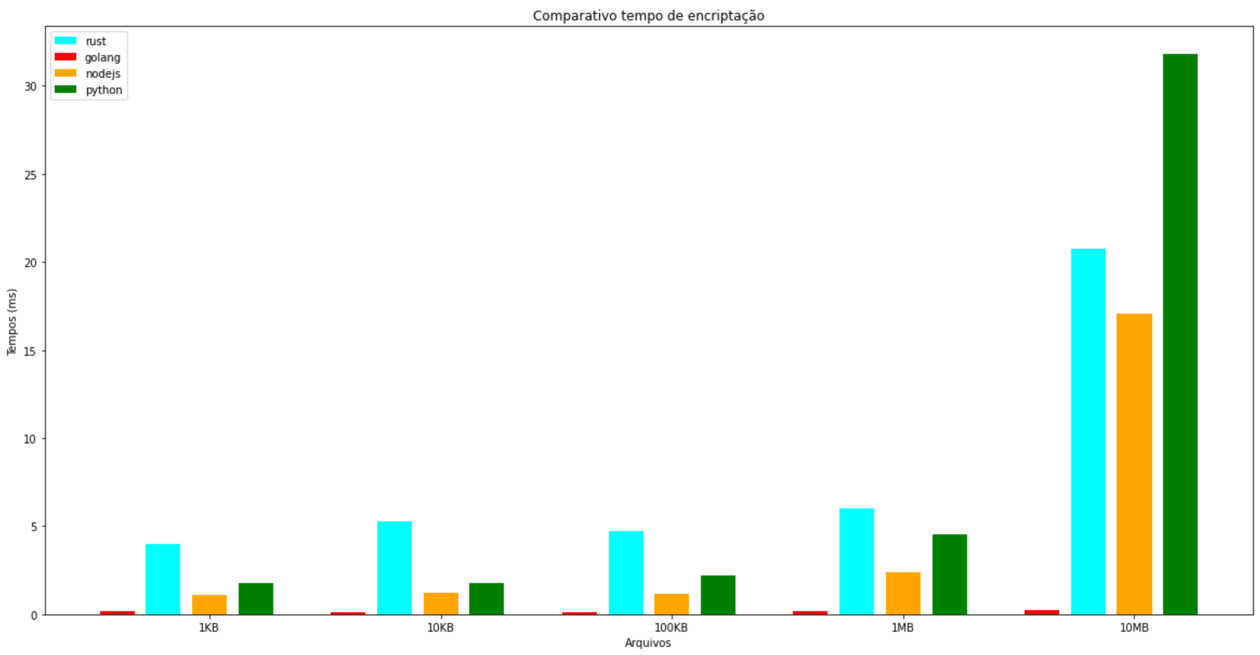
```
x = np.arange(5)
width = 0.15
```

Grafico de tempo de Encrypt

```
plt.figure(figsize=(20,10))

plt.bar(x-0.2, rustEncrypt, width, color='cyan')
plt.bar(x-0.4, golangEncrypt, width, color='red')
plt.bar(x, nodeEncrypt, width, color='orange')
plt.bar(x+0.2, pythonEncrypt, width, color='green')
# plt.bar(x+0.4, javaEncrypt, width, color='purple')

plt.xticks(x, ['1KB', '10KB', '100KB', '1MB', '10MB'])
plt.xlabel("Arquivos")
plt.ylabel("Tempos (ms)")
plt.title("Comparativo tempo de encriptação")
plt.legend(["rust", "golang", "nodejs", "python", "java"])
plt.show()
```



Adicionando tempo do java

```
plt.figure(figsize=(20,10))

plt.bar(x-0.2, rustEncrypt, width, color='cyan')
plt.bar(x-0.4, golangEncrypt, width, color='red')
plt.bar(x, nodeEncrypt, width, color='orange')
plt.bar(x+0.2, pythonEncrypt, width, color='green')
plt.bar(x+0.4, javaEncrypt, width, color='purple')

plt.xticks(x, ['1KB', '10KB', '100KB', '1MB', '10MB'])
plt.xlabel("Arquivos")
plt.ylabel("Tempos (ms)")
plt.title("Comparativo tempo de encriptação")
plt.legend(["rust", "golang", "nodejs", "python", "java"])
plt.show()
```

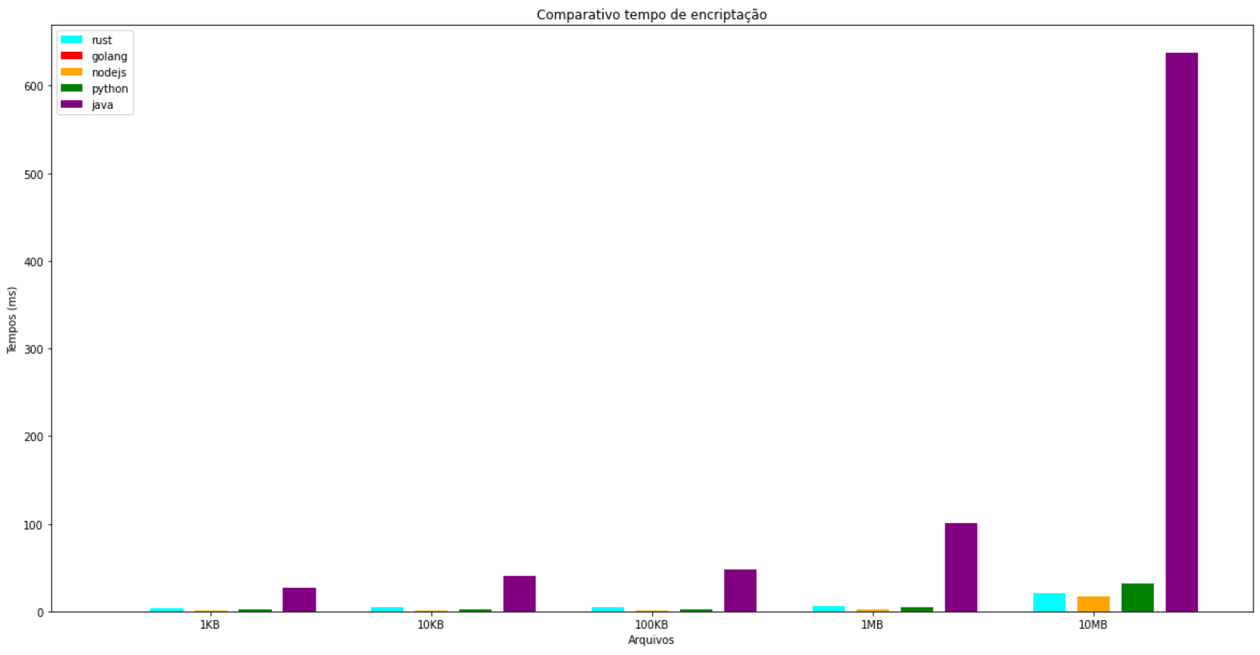
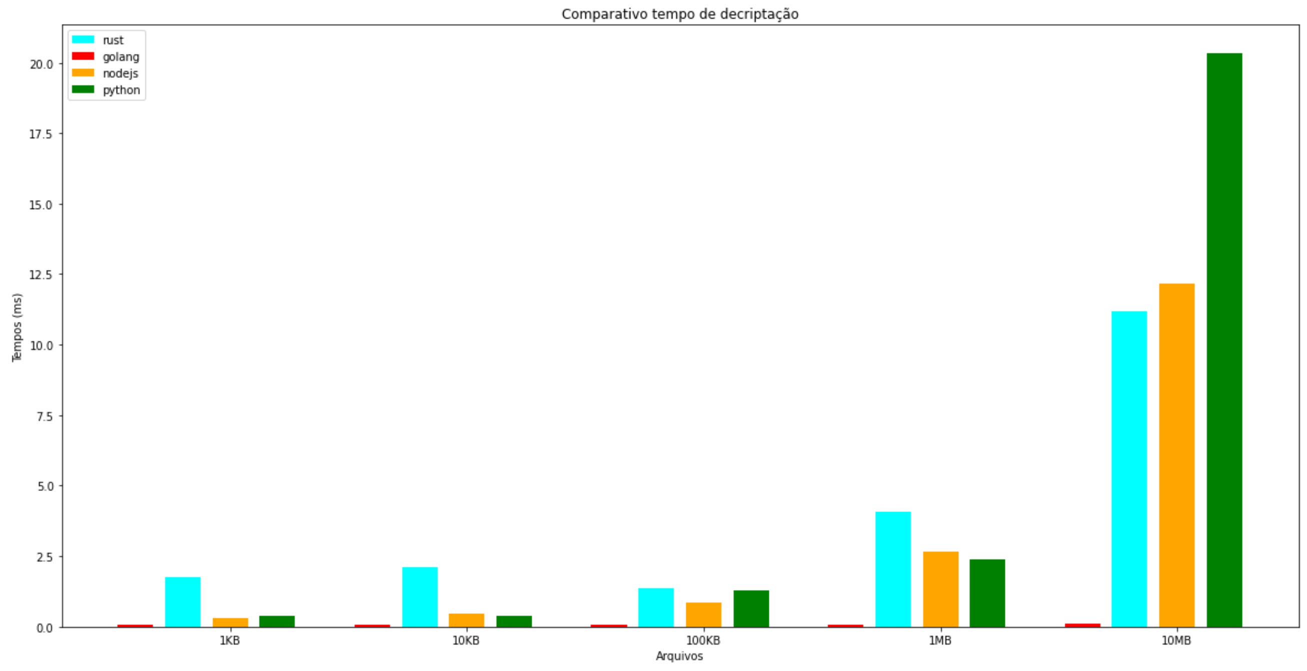


Grafico de tempo de Decrypt

```
plt.figure(figsize=(20,10))

plt.bar(x-0.2, rustDecrypt, width, color='cyan')
plt.bar(x-0.4, golangDecrypt, width, color='red')
plt.bar(x, nodeDecrypt, width, color='orange')
plt.bar(x+0.2, pythonDecrypt, width, color='green')
# plt.bar(x+0.4, javaDecrypt, width, color='purple')

plt.xticks(x, ['1KB', '10KB', '100KB', '1MB', '10MB'])
plt.xlabel("Arquivos")
plt.ylabel("Tempos (ms)")
plt.title("Comparativo tempo de decriptação")
plt.legend(["rust", "golang", "nodejs", "python", "java"])
plt.show()
```

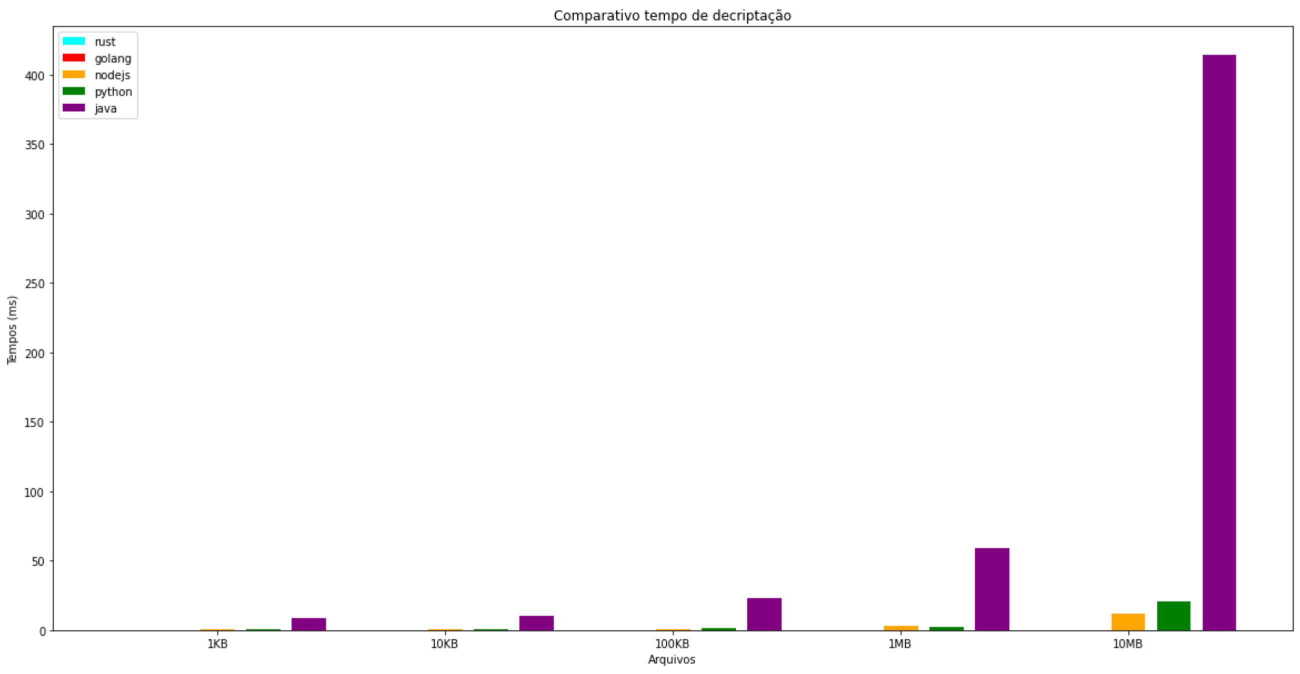


Adicionando tempo do java

```
plt.figure(figsize=(20,10))

plt.bar(x-0.2, rustDecrypt, width, color='cyan')
plt.bar(x-0.4, golangDecrypt, width, color='red')
plt.bar(x, nodeDecrypt, width, color='orange')
plt.bar(x+0.2, pythonDecrypt, width, color='green')
plt.bar(x+0.4, javaDecrypt, width, color='purple')

plt.xticks(x, ['1KB', '10KB', '100KB', '1MB', '10MB'])
plt.xlabel("Arquivos")
plt.ylabel("Tempos (ms)")
plt.title("Comparativo tempo de decriptação")
plt.legend(["rust", "golang", "nodejs", "python", "java"])
plt.show()
```



Colab paid products - Cancel contracts here

✓ 0s completed at 11:11 AM