

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE**

**DEPARTAMENTO DE COMPUTAÇÃO  
CIÊNCIA DA COMPUTAÇÃO**

**ELOY DE FREITAS ALMEIDA, MATHEUS PAULO BASTOS**

**Relatório do trabalho sobre program de busca de assinatura de  
arquivos**

**ALEGRE - ES**

**ABRIL 2023**

# **Relatório do trabalho sobre program de busca de assinatura de arquivos**

Relatório sobre o desenvolvimento de um aplicativo para identificação de assinaturas de arquivos. Disciplina COM10613 do curso de Ciência da Computação do Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo.

por

**ELOY DE FREITAS ALMEIDA, MATHEUS PAULO BASTOS**

Professor

Jacson Rodrigues Correia da Silva

Universidade Federal do Espírito Santo

ALEGRE - ES

ABRIL 2023

# SUMÁRIO

- 1 INTRODUÇÃO..... 3**
  - 1.1 Objetivos..... 3
- 2 Metodologia ..... 4**
  - 2.1 Dataset de assinaturas ..... 4
  - 2.2 Busca de assinaturas ..... 5
  - 2.3 Execução do programa ..... 7
- Referências ..... 8**

# 1 INTRODUÇÃO

Segundo [Wikipedia contributors \(2023\)](#) nem todo arquivo pode ser visualizado por um editor de texto, mas assinaturas de arquivos podem ser reconhecidas quando interpretadas como texto. Dessa forma, o objetivo deste trabalho é construir uma aplicação que seja capaz de identificar assinaturas conhecidas de arquivos em um arquivo de entrada com base nas técnicas aprendidas na disciplina.

## 1.1 Objetivos

1. Desenvolver uma função que faça a leitura de arquivo de forma binária.
2. Desenvolver uma função que faça a leitura de um dataset com metadados de assinaturas de arquivos conhecidos.
3. Desenvolver uma função busque por assinaturas com base no dataset de metadados.

## 2 METODOLOGIA

### 2.1 Dataset de assinaturas

Para desenvolver a solução do problema, foi utilizado a linguagem de programação Python3.9 e o pacote pandas para facilitar a manipulação e análise dos dados.

Com base em pesquisas na internet foi construído um dataset com assinaturas de arquivos conhecidos. O dataset contém uma lista de objetos JSON contendo:

1. Descrição do arquivo
2. Cabeçalho
3. Cabeçalho no formato de número hexadecimal da linguagem python
4. Classe do arquivo
5. Deslocamento de cabeçalho
6. Trailer

O código a seguir apresenta o desenvolvimento da função que faz a leitura do JSON e retorna um objeto pandas.DataFrame, no qual possui recursos que facilitam a manipulação de datasets.

```
def read_dataset(path: str):  
    try:  
        with open(path, 'r') as file:  
            data = js.load(file)  
    except IOError as e:  
        raise IOError(  
            FILE_NOT_FOUND_ERROR_MESSAGE.format(e)
```

```

    )
    dataset = pd.DataFrame(data)
    return dataset

```

Porém, como o dataset é muito grande, foi desenvolvida uma função para que seja possível filtrar uma lista de cabeçalhos para pesquisa. Permitindo que o usuário faça uma busca mais específica.

```

def filter_dataset(
    dataset:pd.DataFrame,
    headers:list[str]=None
):
    if headers:
        dataset = dataset.query(f"Hex in ({headers})")
    return dataset

```

## 2.2 Busca de assinaturas

Para buscar uma assinatura em arquivo foi desenvolvida uma função que recebe o caminho de um arquivo e a uma assinatura no formato de número hexadecimal. Dessa forma o programa faz a leitura do arquivo de forma binária e o percorre em lotes de 8 bytes e os compara com a assinatura passada por parâmetro. Caso o match ocorra um contador é incrementado até que todo o arquivo seja percorrido. Em seguida, ao final do loop o contador é retornado.

```

def find_file_sign(header:str, path:str):
    count = 0
    try:
        with open(path, 'rb') as file:
            while content := file.read(8).hex():
                hex_string = hex(int(content, 16))
                if hex_string == header:
                    count += 1
    except IOError as e:
        raise IOError(FILE_NOT_FOUND_ERROR_MESSAGE.format(e))
    return count

```

A função a seguir tem o objetivo de percorrer todos os cabeçalhos contidos no dataset de assinaturas e fazer a chamada da função *find\_file\_sign* para fazer a busca com base no path do arquivo. E com base nas assinaturas que foram encontradas o programa escreve na saída padrão os metadados do tipo de assinatura encontrada e a quantidade de matches.

```
def search_headers(dataset:pd.DataFrame, path:str):
    dataset['Found'] = 0
    for row in dataset.itertuples():
        hex_str = row[3]
        qtd_files = find_file_sign(hex_str, path)
        (
            dataset
                .loc[dataset['Hex'] == hex_str, ['Found']]
        ) = qtd_files
        if qtd_files > 0:
            print(
                dataset
                .loc[dataset['Hex'] == hex_str]
                .to_string(index=False)
            )
```

Por fim a função principal que recebe o caminho do arquivo alvo da análise, o path do arquivo JSON com as assinaturas e uma lista de cabeçalhos para filtrar a busca. No final o programa retorna os arquivos que encontrou e o tempo de execução

```
def run(path:str, path_dataset:str, headers:list[str]=None):
    start = time.time()
    dataset = read_dataset(path_dataset)
    dataset = filter_dataset(dataset, headers)
    search_headers(dataset, path)
    end = time.time() - start
    print("Tempo de execucao: {:.2f} s".format(end))
```

## 2.3 Execução do programa

Para executar o programa basta instalar o pacote pandas no interpretador Python3.9 que vai ser utilizado. E fazer e fazer a seguinte chamada:

Para fazer a busca por todas as assin

```
python3.9 main.py <path>
```

Exemplo:

```
python3.9 main.py ./imagem.img
```

```
python3.9 main.py <path> <lista de headers>
```

Exemplo:

```
python3.9 main.py ./imagem.img 0x89504e470d0a1a0a
```



## REFERÊNCIAS

Wikipedia contributors. *List of file signatures* — *Wikipedia, The Free Encyclopedia*. 2023. [Online; accessed 7-June-2023]. Disponível em: <[https://en.wikipedia.org/w/index.php?title=List\\_of\\_file\\_signatures&oldid=1156371620](https://en.wikipedia.org/w/index.php?title=List_of_file_signatures&oldid=1156371620)>. Citado na página 3.