

**Relatório**

**Processamento de Linguagens**

**Trabalho Prático 1 – Exercício 1**

**Alunos:**

**a16443 – Francisco Moreira Rebêlo**

**a21958– Gabriel Jablonsky**

**a23525 – Diogo Parente**

**Docente:**

**Óscar Ribeiro**

**Escola Superior de Tecnologia**

**Licenciatura em Engenharia de Sistemas Informáticos - PL**

Barcelos, 15 de abril de 2023

**Índice**

[1. Alínea A 4](#_Toc132558708)

[1.1. Expressão regular definida: 4](#_Toc132558709)

[1.1.1. Explicação: 4](#_Toc132558710)

[1.1.2. Parte 1 4](#_Toc132558711)

[1.1.3. Parte 2 4](#_Toc132558713)

[1.1.4. Parte 3 4](#_Toc132558714)

[1.1.5. Parte 4 4](#_Toc132558715)

[1.1.6. Teste com programa 4](#_Toc132558716)

[2. Alínea B + D 5](#_Toc132558717)

[3. Alínea C 5](#_Toc132558718)

[3.1. Tabela de transição 5](#_Toc132558719)

[3.2. Algoritmo para testes. 6](#_Toc132558720)

[3.2.1. Resultados obtidos 6](#_Toc132558721)

# Alínea A

## Expressão regular definida:

(((\+|-)[\d]+.[\d]+[E]+[\d]+)|([\d]+.[\d]+[E]+[\d]+)|((\+|-)([\d]+.[\d]+[E](\+|-)+[\d]+)|([\d]+.[\d]+[E](\+|-)[\d]+)))

### Explicação:

### Parte 1

### ((\+|-)[\d]+.[\d]+[E]+[\d]+) Números reconhecidos: +10.5E45

### Parte 2

([\d]+.[\d]+[E]+[\d]+)Números reconhecidos: 10.10E45

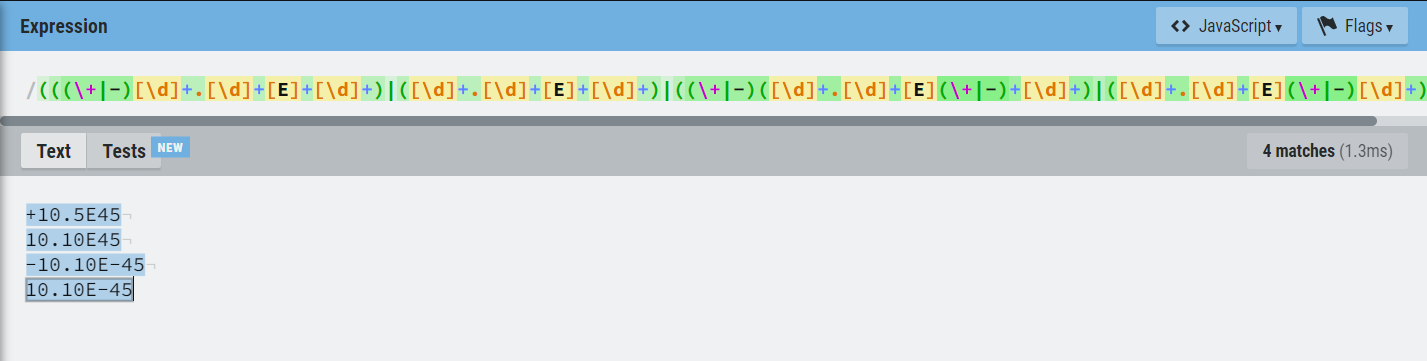
### Parte 3

(\+|-)([\d]+.[\d]+[E](\+|-)+[\d]+) Números reconhecidos: -10.10E-45

### Parte 4

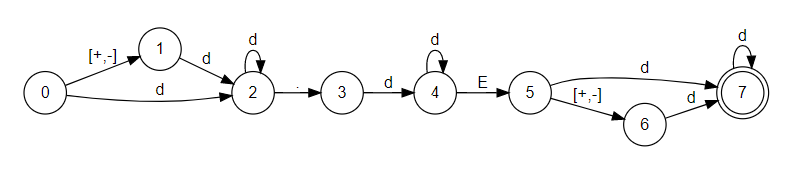
([\d]+.[\d]+[E](\+|-)[\d]+) Números reconhecidos: 10.10E-45

### Teste com programa rexegr



# Alínea B + D

## Autómato determinista em graphviz



# Alínea C

## Tabela de transição

## Algoritmo para testes.

V = {"+", "-", "range (10)", "."}

Q = {"q0", "q1", "q2", "q3", "q4", "q5", "q6", "q7"}

delta = {"q0": {"+": "q1", "-": "q1", "range (10)": "q2"},

         "q1": {"range (10)": "q2"},

         "q2": {"range (10)": "q2", ".": "q3"},

         "q3": {"range (10)": "q4"},

         "q4": {"range (10)": "q4", "E": "q5"},

         "q5": {"+": "q6", "-": "q6", "range (10)": "q7"},

         "q6": {"range (10)": "q7"},

         "q7": {"range (10)": "q7", "E": "q5"},

        }

q0 = "q0"

F = {"q7"}

def reconhece(palavra:str):

    def reconhecedigitos0\_9(c):

        return c.isdigit() and c in "0123456789"

    estado\_atual= "q0"

    tam = len(palavra)

    i = 0

    while (i<tam) and (estado\_atual != "Erro"):

        simbolo\_atual = palavra[i]

        if simbolo\_atual in V:

            estado\_atual = delta[estado\_atual][simbolo\_atual]

        elif  reconhecedigitos0\_9(simbolo\_atual):

            estado\_atual = delta[estado\_atual]["range (10)"]

        elif simbolo\_atual == "E":

            estado\_atual = delta[estado\_atual]["E"]

        else:

            estado\_atual = "Erro"

        i+=1

    return (estado\_atual in F) and (i==tam)

for exemplo in ["12.5", "-12.5", "12.55E-10", "-12.45E-10", "125", "12.55E10"]:

    print(f"'{exemplo}'\t{reconhece(exemplo)}")

### Resultados obtidos

