

# Plataforma Machine.io

## Visualizando Maquinas de Turing

Autor: Tony Esaú de Oliveira

### Introduzindo

[TuringMachine.io](https://turingmachine.io) é um simulador web interativo de Máquinas de Turing que permite criar, testar e visualizar o funcionamento de máquinas de forma intuitiva. A plataforma oferece uma interface gráfica onde é possível definir estados, transições e acompanhar a execução passo a passo sobre a fita, facilitando a compreensão prática dos conceitos teóricos de computabilidade. Além disso, permite exportar e importar configurações, tornando simples o compartilhamento de implementações através de arquivos ou URLs.

### Linguagem da Plataforma

A plataforma utiliza uma sintaxe baseada em YAML para definir Máquinas de Turing de forma clara e estruturada. Abaixo está a explicação de cada elemento:

#### 1. Estrutura Básica

1.1 Define a **entrada inicial** na fita da máquina.

```
input:  'aaaa'
```

1.2 Especifica o **símbolo branco** (célula vazia) da fita.

```
blank:  '  '
```

1.3 Indica o **estado inicial** da máquina.

```
start state:  q
```

#### 2. Tabela de Transições

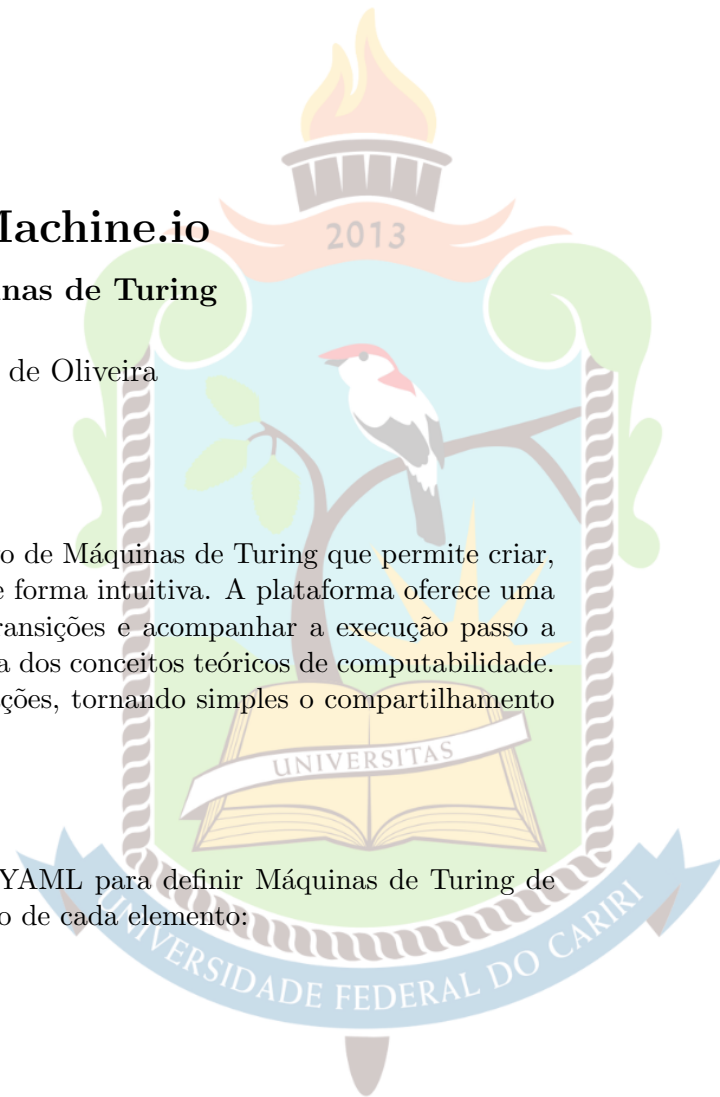
2.1 Contém todas as **funções de transição** da máquina, organizadas por estado.

```
table:
```

#### 3. Formato das Transições

3.1 Transição sem Escrita

```
qi:
  a: {R: q2}
```



Cada transição segue o formato:

- **qi**: estado atual;
- **a**: símbolo lido pela cabeça de leitura;
- **{R: qi+1}** Ação e próximo estado
  - R (Right/Direita), move a cabeça para a direita
  - **qi+1**, próximo estado

### 3.2 Direções Possíveis

- R – Direita (Right)
- L – Esquerda (Left)

### 3.3 Transição Completa

**qi**:

**a**: {**write**: b, **R**: **qi+1**}

- **write**: **b**: escreve 'b' na célula atual (opcional)
- **R**: **qi+1**: move para direita e vai para q2

### 3.4 Lista de Transições

Se em um estado um conjunto de símbolos têm a mesma transição, pode-se utilizar uma notação de lista ao invés de repeti-la várias vezes.

**qi**:

[a,b,c]: {**write**: d, **R**:q2}

## 4. Estados de Parada

Estados sem transições definidas são **estados de aceitação** (*halt states*). A máquina para e aceita ou rejeita a entrada.

**qa**:

## 5. Comentários

Linhas iniciadas com # são comentários explicativos.

# Este é um comentário



## Exemplo Concreto: [Exercício B01](#)

Exemplos de cadeia de entrada:

- Entrada: 000000100 → **Aceita** (contém um '1')
- Entrada: 000000000 → **Rejeita** (não contém '1')
- Entrada: 111111111 → **Aceita** (contém '1')
- Entrada: 1 → **Aceita**

Código da máquina:

```
input: '000000100'
blank: ' '
start state: q_0
table:

# q_0 possui um laço para buscar 0's.
q_0:
# Permanece em q_0 enquanto tiver 0's e não achar um 1.
0: {R: q_0}
# Se achar um 1, pule para aceitação.
1: {R: q_aceita}
# Se achar um ' ', percorreu toda a fita sem achar 1, rejeite a cadeia.
' ': {R: q_rejeita}

q_aceita:

q_rejeita:
```

Copie e cole esse código na plataforma, carregue a máquina, e veja o comportamento acontecendo na fita.

