

# Algoritmo

```
// Função para iniciar execução da Maquina de Turing
function executaMT(){
  const obj = JSON.parse(texto);
  var fita = palavra.split("");
  var estado = obj["EstadoInicial"];
  var pos = 0; // identifica posicao na fita
  var proxElemento = "";
  var direcao = 0; // Qual direcao vai a fita
  var msgFinal;
  var i = 0; // Contabilizador de tempo na execução das transições na tela
```

```
while(1){
  proxElemento = fita[pos];
  // Retorna resultado da transicao
  var transicao = transicaoSaida(obj, estado, proxElemento);
  var entrada = transicao[0];
 var saida = transicao[1];
  estado = saida[0];
  fita[pos] = saida[1];
  if (transicao == "erro"){
      msgFinal = "Palavra não aceita";
      break;
  console.log(entrada);
  marcaCelula(pos, direcao, entrada, saida, i);
  if (saida[2] == 'R'){
    direcao = 1;
    pos++;
  } else{
    direcao = -1;
    pos--;
  if (obj["EstadosFinais"].indexOf(estado) > -1){
    msgFinal = "Palavra aceita";
    break;
  i++;
```

```
// Identifica a proxima transição da MT
function transicaoSaida(obj, estado, proxElemento){
 var result = [];
 for (var [key, value] of Object.entries(obj["Transicao"])){
    var entrada = key.split("-");
    var saida = value.split(",");
    if (entrada[0] == estado && entrada[1] == proxElemento){
      result.push(entrada);
      result.push(saida);
      return result;
 return "erro";
```

# Sétupla

- Arquivo no formato JSON.
- Organizado por Chave: Valor.

# MTs Testadas

1.  $L=\{0^n1^n / n \ge 0\}$ 

Aceita

### Palavra

0011

Iniciar Simulação

#### Fita



## Transição

q3,B -> q4,B,R

# MTs Testadas

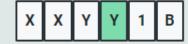
1.  $L=\{0^n1^n / n \ge 0\}$ 



### Palavra

00111 Iniciar Simulação

#### Fita



#### Transição

q3,Y -> q3,Y,R

## MTs Testadas

2. Max(m-n,0), m-n se m  $\ge$  n e 0 se m < n.

### **Palavra**

0000100

Iniciar Simulação

#### Fita



Transição

q4,B -> q6,0,R