



# Linguagens Formais e Autômatos

**Centro Universitário Senac**

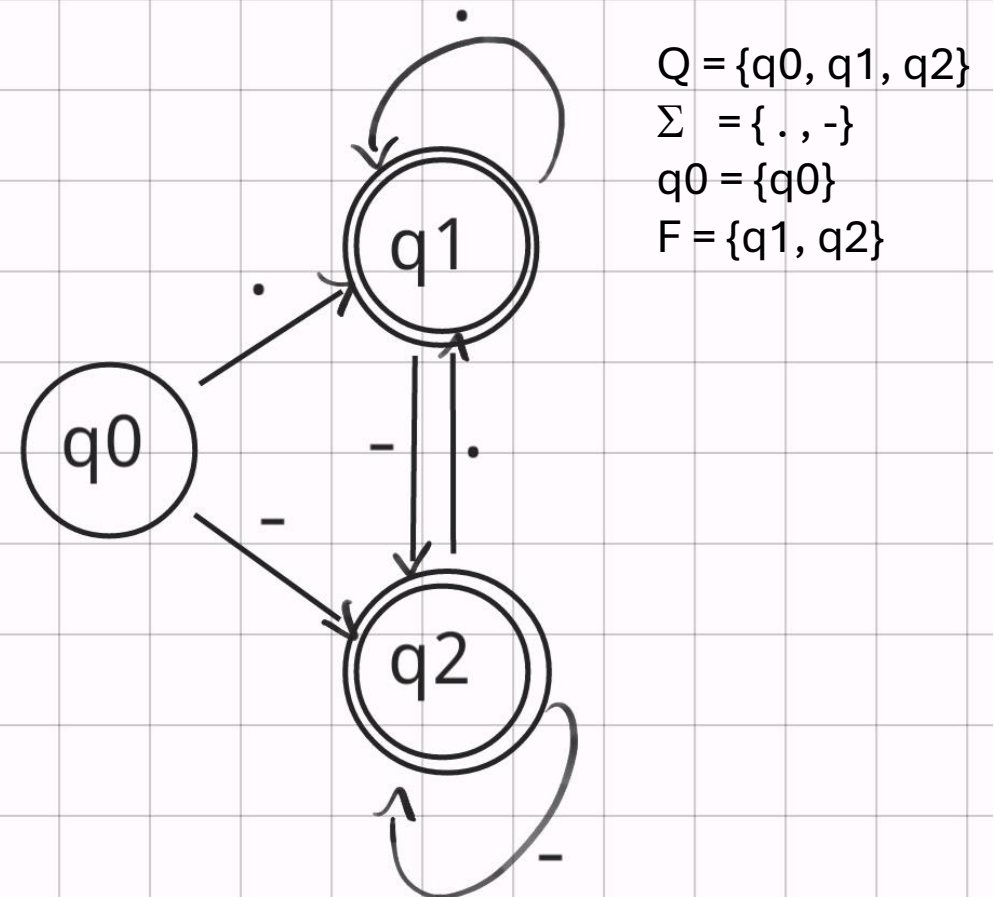
**Aluno: Vinycius Zanardi**

Ep 01 – Código Morse

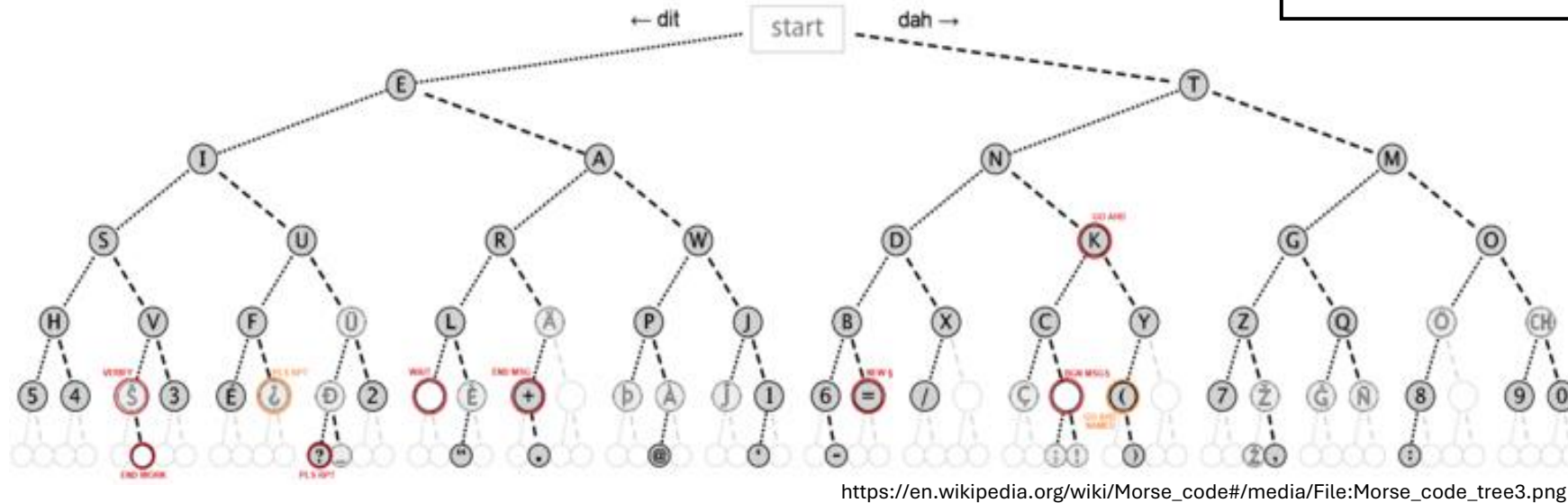
# Modelando o problema

A ideia parte de um ciclo de repetição originado de um ponto ou traço, tendo que não podemos ter uma palavra vazia. O autômato executa até encontrar um ponto de parada satisfatório.

Como não era permitido a utilização de dicionário, foi então pensado em outro método para traduzir a palavra para o seu respectivo símbolo (no próximo slide)



Alfabeto de Entrada: {., -}  
Alfabeto de Saída: {A..Z, 0..9, +, -, /, ?}



Com base na referência, foi tomada então uma nova estratégia onde cada possível estado seria um símbolo diferente de acordo com a palavra de entrada inserida, sendo dividida na árvore por ponto (.) ao lado esquerdo e traço (-) ao lado direito. Para formar palavras também foi adaptado a

utilização de espaço ( ) para o próximo símbolo e barra (/) para o espaço entre palavras. A tarefa exigia **todas** as transições de estados, então para todo o alfabeto, incluindo número e símbolos, foram necessários ao todo 41 estados diferentes para representação de cada um

# O código

Com base no código do professor, temos uma função que vai receber a cadeia, depois uma função para reconhecer caracter por caracter da cadeia e então iniciamos nosso ADF em q0 e damos continuidade nas transições de estados.

MorseTransdutor.rb

```
1 class ADF
2   def initialize(cadeia)
6     end
7
8     # pega apenas um caracter
9   def proximo
16    end
17
18
19   def iniciar
595    end
596  end
597
598
599  morse = ".-.-. ---.. .. --- / ---.. . / -.. . / . -.-. - - - . . . - -"
600  #Digite seu código entre "aspas" acima
601
602  adf = ADF.new(morse)
603  adf.iniciar
```

```
def iniciar
  input = ""
  texto = ""
  estado = "q0"
  puts "Máquina iniciou no estado: " + estado

  loop do
    case [proximo, estado]
    #estados de q0
    in [".", "q0"] # .
      estado = "q1"
      input = "E"
    in ["-", "q0"] # -
      estado = "q2"
      input = "T"
    in ["/", "q0"]
      estado = "q0"
      texto += " "
    in [" ", "q0"]
      estado = "q0"
```

## Entrada e Saida Válidas

```
Entre com o próximo símbolo:  
Estado: q6  
Entre com o próximo símbolo:  
Estado: q14
```

Você inseriu:

```
..- .- -.- / - - - - . . . / - . . / . - - - - - - - - -
```

Tradução: CODIGO MORSE DE EXEMPLO

## Entrada e Saida Inválidas

Output:

```
Máquina iniciou no estado: q0  
Entre com o próximo símbolo:  
Estado: q1  
Entre com o próximo símbolo:  
Estado: q3  
Entre com o próximo símbolo:  
Estado: q8  
Entre com o próximo símbolo:  
Estado: q18  
Entre com o próximo símbolo:  
Estado: q31  
Entre com o próximo símbolo:
```

Você inseriu:

```
..- - - .
```

Tradução: Erro! (Código Invalido)

# Links e Referências:

<https://crivelaro.notion.site/Transdutor-Tradutor-de-C-digo-Morse-358ed524e27142e9984ff7a9cb2df164#11a5c18272c045a181fc0457808f9e3b>

[https://en.wikipedia.org/wiki/Morse\\_code#/media/File:Morse\\_code\\_tree3.png](https://en.wikipedia.org/wiki/Morse_code#/media/File:Morse_code_tree3.png)

<https://morsecode.world/international/translator.html>

<https://replit.com/@celsosenac/automato-finito>