# Máquina de Estados Determinística: Uma implementação genérica

Daniel Vieira – 1257 Leandro Pessoa – 741 Délio Costa –

## **Objetivo**

Este trabalho tem como objetivo implementar uma máquina de estados finito determinística, recebendo como entrada um arquivo XML que contém todas as informações da máquina de estados que será usada, e uma palavra. Deve ser dado uma resposta do tipo SIM ou NÃO aceitando ou rejeitando a palavra recebida como entrada.

### Organização e estrutura dos dados

Para se armazenar as informações da máquina de estados, usou-se de duas classes chamadas *Estado*, que contém todas as informações de um estado como prefixos representados e seus respectivos próximos estados dado um prefixo, bem como o nome do próprio estado, e a classe *MaquinaDeEstados*, tendo como objetivo guardar todos os estados existentes, tamanho dos prefixos, e os estados final e inicial da máquina de estados.

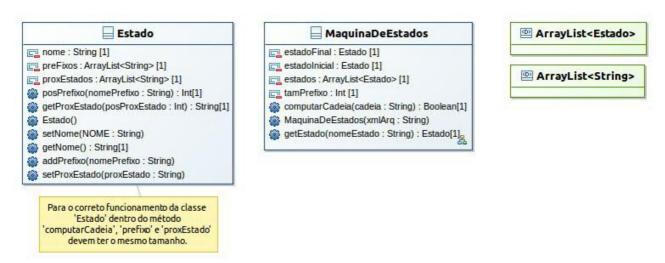


Figura 1: Diagrama de classes

### Lógica para consumir a palavra

A palavra é processada prefixo por prefixo, tendo um tamanho definido pela máquina de estados. Tem-se em anexo o diagrama de atividade do método.

- Enquanto a palavra n\u00e3o for totalmente consumida, \u00e9 retirado da palavra o prefixo ainda existente.
- 2. Procura o mesmo na lista de prefixos do estado atual (que na primeira iteração é o estado inicial).
- 3. Utiliza o índice retornado no passo anterior para obter o nome do próximo estado.
- 4. Caso o nome do próximo estado seja "" (definido como estado de erro), a palavra não é aceita, a iteração continua caso contrário
- 5. Buscar o próximo estado utilizando o nome recebido no passo anterior e atualizar o próximo estado.
- 6. Caso a palavra tenha sido totalmente consumida, testa-se se o estado atual é o estado final, caso verdadeiro, aceita-se a palavra, rejeita-se a palavra caso contrário.

### Organização e estrutura do arquivo XML

A estrutura do arquivo XML foi criada de acordo com os parametros presentes em uma tabela que representa a função de transição de estados onde estão presentes os nomes dos estados, prefixos e próximos estados associados a cada prefixo. Ainda se fez nescessario a criação de parametros para especificar o estado inicial, estado final e tamanho do prefixo, tambem é a representação do tipo de dados Estado criado neste trabalho..

O nó raiz possui os atributos em que os parametros estão presentes especificando estado inicial, estado final e tamanho de prefixo. Os nós filhos representam cada estado e internamente a esses nós existem nós mais internos especificando prefixos associados ao estado corrente e próximos estados associados a esses prefixos.

A estrutura criada para o XML permite que qualquer maquina de estados finitos seja implementada.