

**UNIVERSIDADE ESTADUAL DO PARANÁ**

**-**

***CAMPUS***

**APUCARANA**

**João Vitor Pastori Leme Batista**

**RELATÓRIO TÉCNICO**

**-**

**LFA**

APUCARANA

–

PR

2023

**João Vitor Pastori Leme Batista**

**RELATÓRIO TÉCNICO – LFA**

Trabalho apresentado à disciplina de Linguagens Formais, Autômatos e Computabilidade do curso de Bacharelado em Ciência da Computação.

**Professor:** Guilherme Henrique de Souza

Nakahata;

**APUCARANA – PR 2024**

## SUMÁRIO

**INTRODUÇÃO** ............................................................................................... 02

**CAPÍTULO 1: OBJETIVOS**  ........................................................................... 03

**CAPÍTULO 2: MOTIVAÇÃO E RECURSOS UTILIZADOS**  ........................... 04

**2.1 Motivação** ................................................................................................. 04

**2.1 Estrutura de Dados** ................................................................................. 05

**2.2 Linguagem de programação e demais informações** ............................ 07

**CAPÍTULO 3: RESULTADOS** ........................................................................ 07

[**CONCLUSÃO**](#_Toc9618) 09

[**REFERÊNCIAS** 09](#_Toc9619)

**INTRODUÇÃO**

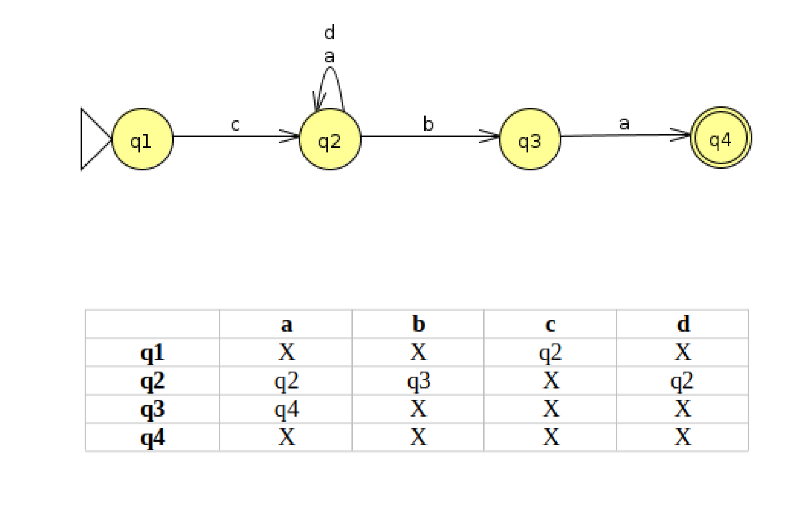
Diferente de outras disciplinas do curso de Ciência da Computação, Linguagens Formais e Autômatos apresentam um grau significativo de praticidade em suas atividades, permitindo que os alunos realizem tarefas "práticas" através da construção de autômatos finitos, autômatos a pilha e máquinas de Turing; focaremos aqui no primeiro tipo, o autômato finito determinístico (AFD). Durante as aulas, podemos perceber que há conexões diretas entre um AFD e a arquitetura e funcionamento dos computadores modernos, onde os estados do AFD correspondem aos estados do processador, as transições de estados a instruções de máquina, e os símbolos de entrada a bits processados.

Assim, surge uma oportunidade interessante para uma aplicação prática dos autômatos finitos determinísticos, dado que eles se relacionam diretamente com a lógica algorítmica básica. A utilização de programação para abordar situações da disciplina de LFA se revela eficaz, pois associa a matéria a outras disciplinas e facilita a construção de um conhecimento mais profundo e prático. Por exemplo, podemos mencionar uma atividade interdisciplinar realizada em uma IES Federal, onde, após o desenvolvimento de um programa que simula o funcionamento de um AFD, obteve-se um resultado positivo. Apesar de ser uma atividade aplicada no primeiro semestre do curso, cerca de 40% dos estudantes relataram dificuldades em aprender uma linguagem de programação e implementar o código; no entanto, mesmo enfrentando desafios, aproximadamente 88% dos alunos não acharam o projeto desestimulante.

Em síntese, existem diversos motivos para crer que a aplicação prática da disciplina, particularmente no que diz respeito a autômatos finitos determinísticos, é altamente benéfica e vantajosa para o aprendizado interdisciplinar. Além disso, o esforço necessário para desenvolver e implementar um código favorece uma assimilação mais profunda do conteúdo abordado em sala de aula.

## CAPÍTULO 1 OBJETIVOS

O objetivo principal do código criado pode ser interpretado como estabelecer o funcionamento prático de um autômato finito determinístico (AFD), mas, em um programa. De modo geral, o código fonte tem como objetivo receber uma descrição formal, ou seja, as características de um conjunto de estados que represente um AFD. Com isso, poderá avaliar a aceitação ou não de palavras mediante a linguagem informada por meio das transições. O AFD deve ser capaz de processar uma cadeia de símbolos e determinar se a palavra pertence à linguagem definida pela estrutura do autômato. Um exemplo de AFD do qual o algoritmo deve ser capaz de reconhecer a linguagem e verificar o pertencimento de palavras está representado na figura 1



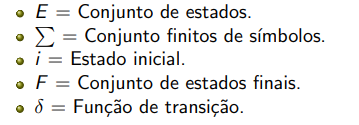
*Figura 1*

## CAPÍTULO 2 MOTIVAÇÃO E RECURSOS UTILIZADOS

Baseando-se no exposto anteriormente, devemos explicitar os motivos para a realização do trabalho, ou seja, o objetivo final e os recursos utilizados para que isso seja cumprido.

**2.1 Motivação**

Como discutido no capítulo que aborda os objetivos do projeto, a motivação também envolve a criação de um código fonte funcional que possa demonstrar o pleno funcionamento de um autômato finito determinístico (AFD). O objetivo é receber as informações necessárias e determinar se as palavras fornecidas pertencem ou não à linguagem definida pelo autômato, com base nas transições especificadas. Para isso, precisamos coletar dados, como a descrição formal e a tabela de transições, e desenvolver uma lógica de programação eficaz para alcançar o resultado desejado. Na figura 2, por exemplo, podemos ver como a descrição formal deve ser recebida e formatada conforme o modelo apresentado



*Figura 2*

Assim, faz-se necessário, com as informações pertinentes acerca dos objetivos e motivações, esmiuçar-se àqueles dados relevantes à Estrutura de Dados, Linguagem de Programação e demais questões acerca da implementação do código em questão.

**2.2 Estrutura de Dados**

Sob uma perspectiva geral, o código fonte é trabalhado com uma estrutura composta pela criação de uma matriz que representa as transições de um estado para outro com base nos símbolos do alfabeto do AFD. Essa matriz contém os estados de destino para cada combinação de estado atual e símbolo do alfabeto.

Visando um melhor aproveitamento e funcionamento, utilizamos uma matriz de transições tabela\_transicao, que armazena os estados futuros para cada combinação de estado atual e símbolo do alfabeto. A matriz é do tipo int, e em cada posição tabela\_transicao[i][j] temos o estado para o qual o AFD deve transitar a partir do estado i ao ler o símbolo alfabeto[j].

Para o recebimento dos dados iniciais, criamos vetores para armazenar os símbolos do alfabeto (alfabeto[]), os estados finais (estados\_finais[]), e a tabela de transição (tabela\_transicao[][]). Além disso, solicitamos ao usuário o estado inicial, a quantidade de estados, a quantidade de estados finais, os símbolos do alfabeto e as transições necessárias para estabelecer a linguagem do AFD. Para cada combinação de estado e símbolo do alfabeto, o usuário digita o estado de destino na tabela de transição.

Olhando para a parte da execução, o código conta com uma função verificar\_palavra que recebe a palavra de entrada e simula o processamento pelo AFD. Esta função percorre cada símbolo da palavra, atualizando o estado atual conforme as transições definidas na matriz. Se um símbolo não pertence ao alfabeto, a função retorna que a palavra contém um símbolo inválido. Caso contrário, verifica-se se o estado final alcançado é um estado de aceitação.

A execução é mantida até que todos os símbolos da palavra tenham sido processados. Se, ao final, o estado atual for um dos estados finais, a palavra é aceita. Caso contrário, a palavra é rejeitada. Outras estruturas de repetição e controle são utilizadas no programa para garantir o funcionamento correto da simulação do AFD.

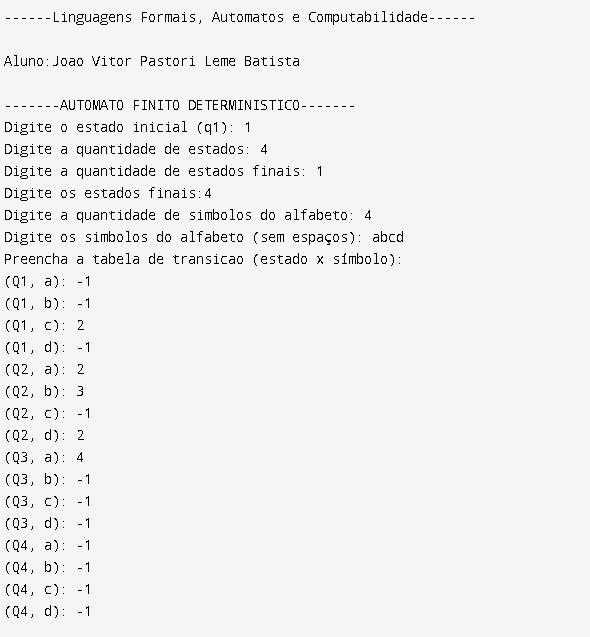
**2.3 Linguagem de Programação e demais informações**

A linguagem escolhida para a implementação do código foi C, uma linguagem de programação amplamente utilizada devido à sua eficiência e versatilidade. Desenvolvida originalmente por Dennis Ritchie nos Laboratórios Bell, C é conhecida por seu desempenho e capacidade de baixo nível.

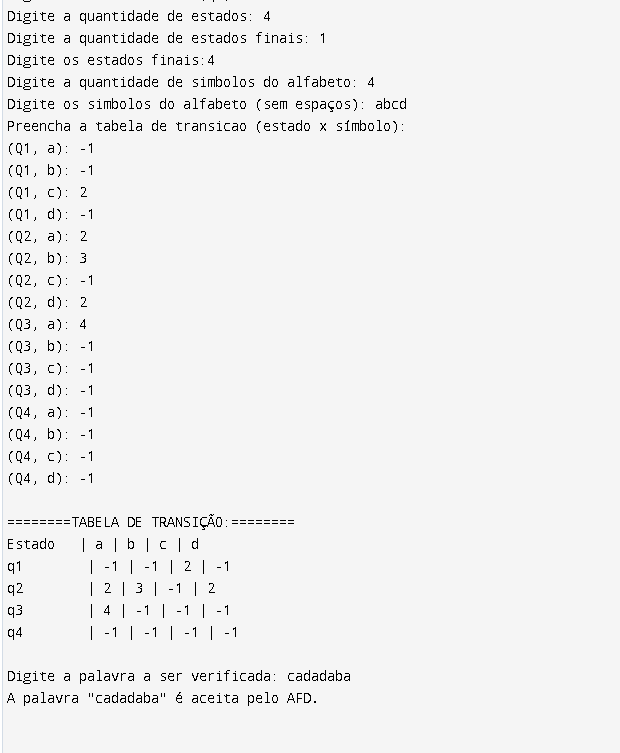
Embora não seja orientado a objetos como Java, o paradigma procedural é empregado para estruturar o programa de forma modular e organizada. Não foram utilizadas bibliotecas externas além das padrões, destacando a capacidade da linguagem de realizar operações fundamentais sem depender de recursos adicionais.

Essa escolha de linguagem não apenas aproveita o conhecimento adquirido no curso de formação, mas também demonstra a robustez e a eficiência de C em lidar com operações de baixo nível, como as envolvidas no benchmark de abertura, fechamento e exclusão de arquivos.

## CAPÍTULO 3 RESULTADOS

Mediante os objetivos apresentados, o resultado esperado seria o pleno funcionamento de um código que exemplifique como um autômato finito determinístico (AFD) reconhece ou não palavras de uma determinada linguagem, dado a sua descrição formal e tabela de transições. Dessa forma, com a implementação completa e revisada, os resultados foram atingidos, resultando em uma aplicação interativa e funcional. O código recebe a quantidade de símbolos no alfabeto, os símbolos, a quantidade de estados, a quantidade de estados finais, o estado inicial, os estados finais e as transições. 

*Figura 3*



*Figura 4*

# CONCLUSÃO

Com base no exposto anteriormente, podemos evidenciar que uma aplicação prática, mesmo em disciplinas que já possuem um caráter prático, pode ser vantajosa e apropriada para a obtenção de conhecimentos adicionais e para uma maior fixação do conteúdo visto em sala de aula. A atividade interdisciplinar citada anteriormente [2] demonstra que disciplinas, mesmo que superficialmente distantes, podem ter relações importantes. Um exemplo disso é a relação entre Linguagens Formais e Autômatos (LFA) e Programação Orientada a Objetos (POO), demonstrada nesta implementação específica pelo uso de estruturas de dados e funções em C.

De um modo geral, métodos diferenciados de ensino, por mais "simples" que sejam, podem resultar em influências significativas na conexão entre estudantes e o conhecimento. Segundo Valente (2002), metodologias ativas de ensino promovem uma aprendizagem mais significativa ao engajar os estudantes em atividades que exigem resolução de problemas e aplicação prática dos conceitos teóricos. Essas metodologias diminuem a abstração e aproximam os estudantes de uma resolução mais íntima dos problemas, uma vez que procuram por si mesmos meios de solução, ao mesmo tempo em que podem se comunicar e auxiliar seus colegas, transmitindo conhecimentos de formas que muitas vezes são difíceis de ocorrer entre discentes e docentes. Em suma, o código apresentado, escrito em C, desempenhou o papel esperado, tanto em termos de funcionamento quanto como objeto de aprendizado ativo e diferenciado.

# REFERÊNCIAS

Valente, J. A. (2002). "Educação a Distância via Internet." In: Educação a Distância: Desafios Contemporâneos. São Paulo: Editora Cortez.