



Prof^a M.A

Simulação de uma Máquina de Doces

Gabriel Toro Genaro – 822154467

Guilherme Cesar de Brito – 8222247316

Lucas Trajano de Araujo – 823160572

Paulo Alberto Abrahão Neto – 822139189

Victor Moreira Cruz – 823128528

Yagho Bastos de Lima - 822132354

1. INTRODUÇÃO

Este documento descreve a implementação de uma simulação de uma máquina de doces utilizando tecnologias web.

O objetivo é demonstrar a relação entre a lógica da máquina e um Autômato Finito Determinístico (AFD), abordando os principais estados e transições.

2. OBJETIVOS

- Implementar uma interface interativa para simular a máquina de doces.
- Gerenciar estados e transições através de um sistema baseado em regras.
- Utilizar conceitos de autômatos finitos para modelar o comportamento da máquina.

3. TECNOLOGIAS UTILIZADAS

A implementação foi realizada utilizando as seguintes tecnologias:

- HTML: Estrutura da interface da máquina.
- CSS: Estilização e layout visual.
- JavaScript: Controle da interatividade e estados da máquina.
- Flask/Python: Backend para processar a lógica e armazenar estados.

4. IMPLEMENTAÇÃO

4.1 Frontend (HTML, CSS, JavaScript)

- HTML: Criamos botões para inserir moedas, selecionar doces e exibir mensagens sobre o estado atual.
- CSS: Aplicamos estilos para tornar a interface amigável e intuitiva.
- JavaScript: Gerenciamos a lógica dos estados da máquina:
 - * Detectamos a inserção de moedas.

- * Atualizamos o display com informações sobre saldo e seleção de doces.
- * Enviamos requisições para o backend via fetch API para processar as transições de estado.

4.2 Backend (Flask/Python)

- Flask foi usado para criar a API que recebe as interações do frontend.
- Estados da máquina são controlados no backend, garantindo consistência na simulação.
- Regras de transição foram implementadas em Python para gerenciar o funcionamento:
 - * Validação de saldo suficiente.
 - * Atualização do estado após a seleção de um doce.
 - * Retorno de troco quando necessário.

5. RELAÇÃO COM AUTÔMATO FINITO DETERMINÍSTICO (AFD)

A lógica da máquina pode ser modelada como um AFD:

- Estados:

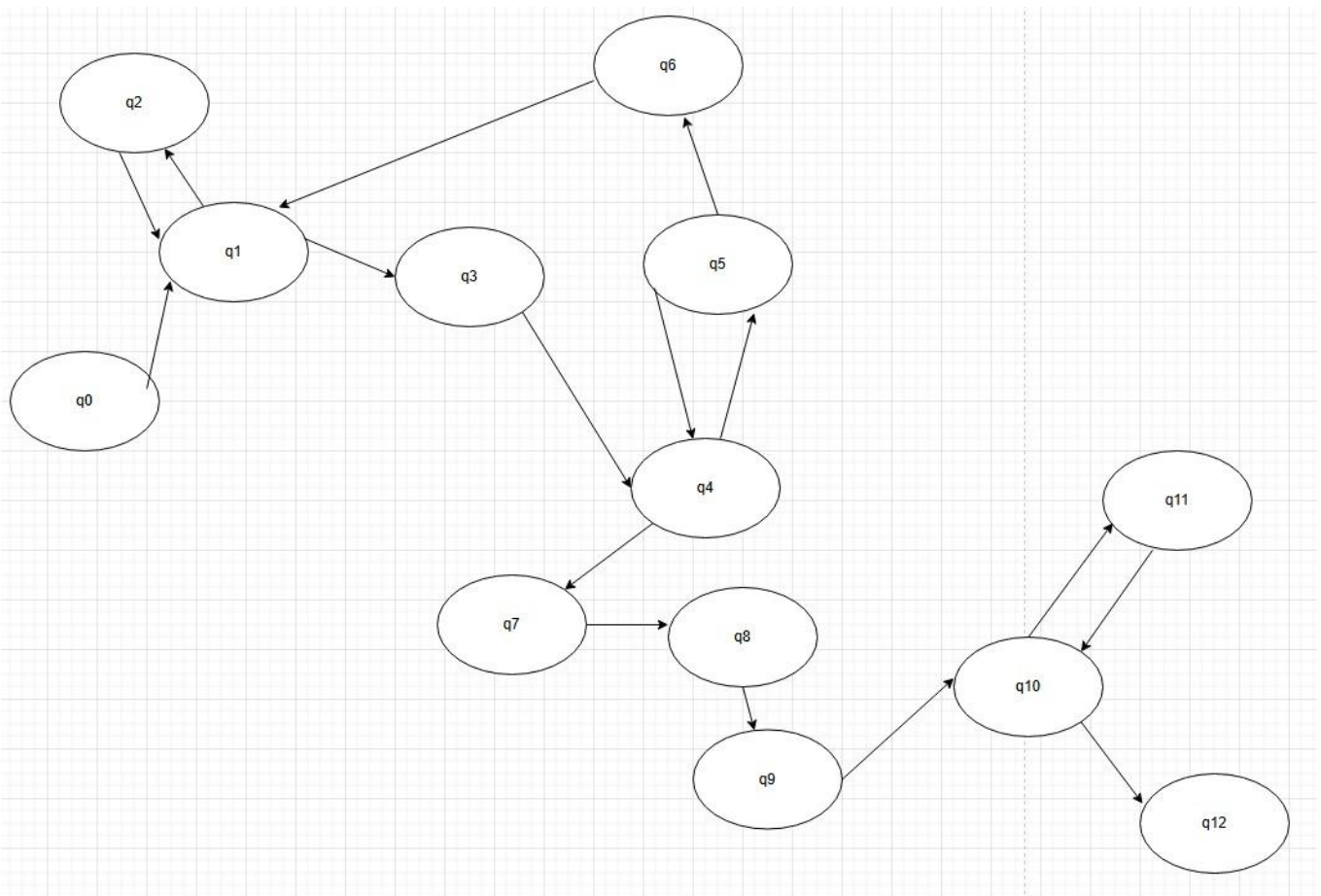
Q0: Início

- * Q1: Insere dinheiro ou moedas
- * Q2: Erro ao inserir
- * Q3: Valor inserido
- * Q4: Efetuar compra * Q5: Erro na compra
- * Q6: Saldo insuficiente
- * Q7: Adicionar ao histórico
- * Q8: Saída do doce
- * Q9: Calcular troco
- * Q10: Regatar troco
- * Q11: Erro ao resgatar
- * Q12: Fim

- Transições:

- * Q0 -> Q1 - O usuário insere dinheiro ou moedas.
- * Q1 -> Q2 - Ocorre um erro ao inserir dinheiro.
- * Q2 -> Q1 - O usuário tenta inserir o dinheiro novamente após o erro.
- * Q1 -> Q3 - O valor foi inserido corretamente.
- * Q3 -> Q4 - O usuário decide efetuar a compra.
- * Q4 -> Q5 - Ocorre um erro na compra.
- * Q5 -> Q4 - O usuário tenta efetuar a compra novamente após o erro.
- * Q5 -> Q6 - O erro foi causado por saldo insuficiente.
- * Q6 -> Q1 - O usuário retorna para inserir mais dinheiro.
- * Q4 -> Q7 - A compra foi bem-sucedida e adicionada ao histórico.
- * Q7 -> Q8 - O doce é liberado pela máquina.
- * Q8 -> Q9 - A máquina calcula o troco.
- * Q9 -> Q10 - O troco está pronto para resgate.
- * Q10 -> Q11 - Ocorre um erro ao tentar resgatar o troco.
- * Q11 -> Q10 - O usuário tenta resgatar o troco novamente após o erro.
- * Q10 -> Q12 - O troco foi resgatado com sucesso e o processo chega ao fim.

Cada uma dessas transições é controlada pelo backend, garantindo que a máquina funcione corretamente conforme os princípios dos autômatos finitos.



6. CONCLUSÃO

A implementação da máquina de doces permitiu demonstrar como conceitos de autômatos finitos podem ser aplicados na prática.

A separação entre frontend e backend garantiu uma estrutura modular e de fácil manutenção, tornando o sistema flexível para futuras melhorias.