

Universidade Federal de Santa Catarina  
Centro Tecnológico  
Departamento de Informática e Estatística  
Ciência da Computação  
INE5411 - Organização de Computadores I

## **Trabalho Final - Bomba de Combustíveis (PRJ04)**

Joshua Cruz do Amaral (24205457)  
Julia Macedo de Castro (23250860)

Florianópolis  
2025

## 1. Descrição do Projeto

O presente projeto consiste no desenvolvimento de um sistema de controle para uma bomba de combustíveis, implementado em linguagem Assembly para a arquitetura MIPS e executado através do simulador MARS. O objetivo principal é simular as funções reais de uma bomba de posto, gerenciando o fornecimento de três tipos distintos de combustíveis: Gasolina Comum, Gasolina Aditivada e Álcool.

O sistema utiliza uma interface híbrida: o **Digital Lab Sim** é empregado para a interação com o menu principal da aplicação, enquanto que para a entrada de valores precisos (como a definição de preços ou quantidade de litros) utilizam-se as janelas de entrada padrão do simulador (**MIPS Keyboard Input**). Essa abordagem facilita o tratamento de números inteiros e em ponto flutuante, oferecendo uma interface mais amigável ao operador.

Além disso, o sistema faz uso do Coprocessador 1 (FPU) para realizar operações aritméticas de precisão com os preços e volume de Litros, ao mesmo tempo que atende ao requisito de simulação de vazão, onde o abastecimento ocorre na velocidade de 1 litro por segundo, por meio da chamada syscall 30 (leitura do relógio do sistema), criando um temporizador (delay) que simula a duração correta do processo de abastecimento antes de sua conclusão.

As funcionalidades finais incluem a capacidade de o operador alterar os preços dos combustíveis, como já mencionado acima, ainda em tempo de execução e também a geração de um cupom fiscal em arquivo de texto (.txt) ao término de cada operação, registrando o tipo de combustível, a quantidade abastecida e o valor total do pagamento.

## 2. Manual do Usuário

A operação do sistema ocorre de forma híbrida, exigindo que o usuário interaja tanto com o componente **Digital Lab Sim** (para navegação no menu principal) quanto com as caixas de diálogo **MIPS Keyboard Input** (para entrada de dados numéricos).

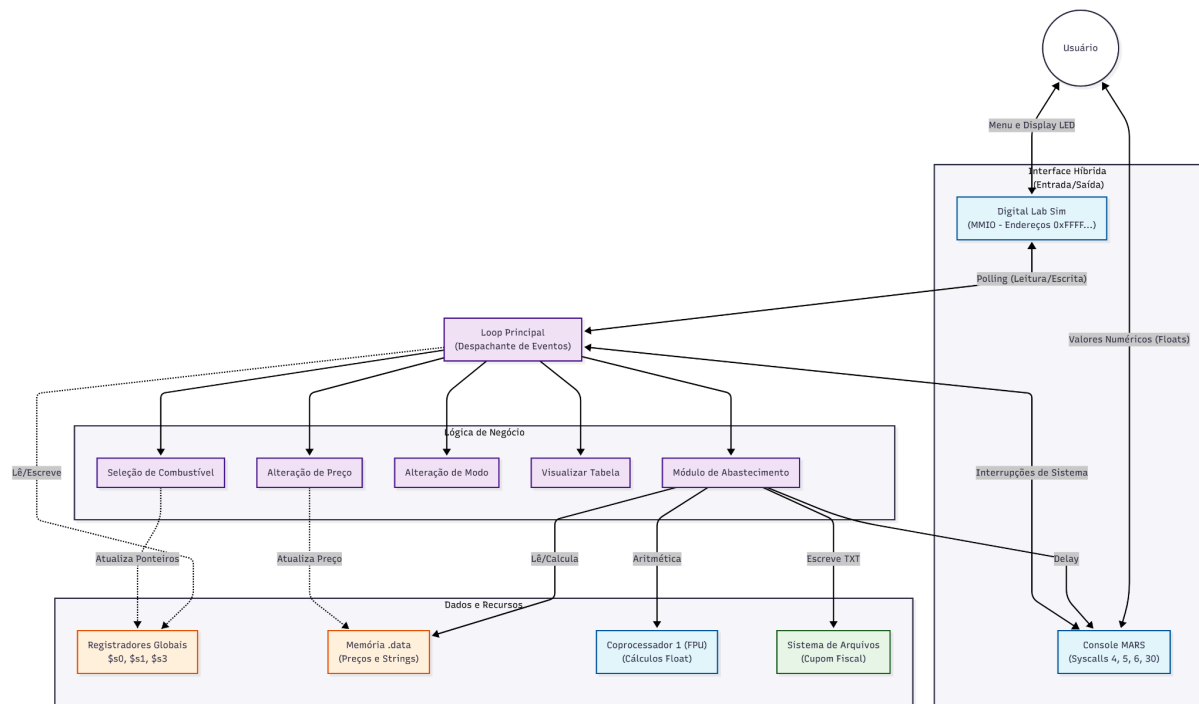
**Menu Principal (Via Digital Lab Sim)** Ao iniciar a execução, o display de sete segmentos exibe a opção selecionada. O usuário deve utilizar o teclado numérico do Digital Lab Sim para escolher uma das seguintes funções:

- **Tecla 1 - Selecionar Combustível:** Acessa a configuração do tipo de combustível a ser bombeado.
- **Tecla 2 - Alterar Preço:** Função para atualizar o valor do litro de qualquer combustível.
- **Tecla 3 - Abastecer:** Inicia o processo de abastecimento com base nas configurações atuais.
- **Tecla 4 - Alterar Modo de Abastecimento:** Alterna a bomba entre o modo "Por Litros" (L) e "Por Valor" (R\$). O modo padrão inicial é "Por Litros".
- **Tecla 5 - Visualizar Tabela de Preços:** Exibe no console a lista atualizada de todos os combustíveis e seus respectivos preços.
- **Tecla 6 - Encerrar Programa:** Finaliza a execução do simulador.

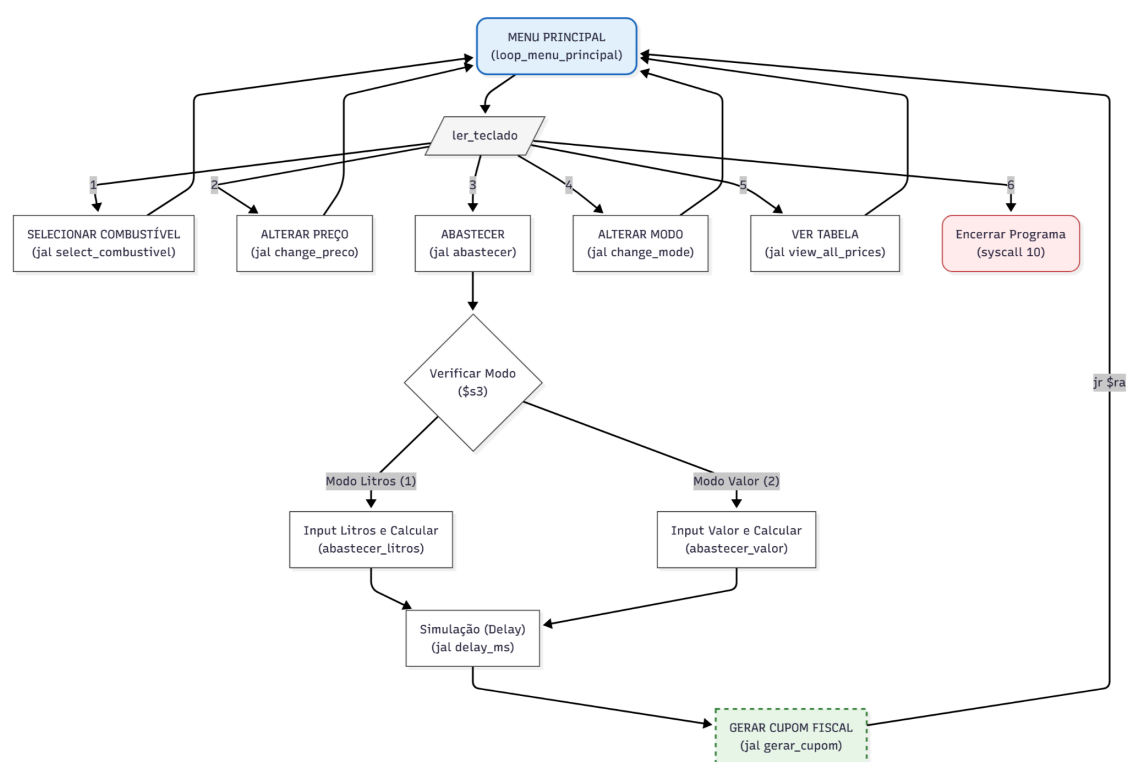
**Entrada de Dados (Via MIPS Keyboard Input)** Após selecionar uma opção no menu principal que exija inserção de valores, uma janela (*pop up*) intitulada "**MIPS Keyboard Input**" aparecerá na tela. O usuário deve atentar-se às seguintes instruções:

- **Seleção de Opções:** Quando solicitado a escolher um tipo de combustível (nas opções 1 e 2), deve-se digitar o número correspondente e pressionar Enter:
  - Gasolina Comum
  - Gasolina Aditivada
  - Álcool
- **Entrada de Valores (Floats):** Para definir novos preços ou a quantidade a ser abastecida, o sistema aceita números decimais.
  - *Atenção:* Deve-se utilizar o **ponto (.)** como separador decimal, e não a vírgula (exemplo: digitar 5.49 e não 5,49).
- **Simulação de Abastecimento:** Ao confirmar os dados na opção 3 (Abastecer), o sistema iniciará uma contagem de tempo simulando a vazão da bomba (1 litro por segundo). O usuário deve aguardar a mensagem "Abastecimento concluído!" antes de realizar nova operação. Ao final, um cupom fiscal em formato de texto (.txt) será gerado no diretório do projeto (pasta onde Mars4\_5.jar está sendo executado), e o nome do arquivo de texto gerado é "cupom.txt".

### 3. Diagrama em blocos



### 4. Fluxograma dos procedimentos



## 5. Considerações finais

O desenvolvimento deste projeto permitiu a consolidação prática dos conceitos fundamentais da arquitetura MIPS, exigindo a integração de diferentes mecanismos de entrada e saída. Uma estratégia essencial para a viabilidade do projeto foi a reutilização de conhecimentos adquiridos em laboratórios anteriores. Especificamente, a lógica de manipulação de arquivos desenvolvida no **Lab 02 (Matrizes)** serviu como base estrutural para a implementação da funcionalidade de geração do **cupom fiscal**, permitindo a escrita dinâmica dos dados da transação em um arquivo de texto.

Ainda no módulo de geração do Cupom Fiscal, optamos por registrar o valor total da transação como um número inteiro. Essa decisão foi motivada pela alta complexidade envolvida na conversão manual de valores em ponto flutuante para cadeias de caracteres (ASCII) em linguagem Assembly, sem o uso de bibliotecas externas. Dessa forma, o sistema garante a integridade dos dados gravados e a estabilidade do código, cumprindo o requisito de registrar o valor da operação de maneira legível e eficiente.

Em questão de aprendizados, a equipe conseguiu desenvolver ainda mais o entendimento sobre o funcionamento da ferramenta Mars e da linguagem Assembly em um projeto real e engajante. Esse trabalho também apresentou desafios técnicos, principalmente na ferramenta Digital Lab Sim, mas que ampliaram a nossa área de conhecimento. É certo que esse projeto corroborou para nosso crescimento tanto como alunos mas também como futuros profissionais da área.