



UNIVERSIDADE FEDERAL DE SANTA CATARINA- UFSC
ENGENHARIA ELETRÔNICA

LUCAS ARMANDO CIELLO

Relatório preliminar projeto final

Florianópolis, 2018

LUCAS ARMANDO CIELLO

Relatório técnico apresentado como requisito parcial para obtenção de aprovação na disciplina Programação C++ para sistemas embarcados, no Curso de Engenharia Eletrônica, na Universidade Federal de Santa Catarina.

Prof. Dr. Eduardo Augusto Bezerra

Florianópolis, 2013

Sumário

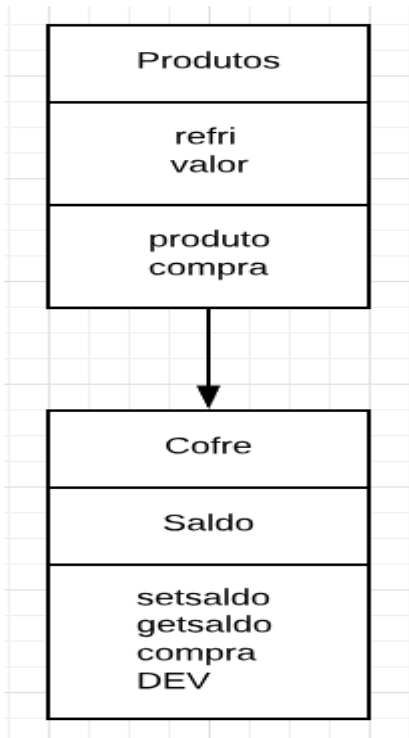
Resumo	3
Diagrama de Classes	4
Maquina de estado	4
Filas	4
Maquina de estado.....	5
Funcionamento da maquina de estado.....	5
Explicação Código	6
Ferramentas utilizadas	6

Resumo

Este relatório apresenta o diagrama de classes preliminar, o funcionamento da maquina de estado, bem como seu código em C++ e a descrição do sistema embarcado a ser utilizado para implementar o projeto.

Diagrama de Classes

Maquina de estado



O Diagrama é referente a maquina de estados, onde temos as classes produtos onde são armazenados os dados dos refrigerantes, bem como a classe cofre que é responsável pela gestão do dinheiro e da liberação do produto.

A classe cofre recebe a classe produto, tendo acesso aos seus atributos e métodos.

Filas

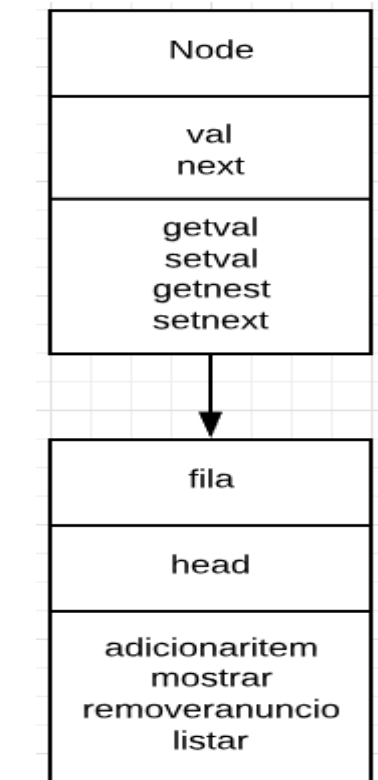


Diagrama referente às filas que serão utilizadas para implementação da dinâmica de propagandas.

A classe fila recebe a classe Node, tendo acesso aos seus atributos e métodos.

O programa final receberá as classes da Máquina de estado bem como da fila, possibilitando a implementação de um programa que fará toda a gestão da máquina de refrigerante e em paralelo serão apresentadas as propagandas cadastradas pelo utilizador.

Maquina de estado

A máquina fornece dois tipos de refrigerantes, MEET e ETIRPS, que estão disponíveis para ser escolhidos no menu do programa. Ambos os refrigerantes custam R\$1,50 e a máquina reconhece moedas de R\$1,00, R\$0,50 e R\$0,25, no código o usuário indica a moeda que está colocando, visto que nesta etapa não será utilizado nenhum tipo de sensor ou botão. A máquina devolve automaticamente o troco quando o saldo da máquina passa de R\$1,50, além disso, durante a compra, o usuário pode desistir da transação, a opção DEV devolve as moedas inseridas até o momento. Somente após acumular um crédito de R\$1,50 o usuário pode obter um refrigerante.

Funcionamento da maquina de estado

Estado atual	Comando de Entrada						
	Nada	M025	M050	M100	DEV	MEET	ETIRPS
S000	S000	S025	S050	S100	S000	S000	S000
S025	S025	S050	S075	S125	S000, D025	S025	S025
S050	S050	S075	S100	S150	S000, D050	S050	S050
S075	S075	S100	S125	S150, D025	S000, D075	S075	S075
S100	S100	S125	S150	S150, D050	S000, D100	S100	S100
S125	S125	S150	S150, D025	S150, D075	S000, D125	S125	S125
S150	S150	S150, D025	S150, D050	S150, D100	S000, D150	S000, DMEET	S000, DETIRPS

Através dessa tabela podemos verificar todas as possibilidades e o comportamento da maquina de estado.

Explicação Código

O código completo da maquina de estado está como anexo, bem como no github (<https://github.com/lucasciello/Projeto-final>).

A classe produto é utilizada para armazenar os dados dos refrigerantes, a classe cofre é filha da classe produto, tendo acesso a seus atributos e métodos. A classe cofre é responsável por cuidar do saldo e da liberação do refrigerante, Ambas as classes têm seus construtores que colocam o programa em estado inicial. A função setsaldo é responsável por adicionar o saldo sempre que o usuário informar que uma moeda foi inserida na maquina, adicionando o saldo anterior ao valor da moeda adicionada, sempre que o saldo ultrapassar o valor de R\$1,50, a função devolve o saldo excedente. A função compra faz um teste para verificar se o saldo é de R\$1,50, caso seja, ela libera o refrigerante escolhido e o valor do refrigerante é descontado do saldo. A função DEV devolve o saldo acumulado na maquina, caso ele exista. Este código é preliminar, podendo ser alterando conforme a necessidade ao longo do projeto.

Ferramentas utilizadas

Para implementação do projeto serão utilizados os conceitos vistos em aula, herança, friends, template, funções virtuais, sobrecarga de operador, polimorfismo, dentro outros métodos que serão necessários para o funcionamento do projeto. O sistema embarcado escolhido para simular o projeto foi o Raspberry PI.