

# Controle integrado e distribuído de tomadas inteligentes

Luca Fachini Campelli

Marcio Monteiro

Rodrigo Pedro Marques

Florianópolis, 2016

## Resumo

aqui vai resumo

**Palavras-chaves:** tomadas inteligentes, sistemas operacionais, eposmote2.

## 1 Introdução

Este trabalho foi realizado por Luca Campelli, Marcio Monteiro e Rodrigo Pedro Marques para a disciplina INE5412 - Sistemas Operacionais I, no semestre 2016.1. O objetivo principal deste trabalho é implementar o descobrimento automático de tomadas em uma rede onde elas possam tomar decisão sobre o consumo energético.

Para a realização deste projeto, foram utilizados três placas EPOSMote II ([LISHA, 2016b](#)) que simularam o comportamento de tomadas inteligentes. Para que isto fosse possível, o projeto foi subdividido em partes. São elas: realizar a comunicação entre as placas via *broadcast*, propor o algoritmo para consumo inteligente de energia, implementação deste algoritmo, validação e testes do algoritmo.

Este relatório está organizado da seguinte forma: em [2](#) são apresentadas as etapas em maior detalhes relacionadas ao desenvolvimento deste projeto; em [3](#) são apresentados os testes realizados após o desenvolvimento; em [4](#) são apresentadas as considerações finais e conclusões em relação a este projeto; finalizando este relatório, na seção [5](#) apresentamos possíveis temas para serem discutidos posteriormente.

## 2 Desenvolvimento

Para que o projeto pudesse ser desenvolvido, primeiramente foi necessário descobrir como era realizada a comunicação entre as placas. Para tal, foram realizados exemplos que constam na documentação do EPOSMote II sobre comunicação entre placas via *broadcast*. No código a seguir é possível observar como foi implementado o código para alcançar este objetivo no projeto.

---

```
int Mensageiro::receberViaNIC(){
    NIC nic;
    NIC::Address src;
    unsigned char prot;
    infoTomadas meg;

    while(1){
        while(!nic.receive(&src, &prot, &meg, sizeof(meg)) > 0);
    }
}
```

```

        if(meg.address != 0){
            gerente->receberMensagem(meg);
        }
        meg.address = 0;
    }
    return 0;
}

Mensagemiro::Mensagemiro(Gerente * gnt){
    gerente = gnt;
    Thread *thread;
    thread = new Thread( \&receberViaNIC );
}

```

Como é possível observar, foi utilizada a função NIC, já presente na no EPOSMoteII. Ela é responsável por prover acesso à redes.

A figura 1 apresenta a modelagem em UML do projeto. A classe *Tomada* representa uma tomada física, inteligente ou não. Neste projeto foram consideradas quatro tipos de tomadas: uma tomada simples, que liga e desliga; uma tomada com *dimer*, que possibilita *dimerizar* (controlar) a energia que passa por ela; uma tomada com sensor, onde é possível medir o seu consumo, colocar em modo de economia de energia, atribuir um limite de consumo, atribuir uma prioridade a ela e verificar o seu consumo mínimo e máximo já registrado; e uma tomada *top* que possui todas as funções das tomadas citadas anteriormente. Cada tomada possui um *Gerente Monitor* responsável por monitorar e atualizar a tomada, além de enviar e receber mensagens de outras tomadas no ambiente. Além disto, ele mantém o endereço de todas as tomadas no ambiente, guarda informações (previsão do consumo de energia, endereço da tomada, prioridade e tipo) da tomada monitorada por ele e também possui um *Mensagemiro* e um *Previsor*. A classe *Mensagemiro* é responsável por receber mensagens via NIC ou UART. O *Previsor* auxilia o gerente realizando as previsões de consumo da tomada gerenciada por ele. Esta classe pode fazer as previsões de consumo diárias, mensal e de todas as tomadas do ambiente.

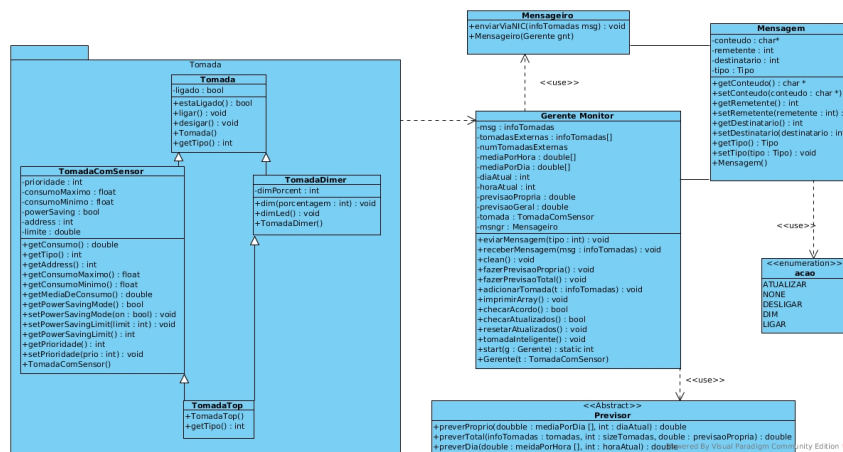


Figura 1 – UML do Projeto.

- 3 Validação e Testes
- 4 Considerações Finais
- 5 Trabalhos Futuros

## Referências

LISHA. *LISHA*. 2016. Acesso em 20 de jun de 2016. Disponível em: <<https://lisha.ufsc.br/HomePage>>. Nenhuma citação no texto.

LISHA. *Welcome to the EPOS Project*. 2016. Acesso em 20 de jun de 2016. Disponível em: <<https://epos.lisha.ufsc.br/HomePage>>. Citado na página 1.