

**UFES - CENTRO TECNOLÓGICO**  
**DEPARTAMENTO DE INFORMÁTICA**  
**Prof. Thomas W. Rauber**  
**Estruturas de Dados I – Engenharia de Computação & Ciência da Computação**

**1º Trabalho 2015/1 – Administração de Leitura de Consumo de Energia Elétrica**

Última atualização: 4 de abril de 2015, 14:55

Data de entrega: veja [www.inf.ufes.br/~thomas](http://www.inf.ufes.br/~thomas)

Linguagem de Programação para Implementação: C

Grupo de até dois alunos

## 1 Objetivo

Representação e manipulação de informação estruturada por linguagem de programação de alto nível.

Como aplicação considere a rotina de um leiturista de uma empresa de energia elétrica. Existem cidades com bairros, ruas, casas e/ou prédios com apartamentos. A empresa tem que administrar o cadastro dos consumidores e medir o consumo através do leiturista que percorre um roteiro programado.

## 2 Estrutura do Sistema

### 2.1 Organização da Infraestrutura

A identificação das unidades envolvidas é ilustrada na tabela 1. Cada unidade tem que ser identificável sem ambiguidade por um identificador numérico e conter informações complementares úteis. Como simplificação assume-se que não existam prédios com múltiplas unidades, somente casas com um único consumidor.

A estrutura das unidades é hierárquica. Uma cidade tem vários bairros. Um bairro tem várias ruas, uma rua tem várias casas. Como simplificação assume-se que não existam ligações entre casas de diferentes ruas e ruas entre diferentes bairros.

Unidade	Parâmetros
Bairro	id, Nome
Rua	id, Nome
Casa	id do consumidor, Número de casa, Consumo, Nome do consumidor

Tabela 1: Unidades do sistema de medição de energia. id = Identificador

### 2.2 Funções

O sistema deve providenciar funcionalidades administrativas e operacionais. Por exemplo como administração, deve ser possível incluir uma nova rua dentro de um bairro (supõe-se que as cidades e bairros sejam invariantes), ou uma nova casa dentro de uma rua de um determinado bairro. Como função operacional deve ser possível, por exemplo, medir o consumo de um consumidor, rua, bairro e da cidade no total.

A funcionalidade das unidades envolvidas é ilustrada na tabela 2, separado por operações administrativas e funções operacionais. Quando uma casa for introduzida no sistema, a ordem da enumeração deve ser preservada, por exemplo se existirem as casas com os números 12 e 18, e a casa 16 for introduzida, deve ser entre as duas existentes. O sistema deve ser robusto em relação à inconsistências, por exemplo retornar uma mensagem de erro, se uma unidade não existir, ou já existir na inclusão.

Unidade	Função	Parâmetros
ADMINISTRATIVAS		
Bairro	Incluir na cidade (somente na inicialização)	veja tabela 1
Rua	Incluir no bairro	id bairro, id rua, nome
Rua	Eliminar do bairro	id bairro, id rua
Casa	Incluir na rua	id bairro, id rua, e tabela 1 (consumo inicial zerado)
Casa	Eliminar da rua	id bairro, id rua, id casa
OPERACIONAIS		
Casa	Consumir	id bairro, id rua, id casa, Consumo
Casa	Medir consumo	id bairro, id rua, id casa
Rua	Medir consumo	id bairro, id rua
Bairro	Medir consumo	id bairro
Cidade	Medir consumo	—

Tabela 2: Funcionalidades das unidades.

### 3 Representação de Informação

As unidades cidade, bairro e rua devem ser organizadas como listas encadeadas, ou seja, a cidade é uma lista de bairros encadeados, um bairro é uma lista de ruas, uma rua uma lista de casas. Ajuda: Cada unidade poderia ser implementada como estrutura que tem as componentes da tabela 1 e eventualmente tem ponteiros para unidades do mesmo tipo subunidades. A cidade pode ser implementada facilmente como ponteiro para bairro. A inicialização da cidade, inserindo vários bairros pode ser uma função de inicialização fixa, pois os bairros não mudam neste modelo.

### 4 Interface

Toda a interação deve funcionar através de arquivos de entrada e saída. A sintaxe dos comandos do arquivo de entrada deve ser conforme a tabela 2. Cada comando tem uma composição fixa da forma

<Unidade> <Ação> <Parâmetros>

Exemplos:

```

rua incluir 17 3 "Av. Fernando Ferrari"
casa eliminar 17 23 8
casa consumir 17 23 8 255
bairro medir 17
cidade medir

```

Os elementos da linha de comando são separados por espaço em branco ou caracter de tabulação ( ' ' ou ' ') na linguagem C. Nomes em forma de cadeias de caracteres devem aparecer entre aspas, como no exemplo acima. O arquivo de saída deve emitir uma confirmação ou rejeição, em caso de inconsistência, para cada comando. Por exemplo o pedido de medir o consumo de uma rua deve retornar, em caso de comando válido, Nome e id do bairro, nome e id da rua, número de casas e consumo total.

### 5 Elaboração e Documentação

A pasta do código contém uma moldura para ser estendida. Pode-se modificar o projeto, por exemplo dividir o código que falta em vários arquivos e modificar o `Makefile` em correspondência. O resultado deve ser um sistema que lê um arquivo de entrada com a lista de comandos, um por linha, e produz o arquivo de saída.

O produto final deve ser um arquivo no formato `zip` com a seguinte sintaxe: `aluno1+aluno2.zip`. Atenção: Nenhum executável ou código objeto pode estar dentro do projeto, pois os serviços de e-mail, como, por exemplo, Hotmail recusam o transporte de tais arquivos por razões de segurança. O aluno se responsabiliza pelo envio

e recepção correta. Em caso de problemas maiores de tráfego de rede (serviços UFES fora do ar), o aluno deve mandar novamente o arquivo original (encaminhamento da mensagem original) quando o serviço voltar. O arquivo deve conter uma **única** pasta com o nome **aluno1+aluno2**. Duas subpastas devem conter o código fonte e a documentação do projeto. O código deve ter um **Makefile** que me permite a compilação facilitada. Devem-se produzir vários desenhos para testar as qualidades do software desenvolvido (subpasta **fig**). O arquivo **aluno1+aluno2.zip** deve ser mandado como anexo **exclusivamente** copiando a hiperligação seguinte no browser ou cliente de E-mail:

`mailto:thomas@inf.ufes.br?subject=Estruturas%20de%20Dados%20I:%20Entrega%20de%20trabalho%20`

A documentação deve ser em forma de descrição de projeto, preferencialmente gerado por  $\text{\LaTeX}$ , contendo os seguintes tópicos:

- Capa do Projeto
  - Título
  - Autoria
  - Data
  - Resumo
- Introdução
- Objetivos
- Metodologia
- Resultados e Avaliação
- Referências Bibliográficas

**Rigidez na administração da memória dinamicamente alocada:** Recomenda-se fortemente usar a ferramenta **valgrind**, e conseqüentemente uma variação do sistema operacional Unix. O vazamento e/ou violação de memória constitui uma degradação de qualidade do software e se refletirá na avaliação do trabalho.

*Bom trabalho!*