

# **Trabalho Final**

## **Sistemas Operacionais I**

Florianópolis, 15 de Abril de 2015.

**Tema do projeto:** aplicação em PC para configuração de rede, captura de tela, instalação de pacotes do SO e outras ações de um sistema de gerenciamento de aula em laboratório de informática.

**Sobre o projeto:** o projeto final da disciplina “*Sistemas Operacionais I*” é denominado *LabSpy* e será disponibilizado de forma gratuita e de código-aberto. O objetivo principal do programa é melhorar o processo de ensino-aprendizagem em laboratórios de computadores fornecendo ao docente ferramentas para monitoramento e controle dos computadores dos discentes.

**Grupo:**

Caique Rodrigues Marques,  
Emmanuel Podestá Junior,  
Fernando Jorge Mota,  
Fernando Paladini.

**Conteúdo abordado:**

1. Requisitos
  - 1.1. Requisitos funcionais
  - 1.2. Requisitos não-funcionais
2. Diagramas
3. Funcionamento planejado (**importante**)

# 1 Requisitos

## 1.1 Requisitos Funcionais

Os requisitos funcionais do projeto *LabSpy* são os que seguem abaixo:

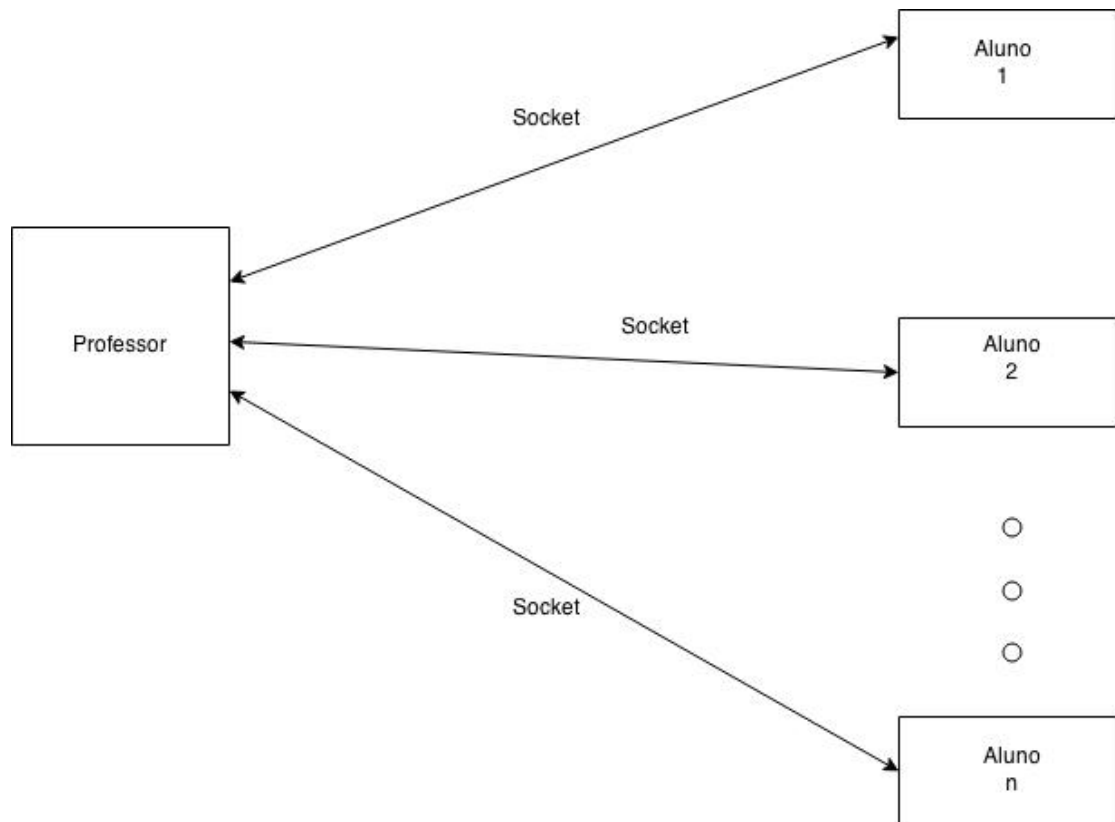
- **Overview das áreas de trabalho:** o programa deve permitir que o docente tenha um *overview* da área de trabalho de todos os computadores conectados na rede, mostrando as imagens dos *desktops* ao vivo ou em uma frequência relativamente alta;
- **Visualização de área de trabalho:** O programa deve permitir que o docente possa ter uma visualização completa (toda a área de tela do programa) da área de trabalho de um computador em específico;
- **Controle remoto do mouse:** O programa deve permitir que o docente controle o *mouse* do computador de um discente a qualquer momento;
- **Controle remoto do teclado:** O programa deve permitir que o docente controle o *teclado* do computador de um discente a qualquer momento;
- **Bloqueio de computadores:** O programa deve permitir que o docente bloqueie e desbloqueie um computador da rede em qualquer dado momento;
- **Proxy definido pelo professor:** O programa deve permitir que o docente possa bloquear sites de serem acessados pelos alunos em qualquer dado momento.
- **Chat:** O programa deve permitir que o docente possa abrir um chat e enviar mensagens instantâneas para todas os estudantes (computadores) da turma.

## 1.2 Requisitos Não-Funcionais

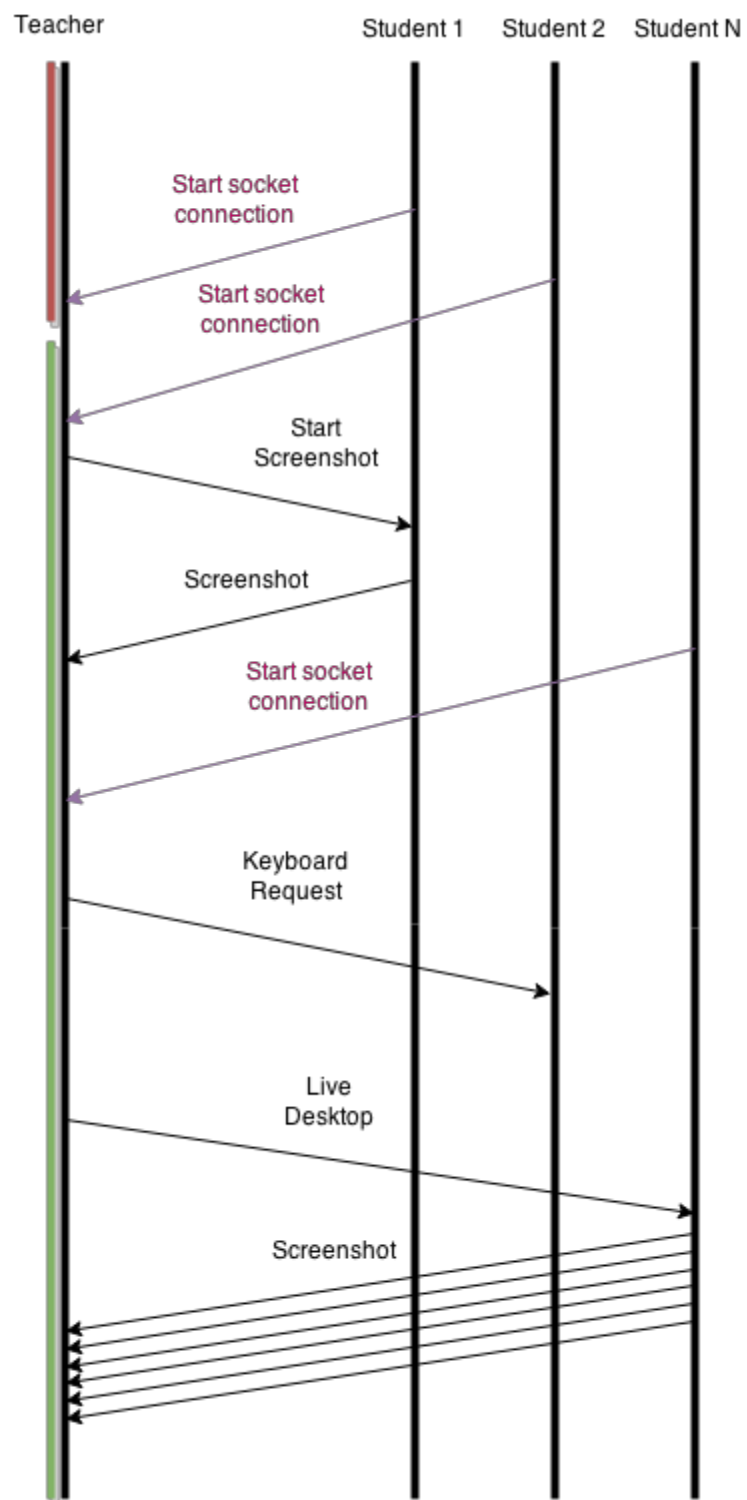
Os requisitos não-funcionais do projeto são os que seguem abaixo:

- O projeto deve ser escrito principalmente na linguagem de programação Java;
- O projeto deve ser disponibilizado de forma gratuita;
- O projeto deve ser *open-source* (código-aberto);
- O projeto deve ser multiplataforma e executar sem maiores problemas em ambientes Linux (principalmente a distribuição Ubuntu) e em ambientes Windows (Windows XP e superiores).

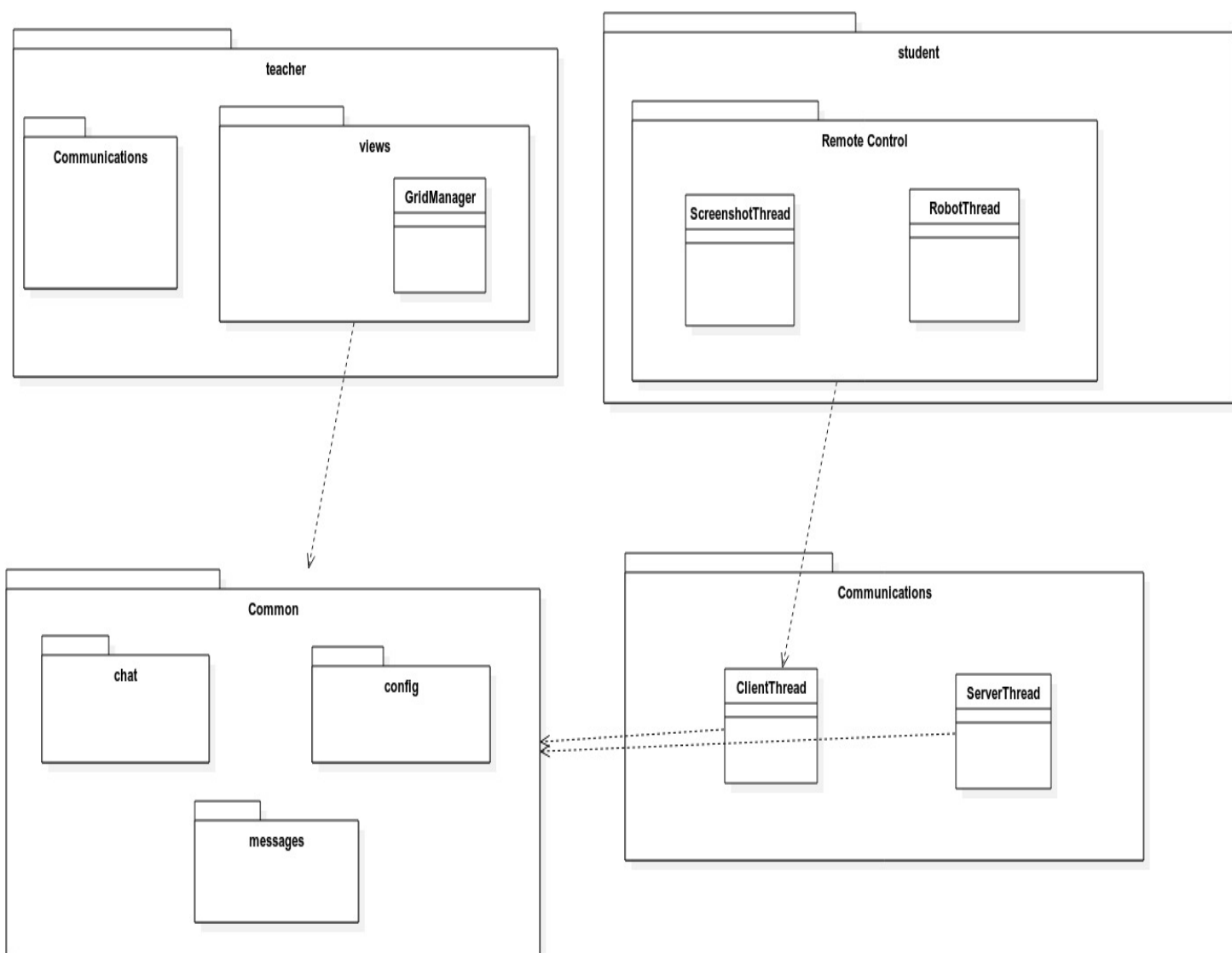
## 2 Diagramas



(Figura 1.1) Abstração da conexão entre computador do professor e dos alunos.



(Figura 1.2) Relação entre requisições e respostas de computadores em Contexto Estudante e em Contexto Professor



(Figura 1.3) Relação entre pacotes e classes do sistema.

### 3 Funcionamento planejado

Para realizar a instalação, configuração e uso do LabSpy em máquinas de um dado laboratório, os seguintes passos se fazem necessários:

1. **Instalação do LabSpy no computador do professor:** haverá um programa de *setup* para configurar o LabSpy no computador do professor. Durante o processo de instalação o administrador será orientado a criar um *hostname* ou um *IP fixo* para o computador em questão (o computador que rodará em Contexto Professor). Após finalizado, o processo fará com que o computador inicie junto com um “processo-servidor” do LabSpy, este processo permitirá que os computadores em Contexto Estudante se conectem, enquanto iniciam, ao computador do professor, estabelecendo assim uma comunicação entre os mesmos.
2. **Instalação do LabSpy no computador do aluno:** haverá um programa de *setup* para configurar o LabSpy nos computadores dos alunos. O *setup* deve ser feito em cada máquina onde é desejado o uso do LabSpy no Contexto Estudante. Esse processo de instalação fará com que o computador em questão sempre inicie com um “processo-cliente” do LabSpy que fará a conexão com o computador em Contexto Professor. Dessa forma, ao iniciar, o sistema em Contexto Estudante começará a comunicação com o computador em Contexto Professor.
3. **Aguardando conexões de computadores em Contexto Estudante:** após ser inicializado, o computador em Contexto Professor terá que aguardar pelo início dos computadores em Contexto Estudante, que ficarão responsáveis por iniciar a comunicação entre computadores *estudante-professor*. Contudo, na ocorrência de uma situação contrária (ou seja, o computador do estudante iniciar antes do computador do professor), o computador em Contexto Estudante terá que esperar até que o computador em Contexto Professor seja iniciado. A partir desse momento, o “servidor” (computador do professor) existirá e será possível que o “cliente” (computador do estudante) faça o início da comunicação. Um computador em Contexto Estudante sempre é responsável por iniciar a comunicação.
4. **Novas conexões de computadores em Contexto Cliente ao computador em Contexto Professor (ações em uma nova conexão):** para cada nova conexão iniciada pelo computador estudante, o computador em Contexto Professor vai criar uma thread cliente. Essas “threads clientes” serão utilizadas pelo computador em Contexto Professor para fazer requisições aos computadores em Contexto Estudante. Através delas será possível separar a parte “servidor” da parte “cliente” no computador em Contexto Professor.

5. **Requisições do professor:** quando uma requisição for feita pelo computador em Contexto Professor, ela será transmitida pela “thread cliente” respectiva ao computador estudante em questão, que tratará essa requisição e fornecerá a sua respectiva resposta ou ação.