

Rede de Compartilhamento de Arquivos - RCA

Este projeto tem por objetivo a implementação da funcionalidade básica de um sistema de troca de arquivos entre pares, frequentemente denominado peer-to-peer, onde os programas de todos os usuários da rede podem agir simultaneamente como cliente e servidor.

Autor

- Jeferson Urbieto da Silva Neto

Funcionalidades

Um programa, ao ser iniciado, deverá identificar os arquivos presentes no seu diretório corrente e se preparar para compartilhá-los. Ao mesmo tempo, ele deve exibir opções ao usuário para listar os usuários e os arquivos existentes na RCA, para procurar um arquivo e para buscar um arquivo.

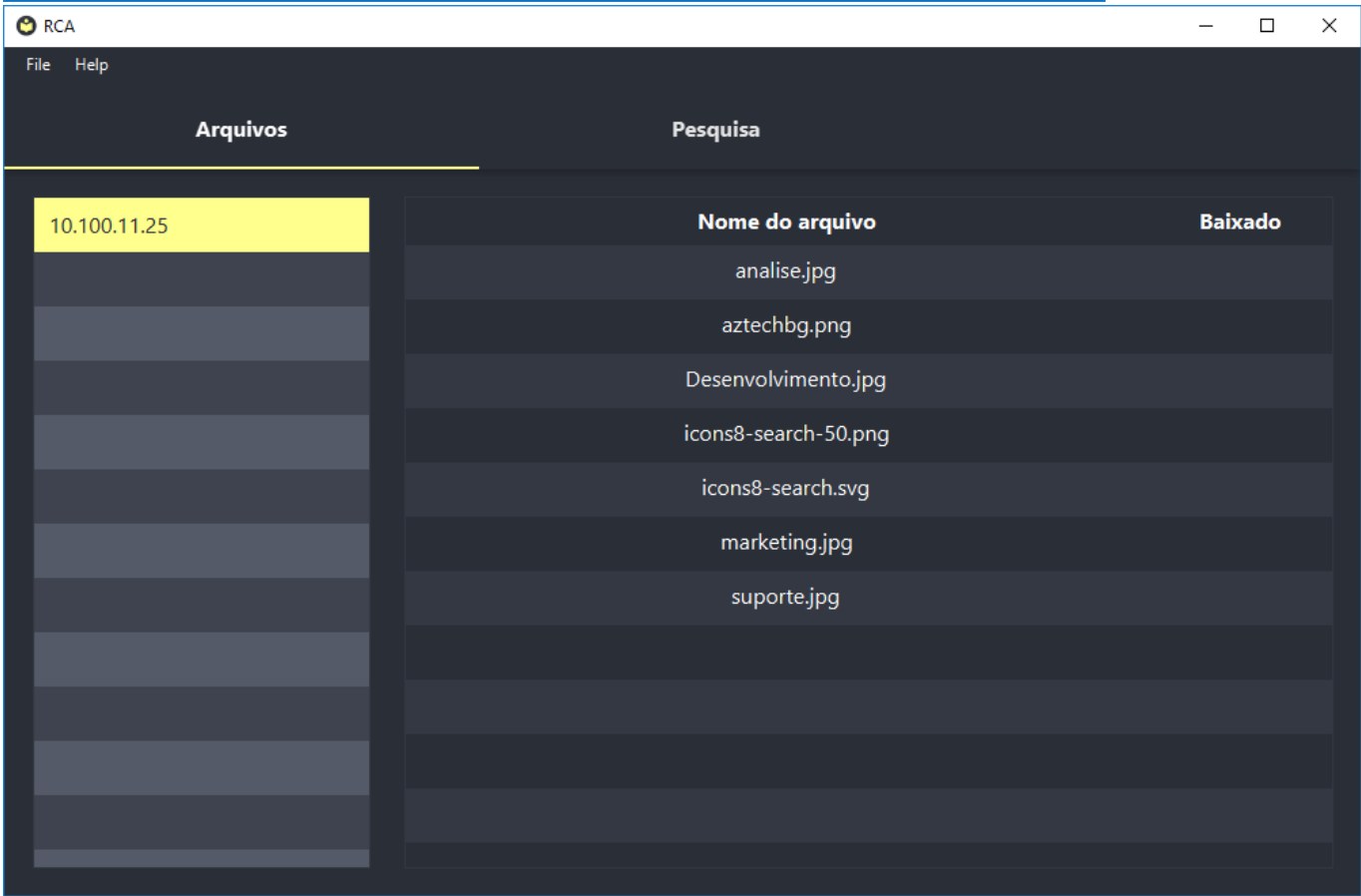
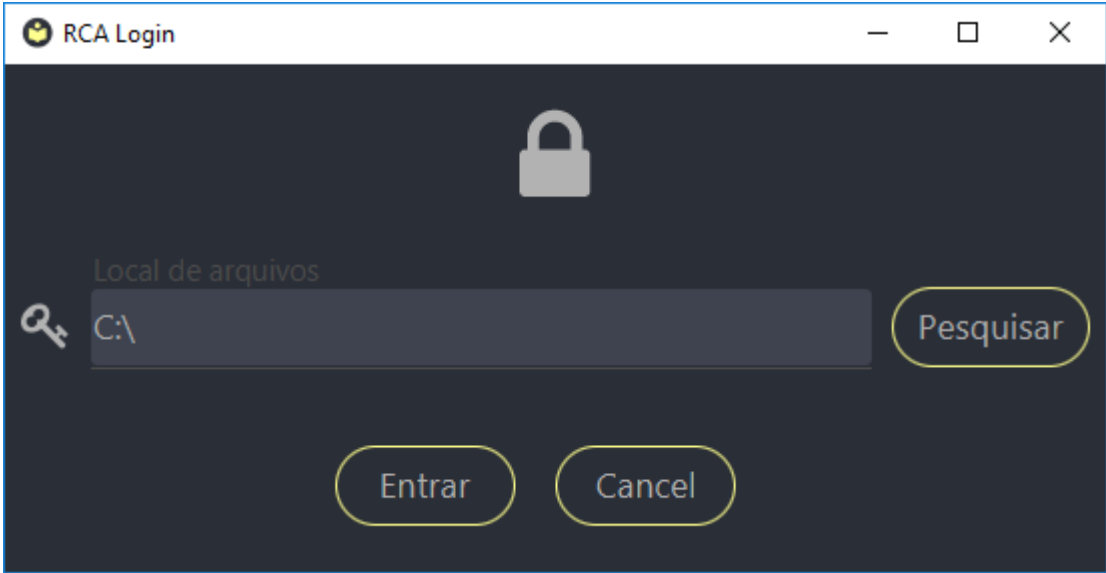
Servidor

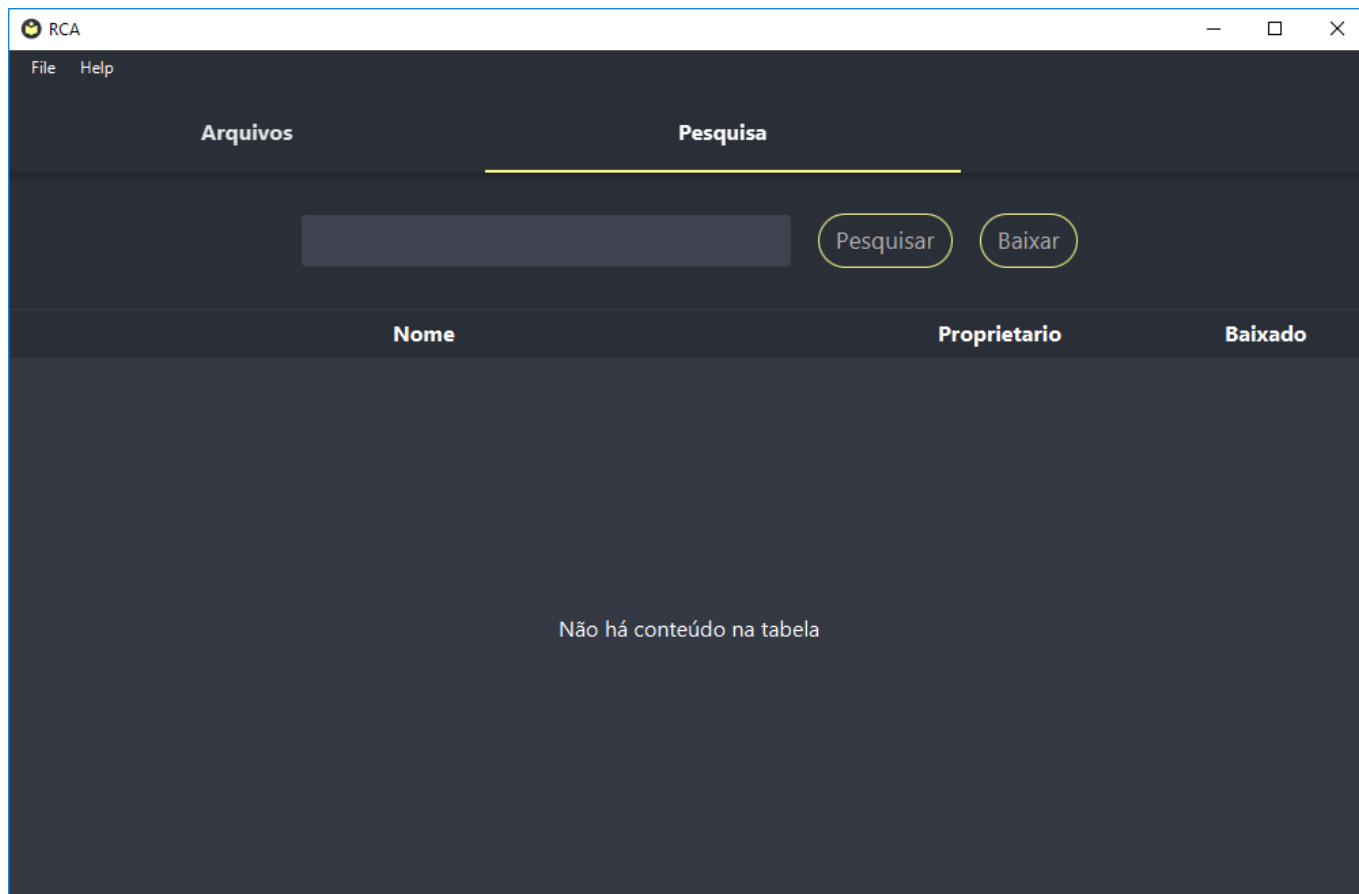
- Armazenar o log das solicitações que foram feitas com: IP, tipo de solicitação, data e hora (não é necessário mostrar na tela, apenas armazene)
- Responder consulta sobre seu IP ("usuário" de rede)
- Fornecer a lista de arquivos contidos em seu diretório
- Responder positivamente caso possua determinado arquivo
- Enviar um determinado arquivo solicitado por outro usuário, cronometrar o tempo gasto para a transferência e o salvar no log

Cliente

- Solicitar (via BROADCAST) a lista de usuários da RCA. Devem ser retornados o IP de cada usuário da rede
- Solicitar (via BROADCAST) a lista de arquivos de todos os programas executando na rede local
- Procurar (via BROADCAST) um arquivo específico na RCA.
- Baixar um arquivo de um determinado servidor.

Screenshot





Backlog

Essa são as demandas solicitas e suas situações atuais no projeto

Funcionalidade	Local	Situação
Solicitar a lista de usuários da RCA	Cliente	Implementada
Solicitar a lista de arquivos de todos os programas executando na rede local	Cliente	Implementada
Procurar um arquivo específico na RCA	Cliente	Implementada
Baixar um arquivo de um determinado servidor	Cliente	Implementada
Armazenar o log das solicitações que foram feitas	Servidor	Implementada
Responder consulta sobre seu IP	Servidor	Implementada
Fornecer a lista de arquivos contidos em seu diretório	Servidor	Implementada
Responder positivamente caso possua determinado arquivo	Servidor	Implementada
Enviar um determinado arquivo solicitado por outro usuário	Servidor	Implementada

Protocolo de Comunicação

A execução do programa na rede funciona através de mensagens UPD que são enviadas pelo cliente, muitas vezes como Broadcast, e respostas pelo servido. Quando é feito o envio de arquivo a comunicação acontece através de mensagens TCP.

Local	Mensagem	Descrição	Tipo	Exemplo
Cliente	FIND_USERS	Envia a mensagem para que todos os usuários de ativos para mandarem o seu IP	Broadcast	FIND_USERS
Cliente	FIND_FILES	Envia a mensagem para que todos os usuários de ativos para mandarem os nomes dos seus arquivos	Broadcast	FIND_FILES
Cliente	FIND_FILE#{NOME ARQUIVO}	Envia o nome do arquivo desejado, para que se algum dos usuários que possuem um arquivo com mesmo nome responda	Broadcast	FIND_FILE#file
Cliente	SEND_FILE#{NOME ARQUIVO}	Envia o nome do arquivo para o servidor que possui o arquivo com mesmo nome para que ele inicie o envio do arquivo	Para o usuário que possui o arquivo	FIND_FILE#file.jpg
Servidor	LIST_USERS#{IP SERVIDOR}	Envia o Ip do servidor para o usuário que solicitou	Usuário que enviou a requisição (FIND_USERS)	LIST_USERS#192.168.0.34
Servidor	LIST_FILES#{NOME ARQUIVO}	Envia o nome dos arquivos armazenados no diretório do servidor para o usuário que solicitou	Usuário que enviou a requisição (FIND_FILES)	LIST_FILES#file.jpg
Servidor	HAVE_FILE#{NOME ARQUIVO}	Envia o nome do arquivo que esta no armazenado no servidor que contem o texto pesquisado na rede	Usuário que enviou a pesquisa (FIND_FILE)	HAVE_FILE#file.jpg
Servidor	UPLOAD_FILE#{NOME ARQUIVO}#{PORTA DE ENVIO}	Envia o nome do arquivo e a porta que sera utilizado para enviar o arquivo	Usuário que enviou a requisição de download (SEND_FILE)	UPLOAD_FILE#file.jpg#55556

Tecnologias

- Java
- Javafx
- Maven
- log4j
- Material Design

Toda a interface foi feita com base no código disponibilizado por [Muhammed Afsal Villan](#) no github, chamado [Library-Assistant](#).

Instalação

Para a instalação apenas precisamos de um ambiente java com maven configurado. Em sua ide escolhida importe o projeto maven e espere fazer download das dependências.

Execução

Para a execução execute o método main da classe Main localizada em `src/main/java/br/com/urbieta/jeferson/ui/main/Main.java`

Log

A aplicação armazena em um arquivo de log todas as suas operações realizadas. Este arquivo está localizado na pasta do projeto com o nome `app.log`.