PROTÓTIPO DE SOFTWARE P2P PARA COMPARTILHAMENTO ANÔNIMO DE ARQUIVOS

Acadêmico: Marcelo Ferreira da Silva
Orientador: M. Sc. - Francisco Adell Péricas

Roteiro

- introdução;
- fundamentação teórica;
- desenvolvimento do protótipo;
- conclusão.

Introdução

- computadores pessoais e a Internet;
- Internet e a liberdade de expressão;
- controle externo da Internet e os ISPs;
- rede flexível e topologicamente hierárquica;
- pesquisas recentes.

Introdução

OBJETIVO GERAL

- pesquisar e desenvolver um protótipo de aplicação para compartilhamento anônimo de arquivos em uma arquitetura de rede descentralizada *Peer-to-Peer* (P2P).

Introdução

OBJETIVOS ESPECÍFICOS

- disponibilizar a conexão entre computadores em uma arquitetura P2P;
- disponibilizar o compartilhamento anônimo e distribuído de arquivos utilizando as implementações básicas fornecida pela plataforma do Freenet;
- disponibilizar um aplicativo para compartilhar arquivos anonimamente.

Fundamentação teórica

REDES P2P

- definição geral;
  - rede distribuída para compartilhamento de recursos de *hardware* entre os nós participantes.
- puramente descentralizadas.
  - conforme definição anterior, além de qualquer entidade terminal da rede poder ser removida sem perda de serviço.

Fundamentação teórica

SEGURANÇA DA INFORMAÇÃO

- confidencialidade;
- integridade;
- disponibilidade;
- (privacidade).

Fundamentação teórica

PRIVACIDADE

- invisibilidade;
- não-rastreamento;
- pseudônimo;
- anonimato.

Fundamentação teórica

FREENET

- privacidade para os produtores de informação, consumidores e titulares;
- resistência à censura de informação;
- alta disponibilidade e confiabilidade através da descentralização;
- armazenamento e roteamento eficiente, escalável e adaptável.

Fundamentação teórica

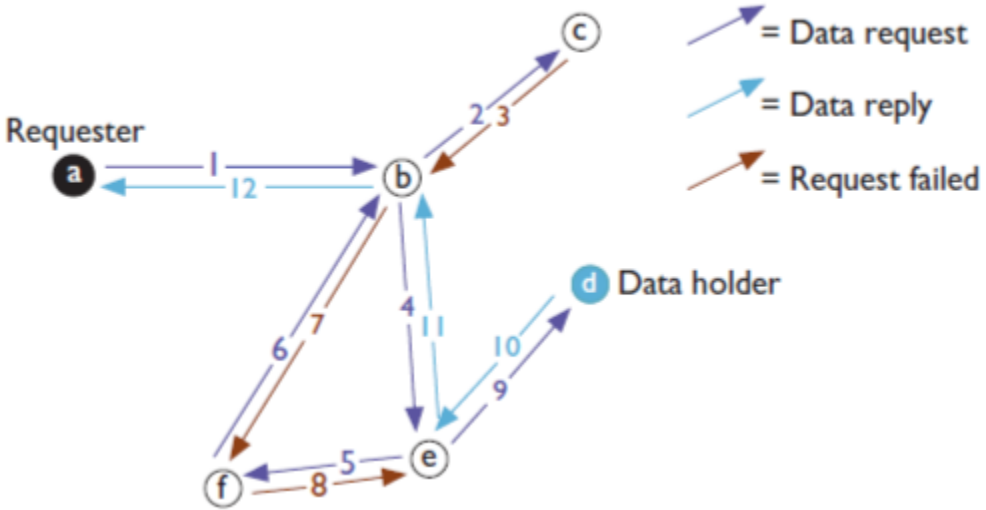
FREENET

- armazenamento dos dados;
- tabela de roteamento;
- *file keys*;
  - *Keyword-Signed Key* (KSK);
  - *Signed-Subspace Key* (SSK);
  - *Content-Hash Key* (CHK).
- *DarkNet e OpenNe*t;
- *Freenet Client Protocol* (FCP);
- *Freenet Proxy* (FProxy);
- *Freesites* e biblioteca para pesquisas.

Fundamentação teórica

FREENET

- Típica requisição realizada pelo Freenet.



Fundamentação teórica

TRABALHO CORRELATO GNUTELLA

- sistema de compartilhamento de arquivos;
- puramente descentralizado P2P;
- protocolo simples;
- alta latência decorrente por uma busca universal;
- grande quantidade de mensagens;
- não visa a garantia da privacidade e do anonimato.

Fundamentação teórica

TRABALHO CORRELATO NAPSTER

- sistema de compartilhamento de MP3;
- P2P híbrido onde há necessidade de servidores centrais;
- possui recurso de pesquisa por metadados com informações sobre artista, título, etc;
- não visa a garantia da privacidade e do anonimato.

Desenvolvimento

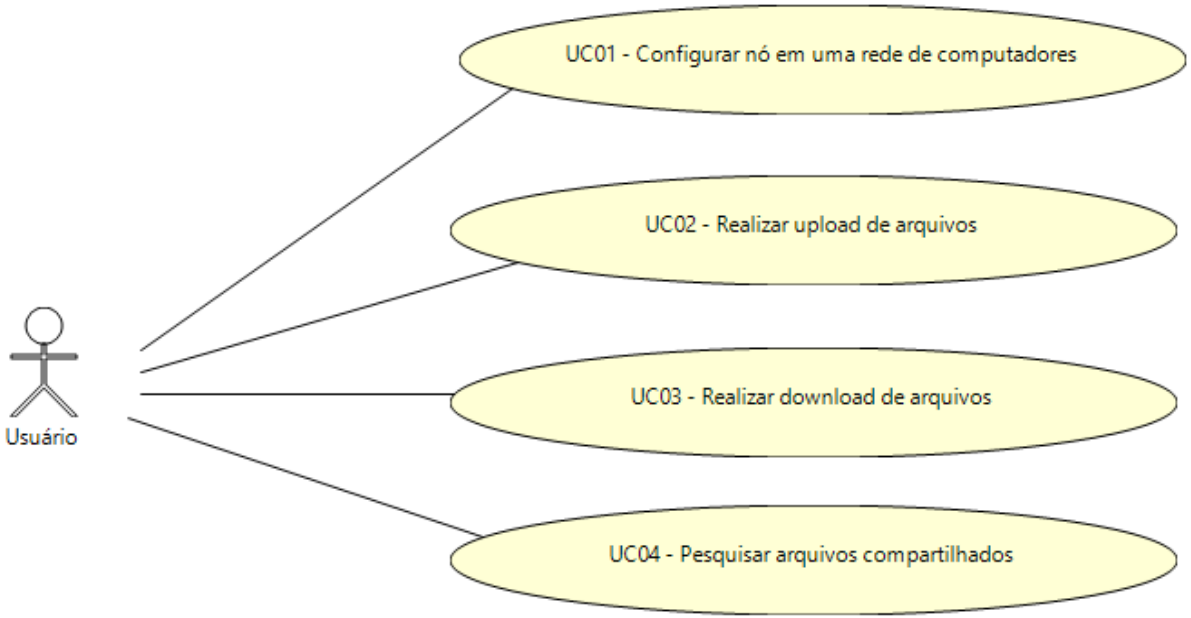
REQUISITOS PRINCIPAIS

O protótipo da aplicação para compartilhamento anônimo de arquivos deverá:

- compartilhar arquivos em uma rede de computadores (Requisito Funcional - RF);
- realizar o *upload* de arquivos compartilhados por meio de uma interface de usuário (RF);
- realizar o *download* de arquivos compartilhados por meio de uma interface de usuário (RF);
- possibilitar a pesquisa de arquivos compartilados por meio de uma interface de usuário (RF);
- garantir a privacidade e anonimato no compartilhamento de arquivos (RF);
- disponibilizar o compartilhamento em uma arquitetura de rede descentralizada P2P baseado na plataforma do Freenet (Requisito Não-Funcional - RNF);
- disponibilizar uma interface *web* de usuário (RNF);
- ser implementado na linguagem de programação Java (RNF).

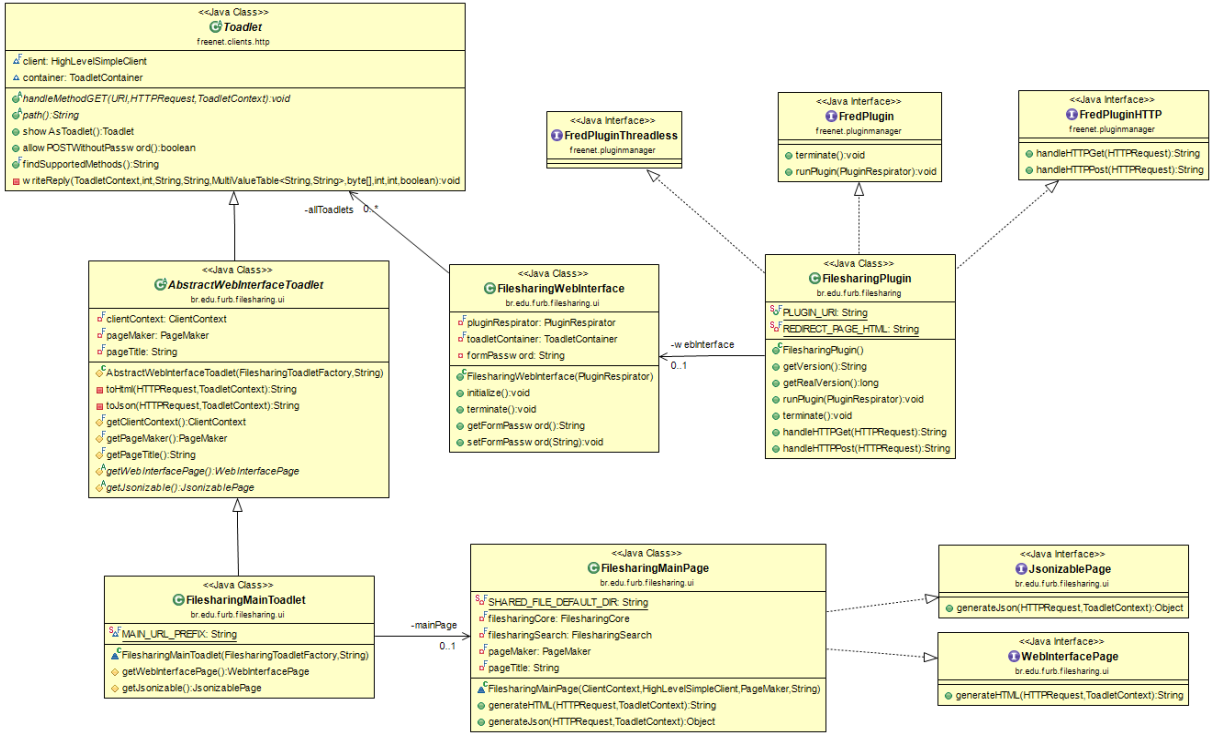
Desenvolvimento

DIAGRAMA DE CASO DE USO



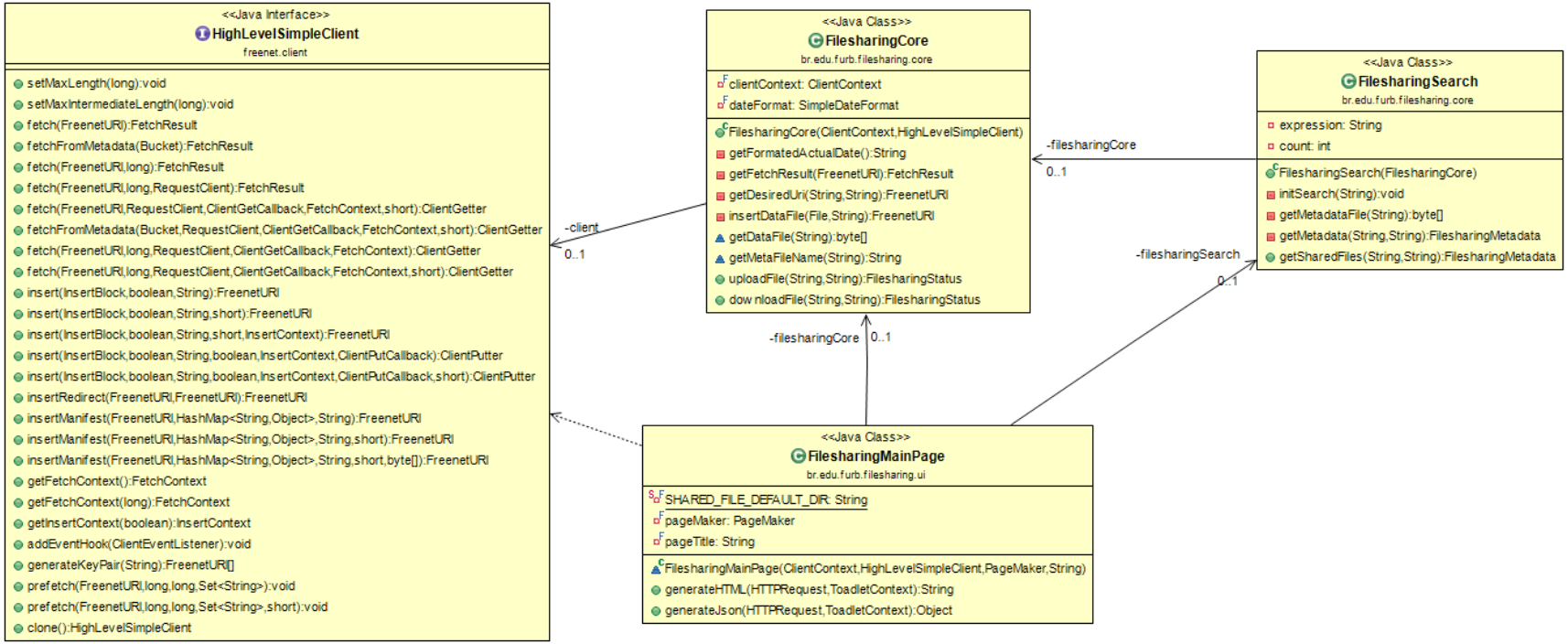
Desenvolvimento

DIAGRAMA DE CLASSES DA CAMADA *VIEW*



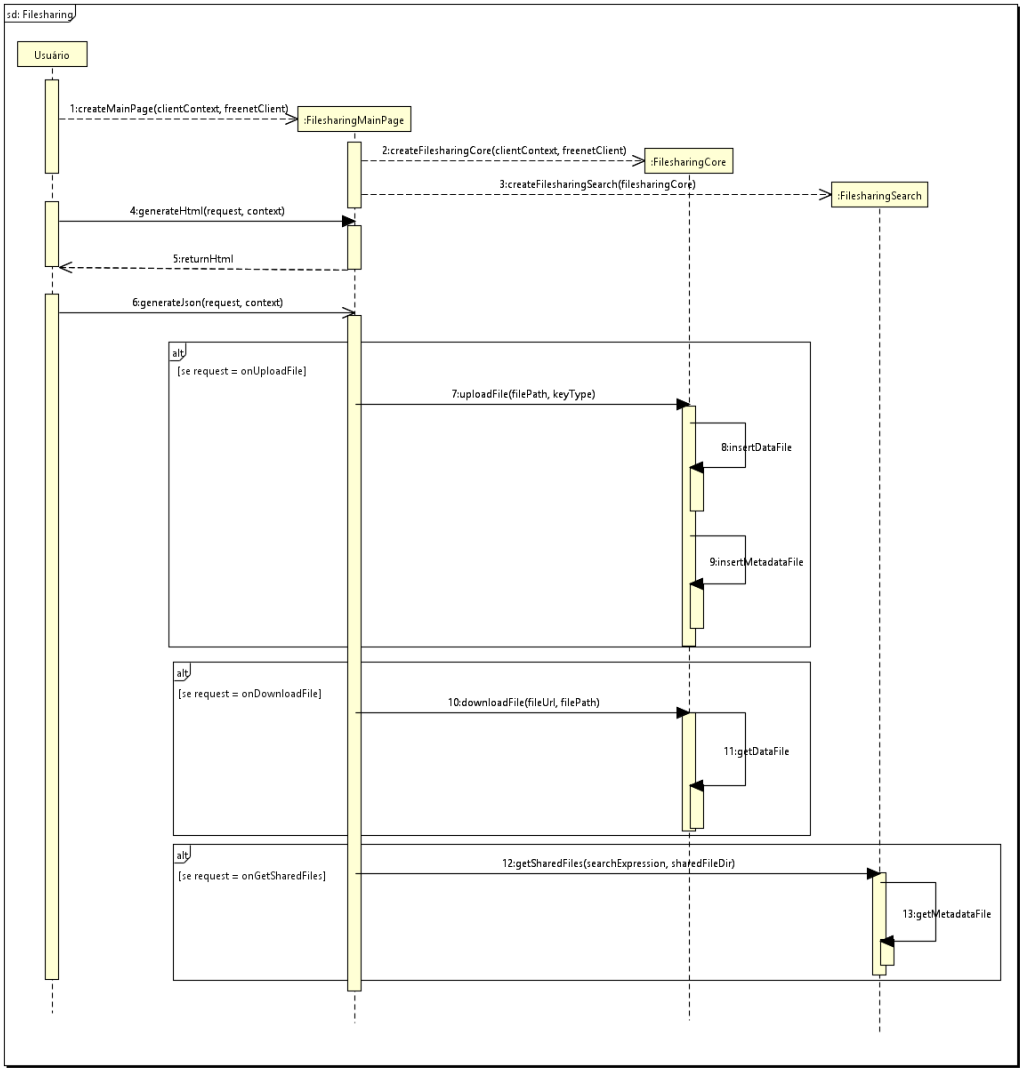
Desenvolvimento

DIAGRAMA DE CLASSES DA CAMADA *CORE*



Desenvolvimento

DIAGRAMA DE SEQUÊNCIA


sd: Filesharing
Usuário
1:createMainPage(clientContext, freenetClient)
:FilesharingMainPage
2:createFilesharingCore(clientContext, freenetClient)
:FilesharingCore
3:createFilesharingSearch(filesharingCore)
:FilesharingSearch
4:generateHtml(request, context)
5:returnHtml
6:generateJson(request, context)
alt
[se request = onUploadFile]
7:uploadFile(filePath, keyType)
8:insertDataFile
9:insertMetadataFile
alt
[se request = onDownloadFile]
10:downloadFile(fileUrl, filePath)
11:getDataFile
alt
[se request = onGetSharedFiles]
12:getSharedFiles(searchExpression, sharedFileDir)
13:getMetadataFile

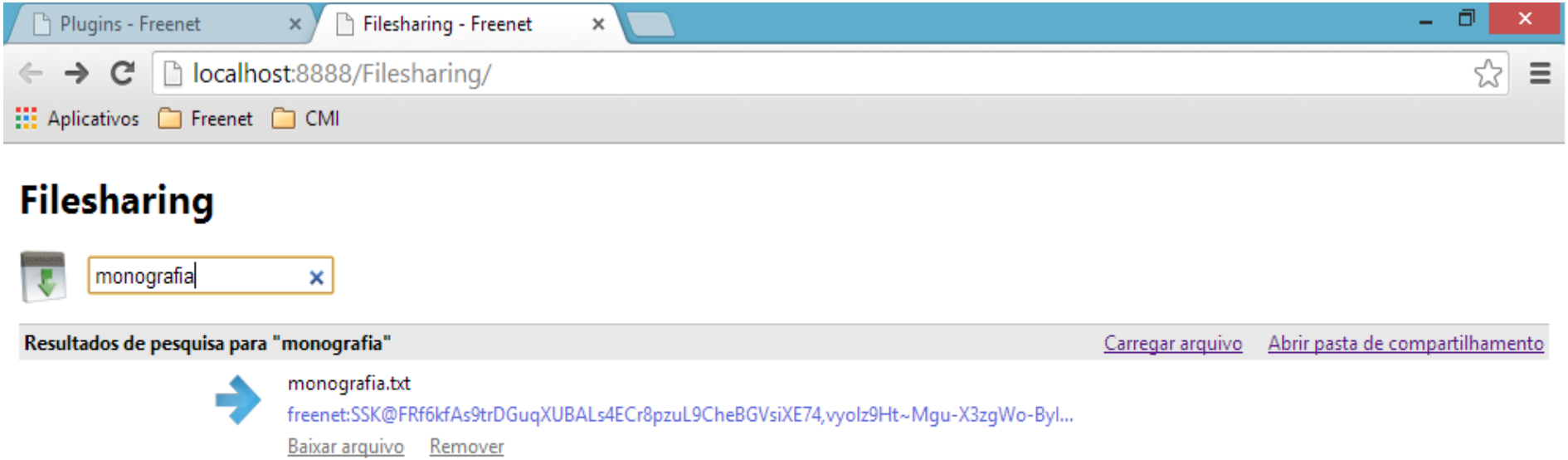
Desenvolvimento

IMPLEMENTAÇÃO

```
01 private FetchResult getFetchResult(FreenetURI freenetUri) throws
02 FetchException {
03       while (true) {
04             try {
05                   return this.client.fetch(uri);
06             } catch (final FetchException e) {
07                   if (e.newURI == null) {
08                         throw e;
09                   }
10                   uri = e.newURI;
11             }
12       }
13 }
```

Desenvolvimento

OPERACIONALIDADE



CONCLUSÃO

RESULTADOS

Principais Características / trabalhos correlatos	Gnutella	Napster	Protótipo Filesharing
Rede de computadores P2P	X	X	X
Rede P2P puramente decentralizada	X		X
Compartilhamento de arquivos	X	X	X
Compartilhamento anônimo de arquivos			X
Replicação dos arquivos na rede			X
Recurso de pesquisa		X	X
Recurso de pesquisa por metadados		X	X
Código fonte aberto			X

CONCLUSÃO

DISCUSSÃO

- mecanismos de anonimato e aplicações da lei;
- os objetivos foram atingidos;
- a depuração do *kernel* do Freenet;
- *plugin* para o FProxy;
- recurso de pesquisa por metadados;
- metadados inseridos pela chave KSK;
- chaves KSK não são seguras a *spams*;
- desempenho na inserção arquivos devido a replicação;
- desempenho na pesquisa e a busca em profundidade;
- pesquisa e *webcrawlers*.

CONCLUSÃO

EXTENSÕES

No decorrer do desenvolvimento do trabalho foram identificados pontos de aprimoramento sugeridos abaixo:

- implementar índices com base nos metadados e um *webcrawler* capaz de pesquisar metadados de forma assíncrona;
- utilizar a biblioteca Apache Lucene para pesquisar palavras chaves nos metadados;
- implementar um *plugin* para o Freenet capaz de recarregar o arquivo na rede com objetivo de mantê-lo disponível caso não seja muito popular;
- implementar validações contra *spam* no caso de metadados inseridos pela chave KSK;
- possibilitar o cancelamento de downloads interrompendo sua execução;
- desenvolver um protótipo de aplicativo *desktop* capaz de compartilhar arquivos anonimamente utilizando o FCP.