

## Trabalho Prático de Redes de Computadores

**Valor:** 10 pontos

**Data da entrega/apresentação:** 29/04/2016 (entrega do código-fonte por e-mail e pequena apresentação de slides sobre o funcionamento do programa, com ênfase na comunicação e protocolos de rede).

### Observações:

- Dupla (temas 1, 2 e 3) ou individual (temas 4 e 5);
- Sem temas repetidos;
- Linguagem de programação: C, Java ou Python;
- Não podem ser utilizadas bibliotecas prontas para manipular as comunicações de rede, apenas sockets.

**Bônus:** 5 pontos extras para os trabalhos que tiverem todas as suas funcionalidades corretamente implementadas em linguagem C.

### Tema 1 (dupla): Proxy para cache e filtro de conteúdo HTTP

- Intermediar as requisições HTTP realizadas pelos clientes (navegadores web);
- Realizar o cache dos arquivos requisitados aos servidores web;
- Utilizar o GET Condicional para verificar se as cópias em cache não expiraram;
- Possuir um arquivo de configuração definindo o tamanho mínimo e máximo dos arquivos que poderão ficar em cache;
- Atuar como filtro de conteúdo, impedindo o acesso a sites (domínio e IP) que estejam definidos como impróprios (em um arquivo de configuração) e também a páginas que possuam palavras proibidas (também definidas em um arquivo de configuração).

### Tema 2 (dupla): Buscador web (*web crawler*)

- A partir de um domínio definido pelo usuário, indexar o conteúdo textual da página. Seguir seus links (internos e externos) e indexá-los, sucessivamente, até atingir um nível de profundidade de busca ou de armazenamento em disco, a ser definido em arquivo de configuração;
- Durante a indexação, o programa deverá respeitar as regras definidas em eventuais arquivos `robots.txt`, páginas com a meta tag `robots` e links com o atributo `rel="nofollow"`;
- Deverá ser possível pesquisar conjuntos de termos ou frases exatas (entre aspas);
- O resultado da busca deverá ser uma página HTML informando os termos pesquisados, o tempo gasto na sua geração, e, para cada resultado da busca, o título, o link e a descrição do mesmo;
- A ordenação dos resultados deverá seguir um critério semântico. Por exemplo, o peso de um termo deverá ser maior, sucessivamente, na URL, no título, na descrição, na tag `h1`, etc.

### **Tema 3 (dupla):** Sala de bate-papo

- Criar um servidor de bate-papo que pode possuir várias salas;
- Ao se conectar ao servidor, o cliente pode:
  - Definir um nome de usuário (*nick*)
  - Listar as salas (para cada sala, listar também o nome dos usuários na mesma)
  - Criar uma sala
  - Entrar em uma sala
  - Sair da sala
- Ao entrar em uma sala, todas as mensagens enviadas por um usuário serão exibidas para todos os outros usuários conectados ao servidor e na mesma sala;
- O usuário que criou a sala é seu *admin*, podendo banir o acesso de outros usuários da sala. Para isso, o servidor identificará cada usuário pelo seu endereço IP.

### **Tema 4 (individual):** Cliente FTP

Funcionalidades mínimas:

- Conectar a um servidor FTP utilizando autenticação e modo anônimo;
- Enviar um arquivo para o servidor;
- Receber um arquivo do servidor;
- Listar os arquivos e pastas do diretório remoto corrente;
- Criar um diretório remoto;
- Remover um diretório remoto (recursivamente);
- Remover um arquivo remoto;
- Acessar outro diretório remoto.

### **Tema 5 (individual):** Cliente FTP

Funcionalidades mínimas:

- Receber conexões autenticadas e anônimas;
- Responder às seguintes requisições do cliente:
  - Enviar um arquivo para o cliente;
  - Receber um arquivo do cliente;
  - Listar os arquivos e pastas do diretório corrente;
  - Criar um diretório;
  - Remover um diretório (recursivamente);
  - Remover um arquivo;
  - Acessar outro diretório.