



Lista 1: Servidor de arquivos

26/09/2022



- Na raiz do repositório crie o arquivo `.gitignore` para ignorar arquivos desnecessários para esse projeto. O conteúdo desse arquivo poderá ser gerado por meio do site <https://gitignore.io>.
- Na raiz do repositório crie o arquivo `Readme.md` e coloque nele uma breve descrição sobre o projeto, as instruções para compilar e para executar as aplicações desse projeto. Também deve indicar quais requisitos foram atendidos e quais não foram.

Desenvolva uma aplicação, seguindo o modelo cliente e servidor, que permita a transferência de arquivos texto ou binários de tamanho arbitrário. O servidor deverá atender somente um cliente por vez e deverá continuar ativo depois de atender um cliente. O aplicativo cliente poderá enviar dois comandos para o servidor:

- `list` – Para listar os arquivos que estão armazenados no servidor;
- `get` – Para receber um arquivo que está armazenado no servidor.

Ao iniciar o servidor deve-se informar, como argumento de linha de comando, o caminho do diretório onde estarão os arquivos que ele irá prover ao cliente. Ex: `java Servidor /tmp/arquivos`.

O cliente deverá receber os comandos como argumentos de linha de comando. Para o comando `get` deve-se fornecer também o nome do arquivo que deseja obter do servidor. O cliente pode jogar os `bytes` recebidos diretamente no dispositivo de saída padrão (terminal). Assim, o usuário deverá fazer uso de redirecionamento de saída para jogar estes `bytes` para dentro de um arquivo. Ou, o cliente poderá fazer uso da API de arquivos, da linguagem de programação escolhida, e salvar os `bytes` diretamente no arquivo. Cabe a você escolher um dos comportamentos apresentados para o cliente. Exemplos:

- `java Cliente IP porta list` – para listar os arquivos ofertados por um servidor;
- `java Cliente IP porta get arquivo.bin > saida.bin` – para obter o arquivo chamado “arquivo.bin” e salvar seu conteúdo dentro de um arquivo chamado “saida.bin” por meio do redirecionamento de saída;
- `java Cliente IP porta get arquivo.bin saida.bin` – para obter o arquivo chamado “arquivo.bin” e salvar seu conteúdo dentro de um arquivo chamado “saida.bin” usando a API de arquivos da linguagem de programação escolhida.

As aplicações cliente e servidora poderão ser desenvolvidas na linguagens C ou Java e devem usar obrigatoriamente a API de `sockets`. Garanta que seja possível executar cliente e servidor em computadores diferentes, ou seja, não defina no código, de forma estática, o endereço IP do cliente ou servidor.



Data para entrega: Até o dia **09/10/2022** via Github Classroom. Veja no SIGAA o endereço para entrega da lista.