

## Servidor de Arquivos

O objetivo deste projeto é desenvolver um pequeno servidor de arquivos usando diferentes metodologias de desenvolvimento distribuído.

### **Ao todo serão 4 Etapas**

- Etapa 1: usando a API SOCKETS para comunicação entre cliente e servidor
- Etapa 2: usando RPC ou RMI (dependendo do que se optou por desenvolver na fase 1)
- Etapa 3: implementar mecanismos de sincronização e exclusão mútua
- Etapa 4: implementar mecanismos de replicação de dados, mantendo a consistência entre os diversos servidores

O servidor de arquivos deverá suportar as seguintes operações:

- Leitura – indicando qual o descritor de arquivo, quantos bytes, a partir de qual posição e o próprio buffer
- Escrita – indicando qual o descritor de arquivo, quantos bytes, quantos elementos e o próprio buffer (se for a operação de append, grava-se no final, caso contrário grava-se no início)
- Abertura de Arquivo (indicando o modo – somente leitura ou leitura/escrita)
- Fechamento de Arquivo
- Criação de Arquivo (cria-se um arquivo vazio)
- Remoção de Arquivo (apaga o arquivo selecionado)
- Listagem de Arquivos (listagem do diretório – retorna um vetor de STRINGS)

Além disso, mensagens de controle deverão ser trocadas entre cliente e servidor, de forma que haja um entendimento entre os processos, tais como:

- Operação executada com sucesso
- Arquivo não encontrado
- Erro de leitura/escrita
- Verificação de disponibilidade do servidor e suas prováveis respostas
- Parâmetros ou códigos de operações inválidos

### **Entregáveis:**

Etapa 1:

Nesta Etapa, você deve especificar o protocolo entre cliente e servidor, indicando todas as mensagens que serão trocadas entre as entidades, suas seqüências e estruturas de dados utilizadas entre ambos. Além disso, os códigos das aplicações cliente e servidor deverão ser entregues para comprovar o funcionamento do sistema.