

# Laboratório 2

## Aplicação cliente/servidor básica

Letícia Freire Carvalho de Sousa - DRE: 118025324

### Atividade 1

1. **Funcionalidades da camada de interface com o usuário:** recebe do usuário o nome do arquivo e a palavra de busca e exibe na tela o resultado do processamento. O resultado do processamento poderá ser: *(i) uma mensagem de erro indicando que o arquivo não foi encontrado; ou (ii) o número de ocorrências da palavra no arquivo.* As mensagens serão editadas nesta camada antes de serem impressas.
1. **Funcionalidades da camada de processamento:** solicita o acesso ao arquivo texto. Se o arquivo for válido, realiza a busca pela palavra informada e prepara a resposta para ser devolvida para a camada de interface. Se o arquivo for inválido, responde com a mensagem de erro. Se o arquivo é inválido esta camada envia o código "-1" para a **camada de interface**, caso contrário ela envia um número inteiro maior ou igual a zero, o número de ocorrências da palavra no arquivo.
1. **Funcionalidades da camada de acesso aos dados:** verifica se o arquivo existe em sua base. Se sim, devolve o seu conteúdo inteiro. Caso contrário, devolve uma mensagem de erro.

### Atividade 2

#### Proposta de arquitetura de sistema

1. **Lado cliente:** implementa a camada de interface com o usuário. O usuário poderá solicitar o processamento de uma ou mais buscas em uma única execução da aplicação: o programa espera pelo nome do arquivo e da palavra

de busca, faz o processamento, retorna o resultado, e então aguarda um novo pedido de arquivo e palavra ou o comando de finalização.

2. **Lado servidor:** implementa a camada de processamento e a camada de acesso aos dados. Projete um servidor iterativo, isto é, que trata as requisições de um cliente de cada vez, em um único fluxo de execução (estudaremos essa classificação depois). Terminada a interação com um cliente, ele poderá voltar a esperar por nova conexão. Dessa forma, o programa do servidor fica em loop infinito (depois veremos como lidar com isso).

## Sequência de mensagens

1. **Camada de interface** envia uma mensagem (no formato "endereço\_do\_arquivo;palavra") para **camada de processamento**.
2. **Camada de processamento** envia mensagem para **camada de acesso de dados** com o nome do arquivo a ser aberto.
3. **Camada de acesso de dados** envia mensagem para **camada de processamento** com o conteúdo do arquivo lido.
  1. Caso não consiga ler o arquivo a **camada de acesso de dados** lança uma exceção a ser tratada pela **camada de processamento**
4. **Camada de processamento** envia mensagem de retorno com o número de ocorrências da palavra no arquivo.
  1. Caso a mensagem enviada pela interface esteja no formato incorreto retorna mensagem com o código "-2"
  2. Caso o arquivo não possa ser aberto retorna mensagem com o código "-1"

## Decisões e implementação

### Cliente

As mensagens para o usuário são editadas nessa camada para dar mais possibilidade de personalização delas para cada aplicação.

As linhas da entrada são lidas até o EOF (representado pelo comando 'ctrl+d').

## Servidor

É dividido em dois arquivos o de **processamento** e o de **acesso de dados**.

### Processamento

Defini um limite de espera por conexão para o socket par evitar ter que forçar a parada da aplicação

Uso o tratamento de exceções para os casos em que a mensagem enviada pela interface está no formato errado ou a aplicação não consegue abrir o arquivo

### Acesso de dados

Caso a camada não consiga abrir o arquivo a lança uma exceção a ser tratada pela **camada de processamento**