

## LABORATÓRIO 2

Lucas Clemente Miranda Braga  
115159700

### Atividade 1

**Objetivo:** Refinar a arquitetura de software – usando o estilo arquitetural em camadas – apresentada abaixo.

Camadas:

1. Funcionalidades da camada de interface com o usuário : recebe do usuário o nome do arquivo 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) a lista de palavras com suas ocorrências. **Recebe da camada de processamento a string pronta para ser impressa.**
2. Funcionalidades da camada de processamento: solicita o acesso ao arquivo texto. Se o arquivo for válido, realiza a contagem das palavras e prepara a resposta para ser devolvida para a camada de interface. Se o arquivo for inválido, responde com a mensagem de erro. **Realiza o processamento do conteúdo inteiro recebido pela camada de acesso aos dados, devolvendo à camada de interface uma string pronta para ser impressa.**
3. Funcionalidades da camada de acesso aos dados: verifica se o arquivo existe em sua base. Se sim, devolve o seu conteúdo inteiro. Caso contrário, envia uma mensagem de erro.

### Atividade 2

**Objetivo:** Refinar a proposta de instanciação da arquitetura de software da aplicação definida na Atividade 1 para uma arquitetura de sistema cliente/servidor de dois níveis, com um servidor e um cliente, apresentada abaixo.

Proposta de arquitetura de sistema:

1. Lado cliente: implementa a camada de interface com o usuário. O usuário poderá solicitar o processamento de um ou mais arquivos em uma única execução da aplicação: o programa espera pelo nome do arquivo, faz o processamento, retorna o resultado, e então aguarda um novo pedido de arquivo ou o comando de finalização. **Ao receber a string pronta para ser impressa da camada de processamento, realiza a conversão para UTF-8 e imprime as ocorrências, após isso espera novas entradas do usuário. Envia a mensagem (nome do arquivo) de forma codificada via socket para o servidor o nome do arquivo, e caso a entrada seja a palavra “encerrar”, fecha a conexão com o servidor.**

2. Lado servidor: implementa a camada de processamento e a camada de acesso aos dados. Implemente 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). **Divide as duas camadas implementadas em métodos, onde a parte principal chama a camada de de Processamento que por sua vez chama a camada de Dados. A camada de processamento possui uma interface que recebe sempre o nome do arquivo desejado, e retorna a string com as ocorrências de cada palavra pronta para ser impressa pela camada de interface. A camada de Acesso aos Dados recebe como parâmetro uma string contendo o nome do arquivo, e devolve, caso o arquivo exista, uma string crua contendo todo o conteúdo deste arquivo, e caso não exista a string "Arquivo não existe". Caso o servidor receba a string "encerrar", ele fecha a conexão com o cliente e fica no aguardo de uma nova conexão.**

O Cliente realiza as requisições para o Servidor enviando strings codificadas, e caso envie a string "encerrar", fecha a conexão.

O Servidor responde ao cliente de duas formas :

1. Com a string a ser impressa formatada para melhor visualização de todas as palavras do arquivo e suas ocorrências.
2. A string "Arquivo não existe" caso não haja o arquivo na base de dados

É importante frisar que o servidor não irá interpretar nenhuma letra que não seja do nosso alfabeto, assim como acentos. Ele realiza uma substituição a partir do regex, permitindo somente caracteres alfanuméricos (0-9a-zA-Z).

Por fim, a camada de processamento só irá processar arquivos .txt para melhor experiência.