Laboratório 2

Joyce Brum

Atividade 1

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:
    a. uma mensagem de erro indicando que o arquivo nao foi encontrado;
    b. ou a lista de palavras com suas ocorrências.

    A camada já receberá as 10 palavras mais frequentes ordenadas por frequência e terá que imprimi-las.
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 invalido, responde com a mensagem de erro.
    A camada de processamento retornará para a camada de interface a lista com as 10 palavras ordenadas por frequência.
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

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.
2. **Lado servidor**: implementa a **camada de processamento** e a **camada de acesso aos dados**. O servidor será um **servidor iterativo**, isto é, que trata as requisições de um cliente de cada vez em um único fluxo de execução. Terminada a interação com o cliente, ele poderá voltar a esperar por nova conexão. Dessa forma, o programa do servidor fica em loop infinito.

Refinamentos:

- A lista de palavras e frequências será representada como uma lista de pares no seguinte formato [(palavra1, frequencia1), (palavra2, frequencia2)].
- A camada de aplicação para solicitar o processamento de um arquivo para a camada de processamento deverá enviar uma **requisição** contendo o nome do arquivo a ser verificado e aguardará a **resposta**

com a lista das 10 palavras mais frequentes do arquivo e suas frequências ou erro.

- A camada de processamento recebe a requisição com o nome do arquivo e envia uma **requisição** também contendo apenas o nome do arquivo à camada de acesso aos dados, aguardando a **resposta** com o conteúdo do arquivo ou erro.
- A camada de acesso aos dados recebe a requisição com o nome do arquivo, procura o arquivo na sua base de dados e retorna uma mensagem no formato de dicionário onde a chave "status" indica o status da busca (0 para sucesso e 1 para erro) e a chave "data" sendo o conteúdo do arquivo caso o mesmo seja encontrado ou "Arquivo não encontrado." caso o contrário.
  formato {'status': 0, 'data': "amor amor vida"} ou
  {'status': 1, 'data': "Arquivo não encontrado."}
- Para encerrar a aplicação, o usuário deve digitar "exit".

Dessa forma temos dois fluxos possíveis: um caso de sucesso e um caso de falha. Segue abaixo uma sequência de mensagens para cada fluxo. no caso de um arquivo de nome amo.txt que só tenha as palavras "amor amor vida"

	Cliente	Processamento	Acesso aos Dados
Sucesso	send-> "amo.txt" rcv <- [(amor, 2), (vida, 1)] imprime: *amor - 2* *vida - 1*	rcv_client <- "amo.txt" send_access -> "amo.txt" rcv_access <- "amor amor vida" send_client -> [(amor, 2), (vida, 1)]	rcv <- "amo.txt" send ->{'status': 0, 'data': "amor amor vida"}
Falha	msg -> amo.xlsx rcv <- "Arquivo não encontrado." imprime: *Arquivo não encontrado.*	rcv_client <- "amo.txt" send_access -> "amo.txt" rcv_access <- "Arquivo não encontrado." send_client -> "Arquivo não encontrado."	rcv <- "amo.txt" send -> {'status': 1, 'data': "Arquivo não encontrado."}