

SISTEMAS DISTRIBUÍDOS: LAB 2 - 01/08/2021, UFRJ

Nome: Breno Trindade Tostes; DRE: 117099396

1ª Atividade:

Camadas:

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.

Refinamento: As mensagens recebidas não serão editadas nessa camada. No máximo, esta camada irá aprimorar a interface que será visualizada pelo usuário, possivelmente incluindo mensagens adicionais para facilitar a interação com o usuário.

2. *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.

Refinamento: O resultado será passado na forma de uma string concisa contendo o resultado da busca:

- 1) Caso de erro - 'Arquivo "nome_do_arquivo" inválido.'
- 2) Caso de sucesso - 'O arquivo "nome_do_arquivo" contém valor_encontrado instâncias da palavra "palavra_solicitada" informada.'

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, devolve uma mensagem de erro.

2ª Atividade:

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).

Refinamentos:

1. Cliente irá enviar uma mensagem de requisição para o servidor contendo o nome do arquivo e a palavra de busca, o server retorna uma resposta informando se o arquivo foi encontrado com sucesso ou não, caso o arquivo seja encontrado o resultado da busca pela palavra é retornado na forma de uma string;

2. Nenhuma estrutura de dados complexa será usada, apenas uma sequência de strings para o envio do nome de arquivo e da palavra de busca; o conteúdo das mensagens será como descrito no item anterior;

3. O lado do cliente ficou responsável por apresentar para o usuário as mensagens com instruções de uso e feedback do status do processo, enquanto o lado do servidor ficou responsável apenas por processar os pedidos e enviar a mensagem final contendo o resultado, além de informar se algum erro foi encontrado no caminho.