

UNIVERSIDADE FEDERAL DE VIÇOSA
CAMPUS FLORESTAL
CCF 355 - SISTEMAS DISTRIBUÍDOS E PARALELOS



Trabalho de Distribuídos
Parte 02

Ana Flávia Marcondes Bomfim - 5082
Carlos Márcio Moreira Costa - 5079
Gabriel Zoéga Fernandes - 5072

Sumário

Introdução:.....	3
Desenvolvimento:.....	3
Modelo Físico.....	3
Modelo de Arquitetura.....	3
Modelos Fundamentais.....	9
Modelo de interação.....	9
Modelo de falhas.....	10
Modelo de segurança.....	10
Referências:.....	11

Introdução:

A segunda parte do trabalho tem como objetivo que os conceitos dados em sala de aula sobre modelagem de um sistema distribuído sejam aplicados. Sendo assim, o grupo teve que pensar no modelo físico, de arquitetura e modelos fundamentais.

Para isso foram utilizados os slides disponibilizados, o trabalho prático 01 e as anotações feitas na sala de aula pelos integrantes do grupo.

Desenvolvimento:

Modelo Físico

Considerando o objetivo do sistema, é possível concluir que um sistema de pequena escala não é adequado devido às suas características, pois utiliza-se de uma rede local e possui heterogeneidade limitada. Sendo assim, considerando que o sistema que está sendo desenvolvido, no qual clientes podem estar geograficamente espalhados e podem ter máquinas com diferentes configurações, um sistema de pequena escala não atende os requisitos.

Pensando que o sistema será usado apenas para um trabalho prático, utilizar a ultra larga escala também não é adequado, afinal, além da complexidade envolvida, ainda existem muitos custos que seriam desnecessários.

Por fim, o melhor modelo físico seria aquele com a escala da internet, que melhor se adequa à heterogeneidade e localização geográfica das máquinas envolvidas no sistema.

Modelo de Arquitetura

- Paradigma de Comunicação: Será utilizado o paradigma de Invocação Remota, pois a comunicação será feita entre processos clientes, que farão requisições para o processo servidor, o qual responderá de acordo com o tipo de requisição

(Requisição-resposta). Sendo assim, as entidades que fazem a requisição sabem quem é a outra entidade envolvida e o tempo de resposta.

- **Modelo de Arquitetura Básico:** Será utilizado o modelo Cliente/Servidor, devido à natureza do sistema, em que os processos cliente farão requisições para o processo servidor, o qual é responsável pela comunicação com o banco de dados, agindo como um servidor central. Essa escolha é a melhor e menos complexa forma de realizar comunicação entre as entidades, sendo assim, é necessário centralizar os dados e operações, de maneira que ambos não interajam com regras semelhantes.
- **Responsabilidades das Entidades:**
 - **Cadastrar-se (CSU01):**
 - **Entidade Cliente:** O cliente será responsável por iniciar a comunicação com o Servidor Central, após o usuário preencher suas informações (email e senha).
 - **Entidade Servidor Central:** Será responsável por enviar as informações fornecidas pelo usuário para o banco de dados, assim, criando um novo usuário. Além disso, envia uma mensagem de confirmação para o cliente.
 - **Fazer Login (CSU02):**
 - **Entidade Cliente:** Após o usuário preencher as informações necessárias, o cliente fará a requisição de autenticação para o Servidor Central.
 - **Entidade Servidor Central:** Recebe as informações enviadas pela requisição do cliente e realiza a verificação com o banco de dados, enviando uma mensagem ao cliente com o resultado da verificação.

- Pesquisar Itens (CSU03):
 - Entidade Cliente: Após o usuário digitar o termo de busca na barra de pesquisa, o Cliente enviará o termo para o Servidor Central e exibirá os resultados da pesquisa quando a resposta chegar.
 - Entidade Servidor Central: O Servidor após receber o termo de pesquisa, irá buscá-lo no banco de dados e enviará os resultados para o Cliente.
- Filtrar anúncios por categoria (CSU04):
 - Entidade Cliente: O cliente envia uma mensagem para o Servidor, com a categoria selecionada pelo usuário, e mostra os produtos retornados pela resposta do Servidor.
 - Entidade Servidor Central: O Servidor recebe uma mensagem com a categoria selecionada e retorna todos os produtos presentes nessa categoria no banco de dados.
- Adicionar itens no carrinho (CSU05):
 - Entidade Cliente: Após o usuário selecionar a opção de adicionar um item no carrinho, o Cliente irá utilizar a lista de produtos anteriormente requisitada ao Servidor Central em outro caso de uso, e irá adicionar o produto ao carrinho.
 - Entidade Servidor Central: O Servidor foi utilizado em outro caso de uso, então não é necessário nesse caso.
- Editar itens do carrinho (CSU06):
 - Entidade Cliente: O usuário seleciona uma opção entre aumentar ou diminuir a quantidade de cada item presente no carrinho. Logo após, o Cliente atualiza o carrinho de acordo com a opção escolhida.
 - Entidade Servidor Central: O Servidor foi utilizado em outro caso de uso, então não é necessário nesse caso.

- Fechar compra (CSU07):
 - Entidade Cliente: O usuário seleciona a opção de fechar compra e o cliente envia uma requisição ao banco de dados para conferir se os produtos presentes no carrinho ainda estão presentes no banco de dados.
 - Entidade Servidor Central: Recebe uma mensagem com a lista de produtos e faz uma pesquisa no banco de dados, retornando ao cliente se os produtos estiverem presentes no estoque e retorna uma mensagem de confirmação.
- Selecionar método de pagamento (CSU08):
 - Entidade Cliente: Após o usuário selecionar a forma que deseja pagar os produtos, o Cliente irá enviar uma confirmação de pagamento para o Servidor Central.
 - Entidade Servidor Central: Após receber a confirmação de pagamento, irá atualizar o estoque do produto no banco de dados e retornar uma mensagem de confirmação.
- Criar loja (CSU09):
 - Entidade Cliente: Após o usuário selecionar a opção de criar uma loja e preencher as informações sobre a loja (nome, descrição e endereço), o Cliente irá enviar essas informações para o Servidor Central.
 - Entidade Servidor Central: Após receber as informações do Cliente, irá armazená-las no banco de dados, assim, criando uma nova loja e retornando uma mensagem de confirmação.
- Editar informações da loja (CSU10):
 - Entidade Cliente: Usuário vendedor insere as informações da loja a serem mudadas e o cliente envia uma mensagem ao Servidor.

- Entidade Servidor Central: Recebe as novas informações da loja e as atualiza no banco de dados e retorna uma mensagem de confirmação.
- Criar anúncio (CSU11):
 - Entidade Cliente: Após o usuário preencher as informações sobre o produto a ser vendido (foto, nome, descrição, estoque e categoria), o Cliente irá enviar essas informações para o Servidor Central.
 - Entidade Servidor Central: Após receber as informações do Cliente, irá armazená-las no banco de dados, assim, criando um novo anúncio e retornando uma mensagem de confirmação.
- Editar o anúncio (CSU12):
 - Entidade Cliente: Usuário vendedor insere as informações do produto a serem mudadas e o cliente envia uma mensagem ao Servidor.
 - Entidade Servidor Central: Recebe as novas informações do produto e as atualiza no banco de dados e retorna uma mensagem de confirmação.
- Mudar visibilidade do anúncio (CSU13):
 - Entidade Cliente: O usuário seleciona a nova visibilidade do produto (visível ou invisível) e o Cliente envia uma requisição alteração para o Servidor Central.
 - Entidade Servidor Central: O Servidor irá atualizar a visibilidade do produto no banco de dados e retornará uma mensagem de confirmação.
- Excluir loja (CSU14):
 - Entidade Cliente: O usuário escolhe a opção de excluir a loja e o cliente envia uma mensagem para o Servidor.

- Entidade Servidor Central: O Servidor recebe a mensagem de exclusão e exclui a loja no banco de dados, retornando uma mensagem de confirmação.
- Excluir anúncios (CSU15):
 - Entidade Cliente: O usuário escolhe a opção de excluir o anúncio e o cliente envia uma mensagem para o Servidor.
 - Entidade Servidor Central: O Servidor recebe a mensagem de exclusão e exclui o anúncio no banco de dados, retornando uma mensagem de confirmação.
- Estrutura em Camadas Físicas: O sistema seria estruturado em uma camada Cliente e uma Servidor, considerando que a camada Cliente será uma aplicação desktop e a camada Servidor guardará todos os dados e responderá às requisições do Cliente. Dessa forma, apenas teríamos a aplicação e o servidor central.

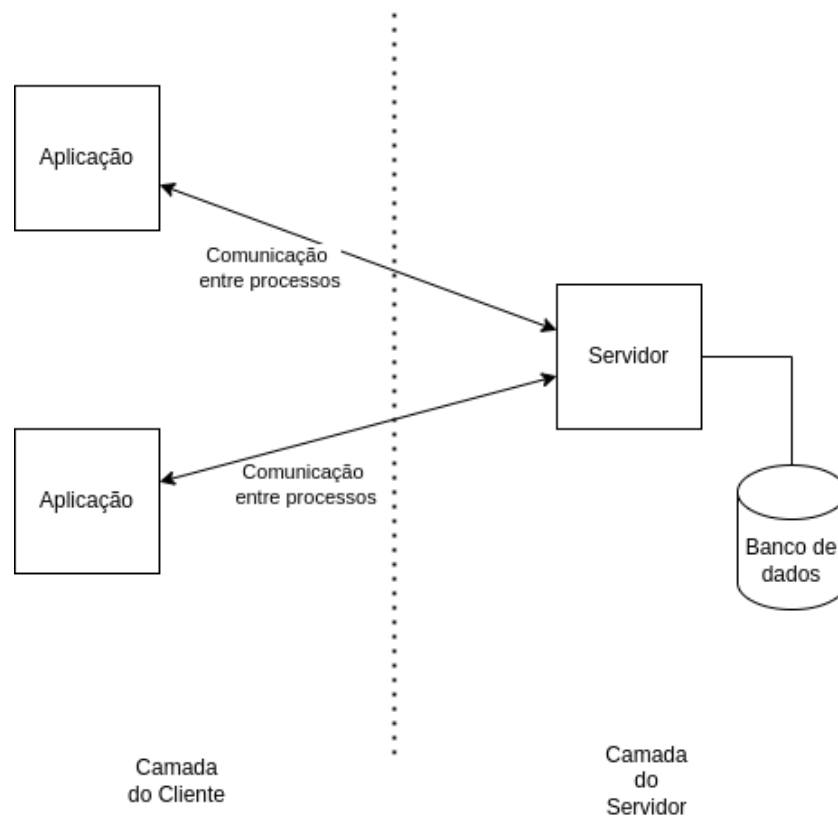


Figura 1: representação da estrutura das camadas físicas.

- Estrutura em Camadas Lógicas: No geral, é importante que haja ao menos três camadas lógicas, sendo um módulo para a funcionalidade, um para a plataforma e um para o middleware, o que vai acontecer na quarta parte do trabalho. No entanto, considerando que não será usado middleware na terceira parte, o sistema será particionado em duas camadas, sendo elas a de funcionalidade e a de plataforma.

Modelos Fundamentais

Modelo de interação

- a) Neste SD, a latência e o jitter implicarão apenas no tempo de resposta entre o cliente e o servidor, podendo interferir negativamente na experiência do usuário e fazendo com que seja necessário ter um atraso arbitrário na transmissão dos

dados, porém, como os processos podem funcionar com diferentes velocidades, isso não irá interferir no funcionamento correto do sistema. Quanto à taxa de transmissão, não terá um impacto grande, devido ao conteúdo das mensagens, que não terão um tamanho grande em bytes.

- b) O sistema será assíncrono, afinal um marketplace não tem a necessidade que processos hajam de forma sincronizada. Além disso, existe uma complexidade muito grande em fazer com que o sistema seja síncrono e seria necessária a informação do tempo tolerável para que uma mensagem fosse transmitida. No entanto, devido ao uso da Internet pública não é possível ter exatidão nesse número, levando a um projeto sem confiabilidade de sincronização.

Modelo de falhas

- Entre as falhas apresentadas em sala, uma falha que poderia atrapalhar o sistema é a falha por omissão de comunicação. Todos os tipos de falhas por omissão de comunicação (por perda de mensagem, por omissão de envio e por omissão de recebimento) poderiam ocorrer, considerando que o modelo cliente-servidor será usado, então a comunicação será de suma importância.
- No modelo assíncrono, não temos uma grande preocupação com a sincronização do sistema, então caso a comunicação atrase dentro de um limite aceitável, nada será feito. Porém, caso o atraso ultrapasse esse limite, o sistema avisará o usuário que a comunicação não foi bem sucedida e perguntará se o usuário deseja tentar novamente, e se ele assim desejar, a mensagem será enviada de novo.

Modelo de segurança

- a) Primeiramente, é válido ressaltar que esse marketplace provavelmente não será alvo de ataques externos por não apresentar dados tão significativos. No geral, não será necessário implementar proteção aos processos, porque o custo de gerenciamento e processamento é maior do que a recompensa pela

implementação da segurança. Os processos clientes, se intrusos, não irão gerar nenhum problema para o sistema no geral, pois não teriam acesso a dados sensíveis do servidos de nenhuma maneira.

- b) Quanto à proteção do canal de comunicação, será feita uma criptografia apenas para as informações sensíveis, no caso, os dados de autenticação do usuário e os dados de transação ao realizar a compra. Isso é necessário, pois com os dados de autenticação do usuário seria possível realizar ações maliciosas indesejadas, como compras e mudanças nas informações dos produtos.

Referências:

[1] UNIVERSIDADE FEDERAL DE VIÇOSA. “Modelos de Sistema (Parte I)”. Disponível em:

<https://ava.ufv.br/pluginfile.php/980196/mod_resource/content/1/4ModelosCCF355Parte1.pdf> . Acesso em: 09 de Abril.

[2] UNIVERSIDADE FEDERAL DE VIÇOSA. “Modelos de Sistema (Parte II)”. Disponível em:

<https://ava.ufv.br/pluginfile.php/980197/mod_resource/content/1/4ModelosCCF355Parte2.pdf> . Acesso em: 09 de Abril.