UNIVERSIDADE FEDERAL DE VIÇOSA - UFV - CAMPUS FLORESTAL

SISTEMAS DISTRIBUÍDOS E PARALELOS

Trabalho Prático 1

Modelagem do Sistema

Emily Lopes Almeida - Emily-Lopes

Ingred Fonseca de Almeida - ingredalmeida1

Iury Martins Pereira - Iurymartins46

Letícia Oliveira Silva - lleticiasilvaa

Florestal - MG

2024

Sumário

1. Modelo Físico

Nosso sistema distribuído é um jogo online que permite a interação de vários usuários em tempo real. Optamos por adotar o modelo Internet-Scale devido à sua capacidade de escalabilidade, distribuição geográfica eficiente e tolerância a falhas. Com esse modelo, podemos lidar com um grande número de jogadores simultâneos e garantir uma experiência consistente em diferentes localidades. Além disso, o modelo Internet-Scale facilita a implementação de atualizações e melhorias contínuas no sistema, garantindo uma evolução constante do jogo e a capacidade de acompanhar as mudanças e demandas dos jogadores.

2. Modelo de Arquitetura

2.1. Paradigma de Comunicação

Durante a fase inicial do desenvolvimento do sistema, as entidades arquitetônicas serão representadas por processos. Nesta fase, adotaremos o paradigma de comunicação direta entre processos para facilitar a troca de informações e o compartilhamento de recursos. Em seguida, na segunda fase do desenvolvimento, iremos trabalhar com abstrações por meio do uso de middleware. Neste contexto, as entidades arquitetônicas serão representadas por objetos, e o paradigma de comunicação adotado será o de invocação remota.

2.2. Modelo de Arquitetura Básico

O modelo de arquitetura básico que será usado no Sistema Distribuído EmotiCards é o modelo Cliente-Servidor (C/S). Neste modelo, os processos assumem os papéis de clientes ou servidores em que os processos clientes interagem com processos servidores, localizados possivelmente em computadores hospedeiros distintos, para acessar os recursos compartilhados que estes gerenciam.

2.3. Funções e Responsabilidades das Entidades do Modelo

Nosso sistema é composto por três entidades: cliente, servidor de aplicação e servidor de dados. Abaixo são especificadas as funcionalidades respectivas a cada uma delas:

2.3.1 Cliente

As responsabilidades e funções de uma entidade Cliente consistem em interagir com o usuário final, realizar operações como criar conta, fazer login, editar perfil, montar baralhos, solicitar a entrada em partidas bem como criar partidas e convidar usuários para as mesmas, além de se comunicar com o Servidor de Aplicação enviando solicitações e processando as respostas, atualizando a interface do usuário conforme necessário.

2.3.2 Servidor de Aplicação

Uma entidade Servidor de Aplicação tem como principal responsabilidade, receber solicitações dos clientes, processá-las e coordenar as operações do sistema como, gerenciar contas de usuário, partidas em andamento, montar baralhos e de fornecer serviços para os clientes, como autenticação, recuperação de senha e gestão de partidas.

2.3.3 Servidor de Dados

O servidor de dados é responsável por gerenciar o acesso e a comunicação com o banco de dados. Sua principal função é lidar com todas as operações relacionadas ao armazenamento, recuperação e manipulação de dados no banco de dados.

2.4. Camadas Físicas e Camadas Lógica

Há um potencial grande de que o nosso sistema distribuído seja estruturado em camadas físicas e lógicas visto que desenvolveremos um sistema robusto, modularizado e acessível.

Na fase inicial de desenvolvimento, focaremos em duas camadas lógicas: aplicações e serviços, e a plataforma. Esta escolha se justifica pelo contato direto com os processos do sistema nessa fase inicial. Já durante a segunda fase de desenvolvimento, adicionaremos uma camada lógica à modelagem: middleware, além das camadas de aplicações e serviços e plataforma. A incorporação do middleware é motivada pelo uso da API de Sockets para abstrair as entidades do sistema.

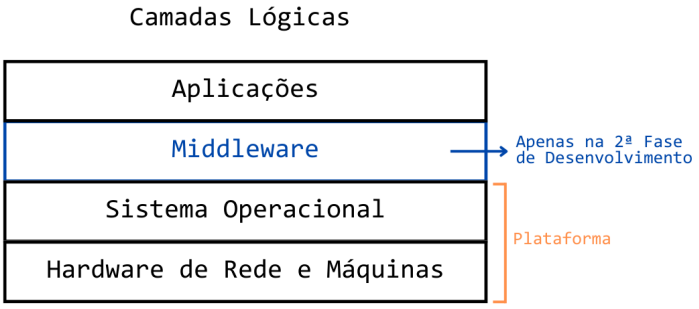

Figura 1 - Diagrama Representativo das Camadas Lógicas

Durante ambas as fases, adotaremos o modelo de camadas físicas composto por três camadas: apresentação, aplicação e dados. Essa escolha visa promover uma estruturação clara e organizada do sistema, facilitando sua manutenção e escalabilidade ao longo do desenvolvimento.

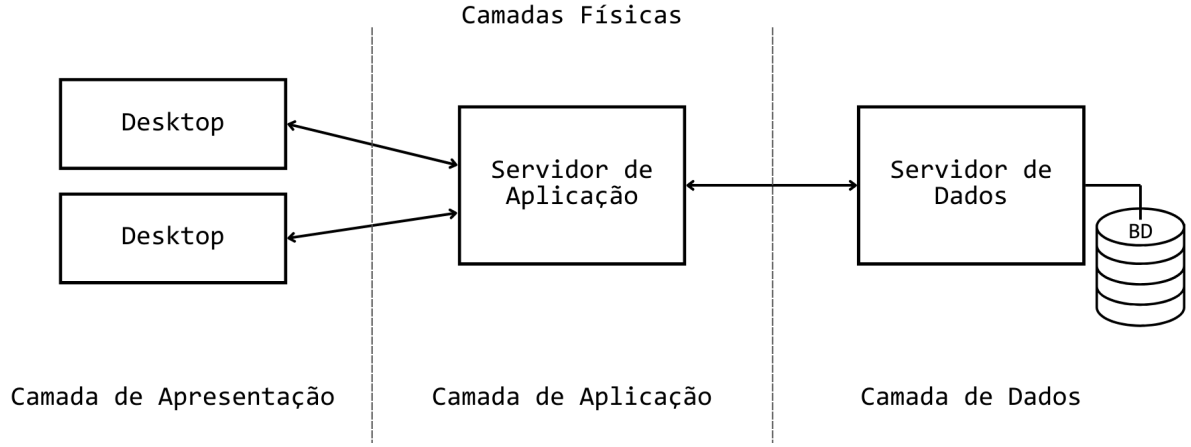

Figura 2 - Diagrama Representativo das Camadas Físicas

3. Modelos Fundamentais

3.1. Modelo de Interação

Para o sistema EmotiCards, a latência, a taxa de transmissão e o jitter são importantes para garantir uma resposta rápida às solicitações dos clientes, especialmente durante uma partida, onde três usuários estarão interagindo em máquinas distintas. No entanto, devido à falta de controle sobre as características do canal de comunicação entre os clientes e o servidor, não é possível oferecer garantias absolutas em relação a esses aspectos.
Nesse contexto, apesar da sincronização ser interessante durante a partida, a escolha do modelo assíncrono de interação entre o cliente e servidor no sistema EmotiCards é justificada pela falta de informações precisas sobre o tempo de execução das etapas do processo, tempo tolerável de transmissão de mensagens e a taxa de desvio de relógio (Clock Drift Rate) dos processos envolvidos no sistema.

3.2. Modelo de Falhas

Uma possível falha que poderia atrapalhar o sistema é a falha por omissão, por exemplo do servidor, onde ele deixa de responder às solicitações dos clientes devido a um erro de software ou hardware. Esta falha pode ser detectada pelo cliente quando ele não recebe uma resposta do servidor dentro de um tempo limite predefinido. Para lidar com essa falha, o cliente pode tentar reenviar a solicitação algumas vezes e, se não obtiver sucesso, informar ao usuário sobre o problema e sugerir tentar novamente mais tarde. Sendo assim, falhas de omissão serão detectadas mas não serão tratadas, apenas mascaradas para o usuário final.

3.3. Modelo de Segurança

O sistema EmotiCards necessita de mecanismos de segurança para proteger os dados dos usuários, prevenir acesso não autorizado e garantir a integridade das partidas. Os mecanismos de segurança que serão implementados são a autenticação de usuário durante o login e a criptografia de dados sensíveis, como senhas e informações pessoais.