

Problema 2: Jogo de Cartas Multiplayer Distribuído

Encontro	Data	Atividade
1	23/09	Apresentação do Problema e Tutorial
2	25/09	Desenvolvimento
3	30/09	Tutorial
4	02/10	Desenvolvimento
5	07/10	Tutorial
6	09/10	Desenvolvimento
7	14/10	Tutorial
8	16/10	Desenvolvimento
9	21/10	Entrega do Barema e Elaboração do Relatório
10	23/10	Entrega e Avaliação do Produto Final

Contexto

O protótipo do "Jogo de Cartas Multiplayer" desenvolvido obteve um sucesso inicial, validando a demanda por experiências online inovadoras e conectividade social. Contudo, a arquitetura centralizada, com um único servidor gerenciando a lógica do jogo, o estado dos jogadores e a comunicação, começou a apresentar limitações de escalabilidade à medida que a base de jogadores crescia. Além disso, a dependência de um servidor centralizado introduz um ponto único de falha, o que compromete a disponibilidade contínua do serviço para uma expansão global. Para atender a um público maior e garantir uma experiência de jogo robusta, a startup de vocês necessita de uma solução mais resiliente e performática.

Problema

Buscando atender a demanda atual e futura, sua startup deve realizar uma reengenharia no sistema do "Jogo de Cartas Multiplayer", migrando de uma arquitetura centralizada para uma distribuída. O objetivo é permitir que múltiplos servidores de jogo colaborem para hospedar partidas e gerenciar recursos compartilhados, visando:

- Melhorar a escalabilidade para suportar um número significativamente maior de jogadores, distribuindo a carga de processamento e gerenciamento;
- Aumentar a tolerância a falhas, eliminando o ponto único de falha do servidor central e garantindo a continuidade do serviço mesmo com a falha de componentes;
- Garantir a consistência do estado do jogo e a justiça na distribuição de cartas em um ambiente distribuído;
- Implementar uma nova funcionalidade, a troca de cartas entre os jogadores.

Restrições

A implementação do sistema deve considerar as seguintes restrições:

- **Contêineres Docker:** Os componentes do jogo como servidores de jogo e clientes devem continuar a ser desenvolvidos e testados em contêineres Docker, permitindo a execução de múltiplas instâncias no laboratório.
- **Arquitetura Descentralizada:** O sistema deve ser implementado com múltiplos servidores de jogo colaborando para gerenciar jogadores e partidas. Cada servidor pode ser responsável por uma região ou grupo de jogadores. Para uma emulação realista do cenário proposto, os elementos da arquitetura devem ser executados em contêineres separados e em computadores distintos no laboratório.

- **Comunicação entre Servidores:** deve ser implementada através de um protocolo baseado em uma API REST. Esta API deverá ser projetada pela sua equipe de desenvolvimento e pode ser testada com softwares como Insomnia ou Postman.
- **Comunicação entre Servidores e Clientes:** deve ser realizada através de um protocolo baseado no modelo publisher-subscriber nesta nova versão, onde podem ser utilizadas bibliotecas de terceiros para implementação.
- **Gerenciamento Distribuído:** A mecânica de aquisição de pacotes de cartas únicos, e que atuam como um "estoque" global deve ser gerenciada agora de forma distribuída. A solução deve garantir que, quando múltiplos jogadores conectados a diferentes servidores tentarem abrir pacotes simultaneamente, a distribuição de cartas seja justa, e que apenas um jogador receba o pacote, evitando duplicações ou perdas de cartas. Nenhuma solução centralizada deve ser empregada para evitar que, caso um servidor apresente falhas, todos os servidores sejam prejudicados ou a integridade dos pacotes seja comprometida.
- **Partidas entre Servidores:** O sistema deve permitir que jogadores conectados a servidores diferentes possam ser pareados para duelos 1v1, mantendo as garantias de pareamento único do protótipo anterior.
- **Tolerância a Falhas:** O sistema deve ser tolerante a falhas de um ou mais servidores de jogo durante uma partida ou durante suas operações, minimizando o impacto nos jogadores e na consistência dos dados.
- **Testes:** Deverá ser desenvolvido um teste de software para verificar a validade da solução em situações de concorrência distribuída e cenários de falha.

Nossas Regras

- Os alunos devem implementar o projeto em grupos de até 2 integrantes;
- O prazo final para entrega e apresentação do trabalho será **23/10/2025**;
- O código-fonte deve ser entregue devidamente comentado por meio da plataforma GitHub, com README explicando como executar o jogo junto e os scripts de testes;
- O aluno deve entregar, junto com o produto, um relatório de no máximo 8 páginas, seguindo o formato padrão da SBC (Sociedade Brasileira de Computação), contendo conceitos e justificativas para a solução adotada;
- O sistema do jogo deve ser desenvolvido, testado e apresentado no Laboratório de Redes e Sistemas Distribuídos (LARSID), onde as apresentações seguirão uma agenda sequencial definida antes da apresentação;
- Cada grupo terá 25 minutos para apresentar o sistema do jogo em funcionamento e responder às questões técnicas sobre a implementação.

Observações

- Trabalhos entregues fora do prazo serão penalizados com 20% do valor da nota + 5% por dia de atraso, dentro da mesma semana da entrega final;
- Trabalhos copiados de qualquer fonte e trabalhos idênticos terão nota ZERO;
- As informações sobre o problema podem ser alteradas no decorrer das sessões.

Avaliação

A nota final será a composição das seguintes três notas:

1. Desempenho tutorial - 30%
2. Relatório do produto (em PDF) - 20%
3. Produto no GitHub (com README) - 50%