Documentação de software
Taverna

Disciplina: Software concorrente e distribuído
Docente: Sérgio Carvalho
Equipe Corvo Manco - Alisson Braz Domingues, Ester Adaianne Oliveira Ferreira, José Carlos Lee, Luís Felipe Ferreira Silva , Vinicíus Prates Araújo

1. Overview do projeto

1.1 Contextualização

Taverna é um software de reservas de mesas e jogos para casas de jogos de tabuleiros, que são locais que oferecem um espaço físico para que seus clientes possam alugar jogos de tabuleiro e de cartas por um determinado período e desfrutar dos jogos alugados. O software foi desenvolvido durante a disciplina Software Concorrente e Distribuído, ministrada no semestre 2024/1.

1.2 Objetivos do projeto

Os objetivos principais do projeto eram estudar e aplicar princípios de concorrência e distribuição de sistemas, criando durante o semestre uma aplicação robusta e funcional que utilize alguns dos princípios estudados. Para o escopo do software de reservas de mesa e jogos desenvolvido, mencionado daqui para frente apenas como Taverna, os objetivos eram aplicar esses princípios de forma a solucionar problemas relacionados à disputa por recursos em comum no sistema, no caso as mesas e jogos para reserva, e a garantir que fosse um software distribuído.

1.3 Discussões de relevância

Taverna soluciona um problema real das casas de jogos de tabuleiros, visto que sistemas de reservas de mesas tradicionais não atendem ou atendem de forma limitada as necessidades desses estabelecimentos (discutidas com mais profundidade na sessão 2 deste documento), e por vezes é inviável para o estabelecimento manter alguém apenas para cuidar das reservas. Além disso, o desenvolvimento do software durante a disciplina serviu como maneira eficiente de aprendizado prático e entendimento mais profundo sobre princípios de concorrência e distribuição.

1.4 Trabalhos relacionados

Alguns sistemas com funcionalidades semelhantes às oferecidas pelo Taverna são sistemas de reserva de mesas utilizados por restaurantes ou outros estabelecimentos comerciais e sistemas de reservas em geral, como reservas de quartos de hotel, por exemplo.

2. Requisitos

2.1 Elicitação

O levantamento inicial de requisitos foi realizado por meio de sessões de brainstorming e estudo das principais regras de negócio dos estabelecimentos a serem atendidos pelo Taverna. Entende-se que sistemas tradicionais de reservas de mesas não atendem completamente às necessidades das casas de jogos de tabuleiro porque as reservas nesses estabelecimentos podem ser de jogos associados à mesas.

Após ampla discussão e análise, chega-se à conclusão de que o sistema a ser desenvolvido é um sistema de reserva de mesas e jogos que deve aplicar princípios de concorrência e distribuição. Não é um software para vendas de jogos de tabuleiros, e qualquer reserva de jogos deve estar obrigatoriamente associada à reserva de uma mesa, uma vez que os estabelecimentos não permitem a saída dos jogos do local e nem o aluguel de jogos sem que haja espaço devido para manuseá-los. Os requisitos funcionais foram traduzidos para a forma de histórias de usuário com critérios de aceitação. Optou-se por escrever os requisitos não funcionais de maneira tradicional.

2.2 Escopo da aplicação

A aplicação é um software para gestão das reservas de mesas na casa de jogos de tabuleiro e para gestão das reservas de jogos disponíveis no catálogo.

A aplicação não é um software para venda ou aluguel de jogos de tabuleiro, um mural de avisos sobre campeonatos realizados na Casa, um software para gestão financeira da casa de jogos de tabuleiro ou para aluguel de jogos de tabuleiro. Não é um software para gestão de outros serviços ofertados pela Casa, como por exemplo lanchonete, venda de drinks, etc.

2.3 Requisitos funcionais

Descritos no formato de histórias de usuário, seguindo o padrão:

[HUXXX - Título da história de usuário]
COMO [persona]
QUERO [o quê?]
PARA [pra que?]
“E” e “MAS” também podem ser utilizados, mas a estrutura original se mantém.

HU001 - Reservar uma Mesa
COMO cliente
QUERO reservar uma mesa em um dia,

PARA garantir que terei um lugar para jogar o meu Jogo

CA001 - Visualizar a disponibilidade das mesas

DADO o cliente está na tela de reservas

QUANDO o cliente seleciona o dia desejado para a Reserva.

ENTÃO o sistema exibe a quantidade atual de mesas disponíveis no salão principal.

CA002 - Confirmar reserva

DADO o cliente está na tela de reservas

QUANDO o cliente clica no botão "Confirmar Reserva"

ENTÃO o sistema valida a disponibilidade da Mesa e dia selecionados e confirma a Reserva.

HU002 - Reservar um Jogo

COMO cliente

QUERO reservar um ou mais Jogos em um Dia,

PARA garantir que jogarei um Jogo de minha escolha

CA001 - Visualizar a disponibilidade dos Jogos

DADO o cliente está na tela de reservas

QUANDO o cliente seleciona o Jogo em um Dia escolhido para a Reserva

ENTÃO o sistema exibe a quantidade atual de jogos disponíveis no catálogo de jogos.

CA002 - Selecionar uma mesa

DADO o cliente está na tela de reservas

QUANDO o cliente Reserva um Jogo

ENTÃO o sistema exibe mesas disponíveis para Reserva.

CA003 - Confirmar reserva

DADO o cliente está na tela de reservas

QUANDO o cliente clica no botão "Confirmar Reserva"

ENTÃO o sistema valida a disponibilidade da Mesa e Dia selecionados e confirma a Reserva.

HU005 - Cancelar reserva

COMO cliente

QUERO cancelar uma ou mais Reservas,

PARA permitir a realocação da reserva

CA001 - Visualizar reservas

DADO o cliente está na tela de reservas

QUANDO o cliente seleciona a Reserva

ENTÃO o sistema exibe a opção de “cancelar a reserva”.

CA002 - Confirmar cancelamento de reservas
DADO o cliente está na tela de reservas
QUANDO o cliente em “cancelar a reserva”
ENTÃO o sistema exibe a opção de confirmação de cancelamento da reserva.

HU006 - Cadastrar usuário
COMO cliente
QUERO cadastrar-me no sistema,
PARA permitir que seja autenticado no sistema.

CA001 - Efetuar cadastro
DADO o cliente está na tela de cadastro
QUANDO o cliente preenche seus dados e clica em cadastrar
ENTÃO o sistema cadastra os dados do cliente no banco de dados

HU007 - Autenticar usuário
COMO cliente
QUERO efetuar login no sistema,
PARA permitir a reserva de mesas e jogos.

2.4 Requisitos não funcionais

RNF01 - O sistema deve possuir uma interface intuitiva e de fácil entendimento e navegação a fim de atender ao público diverso das casas de jogo de tabuleiros.

RNF02 - A interface do sistema deve ser responsiva, a fim de que possa ser utilizado tanto em dispositivos móveis quanto em computadores ou notebooks.

RNF03 - O sistema deve utilizar princípios de concorrência de software para garantir que as mesmas mesas não sejam reservadas por mais de um cliente em um determinado horário em um determinado dia.

RNF04 - O sistema deve utilizar princípios de concorrência de software para garantir que os mesmos jogos não sejam reservados por mais de um cliente em um determinado horário em um determinado dia.

RNF05 - O sistema deve utilizar princípios de sistemas distribuídos para atender à demanda de reservas.

RNF06 - O sistema deverá ser desenvolvido em Java, utilizando o framework SpringBoot e as ferramentas Spring Data JPA, Spring Authentication, Lombok.

RNF07 - O gerenciamento das dependências do sistema deverá ser realizado com o Maven.

RNF08 - O sistema deverá ser hospedado no servidor

RNF09 - O sistema deverá implementar o balanceador de carga

3. Fundamentos de SCD relacionados ao projeto

3.1 Princípios de softwares concorrentes

3.1.1 - Sincronização: as tarefas concorrentes devem se coordenar entre si para evitar condições de corrida e garantir a consistência dos dados.

3.1.2 - Compartilhamento de recursos: os recursos compartilhados e concorridos por tarefas devem ser gerenciados para evitar conflitos.

3.1.3 - Isolamento: as tarefas concorrentes não devem interferir umas nas outras.

Esses princípios de softwares concorrentes foram abordados e tratados no sistema a fim de garantir a consistência dos dados sobre disponibilidade de mesas e jogos para reserva, evitando assim que reservas duplicadas ou com o mesmo recurso a ser utilizado por dois clientes fossem realizadas e evitando também erros na hora de realizar ou cancelar reservas ou na exibição da disponibilidade de mesas ou jogos. Foi utilizado um modelo de concorrência por mensagens, em que os processos se comunicam entre si através de websockets. O gerenciamento dos recursos como mesas e jogos foi feito com o uso de pessimistic lock e mutex e estruturas de dados que podem ser usadas com threads concorrentes, seguramente como listas.

A devida aplicação dessas ferramentas e respeito aos princípios de concorrência de software garantiu o desenvolvimento de um software que evita situações em que duas ou mais threads ficam permanentemente bloqueadas esperando por recursos, também conhecidas como deadlocks, e situações em que uma thread não consegue acesso aos recursos necessários para continuar a execução porque outras threads estão constantemente usando esses recursos, também conhecidas como starvation.

3.2 Princípios de sistemas distribuídos

3.2.1 - Transparência: a complexidade do sistema e funcionalidades como localização de recursos e migração de processos são escondidos do usuário final e, se pertinente, dos desenvolvedores.

3.2.2 - Escalabilidade: Capacidade do sistema de crescer e se adaptar ao aumento de carga de trabalho sem perder desempenho.

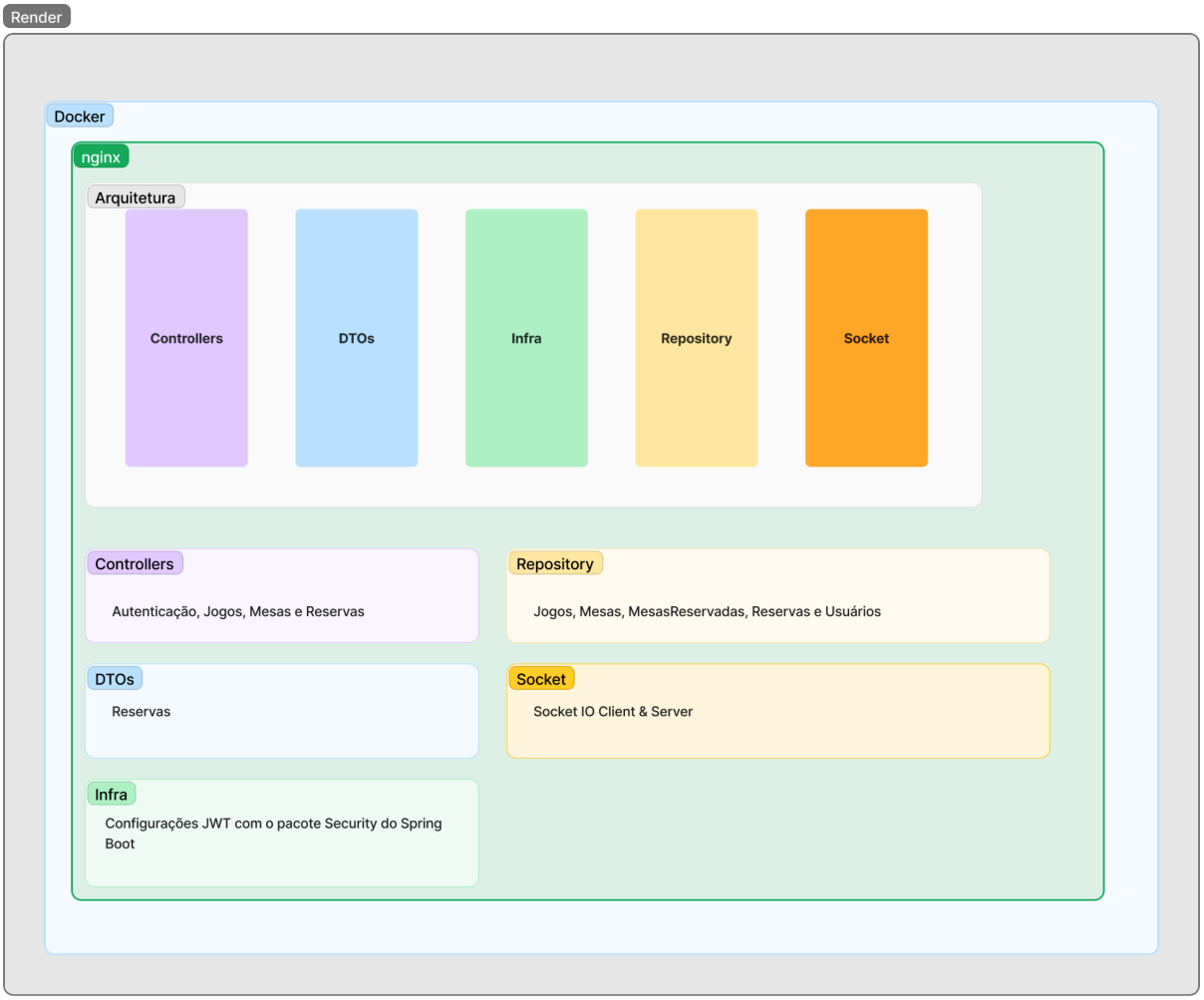
3.2.3 - Tolerância a falhas: Capacidade do sistema de continuar funcionando corretamente mesmo quando algumas de suas partes falham.

3.2.4 - Desempenho: Otimização da utilização dos recursos a fim de reduzir a latência de comunicação entre os componentes do sistema.

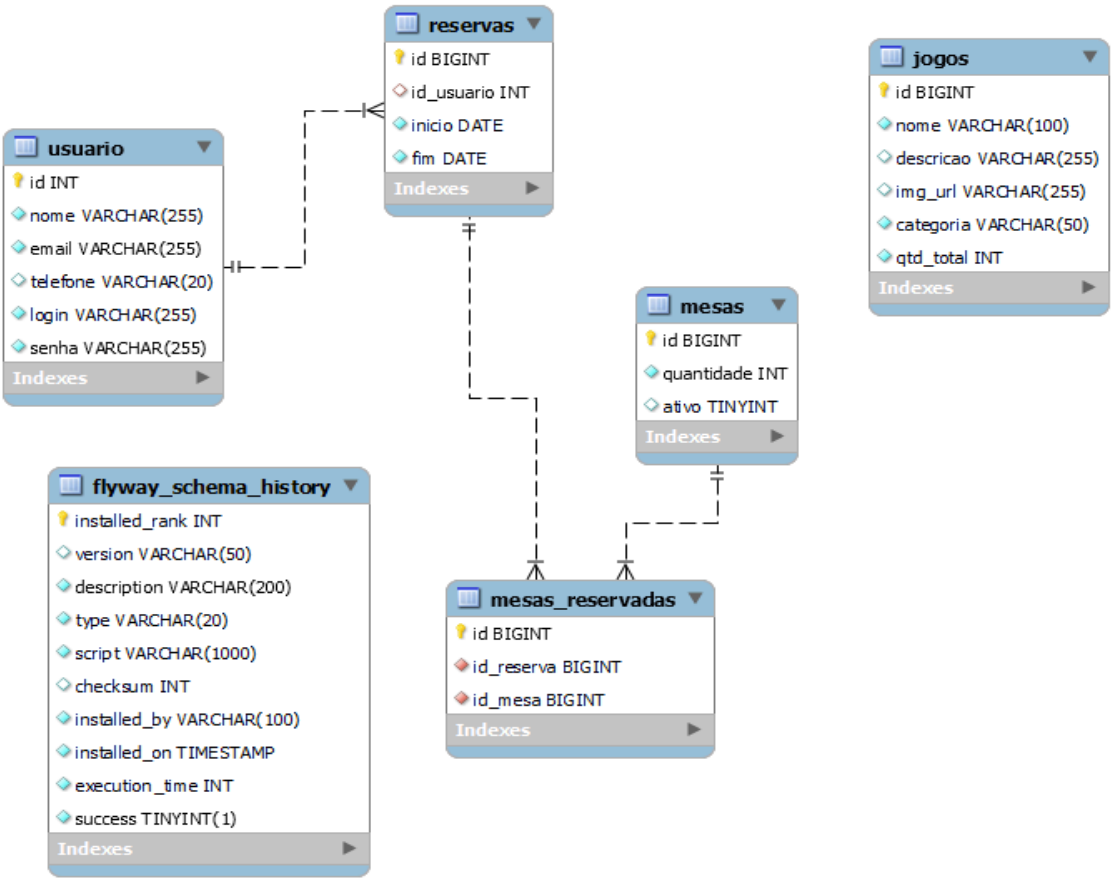
Esses princípios de sistemas distribuídos foram abordados e tratados no sistema a fim de garantir o desenvolvimento de um software funcional e com desempenho satisfatório, que atenda de maneira eficiente à sua função. Com uma arquitetura Cliente-Servidor, o sistema foi hospedado em um servidor para poder ser acessado por diversos usuários ao mesmo tempo e implementa um balanceador de carga para garantir a devida distribuição das requisições por recursos. Foi também implementada a necessidade de cadastro e autenticação do usuário para que possa realizar uma reserva, o que colabora com a segurança dos recursos.

4. Resultados

4.1 Design arquitetural



4.2 Design dos dados



4.3 Limitações

As principais limitações encontradas durante o desenvolvimento do trabalho foram relacionadas à questão do tempo de desenvolvimento. Foi necessário reduzir o escopo e retrabalhar alguns dos requisitos inicialmente definidos a fim de entregar o software funcional em tempo hábil e com os conceitos necessários devidamente aplicados. Uma das limitações de aplicação do software é que para ser utilizado por estabelecimentos comerciais reais, ele precisaria atender às regras definidas na Lei Geral de Proteção de Dados, uma vez que exige dados como email e senha dos usuários para que as reservas sejam realizadas. Dentro do escopo do desenvolvimento para a disciplina de Software Concorrente e Distribuído essa questão não foi considerada prioritária.

4.4 Trabalhos futuros

Há algumas melhorias e implementação de funções que podem ser desenvolvidas para o software Taverna em trabalhos futuros, como a implementação da parte de gestão de mesas e jogos no catálogo por parte de administradores devidamente cadastrados e autorizados pelos estabelecimentos contratantes e a implementação de medidas de segurança de dados para que o software se adeque à LGPD.