



Orientação a Objetos | FGA0158 | Turma 01 (14:00-15:50) | 2025.1

Professor: André Luiz Peron Martins Lanna

Atividade: Trabalho Prático Orientação a Objetos

Nome: Eduardo Rodrigues Reis

Matrícula: 242015595

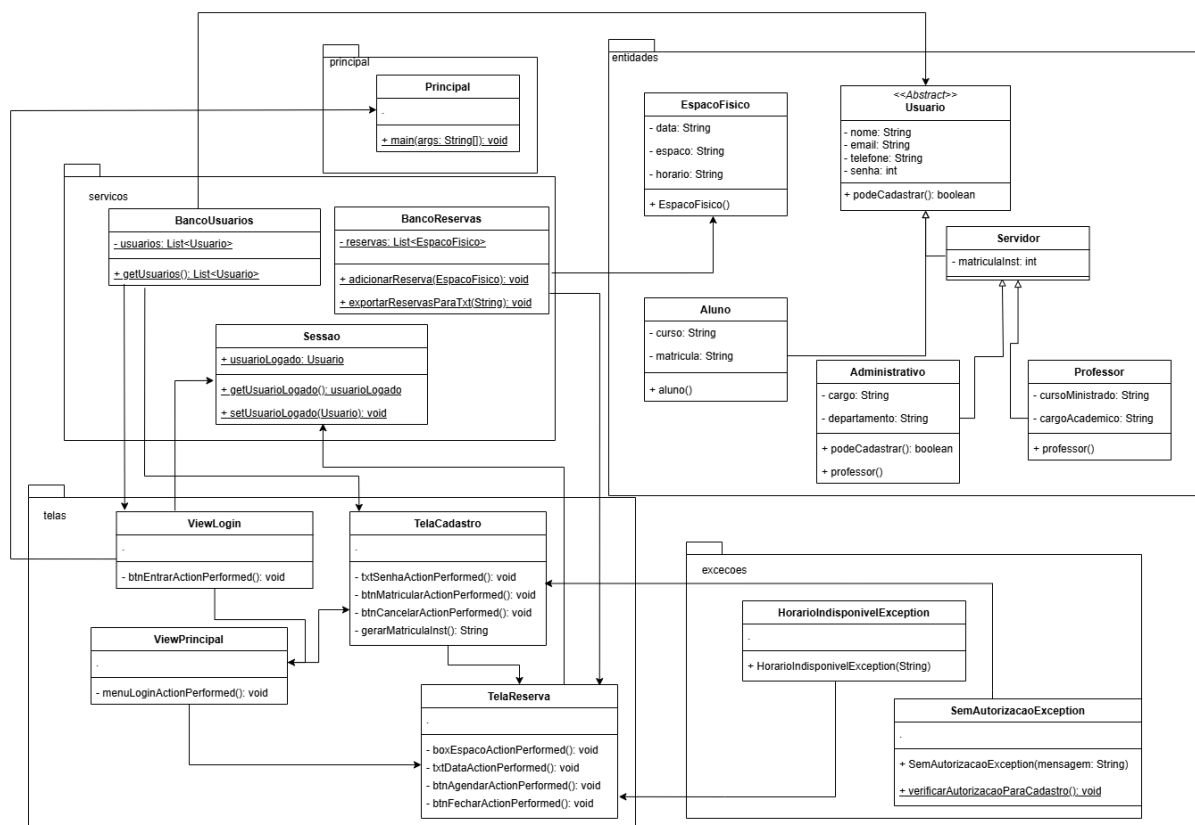
Nome: Luis Gabriel dos Santos Galeno

Matrícula: 242015906

Nome: Marcos Vinicius Monteiro de Aguiar

Matrícula: 190092394

Diagrama de Classes UML



1. Explicação das associações, heranças e polimorfismos aplicados e das exceções personalizadas

Foi utilizado herança no pacote entidades, onde a classe Usuário é uma classe abstrata, pois só pode-se criar objetos da classe Aluno, Administrativo ou Professor. Ainda nesse pacote, foi utilizado também um polimorfismo por sobrescrita no método podeCadastrar(), porque dessa forma, todos os usuários estão sem autorização para realizar cadastro de novos usuários, somente o Administrativo, classe que foi aplicado o polimorfismo por sobrescrita que pode cadastrar retornando true. Caso o usuário sem autorização tentar realizar o cadastro “cairá” na classe SemAutorizacaoException dentro do pacote excecoes. Nessa classe é herdada a Exception, isso significa que deve ser tratada com try/catch ou deve ser declarada com throws em métodos, possui um construtor personalizado chamando o super, da sua SuperClasse. Ainda nessa classe ela possui o método static verificarAutorizacaoParaCadastro(), que verifica se o usuário logado tem permissão para cadastrar, ela é static para poder usar diretamente sem precisar instanciar a classe, e ainda possui a integração com a classe Sessao.

A classe Sessao é responsável por controlar o usuário atualmente logado no sistema, ela está no pacote servicos, ela armazena e fornece acesso ao usuário logado em qualquer parte do sistema, ela possui um atributo static, que significa que pertence à classe e não a uma instância, e só pode haver um usuário autenticado na sessão atual. Possui um método static getUsuarioLogado() que é usado para acessar de qualquer lugar do sistema, sem precisar instanciar a classe, também possui um setUsuarioLogado(Usuario usuario) que define quem é o usuário logado no momento, que é chamado logo após o login, com o objeto Usuario recuperado do banco.

A classe BancoUsuarios no pacote de servicos simula um BD e fornece uma lista de usuários. possui um atributo usuarios que é uma lista static e final, porque a lista pertence à classe e a referência da lista não pode ser trocada, mas o conteúdo sim. Possui um bloco de inicialização estático, ele é executado automaticamente uma vez e tem três usuários de exemplo.

Seguindo o mesmo objetivo, a classe BancoReservas simula um BD em memória que armazena reservas feitas evitando conflito de horários e permite consultar as reservas já feitas além de exportar para um arquivo .txt. Nela temos o método adicionarReserva que verifica se já existe uma reserva no mesmo espaço, data e horário e se houver lança uma exceção HorarioIndisponivelException, se disponível, adiciona à lista. Temos também um método para exportarReservasParaTxt(String caminhoArquivo). É utilizado generics para evitar a necessidade de casting manuais.

No pacote telas, foram implementadas utilizando Java Swing. Temos a tela login que possui a lógica de login que verifica se o email e senha corresponde a que temos na classe BancoUsuarios. Na tela principal, temos uma box onde pode-se cadastrar usuários alunos, administrativo ou professor. Na classe TelaCadastro quando o btnMatricular é clicado ele captura os campos preenchidos cria um objeto aluno, administrativo ou professor e gera uma matrícula automática utilizando um método auxiliar gerarMatricula(), para alunos possui a estrutura: dois primeiros dígitos ano de ingresso, terceiro dígito semestre de ingresso, e os últimos seis dígitos gerados aleatoriamente. Para servidores temos o gerarMatriculaInst(), que possui a mesma lógica, exceto a dos três primeiros dígitos. Após isso adiciona o aluno ao BancoUsuarios.

Ainda na classe ViewPrincipal temos dois botões btnAgendar e btnHistorico, caso o usuário clique em agendar abre uma TelaReserva temos ação do botão agendar, ele lê os dados digitados pelo usuário e cria um objeto EspacoFisico caso o horário, data e espaço não estiverem ocupado. No campo de data ele converte os quatro dígitos digitados pelo usuário em formato de data dd/MM/YYYY.