

Relatório Trabalho Prático

Orientação a Objetos

Prof. Andre Lanna

Alunos:

Artur Fernandes Galdino - 241010923

Fábio Alessandro Santos Vieira - 241011804

Pedro Augusto Macedo Del Castilo – 241025354

João Eduardo de Souza Leles – 241025274

Este projeto foi desenvolvido como parte da disciplina de Orientação por Objetos (OO) na Faculdade do Gama (FGA) da Universidade de Brasília (UnB). O sistema permite o cadastro de usuários (alunos, professores e servidores) e a reserva de espaços físicos como laboratórios, salas de aula e auditórios.

- Foi utilizado associações em diversas partes do código
- A utilização de associações é de extrema importância, pois cada classe pode assumir sua função e se comunicar através delas
- Um exemplo de utilização de associações é a instanciação dos objetos SalaDeAula, Laboratorio, Auditorio na classe MostrarEspacos

2.2 Herança

- Foi aplicada herança para evitar duplicação de código (atributos como nome, email, matrícula)
- Permite tratamento polimórfico de diferentes tipos de usuários
- Facilita a extensão do sistema para novos tipos de usuários

2.2.1 Usuarios

- Alunos, Professores, Servidores são subclasses.

2.2.2 EspacosFisicos

- Laboratorio, SalaDeAula, Auditorio são subclasses.

2.3 Polimorfismos

2.3.1 Sobrescrita

- Pacote Entidades

- toString();
- getAnoDeIngresso();
- validaEmail(String matricula, String email);

- Pacote Servicos

- getEquipamentoDisponivel();

2.3.2 Sobrecarga

- Pacote Cadastro

- montaRelatorio(Usuarios usuarioLogado) e montaRelatorio()

- Pacote Servicos

- exibirReservas(Usuarios usuarioLogado, String reservas) e exibirReservas(EspacosFisicos espacosFisicos)

2.3.3 Paramétrico

- ArrayList<Usuarios> listaUsuarios
- ArrayList<EspacosFisicos> listaEspacos
- Map<String, String> reservasPorHorario
- Map<String, ArrayList<String>> reservasPorUsuario
- Map<String, String> reservasPorEspaco
- ArrayList<String> equipamentosV

3. Exceções Customizadas

3.1 DiasExcedidosException

Propósito: Impedir que alunos reservem espaços em dias consecutivos

Justificativa:

- Regra de negócio específica para alunos
- Evita monopolização de recursos
- Facilita a manutenção da regra em uma única linha

3.2 JaReservouException

Propósito: Impedir que um usuário reserve o mesmo horário múltiplas vezes

Justificativa:

- Evitar conflitos de agendamento
- Melhora distribuição de recursos
- Fornece feedback claro ao usuário