

Universidade Federal de Santa Catarina

Disciplina: INE5605-02238B (20242) - Desenvolvimento de Sistemas Orientados a Objetos I

Professora: Thais Bardini Idalino

Alunos: Júlio César De Freitas Raposo(24105042) e Reni Rogete Rosa Ferreira Júnior(24104772)

Data: 30/09/2024

Entrega parcial 1

Parte 1 (Parcial): Deve ser apresentado para o(a) professor(a) em sala de aula no dia 30/09/2024:

- Um documento em formato PDF descrevendo como será a divisão das atividades do trabalho entre os membros da equipe;
- Uma figura com o diagrama de classes somente das entidades (Model), seguindo a notação UML 2;
- Código-fonte orientado a objetos em Python somente com a implementação das entidades (Model).

Descrição do sistema:

Este sistema de gestão de cinema permite o controle sobre filmes, sessões, salas e vendas de ingressos. Ele oferece funcionalidades para o cadastro e gerenciamento de filmes, clientes, funcionários e sessões de exibição. Além disso, o sistema registra a compra de ingressos, associando clientes a assentos específicos em cada sessão. Por fim, gera relatórios de vendas, mostrando ocupação de salas, receita gerada e popularidade de filmes e sessões, facilitando o acompanhamento e otimização das operações do cinema.

Divisão das atividades do trabalho entre os membros da equipe:

Modelagem UML: Reni

Revisão Modelagem UML: ambos

Implementação das entidades:

1- class Filme: Júlio

Revisão: ambos

2- class Funcionario: Reni

Revisão: ambos

3- class Ingresso: Júlio

Universidade Federal de Santa Catarina

Revisão: ambos

4- class Sessão: Júlio

Revisão: ambos

5- class Cliente: Reni

Revisão: ambos

6- class Sala: Júlio

Revisão: ambos

7- class Caixa: Júlio

Revisão: ambos

8- class Genero: Reni

Revisão: ambos

9- class Ingresso: Júlio

Revisão: ambos

10- class sala: Júlio

Revisão: ambos

11- class Sessao: Ambos

Revisão: ambos

Entrega completa - Parte 3:

UML:

```

classDiagram
    class Sistema {
        + lista_apresentacoes()
        + mostra_mensagem()
        + pega_preco_ingr()
    }
    class Sala {
        + idSala int
        + nomeSala string
        + capacidade int
        + <<criar>> _insr_idSala int, nomeSala string, capacidade int
        + <<atualizar>> _atual_idSala int
        + <<deletar>> _delSala(string)
        + <<buscar>> _buscSala(int)
    }
    class Filme {
        + idFilme int
        + titulo string
        + duracao int
        + genero string
        + tipoExibicao TipoExibicao
        + <<criar>> _insr_idFilme int, titulo string, duracao int, genero string, tipoExibicao TipoExibicao
        + <<atualizar>> _atual_idFilme int
        + <<deletar>> _delFilme(string)
        + <<buscar>> _buscFilme(int)
    }
    class Ingresso {
        + idIngresso int
        + valor float
        + cliente Cliente
        + sessao Sessao
        + <<criar>> _insr_idIngresso int, valor float, cliente Cliente, sessao Sessao
        + <<atualizar>> _atual_idIngresso int
        + <<deletar>> _delIngresso(string)
    }
    class Funcionario {
        + cpf int
        + idFuncionario int
        + nome string
        + cargo string
        + salario float
        + periodo string
        + <<criar>> _insr_idFuncionario int, nome string, cargo string, salario float, periodo string
        + <<atualizar>> _atual_idFuncionario int
        + <<deletar>> _delFuncionario(string)
    }
    class Cliente {
        + cpf int
        + nome string
        + time string
        + tempoValido int
        + sessaoAguardando List-Sessao
        + <<criar>> _insr_idCliente int, nome string
        + <<atualizar>> _atual_idCliente int
        + <<deletar>> _delCliente(string)
    }
    class Sessao {
        + idSessao int
        + horario string
        + assentoCapacidade int
        + valorIngresso float
        + time string
        + sala Sala
        + <<criar>> _insr_idSessao int, horario string, time string, sala Sala, valorIngresso float, assentoCapacidade int
        + <<atualizar>> _atual_idSessao int
        + <<deletar>> _delSessao(string)
    }
    class ControllerSala {
        + listaSala()
        + <<criar>> _insr_idSala()
        + <<atualizar>> _atual_idSala()
        + <<deletar>> _delSala()
        + <<buscar>> _buscSala(int)
    }
    class ControllerFilme {
        + listaFilme()
        + <<criar>> _insr_idFilme()
        + <<atualizar>> _atual_idFilme()
        + <<deletar>> _delFilme()
        + <<buscar>> _buscFilme(int)
    }
    class ControllerIngresso {
        + listaIngresso()
        + <<criar>> _insr_idIngresso()
        + <<atualizar>> _atual_idIngresso()
        + <<deletar>> _delIngresso()
        + <<buscar>> _buscIngresso(int)
    }
    class ControllerFuncionario {
        + listaFuncionario()
        + <<criar>> _insr_idFuncionario()
        + <<atualizar>> _atual_idFuncionario()
        + <<deletar>> _delFuncionario()
        + <<buscar>> _buscFuncionario(int)
    }
    class ControllerCliente {
        + listaCliente()
        + <<criar>> _insr_idCliente()
        + <<atualizar>> _atual_idCliente()
        + <<deletar>> _delCliente()
        + <<buscar>> _buscCliente(int)
    }
    class ControllerSessao {
        + listaSessao()
        + <<criar>> _insr_idSessao()
        + <<atualizar>> _atual_idSessao()
        + <<deletar>> _delSessao()
        + <<buscar>> _buscSessao(int)
    }
    class ControllerSistema {
        + ControllerSala()
        + ControllerFilme()
        + ControllerIngresso()
        + ControllerFuncionario()
        + ControllerCliente()
        + ControllerSessao()
    }

    Sistema --> Sala : sala
    Sistema --> Filme : filme
    Sistema --> Ingresso : ingresso
    Sistema --> Funcionario : funcionario
    Sistema --> Cliente : cliente
    Sistema --> Sessao : sessao

    Sala --> ControllerSala : sala
    Filme --> ControllerFilme : filme
    Ingresso --> ControllerIngresso : ingresso
    Funcionario --> ControllerFuncionario : funcionario
    Cliente --> ControllerCliente : cliente
    Sessao --> ControllerSessao : sessao

    ControllerSistema --> ControllerSala : sala
    ControllerSistema --> ControllerFilme : filme
    ControllerSistema --> ControllerIngresso : ingresso
    ControllerSistema --> ControllerFuncionario : funcionario
    ControllerSistema --> ControllerCliente : cliente
    ControllerSistema --> ControllerSessao : sessao

```

https://drive.google.com/file/d/1Z3vun_EONPynVJLinsb_PNBkRAIUdPS_/view?usp=sharing

<https://github.com/Reniexp/trabalhoDSO>

https://drive.google.com/file/d/1JtnMRtNmP6n_JJ6V10poKMvQJuJnZY3r/view?usp=sharing