

Sala // cria uma sala

- | | |
|--|---|
| <ul style="list-style-type: none">• IDSala; // cada sala vai ter um ID de identificação• lAssento; // número de linhas de assentos que a sala possui• cAssento; // número de assentos que cada linha da sala possui• Sala(); // método construtor da sala, atribuir um número para cada cadeira• tamanhoSala(); // método para calcular o tamanho total da sala (número de assentos)• exibirID(); // exibe o ID da sala | <ul style="list-style-type: none">• RepositorioSala; // banco de dados para armazenar os objetos de Sala• ServicoSala; // classe de serviços para ações intermediárias |
|--|---|

ServicoSala // classe de serviços para ações intermediárias

- | | |
|--|--|
| <ul style="list-style-type: none">• removerSala(IDSala); // remove uma sala pelo ID• criarSala(Sala()); // chama o método construtor da classe Sala para criar uma sala• editarSala(IDSala); // edita uma sala pelo ID• inativarSala(IDSala); // deixa uma sala inativa pelo ID | <ul style="list-style-type: none">• RepositorioSala; // banco de dados para armazenar os objetos de Sala |
|--|--|

Filme // cria um filme novo

- | | |
|--|--|
| <ul style="list-style-type: none">• IDFilme; // cada filme vai ter um ID de identificação• tituloFilme; // nome do filme• classificacaoFilme; // classificação do filme• duracaoFilme; // duração do filme• generoFilme; // gênero do filme• precoFilme; // preço de cada ingresso para o filme• Filme(); // método construtor do filme• exibirID(); // exibe o ID do filme | <ul style="list-style-type: none">• RepositorioFilme; // banco de dados para armazenar os objetos de Filme• ServicoFilme; // classe de serviços para ações intermediárias |
|--|--|

ServicoFilme // classe de serviços para ações intermediárias

- | | |
|--|--|
| <ul style="list-style-type: none">• cadastrarFilme(Filme(tituloFilme, classificacaoFilme, duracaoFilme, generoFilme, precoFilme)); // chama o método construtor de Filme para cadastrar um filme novo• editarFilme(IDFilme); // edita um filme pelo ID• escondermostrarFilme(IDFilme); // retira ou mostra um filme em cartaz pelo ID• excluirFilme(IDFilme); // exclui o filme do banco de dados | <ul style="list-style-type: none">• RepositorioFilme; // banco de dados para armazenar os objetos de Filme |
|--|--|

Sessao // atribui um filme em uma sessão (sala e horário)

- | | |
|---|---|
| <ul style="list-style-type: none">• IDSessao; // cada sessão vai ter um ID de identificação• dataSessao; // escolhe a data da sessão• horaSessao; // escolhe a hora da sessão• Sessao(IDSala, IDFilme); // método construtor da sessão• exibirID(); // exibe o ID da sessão | <ul style="list-style-type: none">• RepositorioSessao; // banco de dados para armazenar os objetos de Sessao• ServicoSessao; // classe de serviços para ações intermediárias• ServicoReserva; |
|---|---|

ServicoSessao // classe de serviços para ações intermediárias

- | | |
|---|--|
| <ul style="list-style-type: none">• cadastrarSessao(Sessao(IDSala, IDFilme, dataSessao, horaSessao));• editarSessao(IDSessao);• invativarSessao(IDSessão);• removerSessao(IDSessão); | <ul style="list-style-type: none">• RepositorioSessao; |
|---|--|

Cliente // cria um novo cliente

- | | |
|---|--|
| <ul style="list-style-type: none">• IDCliente; // cada cliente vai ter um ID de identificação• nomeCliente; // nome do cliente• emailCliente; // email do cliente• idadeCliente; // idade do cliente• cpfCliente; // CPF do cliente• Cliente(); // método construtor de Cliente• exibirID(); // exibe o ID do cliente | <ul style="list-style-type: none">• RepositorioCliente; // Banco de dados para armazenar os objetos de Cliente• ServicoCliente; // classe de serviços para ações intermediárias |
|---|--|

ServicoCliente // classe de serviços para ações intermediárias

- | | |
|---|--|
| <ul style="list-style-type: none">• cadastrarCliente(Cliente(nomeCliente, emailCliente, idadeCliente, cpfCliente)); // chama o método construtor de Cliente para cadastrar um cliente novo• editarCliente(IDCliente); // edita cliente pelo ID• invativarCliente(IDCliente); // invativa um cliente pelo ID• removerCliente(IDCliente); // remove um cliente do banco de dados | <ul style="list-style-type: none">• RepositorioCliente; // Banco de dados para armazenar os objetos de Cliente• ServicoReserva; // classe de serviços para ações intermediárias |
|---|--|

Funcionario // cria um novo funcionario

- | | |
|---|--|
| <ul style="list-style-type: none">• IDFuncionario; // cada funcionário vai ter um ID de identificação• nomeFuncionario; // nome do funcionário• senhaFuncionario; // senha do funcionário• cargoFuncionario; // cargo do funcionário• Funcionario(); // método construtor do funcionário• exibirID(); // exibe o ID do funcionário | <ul style="list-style-type: none">• RepositorioFuncionario; // banco de dados para armazenar os objetos de Funcionario• ServicoFuncionario; // classe de serviços para ações intermediárias |
|---|--|

ServicoFuncionario // classe de serviços para ações intermediárias

- | | |
|---|--|
| <ul style="list-style-type: none"> • cadastrarFuncionario(Funcionario(nomeFuncionario, senhaFuncionario, cargoFuncionario); // chama o método construtor de Funcionario para cadastrar um novo funcionário • editarFuncionario(IDFuncionario); // edita um funcionário pelo ID • demitirFuncionario(IDFuncionario); // demite um funcionário pelo ID • mudarNome(IDFuncionario); // muda o nome de um funcionário pelo ID • mudarSenha(IDFuncionario); // muda a senha de um funcionário pelo ID | <ul style="list-style-type: none"> • RepositorioFuncionario; // banco de dados para armazenar os objetos de Funcionario |
|---|--|

Reserva // cria uma nova reserva

- | | |
|--|---|
| <ul style="list-style-type: none"> • IDReserva; // cada reserva vai ter um ID de identificação • Reserva(IDSessao, IDCliente); // método construtor da reserva, reserva em número de cadeira caso não seja duplicata • duplicataReserva(IDSessão, Sala()); // checa duplicata da sessão e retorno se é ou não duplicada, ele usa o método sala que atribui um valor para cada cadeira • exibirID(); // exibe o ID da reserva | <ul style="list-style-type: none"> • RepositorioReserva; // banco de dados para armazenar os objetos de Reserva • ServicoReserva; // classe de serviços para ações intermediárias |
|--|---|

Compra

- | | |
|--|---|
| <ul style="list-style-type: none"> • IDCompra; // cada compra vai ter um ID de identificação • pagamentoCompra; // tipo de pagamento da compra • quantidadeCompra; // quantidade de ingressos comprados • calculaCompra(); // calcula valor total da compra • Compra(Reserva()); // método construtor de compra, confirma uma reserva somente se o pagamento for concluído • imprimirCompra(); // imprime a compra em PDF para enviar por email ou imprimir na impressora • exibirID(); | <ul style="list-style-type: none"> • RepositorioCompra; // banco de dados para armazenar os objetos de Reserva |
|--|---|