

ID	CLASSE TESTE	DATA	VERSÃO	TIPO	NOME TESTE	DESCRIÇÃO	ENTRADA	SAÍDA	GRAVIDADE	EXECUTADO	MOTIVO DA FALHA	CORRIGIDO
1	EventosTest	12/05/2024	1.0	Unidade	naoValidarEventoComNomeNulo()	Verifica se a aplicação impede a criação de um evento com nome nulo e retorna uma mensagem de erro específica.	Nome: NULL, Local: 'Rio de Janeiro', Data: '2025-01-01'	Mensagem "Falha na validação: O nome do evento é inválido."	Grave	Sim	-	-
2		12/05/2024	1.0	Unidade	naoValidarEventoComDataPassada()	Assegura que o sistema rejeita a tentativa de criar um evento com uma data anterior à data atual, retornando uma mensagem de erro adequada.	Nome: 'Concerto', Local: 'São Paulo', Data: '2020-01-01'	Mensagem "Falha na validação: A data do evento não é futura."	Grave	Sim	-	-
3		12/05/2024	1.0	Unidade	naoValidarEventoComLocalNulo()	Testa se a criação de um evento é bloqueada quando o local é nulo, e se uma mensagem de erro é corretamente gerada.	Nome: 'Festival', Local: NULL, Data: '2025-05-01'	Mensagem "Falha na validação: O local do evento é inválido."	Grave	Sim	-	-
4	EventosDAOTest	12/05/2024	1.0	Integração	inserirERecuperarEvento()	Confirma se um evento pode ser corretamente inserido e depois recuperado do banco de dados, verificando a integridade dos dados retornados.	Nome: 'Palestra', Local: 'Curitiba', Data: '2025-08-20'	ID, Nome, Local e Data do evento correspondem aos dados inseridos.	Médio	Sim	-	-
5		12/05/2024	1.0	Integração	excluirEventoTest()	Verifica se um evento é inserido e depois corretamente excluído, com uma confirmação de exclusão.	Nome: 'Conferência', Local: 'Belo Horizonte', Data: '2025-09-15'	Mensagem "Evento excluído com sucesso."	Médio	Não	O evento ainda existe no banco de dados	Não
6	ClientesTest	19/05/2024	1.0	Unidade	emailInvalidoDeveLancarExcecao()	Verifica se o email foi digitado corretamente e se não, lança exceção, espera emails como user@example.com ou info.test@domain.net, mas não user@.com ou @example.com.	E-mail: 'user@example.com'	Mensagem "E-mail inválido"	Médio	Sim	-	-
7		19/05/2024	1.0	Unidade	telefoneInvalidoDeveLancarExcecao()	Verifica se o telefone tem expressão regular diferente de (xxxxxx-xxxx)	Telefone: '4699109-5696'	Mensagem "Telefone inválido"	Médio	Sim	-	-
8	IngressosTest	19/05/2024	1.0	Unidade	precoInvalidoDeveLancarExcecao()	Verifica se o valor do ingresso tem formatação diferente de decimal (5,2)	Preço: '100,1234'	Mensagem "Preço inválido"	Grave	Sim	-	-
9	IngressoDAOTest	19/05/2024	1.0	Integração	inserirIngressoComEventoInvalidoDeveFalhar()	Confirma que a inserção de um ingresso com um ID de evento não existente resulta em uma falha devido à violação de chave estrangeira.	ID Ingresso: 1, Evento ID: 57, Tipo: 'VIP', Preço: 250.00	Mensagem "Conflito de FK"	Grave	Não	Necessário setup inicial com @Before	Sim
10		19/05/2024	1.0	Integração	inserirEVerificarIngresso()	Testa se um ingresso inserido é recuperado corretamente, verificando se os dados recuperados correspondem aos inseridos.	ID Ingresso: 101, Evento ID: 201, Tipo: 'VIP', Preço: 150.00	Id, IdEvento, Tipo e Preço.	Médio	Não	Ajuste em método MapearParaEntidade() Linha de código que ocasionava erro foi removida. ERRO - SEVERE: Erro ao mapear para entidade: A nome da coluna idEvento não foi encontrado neste ResultSet.	Sim
11	ClientesDAOTest	19/05/2024	1.0	Integração	deletarClienteTest()	Testa se o cliente é corretamente deletado do banco de dados.	ID Cliente: 57	Mensagem de exclusão confirmada	Médio	Não	O método buscarPorCodigo não retornou null como esperado após a exclusão do cliente.	Não
12		19/05/2024	1.0	Integração	atualizarClienteTest()	Verifica se as atualizações no registro de um cliente são efetivamente aplicadas e mantidas no banco de dados.	ID Cliente: 58, nome: Maria, email: maria@email.com, telefone: 2199876-5432, endereco: Avenida X, 456	Dados do cliente devem ser atualizados no banco de dados	Grave	Sim	-	-
13		19/05/2024	1.0	Integração	buscarClientePorNomeTest()	Insere um nome e verifica a existência deste cliente no banco de dados.	Nome: Ana Flavia Perin	Cliente com o nome especificado deve ser encontrado no banco de dados	Grave	Sim	-	-
14		20/05/2024	1.0	Integração	inserirReservaNoBanco()	Testa se a inserção de uma reserva é executada corretamente no banco de dados.	ID Reserva: 101, cliente_id, ingresso_id, evento_id, quantidade, data_reserva	Mensagem "Reserva cadastrada com sucesso"	Grave	Não	"Cannot commit when autoCommit is enabled" e "Cannot rollback when autoCommit is enabled" o controle de transação na conexão de banco de dados está configurado para autoCommit precisa ser desabilitado ao início da operação e só reabilitado ao final do processo. Ajuste na função inserir em ReservasDAO	Sim

15	ReservasDAOTest	20/05/2024	1.0	Integração	removerReservaDoBanco()	Verificar a funcionalidade de remover uma reserva do banco de dados.	ID Reserva: 102	Mensagem "O evento foi removido com sucesso"	Médio	Não	Alteração nos métodos buscarPorCodigo e mapearParaEntidade para retornar null e não os valores zerados	Sim
16		20/05/2024	1.0	Integração	insercaoFalhaQuandoFkInvalida()	Verificar o comportamento da aplicação quando tenta inserir ou atualizar reservas com IDs inválidos para cliente, ingresso ou evento.	idReserva errada, cliente_id errada, ingresso_id errada, evento_id errada, quantidade, data_reserva	Mensagem "Evento cadastrado com sucesso"	Grave	Não	Exceção não está sendo capturada ou transformada em uma RuntimeException conforme esperado pelo teste. Alteração no método inserir dentro do catch lançando throw new RuntimeException	Sim
17	ReservasTest	20/05/2024	1.0	Unidade	naoValidarReservaComDataPassada()	Verifica se a data adicionada em uma reserva é anterior à data atual e impede a criação da reserva.	idReserva, cliente_id, ingresso_id, evento_id, quantidade, data_reserva errada	Mensagem "Falha na validação: A data do evento não é futura."	Grave	Sim	-	-
18		20/05/2024	1.0	Unidade	quantidadeDeveSerPositiva()	Verifica se a quantidade de ingressos em uma reserva é positiva e impede a criação com quantidades negativas.	Quantidade: -1	Mensagem "Quantidade de ingressos deve ser positiva"	Grave	Não	Faltava implementar validação	Sim
19	ConexaoDAOTest	20/05/2024	1.0	Unidade	testarSingleton()	Verificar se o padrão Singleton está sendo corretamente implementado, garantindo que apenas uma instância de ConexaoDAO seja criada.	-	Mensagem "INFO: Conexao com o banco de dados estabelecida."	Grave	Sim	-	-
20		20/05/2024	1.0	Integração	testarConexaoComBanco()	Testa se a conexão com o banco de dados é estabelecida corretamente.	-	Mensagem "INFO: Conexao com o banco de dados estabelecida."	Grave	Sim	-	-
21		20/05/2024	1.0	Integração	testarFechamentoDeConexao()	Verifica se a conexão com o banco de dados é corretamente fechada após o uso.	-	Mensagem "INFO: Conexao com banco de dados fechada."	Grave	Sim	-	-