

LUANA:

Teste: EquipamentoTest

Classe testada: Equipamento **Pacote:** br.unicentro.criadorpers

Este teste verifica se o construtor da classe Equipamento atribui corretamente os valores passados e se os métodos getId(), getNome(), getTipo(), getDescricao(), getBonusAtaque(), getBonusDefesa() e toString() retornam o resultado esperado.

Ele confirma a integridade dos dados dos equipamentos cadastrados no sistema, assegurando que o modelo funciona conforme planejado.

Teste: EquipamentoDAOTest

Classe testada: EquipamentoDAO **Pacote:** br.unicentro.criadorpers

Este teste chama o método EquipamentoDAO.listarEquipamentos() e verifica se ele retorna uma lista válida e não nula.

A asserção garante que o sistema está conseguindo acessar o banco de dados, executar a consulta SQL e criar corretamente os objetos Equipamento a partir dos registros armazenados.

Teste: RaceTest

Classe testada: Race **Pacote:** b.unicentro.criadorpers

O teste avalia se o construtor da classe Race cria objetos de forma correta e se os métodos getId(), getNome() e toString() retornam os valores apropriados.

Ele assegura que o modelo de dados referente à raça dos personagens está implementado de forma consistente.

Teste: ClasseTest

Classe testada: Classe **Pacote:** br.unicentro.criadorpers

Este teste verifica se o construtor da classe Classe cria o objeto com os dados corretos e se os métodos getId(), getNome() e toString() retornam os valores esperados.

O teste confirma a integridade da unidade que representa as classes dos personagens (como Mago, Guerreiro, etc.), garantindo que elas serão corretamente manipuladas em toda a aplicação.

GABRIELA:

Teste: DatabaseTest

Classe testada: Database

Pacote: br.unicentro.criadorpers

Resumo: Este teste verifica se o método responsável por estabelecer a conexão com o banco de dados retorna um objeto de conexão válido. Ele garante que as configurações do banco estão corretas e que o sistema consegue se comunicar com o PostgreSQL, sendo fundamental para todas as operações de CRUD.

Teste: RaceDAOTest

Classe testada: RaceDAO

Pacote: br.unicentro.criadorpers

Resumo: Este teste executa o método listarRacas() e verifica se ele retorna uma lista válida e não nula. Ele assegura que o sistema consegue acessar o banco de dados, executar a consulta SQL e construir corretamente os objetos Race com base nos registros armazenados. Além disso, valida que existe pelo menos uma raça cadastrada, garantindo o funcionamento da parte de seleção de raças no sistema.

Teste: CharacterDAOTest

Classe testada: CharacterDAO

Pacote: br.unicentro.criadorpers

Resumo: O teste avalia o método inserirPersonagem(), verificando se ele retorna um ID válido após a inserção de um novo personagem no banco. Ele confirma que o processo de criação de personagens está funcionando corretamente, desde a inserção no banco até o retorno do identificador do registro recém-criado.

Teste: CharacterTest

Classe testada: Character

Pacote: br.unicentro.criadorpers

Resumo: Este teste avalia se o construtor da classe Character atribui corretamente os valores passados e se os métodos getNome(), getForca(), getRaca(), getClasse() e demais getters retornam o resultado esperado.

Teste: HabilidadeDAOTest

Classe testada: HabilidadeDAO

O teste HabilidadeDAOTest verifica se o método listarHabilidades() retorna corretamente uma lista de habilidades do banco de dados. Ele assegura que a lista não seja nula e contenha pelo menos um registro, confirmando que a consulta SQL foi executada com sucesso. Dessa forma, garante-se que o sistema está acessando o banco e criando adequadamente os objetos Habilidade a partir dos dados armazenados.

Teste: HabilidadeTest

Classe testada: Habilidade

O teste HabilidadeTest verifica se o construtor e os métodos da classe Habilidade funcionam corretamente. Ele garante que os atributos id, nome e descricao são armazenados e retornados conforme esperado. Além disso, valida que o método toString() retorna o nome da habilidade, assegurando o comportamento adequado da classe de modelo.

Teste: ClassDAOTest

Classe testada: ClassDAO

O teste ClassDAOTest verifica se o método listarClasses() retorna uma lista válida e não nula. Ele confirma que a consulta SQL é executada corretamente e que há pelo menos uma classe cadastrada no banco de dados. Dessa forma, garante-se que o sistema consegue acessar o banco e criar adequadamente os objetos Classe a partir dos registros armazenados.

Teste: MainTest

Classe testada: Main

O teste MainTest verifica se o método main() da aplicação pode ser executado sem lançar exceções e confirma que a classe Main herda corretamente de javafx.application.Application.

Dessa forma, garante-se que o ponto de entrada da aplicação está configurado adequadamente e que o carregamento inicial da interface pode ocorrer sem erros.