

Relatório de Testes

Integrante: Hugo Andrade de Oliveira

Projeto: Sistema de Gerenciamento de Psicólogos e Pacientes

Pacote: `br.unicentro.trabalho1.dao`

Classe: `PsicologoDBDAOTest`

Método de Teste: `insere()`

- **Unidade testada:** Inserção de novo registro no banco de dados.
- **Explicação:**
O teste cria um objeto `Psicologo` e insere no banco através do método `dao.insere(p)`. Em seguida, busca o mesmo registro com `dao.buscaPorCodigo(0)` e verifica se os dados retornados (ID, nome, tempo de atuação e salário) correspondem ao esperado. Por fim, o registro é removido para manter o banco limpo.
- **Objetivo:** Garantir que o método de inserção grava corretamente os dados de um psicólogo no banco e que a busca posterior retorna exatamente o que foi inserido.

Método de Teste: `atualiza()`

- **Unidade testada:** Atualização de registro existente.
- **Explicação:**
O teste insere um psicólogo, altera o nome para "TesteAtualizado" e chama `dao.atualiza(p)`. Depois, busca novamente pelo ID e verifica se o nome foi realmente modificado, mantendo os demais atributos inalterados.
- **Objetivo:** Verificar se o método `atualiza` altera corretamente os dados de um psicólogo já existente no banco.

Método de Teste: remove()

- **Unidade testada:** Exclusão de registro no banco de dados.
- **Explicação:**
O teste insere um psicólogo e, em seguida, o remove usando `dao.remove(p)`. Depois, tenta buscá-lo novamente e espera que ocorra uma exceção (`SQLException`), confirmando que o registro foi realmente deletado.
- **Objetivo:** Assegurar que o método `remove` exclui permanentemente o registro do banco.

Método de Teste: buscaPorCodigo()

- **Unidade testada:** Recuperação de registro específico.
- **Explicação:**
Um psicólogo é inserido e depois recuperado usando o código de identificação. O teste valida todos os campos (ID, nome, tempo de atuação e salário) usando `assertAll`.
- **Objetivo:** Comprovar que o método `buscaPorCodigo` retorna corretamente os dados do psicólogo referente ao código informado.

Método de Teste: listaTodos()

- **Unidade testada:** Listagem de registros do banco.
- **Explicação:**
Após inserir um psicólogo, o teste chama `dao.listaTodos()` e verifica se a lista retornada não está vazia, garantindo que o método consegue acessar e listar os registros.
- **Objetivo:** Validar que o método `listaTodos` retorna corretamente todos os psicólogos armazenados.

Pacote: br.unicentro.trabalho1.dao

Classe: PacienteDBDAOTest

Método de Teste: insere()

- **Unidade testada:** Inserção de paciente no banco.
- **Explicação:**
O teste cria um Paciente e o insere com `dao.insere(p)`. Em seguida, busca o mesmo registro e verifica se os campos (ID, nome, CPF e ID do psicólogo) correspondem ao esperado.
- **Objetivo:** Confirmar que o método `insere` cadastra corretamente um paciente e que os dados permanecem consistentes na busca.

Método de Teste: atualiza()

- **Unidade testada:** Atualização de paciente.
- **Explicação:**
Insere um paciente, altera o nome e executa `dao.atualiza(p)`. Depois, busca novamente e verifica se a atualização foi efetivada, mantendo os outros campos inalterados.
- **Objetivo:** Verificar o funcionamento correto do método `atualiza` para registros de pacientes.

Método de Teste: remove()

- **Unidade testada:** Exclusão de paciente.
- **Explicação:**
Após inserir um paciente, o teste executa `dao.remove(p)` e tenta buscá-lo. Espera-se que uma exceção `SQLException` seja lançada, indicando que o paciente foi removido.
- **Objetivo:** Garantir que o método `remove` elimina corretamente o paciente do banco.

Método de Teste: buscaPorCodigo()

- **Unidade testada:** Consulta por código do paciente.
- **Explicação:**
O teste insere um paciente e realiza uma busca pelo mesmo código, conferindo se todos os atributos retornados correspondem ao esperado.
- **Objetivo:** Verificar se o método buscaPorCodigo está retornando o paciente correto conforme o código informado.

Método de Teste: listaTodos()

- **Unidade testada:** Listagem de pacientes.
- **Explicação:**
Após inserir um paciente, o teste chama `dao.listaTodos()` e confirma que a lista não está vazia.
- **Objetivo:** Assegurar que o método retorna corretamente a lista de pacientes cadastrados.

Pacote: br.unicentro.trabalho1.model

Classe: PsicologoTest

Método de Teste: testGetAndSetPsicologoID()

- **Unidade testada:** Métodos get e set do atributo psicologoID.
- **Explicação:**
Verifica se o ID inicial é zero e se a atualização via `setPsicologoID()` reflete corretamente na instância.
- **Objetivo:** Confirmar o funcionamento básico dos métodos de acesso ao ID.

Método de Teste: testGetAndSetName()

- **Unidade testada:** Atributo nome.
- **Explicação:**
Testa a leitura e modificação do nome de um psicólogo, garantindo consistência no valor armazenado.
- **Objetivo:** Garantir que o método setName atualiza corretamente o valor e que getName o retorna adequadamente.

Método de Teste: testGetAndSetCpf()

- **Unidade testada:** Atributo tempoAtuacao.
- **Explicação:**
Avalia se o tempo de atuação pode ser obtido e alterado corretamente.
- **Objetivo:** Assegurar que o atributo de tempo de atuação é manipulado sem erros.

Método de Teste: testGetAndSetSalario()

- **Unidade testada:** Atributo salario.
- **Explicação:**
Testa o valor inicial e a atualização do salário de um psicólogo.
- **Objetivo:** Confirmar o funcionamento correto dos métodos getSalario e setSalario.

Método de Teste: testToString()

- **Unidade testada:** Representação textual do objeto.
- **Explicação:**
Verifica se o método toString() retorna uma string formatada conforme o esperado, contendo todos os campos.

- **Objetivo:** Garantir a integridade da saída textual do objeto para exibição ou logs.

Pacote: `br.unicentro.trabalho1.model`

Classe: `PacienteTest`

Método de Teste: `testGetAndSetPacienteID()`

- **Unidade testada:** Atributo `pacienteID`.
- **Explicação:**
Confere se o ID inicial é 0 e se é possível alterá-lo corretamente.
- **Objetivo:** Assegurar o correto funcionamento dos métodos `get` e `set` para o ID.

Método de Teste: `testGetAndSetNome()`

- **Unidade testada:** Atributo `nome`.
- **Explicação:**
Valida o nome inicial e testa sua alteração para outro valor.
- **Objetivo:** Garantir a consistência da manipulação do campo `nome`.

Método de Teste: `testGetAndSetCpf()`

- **Unidade testada:** Atributo `cpf`.
- **Explicação:**
Testa o valor inicial e a atualização do CPF de um paciente.
- **Objetivo:** Confirmar o comportamento correto dos métodos `getCpf` e `setCpf`.

Método de Teste: testGetAndSetPsicologo_ID()

- **Unidade testada:** Relacionamento com o psicólogo.
- **Explicação:**
Verifica se o ID do psicólogo associado pode ser lido e modificado corretamente.
- **Objetivo:** Validar o controle de associação entre paciente e psicólogo.

Método de Teste: testToString()

- **Unidade testada:** Método toString().
- **Explicação:**
Verifica se a saída textual do paciente contém todos os dados esperados no formato definido.
- **Objetivo:** Garantir que a representação textual do objeto está completa e correta.