

TESTE UNITÁRIOS DAS FUNÇÕES DE ORÇAMENTOS COM PHP UNIT

OBJETIVO:

Este documento fornece uma visão detalhada dos testes das funcionalidades relacionadas ao Orçamentos implementados no projeto, incluindo os resultados esperados e obtidos para cada teste. O objetivo é garantir que as funcionalidades de manipulação de orçamentos (como cadastro, atualização, exclusão, etc.) estejam funcionando corretamente.

PRÉ-REQUISITOS:

Necessário ter o PHP 8.1.25 ou superior no PC; instalar o Composer para gerenciamento de dependências do PHP; e PHPUnit para execução dos testes.

INSTALAR AS DEPENDÊNCIAS DO PROJETO COM O COMPOSER (O PHPUnit será instalado também):

No diretório raiz do projeto, execute o seguinte comando no terminal (usei no VSCode): `composer install`

VERIFICAR A INSTALAÇÃO DO PHPUnit NO PROJETO:

Utilize o comando no terminal: `php vendor/bin/phpunit --version`

CONFIGURAR O BANCO DE DADOS:

Edite o arquivo `configBanco.php` e atualize as configurações do banco de dados conforme necessário.

ESTRUTURA DOS TESTES:

O arquivo `"testeOrcamento.php"` contém 5 métodos de testes para verificar a funcionalidade de manipulação de orçamentos. Cada teste foca em diferentes cenários, como cadastro, atualização de status e busca de orçamentos.

DESCRIÇÃO DOS MÉTODOS DE TESTES:

1 -> Teste de Cadastro de Orçamento Válido:

O método "testCadastroOrcamentoValido()" verifica se um orçamento válido pode ser cadastrado corretamente. O teste espera que o cadastro retorne true.

2 -> Teste de Busca de Todos os Orçamentos:

O método "testBuscarTodosOrcamentos()" verifica se todos os orçamentos podem ser recuperados corretamente. O teste espera que a função de busca retorne um array associativo contendo os orçamentos.

3 -> Teste de Atualização de Status de Orçamento:

O método "testAtualizarStatusOrcamento()" verifica se o status de um orçamento pode ser atualizado corretamente. O teste espera que a atualização retorne true.

4 -> Teste de Busca de Orçamentos por Cliente:

O método "testBuscarOrcamentosPorCliente()" verifica se os orçamentos de um cliente específico podem ser recuperados corretamente. O teste espera que a função de busca retorne um array associativo com os orçamentos.

5 -> Teste de Busca de Orçamento por Número:

O método "testBuscarOrcamentoPorNumero()" verifica se os detalhes de um orçamento específico podem ser recuperados corretamente pelo número do orçamento. O teste espera que a função de busca retorne um array associativo com os dados do orçamento.

COMANDO PARA RODAR OS TESTES:

php vendor/bin/phpunit PHP/tests/Orcamento/TesteOrcamento.php

RESULTADO DOS TESTES:

1 -> testCadastroOrcamentoValido()

Descrição: verifica se um orçamento válido é cadastrado corretamente.

Resultado Esperado: true.

Resultado Obtido: true.

2 -> testBuscarTodosOrcamentos()

Descrição: verifica se todos os orçamentos podem ser recuperados.

Resultado Esperado: Array de orçamentos.

Resultado Obtido: Array de orçamentos.

3 -> testAtualizarStatusOrcamento()

Descrição: verifica se o status de um orçamento é atualizado corretamente.

Resultado Esperado: true.

Resultado Obtido: true.

4 -> testBuscarOrcamentosPorCliente()

Descrição: verifica se os orçamentos de um cliente específico podem ser recuperados.

Resultado Esperado: Array de orçamentos.

Resultado Obtido: Array de orçamentos.

5 -> testBuscarOrcamentoPorNumero()

Descrição: verifica se os detalhes de um orçamento podem ser recuperados pelo número do orçamento.

Resultado Esperado: Array com dados do orçamento.

Resultado Obtido: Array com dados do orçamento.

CONCLUSÃO DOS TESTES:

Todos os testes foram executados com sucesso, confirmando que as funcionalidades de manipulação de orçamentos estão funcionando conforme o esperado. Não foram encontrados problemas durante a execução dos testes, e todos os resultados obtidos estavam de acordo com os resultados esperados.