

# Documentação do arquivo

## ChangePasswordControllerTest.cls

### Introdução

Este arquivo contém o código de teste para o controlador de página Apex que expõe a funcionalidade de alteração de senha. Ele verifica se o método `changePassword` funciona corretamente ao alterar a senha do usuário.

### Descrição

O arquivo `ChangePasswordControllerTest.cls` é uma classe de teste escrita em Apex para validar o comportamento do controlador de alteração de senha (`ChangePasswordController`). Ele simula o processo de alteração de senha, configurando os valores necessários e verificando se o método `changePassword` retorna o resultado esperado.

### Estrutura

A classe contém um único método de teste chamado `testChangePasswordController`, que realiza as seguintes ações:

1. Instancia o controlador `ChangePasswordController`.
2. Define os valores para as variáveis de senha antiga, nova senha e confirmação da nova senha.
3. Chama o método `changePassword` e verifica se o retorno é `null`.

### Dependências

A classe depende do seguinte:

- `ChangePasswordController`: O controlador que implementa a funcionalidade de alteração de senha.
- Framework de teste Apex (`@IsTest`): Utilizado para criar e executar testes unitários.

### Imports

Não há importações explícitas no código, pois o Apex não utiliza declarações de importação como outras linguagens.

### Variáveis

**Variáveis utilizadas no teste:**

- `controller.oldPassword`: Representa a senha antiga do usuário.
- `controller.newPassword`: Representa a nova senha que o usuário deseja configurar.
- `controller.verifyNewPassword`: Representa a confirmação da nova senha.

### Métodos

**`testChangePasswordController`**

Este é o único método na classe e realiza o teste unitário do controlador de alteração de senha. Ele:

1. Instancia o controlador `ChangePasswordController`.
2. Define os valores das variáveis de senha.
3. Chama o método `changePassword` e verifica se o retorno é `null`.

**Parâmetros:**

Nenhum.

## Retorno:

Nenhum retorno explícito, pois é um método de teste.

## Comportamento:

- Simula o processo de alteração de senha.
- Verifica se o método `changePassword` retorna `null`, indicando que a operação foi bem-sucedida.

## Exemplo

Abaixo está um exemplo de como o teste é executado:

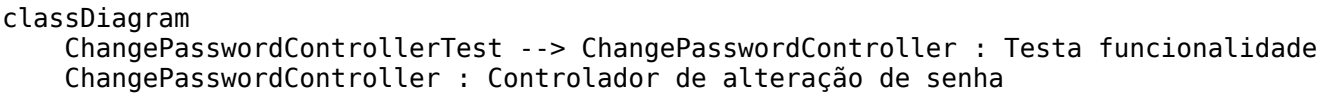
```
@IsTest public with sharing class ChangePasswordControllerTest {
    @IsTest(SeeAllData=true) public static void testChangePasswordController() {
        // Instancia o controlador
        ChangePasswordController controller = new ChangePasswordController();

        // Define os valores das senhas
        controller.oldPassword = '123456';
        controller.newPassword = 'qwerty1';
        controller.verifyNewPassword = 'qwerty1';

        // Verifica se o método retorna null
        System.assertEquals(controller.changePassword(), null);
    }
}
```

## Diagrama de Dependência

Abaixo está um diagrama de dependência que ilustra a relação entre a classe de teste e o controlador:



## Notas

- O teste utiliza a anotação `@IsTest(SeeAllData=true)`, o que permite acesso a todos os dados na organização. Isso pode ser útil em alguns casos, mas deve ser usado com cuidado para evitar dependências de dados reais.
- O método `changePassword` não possui validações explícitas no teste. Certifique-se de que o controlador implementa as verificações necessárias para garantir a segurança e integridade dos dados.

## Vulnerabilidades

- **Uso de `SeeAllData=true`:** Esta configuração pode introduzir dependências de dados reais na organização, o que pode levar a resultados inconsistentes ou falhas no teste. É recomendável evitar o uso de `SeeAllData=true` sempre que possível.
- **Validação de senha:** O teste não verifica se o método `changePassword` realiza validações adequadas, como verificar a força da senha ou garantir que a nova senha seja diferente da antiga. Isso pode ser uma vulnerabilidade se não for tratado no controlador.