

# Documentação do arquivo SiteRegisterControllerTest.cls

## Introdução

Este arquivo contém a classe SiteRegisterControllerTest, que é responsável por realizar testes unitários para a classe SiteRegisterController. O objetivo principal é garantir que os métodos da classe SiteRegisterController funcionem conforme o esperado, validando cenários específicos relacionados ao registro de usuários.

## Descrição

A classe SiteRegisterControllerTest é uma classe de teste escrita em Apex, usada para verificar o comportamento da classe SiteRegisterController. Ela utiliza a anotação @IsTest para indicar que é uma classe de teste e contém métodos que simulam cenários de registro de usuários. O método principal, testRegistration, testa o comportamento do método registerUser em diferentes condições, como a ausência de senha ou a incompatibilidade entre senha e confirmação de senha.

## Estrutura

A estrutura do arquivo é composta por:

- Uma classe de teste chamada SiteRegisterControllerTest.
- Um método de teste estático chamado testRegistration.

## Dependências

A classe depende da seguinte classe:

- SiteRegisterController: A classe que está sendo testada.

## Imports

Não há imports explícitos no código, pois o Apex não utiliza a mesma estrutura de imports que outras linguagens como Java. No entanto, a classe depende implicitamente da classe SiteRegisterController.

## Variáveis

As variáveis utilizadas no código são:

- controller: Uma instância da classe SiteRegisterController usada para acessar e testar seus métodos.
- username: Variável da classe SiteRegisterController que armazena o nome de usuário.
- email: Variável da classe SiteRegisterController que armazena o e-mail do usuário.
- communityNickname: Variável da classe SiteRegisterController que armazena o apelido da comunidade.
- password: Variável da classe SiteRegisterController que armazena a senha do usuário.
- confirmPassword: Variável da classe SiteRegisterController que armazena a confirmação da senha.

## Métodos

### testRegistration

Este é o único método de teste presente na classe. Ele realiza os seguintes passos:

1. Cria uma instância da classe SiteRegisterController.
2. Define valores para as variáveis username, email e communityNickname.
3. Chama o método registerUser e verifica se ele retorna null quando a página não é acessada como um usuário convidado.
4. Define valores para as variáveis password e confirmPassword com senhas incompatíveis.

5. Chama novamente o método `registerUser` e verifica se ele retorna `null` devido à incompatibilidade das senhas.

O método utiliza a anotação `@IsTest(SeeAllData=true)` para permitir o acesso a dados reais durante o teste, embora isso não seja uma prática recomendada em muitos casos.

## Exemplo

Abaixo está um exemplo de como a classe `SiteRegisterControllerTest` é usada para testar o comportamento do método `registerUser`:

```
@IsTest public with sharing class SiteRegisterControllerTest {
    @IsTest(SeeAllData=true) static void testRegistration() {
        SiteRegisterController controller = new SiteRegisterController();
        controller.username = 'test@force.com';
        controller.email = 'test@force.com';
        controller.communityNickname = 'test';
        System.assert(controller.registerUser() == null);

        controller.password = 'abcd1234';
        controller.confirmPassword = 'abcd123';
        System.assert(controller.registerUser() == null);
    }
}
```

## Diagrama de Dependência

O diagrama abaixo ilustra a dependência entre a classe `SiteRegisterControllerTest` e a classe `SiteRegisterController`:

```
classDiagram
    class SiteRegisterControllerTest {
        +testRegistration() : void
    }
    class SiteRegisterController {
        +username : String
        +email : String
        +communityNickname : String
        +password : String
        +confirmPassword : String
        +registerUser() : Object
    }
    SiteRegisterControllerTest --> SiteRegisterController
```

## Notas

- A anotação `@IsTest(SeeAllData=true)` permite o acesso a dados reais, mas isso pode introduzir dependências indesejadas nos testes. É recomendável usar dados simulados sempre que possível.
- O método `registerUser` retorna `null` quando a página não é acessada como um usuário convidado. Isso deve ser levado em consideração ao escrever testes adicionais.

## Vulnerabilidades

- O uso de `SeeAllData=true` pode expor dados reais durante os testes, o que não é uma prática recomendada. Isso pode levar a falhas nos testes se os dados reais forem alterados.
- O teste não cobre cenários em que o método `registerUser` é chamado com dados válidos e a página é acessada como um usuário convidado. Isso pode ser uma lacuna na cobertura de testes.