

# Documentação do arquivo SiteRegisterController.cls

## Introdução

O arquivo `SiteRegisterController.cls` é uma classe Apex que tem como objetivo criar usuários de portal em uma instância Salesforce. Ele permite que os usuários sejam registrados e, caso necessário, sejam autenticados automaticamente após o registro. A classe também valida senhas e gerencia a criação de usuários no contexto de um portal.

## Descrição

A classe `SiteRegisterController` é usada para registrar usuários no portal Salesforce. Ela contém métodos para validar senhas, criar usuários e redirecionar para páginas específicas após o registro. A classe utiliza a funcionalidade de criação de usuários de portal fornecida pelo Salesforce, além de realizar verificações básicas, como a correspondência entre senha e confirmação de senha.

## Estrutura

A estrutura da classe é composta por:

- Variáveis privadas e públicas para armazenar informações do usuário.
- Um método privado para validação de senha.
- Um método público para registrar o usuário no portal.

## Dependências

A classe depende das seguintes funcionalidades e objetos do Salesforce:

- `Site.createPortalUser`: Método para criar usuários de portal.
- `Site.login`: Método para autenticar usuários no portal.
- `ApexPages.Message`: Para exibir mensagens de erro.
- `System.Page`: Para redirecionar para páginas específicas.

## Imports

Não há importações explícitas no código, pois o Apex utiliza namespaces internos do Salesforce.

## Variáveis

### Variáveis privadas

- `PORTAL_ACCOUNT_ID`: ID da conta na qual o contato será criado e habilitado como usuário do portal. É uma constante estática.

### Variáveis públicas

- `username`: Nome de usuário do portal.
- `email`: Endereço de e-mail do usuário.
- `password`: Senha do usuário. É automaticamente ajustada para remover espaços em branco.
- `confirmPassword`: Confirmação da senha. Também ajustada para remover espaços em branco.
- `communityNickname`: Apelido do usuário na comunidade. Ajustado para remover espaços em branco.

## Métodos

### Construtor

- `SiteRegisterController()`: Construtor padrão da classe. Não realiza nenhuma operação específica.

## Métodos privados

- `isValidPassword()`: Verifica se a senha e a confirmação de senha correspondem. Retorna `true` se forem iguais, caso contrário, retorna `false`.

## Métodos públicos

- `registerUser()`: Registra um novo usuário no portal. Este método:
  - Valida se a senha e a confirmação de senha correspondem.
  - Cria um objeto `User` com as informações fornecidas.
  - Utiliza o método `Site.createPortalUser` para criar o usuário no portal.
  - Redireciona o usuário para a página de confirmação ou realiza o login automático, dependendo da situação.

## Exemplo

Aqui está um exemplo de como usar a classe `SiteRegisterController`:

```
SiteRegisterController controller = new SiteRegisterController();
controller.username = 'usuario_teste';
controller.email = 'usuario@teste.com';
controller.password = 'senha123';
controller.confirmPassword = 'senha123';
controller.communityNickname = 'UsuarioTeste';
```

```
PageReference result = controller.registerUser();
if (result != null) {
    System.debug('Usuário registrado com sucesso!');
} else {
    System.debug('Erro ao registrar o usuário.');
```

## Diagrama de Dependência

O diagrama abaixo ilustra as dependências entre os métodos e variáveis da classe:

```
classDiagram
    class SiteRegisterController {
        - static Id PORTAL_ACCOUNT_ID
        + String username
        + String email
        + String password
        + String confirmPassword
        + String communityNickname
        - boolean isValidPassword()
        + PageReference registerUser()
    }
    SiteRegisterController --> Site : createPortalUser
    SiteRegisterController --> Site : login
    SiteRegisterController --> ApexPages : Message
    SiteRegisterController --> System : Page
```

## Notas

- Certifique-se de que o ID da conta (`PORTAL_ACCOUNT_ID`) seja válido e que o proprietário da conta esteja na hierarquia de funções, conforme as instruções de configuração do portal do cliente.
- O campo `lastName` é obrigatório para o objeto `User`. Caso não seja especificado, ele será definido como o nome de usuário.

## Vulnerabilidades

- **Validação de senha:** A validação de senha é básica e não verifica critérios de segurança, como comprimento mínimo ou complexidade.
- **Hardcoded Account ID:** O ID da conta está codificado diretamente no código, o que pode causar problemas de manutenção e segurança. Recomenda-se armazená-lo em um campo de configuração ou variável personalizada.
- **Mensagens de erro:** As mensagens de erro dependem de rótulos personalizados (`Label.site.passwords_dont_match`). Certifique-se de que esses rótulos estejam configurados corretamente no Salesforce.

Esta documentação detalhada deve ajudar qualquer desenvolvedor a entender e utilizar a classe `SiteRegisterController` de forma eficaz.