

# Documentação do arquivo ContactSearch.cls

## Introdução

O arquivo `ContactSearch.cls` contém uma classe escrita em Apex (linguagem utilizada no Salesforce) que realiza a busca de contatos no banco de dados com base no sobrenome e no código postal fornecidos como parâmetros. Essa classe é útil para filtrar contatos de forma eficiente e retornar uma lista de resultados que atendam aos critérios especificados.

## Descrição

A classe `ContactSearch` possui um método estático chamado `searchForContacts`, que executa uma consulta SOQL (Salesforce Object Query Language) para buscar contatos no banco de dados. O método aceita dois parâmetros: o sobrenome (`aLastName`) e o código postal (`aPostalCode`). Ele retorna uma lista de objetos do tipo `Contact` que correspondem aos critérios fornecidos.

## Estrutura

A estrutura do arquivo é simples e consiste em:

- Uma classe pública chamada `ContactSearch`.
- Um método estático chamado `searchForContacts`.

## Dependências

A classe depende dos seguintes elementos:

- Objeto padrão `Contact` do Salesforce.
- Linguagem SOQL para realizar consultas no banco de dados.

## Imports

Não há importações explícitas no código, pois o Salesforce Apex já fornece acesso ao objeto `Contact` e à linguagem SOQL.

## Variáveis

### Variáveis Locais

- `ctSelected`: Lista de objetos do tipo `Contact` que armazena os resultados da consulta SOQL.

### Parâmetros do Método

- `aLastName`: String que representa o sobrenome do contato a ser buscado.
- `aPostalCode`: String que representa o código postal do contato a ser buscado.

## Métodos

### `searchForContacts`

#### Descrição

Este método realiza uma busca no banco de dados Salesforce para encontrar contatos que correspondam ao sobrenome e ao código postal fornecidos.

## Parâmetros

- `aLastName` (String): Sobrenome do contato.
- `aPostalCode` (String): Código postal do contato.

## Retorno

- `List<Contact>`: Retorna uma lista de objetos do tipo `Contact` que atendem aos critérios de busca.

## Funcionamento

1. O método recebe dois parâmetros: `aLastName` e `aPostalCode`.
2. Executa uma consulta SOQL para buscar contatos cujo sobrenome seja igual ao valor de `aLastName` e cujo código postal seja igual ao valor de `aPostalCode`.
3. Retorna a lista de contatos encontrados.

## Exemplo

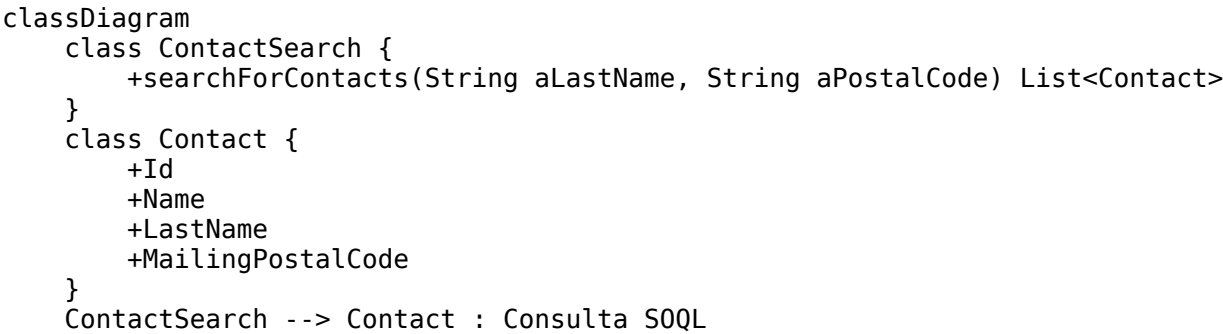
Aqui está um exemplo de como usar a classe `ContactSearch`:

```
String lastName = 'Silva';
String postalCode = '12345';
List<Contact> contacts = ContactSearch.searchForContacts(lastName, postalCode);

for (Contact contact : contacts) {
    System.debug('Contato encontrado: ' + contact.Name);
}
```

## Diagrama de Dependência

O diagrama abaixo ilustra a relação entre a classe `ContactSearch` e o objeto `Contact`:



## Notas

- Certifique-se de que os parâmetros fornecidos ao método `searchForContacts` sejam válidos e correspondam aos dados existentes no banco de dados.
- O método utiliza consultas SOQL, que podem estar sujeitas a limites de governança do Salesforce. É importante considerar o volume de dados ao utilizar este método.

## Vulnerabilidades

- **Injeção SOQL:** Embora o código utilize variáveis `bind` (`:aLastName` e `:aPostalCode`), é importante garantir que os valores passados para esses parâmetros sejam validados para evitar possíveis ataques de injeção SOQL.
- **Limites de Governança:** Caso o número de contatos retornados seja muito grande, pode haver problemas relacionados aos limites de governança do Salesforce. Considere paginar os resultados ou limitar a consulta, se necessário.