

Documento de implementação: CRUD de chaves em Python com Flask

1. Buscar uma chave pelo nome

Para implementar a funcionalidade de busca de uma chave pelo nome, precisamos realizar as seguintes etapas:

1. Criar uma nova rota no arquivo `app.py` para o endpoint `/chaves/<nome>`.
2. Implementar uma função para essa rota que retorne a chave correspondente ao nome fornecido.
3. Atualizar o layout do frontend para incluir um campo de texto para o nome da chave e um botão de busca.

Implementação da rota

Aqui está um exemplo de implementação da rota para a busca de chaves:

```
@app.route('/chaves/<nome>', methods=['GET'])
def buscar_chave(nome):
    # Busca a chave no banco de dados
    chave = Chaves.query.filter_by(nome=nome).first()
    # Retorna a chave
    if chave is not None:
        return chave.to_dict()
    else:
        return None
```

Essa função usa o método `query.filter_by()` do SQLAlchemy para buscar a chave no banco de dados com o nome fornecido. Se a chave for encontrada, a função retorna seu dicionário de dados. Caso contrário, a função retorna `None`.

Implementação da função de busca

Aqui está um exemplo de implementação da função de busca:

```
def buscar_chave(nome):
    ***# Busca a chave no banco de dados***
    chave = Chaves.query.filter_by(nome=nome).first()
    ***# Exibe a chave no frontend***
    if chave is not None:
        ***# Exibir os dados da chave***
```

```
return f'<h1>{chave.nome}</h1>' \ f'<h2>{chave.valor}</h2>'
else:
    ***# Exibir mensagem de erro***
    return 'Chave não encontrada'
```

Essa função primeiro usa a função `buscar_chave()` implementada anteriormente para buscar a chave no banco de dados. Se a chave for encontrada, a função exibe seus dados no frontend. Caso contrário, a função exibe uma mensagem de erro.

Atualização do layout

Para adicionar um campo de texto para o nome da chave e um botão de busca ao layout do frontend, podemos usar o seguinte código:

```
<input type="text" id="nome" placeholder="Nome da chave"> <button id="busc
```

Esse código adiciona um campo de texto com o id `nome` e um botão com o id `buscar` ao layout.

Quando o usuário digita um nome no campo de texto e clica no botão, o evento `click` do botão é disparado. Esse evento pode ser usado para chamar a função `buscar_chave()` com o nome fornecido pelo usuário.

2. Alterar uma chave

Para implementar a funcionalidade de alteração de uma chave, precisamos realizar as seguintes etapas:

1. Criar uma nova rota no arquivo `app.py` para o endpoint `/chaves/<nome>/editar`.
2. Implementar uma função para essa rota que permita ao usuário alterar os dados da chave.
3. Atualizar o layout do frontend para incluir um formulário para editar os dados da chave.

Implementação da rota

Aqui está um exemplo de implementação da rota para a alteração de chaves:

```
@app.route('/chaves/<nome>/editar', methods=['GET', 'POST'])
def editar_chave(nome):
    # Busca a chave no banco de dados
```

```
    chave = Chaves.query.filter_by(nome=nome).first()
# Obtém os dados da chave do formulário
if request.method == 'POST':
    chave.nome = request.form['nome']
    chave.valor = request.form['valor']
# Salva as alterações no banco de dados
db.session.commit()
# Retorna os dados da chave
return chave.to_dict()
```

Essa função usa o método `query.filter_by()` do SQLAlchemy para buscar a chave no banco de dados com o nome fornecido. Se a chave for encontrada, a função obtém os dados da chave do formulário. Caso contrário, a função retorna `None`