

**08**

# **DESENVOLVIMENTO DE APLICAÇÃO**

## **PARTE 2**



# Criando um CRUD em Laravel

03

## Criando a Rota para Editar e Deletar

- Primeiro, no arquivo **routes/web.php**, vamos definir as rotas para editar e atualizar o usuário.

```
// Novas rotas para editar e atualizar:  
Route::get('/usuarios/{id}/edit', [UsuariosController::class, 'edit'])->name('usuarios.edit');  
Route::put('/usuarios/{id}', [UsuariosController::class, 'update'])->name('usuarios.update');
```

- A rota usuarios.edit é usada para carregar o formulário de edição.
- A rota usuarios.update é usada para atualizar os dados do usuário no banco de dados.



# Criando um CRUD em Laravel

03

## Criando a Rota para Editar e Deletar

- E no seu **web.php** (rotas), adiciona a rota de destruir:

```
// Rota para excluir usuário  
Route::delete('/usuarios/{id}', [UsuariosController::class, 'destroy'])->name('usuarios.destroy');
```

- Vamos repetir a definição da rota de delete.



# Criando um CRUD em Laravel

03

## Criando os Métodos no Controller

- Agora, vamos para os métodos do Controller. No seu **UsuariosController.php**, você precisa criar os métodos edit, update e destroy.

Método **edit**: Este método será responsável por carregar o formulário de edição com os dados do usuário selecionado.

```
// --- Adicionando função de editar ---  
public function edit($id)  
{  
    $usuario = Usuarios::findOrFail($id);  
    return view('Usuarios.edit', compact('usuario'));  
}
```

Usamos **findOrFail** para buscar o usuário pelo id.



# Criando um CRUD em Laravel

03

## Criando os Métodos no Controller

- Método **update**: Este método será responsável por atualizar os dados do usuário no banco de dados após o envio do formulário.

```
// --- Adicionando função de atualizar ---  
public function update(Request $request, $id)  
{  
    $usuario = Usuarios::findOrFail($id);  
  
    $usuario->nome = $request->input('nome');  
    $usuario->email = $request->input('email');  
  
    if ($request->filled('senha')) {  
        $usuario->senha = bcrypt($request->input('senha'));  
    }  
  
    $usuario->save();  
  
    return redirect()->route('usuarios.read')->with('success', 'Usuário atualizado com sucesso!');  
}
```



# Criando um CRUD em Laravel

03

## Criando os Métodos no Controller

- Método **destroy**: Este método será responsável por deletar o usuário selecionado.

```
public function destroy($id)
{
    $usuario = Usuarios::findOrFail($id);
    $usuario->delete();

    return redirect()->route('usuarios.read')->with('success', 'Usuário excluído com sucesso!');
}
```

O método destroy encontra o usuário pelo id e, se encontrado, o deleta com \$usuario->delete(). Após a exclusão, o usuário é redirecionado de volta para a lista com uma mensagem de sucesso.



# Criando um CRUD em Laravel

03

## Criando as Views de Editar e Confirmar Exclusão



```
<h1>Editar Usuário</h1>

@if ($errors->any())
    <div>
        @foreach ($errors->all() as $erro)
            <p>{{ $erro }}</p>
        @endforeach
    </div>
@endif

<form action="{{ route('usuarios.update', $usuario->id) }}" method="POST">
    @csrf
    @method('PUT') <!-- Importantíssimo para o PUT -->

    <label for="nome">Nome:</label>
    <input type="text" name="nome" value="{{ $usuario->nome }}" required><br>

    <label for="email">Email:</label>
    <input type="email" name="email" value="{{ $usuario->email }}" required><br>

    <label for="senha">Senha (deixe em branco para manter):</label>
    <input type="password" name="senha"><br>

    <button type="submit">Atualizar</button>
</form>
```

### View **edit.blade.php**

Essa será a tela onde o usuário pode editar seus dados. Ela pode ficar parecida com a tela de criação, mas com os campos já preenchidos. Nessa tela, o formulário já é preenchido com os dados do usuário ( $\$usuario->nome$ ,  $\$usuario->email$ ).



# Criando um CRUD em Laravel

03

## Criando as Views de Editar e Confirmar Exclusão

- **Atualizando o botão Editar:** Para chamar a rota de edição.

O que falta agora é colocar os links corretos para Editar e Excluir, certo? Vou te mostrar como corrigir esses dois `<a href="#">` que você deixou em branco:

```
<!-- Ações: editar e excluir -->  
<a href="{{ route('usuarios.edit', $usuario->id) }}" class="btn btn-info btn-sm">Editar</a>
```





# Criando um CRUD em Laravel

03

## Criando as Views de Editar e Confirmar Exclusão

### Confirmar Exclusão

Você pode implementar um sistema simples de confirmar exclusão via Modal ou um simples alert de JavaScript. Aqui, vou te mostrar o básico: No seu **read.blade.php**, no botão de Excluir, modifique o formulário para que ele peça uma confirmação ao usuário:

```
<form action="{{ route('usuarios.destroy', $usuario->id) }}" method="POST" style="display:inline-block;">
    @csrf
    @method('DELETE')
    <button type="submit" class="btn btn-danger btn-sm"
        onclick="return confirm('Tem certeza que deseja excluir este usuário?')">Excluir</button>
</form>
```



# Criando um CRUD em Laravel

03

## Criando as Views de Editar e Confirmar Exclusão

### Confirmar Exclusão

Esse código utiliza um `onclick="return confirm()"` para exibir uma janela de confirmação. Se o usuário confirmar, o formulário será enviado e o usuário será excluído.

Quando você usa a diretiva `@csrf`, o Laravel automaticamente adiciona um token único ao seu formulário. Esse token é verificado a cada requisição para garantir que a requisição foi realmente feita pelo usuário e não por um atacante.

A diretiva `@method('DELETE')` no Laravel permite que você "simule" métodos DELETE ou PUT em um formulário, já que o HTML só suporta GET e POST. Então, mesmo que você esteja enviando o formulário via POST, o Laravel irá "interpretar" o formulário como um DELETE graças ao `@method('DELETE')`.



# Criando um CRUD em Laravel

03

## Testando tudo

### Cadastrar um novo usuário

Nome

E-mail

Senha

### Lista de Usuários

ID	Nome	Email	Ações
3	Marcia Silva	marciaprof2023@gmail.com	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
4	Laura Silva	laura@gmail.com	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
5	Dioneide Sales	dioneide@gmail.com	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
6	Grazi	grazi@gmail.com	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
8	Laura Silva	laura@gmail.com	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
9	Laura Silva	laura123@gmail.com	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
11	Grazi	marciaprof2023@gmail.com	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
33	zulu	zulu@gmail.com	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

### Editar Usuário

Nome:

Email:

Senha (deixe em branco para manter):

127.0.0.1:8000 diz

Tem certeza que deseja excluir este usuário?

OK

Cancelar

**Até a  
Próxima  
Aula!**

