

# CRUD - Operações Update e Delete

Usabilidade, desenvolvimento web, mobile e jogos

Prof. Me. Gustavo Torres Custódio  
gustavo.custodio@ulife.com.br

# Conteúdo

**CRUD - UPDATE**

**CRUD - DELETE**

# Introdução

- Vamos continuar desenvolvendo nosso *backend* com operações CRUD.
  - Baixe os arquivos com o código e com o banco de dados no site da disciplina.

# Introdução

- Inicie a aplicação Flask:
  - `set FLASK_ENV=development`
  - `set FLASK_APP=crud.py`
  - `flask run`

# Introdução

- Acesse a rota `/listarfuncionarios`
- Serão listados todos os funcionários disponíveis.

# Introdução

Id: 1

Nome: Gustavo

---

Id: 2

Nome: Michael

---

Id: 3

Nome: Gabriela

---

Id: 4

Nome: Alfredo

---



CRUD ~ Operações Update e  
Delete

## CRUD - UPDATE

# Update

- A operação de *update* (atualização) modifica uma ou mais linhas de uma tabela.
  - O registro já precisa existir na tabela.
  - A **cláusula WHERE** determina quais linha serão afetadas.



# Update

- Forma geral:

```
UPDATE <Tabela>  
SET <atributo1>='<valor1>'  
    <atributo2>='<valor2>'  
    ...  
    <atributoN>='<valorN>'  
WHERE <condição>;
```

## Update

- Vamos criar um formulário com 4 campos:
  - id;
  - nome;
  - idade;
  - endereço.
- O id será utilizado para **determinar qual linha será atualizada.**
- Os demais campos serão atualizados.

## Update

- Crie um formulário chamado formulario\_atualiza.html (dentro da pasta templates).

```
<h2>Atualizar</h2>
<form action="/atualizarfuncionario" method="POST">
  <input type="number" name="id" placeholder="id">
    <br> <br>
  <input type="text" name="nome" placeholder="Nome">
    <br> <br>
  <input type="number" name="idade" placeholder="Idade">
    <br> <br>
  <input type="text" name="endereco" placeholder="Endereço">
    <br> <br>
  <input type="submit" value="Atualizar">
</form>
```

# Update

- Criamos um *endpoint* para abrir o formulário de atualização

```
1  @app.route('/update')
2  def formulario_atualiza():
3      return render_template('formulario_atualiza.html')
```

# Update

- E um *endpoint* que processa os dados enviados pelo formulário.

```
1  @app.route('/atualizarfuncionario', methods = ['POST'],)
2  def atualizar_funcionario():
3      banco = carregar_banco()
4      cur = banco.cursor()
5
6      id = request.form["id"]
7      nome = request.form["nome"]
8      idade = request.form["idade"]
9      endereco = request.form["endereco"]
10
11     cur.execute("""
12         UPDATE Funcionario
13         SET nome=?, idade=?, endereco=?
14         WHERE id = ?
15         """, (nome, idade, endereco, id)
16     ).fetchall()
17     banco.commit()
18     banco.close()
19     return '<h1>Funcionário Atualizado com Sucesso.</h1>'
```

## Update

- No final, será buscado um funcionário por id e suas informações (exceto o id) serão atualizadas.
  - De acordo com os valores dos campos no formulário.
- Listamos os funcionários novamente para ver o resultado.



CRUD - Operações Update e  
Delete

## CRUD - DELETE

# Delete

- A operação de *delete* (remover) remove uma ou mais linhas de uma tabela.
  - Assim como o *update*, a **cláusula WHERE** determina quais linha serão afetadas.
    - Sem ela, todas as linhas da tabela são removidas.



# Delete

- Forma geral:

```
DELETE  
FROM <Tabela>  
WHERE <condição>;
```

## Delete

- Crie um formulário chamado formulario\_deleta.html (dentro da pasta templates).
  - Ele possui apenas o campo id.
  - O funcionário será buscado pelo id e sua linha na tabela será removida.

```
<h2>Remover</h2>
<form action="/deletarfuncionario" method="POST">
  <input type="number" name="id" id="id" placeholder="id">
  <input type="submit" value="Remover Funcionário">
</form>
```

# Delete

- Criamos um *endpoint* para abrir o formulário.

```
1  @app.route('/delete')
2  def formulario_deleta():
3      return render_template('formulario_deleta.html')
```

# Delete

- E um *endpoint* que recebe o id
  - e aplica a operação SQL de remover o funcionário com o id correspondente.

```
1  @app.route('/deletarfuncionario', methods = ['POST',])
2  def deletar_funcionario():
3      banco = carregar_banco()
4      cur = banco.cursor()
5
6      id = request.form["id"]
7      cur.execute("""
8          DELETE FROM Funcionario
9          WHERE id = ?""", (id)
10     ).fetchall()
11     banco.commit()
12     banco.close()
13     return '<h1>Funcionário Removido com Sucesso.</h1>'
```

## Delete

- No final, será buscado um funcionário por id e ele será removido da tabela.
- Listamos os funcionários novamente para ver o resultado.

## Exercício

- Adicione uma tela inicial com 4 *links*.
- Cada link direciona a página para um dos formulários:
  - Busca;
  - Inserir;
  - Atualizar;
  - Deletar.

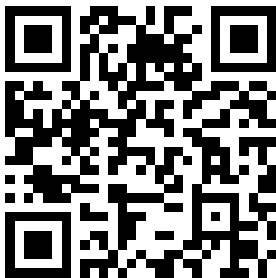
## Referências



Grinberg, M. (2018).

*Flask web development: developing web applications with python.*  
"O'Reilly Media, Inc.".

## Conteúdo



<https://gustavotcustodio.github.io/usabilidade.html>



Obrigado

[gustavo.custodio@ulife.com.br](mailto:gustavo.custodio@ulife.com.br)