



# Programação de Sistemas para Internet

Prof. Romerito Campos

# Plano de Aula

- Objetivo:
  - Compreender o funcionamento de formulários, métodos GET e POST e strings de consulta
- Conteúdo:
  - Métodos POST e GET
  - Formulários HTML
  - Strings de Consulta

# Metodos GET e POST

- Métodos **GET** tem por objetivo permitir que uma representação de recurso especificado seja recuperado
- Método **POST** tem por objetivo enviar dados para o servidor.
  - Este método tem detalhes adicionais em relação ao método GET
  - Uma forma comum de enviar requisições POST é via **Forms HTML**
  - Também é possível realizar via javascript

# Método POST

# Método POST

- Request são geralmente enviadas via formulários HTML
  - `<form> ... </form>`
- Quando precisamos realizar algum tipo de alteração no servidor, utilizamos POST.
  - Salvar algum dado, por exemplo
  - Login/logout de usuários também são exemplos (*mesmo ser alterar dados no servidor*)

# Metodo POST

- Podemos indicar o tipo de conteúdo a ser enviado
  - **application/x-www-form-urlencoded**
  - **multipart/form-data**
- Estas opções podem ser passadas via `enctype` no html

```
<form enctype="multipart-form-data">  
  <input type="file">  
</form>
```

# Método POST

- **application/x-www-form-urlencoded (PADRÃO)**: codifica a informação entre pares de valores separados por &
- Dado um formulário com três campos: nome, cidade e idade. A informação é codificada como abaixo:

```
Nome=rafael&cidade=campinas&idade=3
```

# Método POST

- **multipart/form-data:**
  - mais flexível [1](#), [2](#).
  - Pode ser utilizado para envio de dados binários como imagens além de texto.
  - Acrescenta overhead no processamento no servidor (pouco indicado para dados textuais)

# String de Consulta

# String de Consulta

- Strings de consulta estão fortemente relacionadas ao método GET
- É possível ver sua composição na URL através da barra de navegação
- Podemos observar também na construção de links

```
http://localhost:5000/profile?name=joao&sname=jj
```

# String de consulta

- Úteis para criação de filtros
- Envio de dados ao servidor (dados não-sensíveis)

```
http://localhost:5000/profile?name=joao&sname=jj
```

- String de consulta
  - Utiliza-se ? seguido de pares **key=value** separados por **&**

```
/profile?name=joao&sname=jj
```

# Prática

- Criar uma aplicação utilizando flask que permite enviar dados via POST(sem armazenamento ainda)
- Esta aplicação também tem uma rota onde podemos filtrar dados com base em requisições GET e String de Consulta

# Pratica

```
python -m venv env # criando ambiente virtual
.\env\Scripts\activate.bat # ativando ambiente

pip install flask # Instalação do flask

# Rodar uma aplicação flask em modo DEBUG
# main refere-se a um arquivo main.py
flask --app main run --debug
```

# Atividade

- Criar uma aplicação que utilize os recursos abordados em sala de aula
- A atividade pode ser feita em dupla
- A dupla fica livre para definir um tema

# Referências

CODINGSCENES. What is the difference between application/x-www-form-urlencoded and multipart/form-data OR text/plain? Disponível em:

<https://medium.com/@codingscenes/application-x-www-form-urlencoded-and-multipart-form-data-are-two-different-formats-for-3678a10073e9>. Acesso em: 6 mar. 2024.

MIME types. Disponível em: [https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Basics\\_of\\_HTTP/MIME\\_types](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Basics_of_HTTP/MIME_types). Acesso em: 5 mar. 2024.

POST. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods/POST>. Acesso em: 5 mar. 2024.