

**Aula 05 – Introdução ao FLASK**

|  |  |
| --- | --- |
| Flask depende/utiliza:   * WSGI, * Werkzeug, * Jinja2 | Uma imagem contendo objeto  Descrição gerada automaticamente |
| Exemplo de código básico |  |
|  |  |
| Organização básica de diretórios |  |
| Instalação básica   1. (ambientes virtuais) 2. pip install virtualenv 3. venv\scripts\activate 4. pip install flask | **pip install Flask** |
| Código básico:   1. Salvar, rodar, rodar internamente no **vscode** 2. Mostrar como rodar e deixar disponível na rede: **flask run host=0.0.0.0** 3. **Descrição da aplicação:** 4. app.route(rule, options) 5. app.run(host, port, debug, options) 6. mostrar a opção **add\_url** |  |
| **Python decorators**   1. Rodar o exemplo **decorators** e **contextmanager** | *O método atual para transformar funções e métodos (por exemplo, declarando-os como classes ou métodos estáticos) é complicado e pode levar a código que é difícil de entender. Idealmente, essas transformações devem ser feitas no mesmo ponto do código onde a própria declaração é feita. Esta PEP introduz uma nova sintaxe para transformações de uma função ou declaração de métodos.* |
| Construir uma url dinamicamente  return “Hello {}”.format(name) |  |
| URL Canônica |  |
| Mais de uma URL por view |  |
| Métodos HTTP |  |
| Tratamento de POST HTTP |  |
| Mais sobre o objeto request | * request.**method**: Informa qual método HTTP foi usado na requisiçao * request.**headers**: headers HTTP da requisição, útil para checar o mimetype e dados de basic auth. * request.**environ**: Variáveis de ambiente do WSGI, navegador, ip do cliente etc * request.**path**, request.url: O path ou a url completa da requisição * request.**is\_xhr**: Informa se é ou não uma requisição Ajax * request.**blueprint**: Nome do blueprint que interceptou o request |
| Exemplo | print(request.environ['REMOTE\_ADDR'])        if 'chrome' in request.user\_agent.browser:          print("Usuario Chrome detectado")        print(request.user\_agent.browser) |
| * Fechamento: * Demonstração de uso de Jquery e JSON retornado pelo FLASK; * Exercícios! | |