

## Implementação do Projeto Final

Considere os seguintes requisitos:

1. Cada grupo deverá orientar sua implementação de acordo com o Tema definido;
2. As temáticas definidas referem-se a *lightweight frameworks* de desenvolvimento Web que trabalham conceitos de melhoria de produtividade, facilidade e melhoria de performance no desenvolvimento de sistemas Web usando linguagens suportadas pela JVM;
3. Para as implementações os Times deverão considerar as seguintes regras de negócio:
  - a. O sistema exemplo a ser implementado é um sistema que permite a postagem de um código fonte desenvolvido em Python através de um serviço Web contendo o seguinte contrato de requisição:  
HTTP Request – **Method:** POST – application/json – **Endpoint:** /maratona  

```
{
    'filename' : 'pradinho.py',
    'problem' : 'A',
    'sourcecode' : '<código_fonte_a_ser_executado>'
}
```
  - b. O sistema deverá considerar os problemas **Zerinho ou Um (A)** e **Mergulho (B)** (PDFs anexo) como problemas a serem processados para a validação do algoritmo – Não é necessária a postagem do problema, apenas a execução dos casos de teste que cada grupo deverá extrair dos seus respectivos PDFs, considerando as entradas e respectivas saídas dos problemas;
  - c. A comparação a ser executada para a saída do processamento é de “**igualdade exata**”, ou seja, o problema será considerado correto se a saída processada for idêntica à saída esperada;
  - d. O sistema deverá fornecer como resposta um JSON respeitando o seguinte contrato:  

```
{
    'filename' : 'pradinho.py',
    'problem' : 'A',
    'status' : 'SUCCESS'
}
```

Sendo, que o status pode assumir 'SUCCESS' ou 'FAIL' como valores.
  - e. O sistema deverá persistir (em memória – Considerar o uso de Collections) um histórico de todas as execuções realizadas, considerando:
    - Data e hora da execução
    - Nome do arquivo enviado
    - ID do problema processado
    - Código fonte processado
    - Status da execução
  - f. O sistema deverá fornecer um serviço Web para consulta de todo o histórico de execução, considerando os seguintes filtros:

HTTP Request – **Method:** GET - application/json – **Endpoint:** /maratona

- Por status;
- Por ID do problema; e
- Por período de execução (data e hora).

Fornecendo como resposta uma lista de registros de histórico respeitando o seguinte contrato:

```
{
    'filename' : 'pradinho.py',
    'problem' : 'A',
    'sourcecode' : '<codigo_fonte>',
    'datetime' : '<data_hora_execucao>',
    'status' : 'SUCCESS'
}
```

- g. Utilizar a ferramenta Postman para implementar os testes dos dois serviços a serem processados.
4. Instrução para a execução do problema:
- a. **python3.exe < arqEntradaCasoTeste1.txt > arqSaidaCasoTeste1.txt** – Devem ser considerados na execução todos os casos de teste existentes no problema, considerando um arquivo por caso de teste; e
  - b. Efetuar a leitura do arquivo gerado e do arquivo de saída existente com o resultado esperado, efetuando as devidas comparações.