

# Relatório INFO7052-CI1061-CI092

## TRABALHO 1

Lucas Gabriel Lima - GRR20150563

### Escolhas de tecnologia:

Foi escolhida a linguagem Python para realizar o parsing do arquivo JSON e gerar o arquivo .java. A escolha da linguagem se deveu principalmente a eu ter uma boa experiência prévia com Python e me sentir confortável transformando dados com ele, uma vez que a utilizo diariamente no meu trabalho a mais de um ano e meio.

Não somente pelo costume, existe um paralelo direto entre o formato JSON e a estrutura de dados 'dict' do Python, facilitando bastante o parsing e a manipulação das classes extraídas do arquivo de entrada para a geração do arquivo .java.

Com isso, o Python foi escolhido para a implementação do trabalho devido à facilidade de se manipular JSONs usando dicts, e por ser uma linguagem extremamente sucinta e legível, garantindo uma maior legibilidade ao código produzido, o que por sua vez facilita o entendimento do problema e futuras alterações.

### Escolhas de implementação:

O programa é utilizado com o comando:

```
python3 javagen.py <json_file_to_parse> <java_output_file>
```

Caso não seja informado um arquivo de saída, o programa imprimirá o código produzido na saída padrão. Caso nenhum parâmetro seja informado, o programa imprimirá como ser utilizado de forma correta.

Todo o funcionamento do programa está contido na função

`generate_code(file)`.

Essa função é responsável por chamar a função que realiza o parse do arquivo de entrada, que então retorna um dict (dicionário) contendo uma lista com todas as classes detectadas. Cada classe neste dicionário, é por sua vez um dicionário contendo seu nome, e o nome de seus atributos, bem como o seu tipo.

A partir daí, a função fica responsável por iterar entre as classes detectadas e imprimir suas respectivas classes em java para a saída.

Para o parse do arquivo de entrada, a função `generate_code()` chama a função `parse_json(file)` que se responsabiliza por transformar o arquivo JSON em um dict, detectar as classes e seus atributos, e retornar um dict no formato acima explicado. Para passear pelo arquivo JSON, utilizou-se uma abordagem recursiva, onde para cada item do dict, ou lista de dicts, é criada uma nova classe, com cada um de seus atributos e seus tipos.

Essa abordagem permite que o parse ocorra de forma genérica para qualquer arquivo JSON, novamente facilitando futuras alterações.