

Mestrado em Engenharia Informática

Experimentação em Engenharia de Software

Projeto: Fase I e II

Este documento descreve o projeto de software da disciplina de Experimentação em Engenharia de Software que deverão resolver em grupos de 3 alunos.

Fase 1: Monitorização da performance de linguagens de programação

Nesta fase pretende-se que os alunos considerem *benchmarks* desenvolvidos pela comunidade de programadores das linguagens de programação e que foram/estão a ser usados em estudos (científicos ou outros) em larga escala.

Como exemplo, apresentam-se três benchmarks muito usados em três linguagens de programação diferentes:

- Python: `pyperformance` (<https://github.com/python/pyperformance>)
- Haskell: `NoFib` (<https://gitlab.haskell.org/ghc/nofib>)
- Java: `Dacapo` (<https://www.dacapobench.org/>)

Os alunos devem desenvolver as seguintes tarefas:

1. Descarregar, instalar, executar estes três benchmarks.
2. Procurar outros benchmarks, quer para outras linguagens de programação quer para as três referidas acima.
3. utilizar o RAPL e a função de monitorização fornecida nas aulas para analisar a performance dos benchmarks (tempo de execução, consumo de energia e de memória).
4. utilizar o PowerCap e a função de monitorização fornecida nas aulas para analisar o impacto de usar um limite no consumo de energia na execução dos benchmarks.
5. Considerar diferentes interpretadores/compiladores da mesma linguagem. Por exemplo, em Python podem usar o compilador `codon` (<https://www.usenix.org/publications/loginonline/codon-python-compiler>).

Fase 2: Análise dos Resultados

Nesta fase pretende-se que os alunos analisem os resultados obtidos na fase anterior. Assim, deverão desenvolver as seguintes tarefas:

1. Utilizarem métodos estatísticos para mostrar que o PowerCap reduz o consumo de energia em cada benchmark estudado. Utilizem para tal significância estatística estudada nas aulas para fazerem esta análise.
2. Utilizarem métodos estatísticos para mostrar que o PowerCap aumenta o tempo de execução em cada benchmark estudado.
3. Compare o desempenho obtido considerando diferentes linguagens. Por exemplo, indicando qual das linguagens tira melhor partido do PowerCap.
4. Como extra, poderá considerar diferentes versões de compiladores/interpretadores (por exemplo python 3.7, 3.10, 3.12, java, etc) e analisarem o impacto que eles têm no desempenho energético e de tempo de execução.
5. Escreva um bom relatório fundamentando todos os resultados obtidos e as conclusões a que chegaram.