

# Análise e Teste de Software: Projeto Prático

João Saraiva & José Nuno Macedo

Ano Lectivo 2023/2024

Considere o projeto da disciplina de Programação Orientada a Objetos proposto no ano lectivo 2022/2023. No projeto prático de Análise e Teste de Software pretende-se usar técnicas de teste de software para testar a(s) aplicações desenvolvidas pelos alunos de POO. Junto com este enunciado disponibilizam-se 3 soluções desenvolvidas pelos alunos de POO.

Considere estas soluções, e eventualmente a solução que desenvolveu com o seu grupo no ano anterior, e realize as seguintes tarefas.

**Data de Entrega/Apresentação: 27 de Maio**

## 1 Tarefas a Desenvolver

Analise as soluções disponibilizadas e atente no facto de que possuem níveis de complexidade diferentes. Deverá escolher projetos entre as soluções disponibilizadas (e eventualmente a solução que desenvolveu com o seu grupo no ano anterior), e realizar as seguintes tarefas:

1. Escrever testes **JUnit** para o sistema de software considerado. Se este já possuir testes unitários, tentar complementar a suite de testes.
2. Utilizar o sistema **EvoSuite** para gerar testes automaticamente.
3. Analisar cobertura dos testes e a qualidade dos testes.
4. Utilizar o sistema de mutação de código para Java **PIT** para criar mutantes do programa Java a testar.
5. Utilizar o sistema de geração automática de casos de teste **QuickCheck** para gerar o ficheiro de logs que a aplicação a analisar recebe como input.
6. Relatório com gráficos e análise das soluções do projeto Vintage!

## 2 Extras

Serão valorizados os trabalhos em que sejam implementadas tarefas extras relevantes à análise e teste dos projetos de POO em consideração. Nesse sentido, alguns extras a considerar:

1. Automatizar partes do processo de análise dos projetos, por exemplo com recurso ao Maven, e/ou a uma Makefile. No seu nível mais extremo, é possível com um único comando correr todas as ferramentas que o projeto envolve e gerar gráficos com a informação relevante agregada.
2. Utilizar Large Language Models, como o ChatGPT ou o GitHub Copilot, para a geração de casos de teste para os projetos. Fazer uma análise da qualidade dos testes gerados por estas ferramentas.
3. Usar outros sistemas de geração de casos de teste para produzir o ficheiro de logs. Um possibilidade será usar o sistema **Hypothesis** em Python.
4. Usar um sistema de Property Based Testing em Java para testar propriedades do projeto de POO.
5. Utilizar o sistema **SonarQube** para analisar a qualidade do código fonte. Este sistema analisa o código-fonte de um projeto em termos de *Code Smells*.
6. Bom relatório!