



### Trabalho Prático Final (Interdisciplinar)

É cada vez mais presente no dia-a-dia dos times de desenvolvimento de *software* nas empresas a preocupação com a Qualidade de *Software*.

Escrever um código de qualidade significa:

- Dominar conceitos básicos e boas práticas de programação para deixar o código limpo, legível e fácil de ser mantido;
- Utilizar ferramentas adequadas para **avaliação** e **manutenção** dessa qualidade durante e após o desenvolvimento do código;
- Dominar princípios de qualidade na aplicação de padrões em seus projetos.

Em nossas aulas, vimos que a atividade de teste de *software* pode estar presente nas diversas etapas do ciclo de vida de desenvolvimento de um sistema e que há, basicamente, 3 níveis/fases de teste: unidade, integração e sistema. Na fase de Teste de Unidade, o foco é na verificação da menor unidade do projeto de *software* (p.e. os métodos de um classe, considerando um código implementado utilizando-se o paradigma de orientação a objetos).

Partindo do que foi exposto e com o intuito de proporcionar a você, aluno, uma visão mais abrangente de parte deste processo de desenvolvimento de um *software*, propõe-se neste Trabalho Prático, o relacionamento de conteúdos de diferentes Unidades Curriculares (UCs) realizando, portanto, o que chamamos de **Interdisciplinaridade**. Trabalharemos a aplicação de alguns conteúdos da UC de Teste de Software nas UCs de Práticas em Desenvolvimento de Sistemas I ou de Programação para Internet I e II ou de alguma outra UC de programação que você já tenha feito no curso de forma que você perceberá que eles estão totalmente relacionados.

Desta maneira, estamos auxiliando você a desenvolver o conhecimento de forma mais integrada, assim como acontece no exercício da sua futura profissão.

Diante deste panorama e para realização deste trabalho prático, você e sua equipe poderão utilizar qualquer trecho de código de qualquer aplicação que tenha sido desenvolvida nas UCs mencionadas anteriormente. Você também pode criar uma pequena aplicação para a realização deste trabalho.

O trabalho poderá ser realizado **individualmente ou em grupos de até 4 alunos**. A data e o formato de entrega deverá estar de acordo com o definido pelo professor. Ficarei à disposição remotamente para possíveis esclarecimentos às suas dúvidas.

Desde já, desejo a todos um **ÓTIMO TRABALHO!**

Regras para a realização do trabalho:

1. Elaborar um breve documento de requisitos que apresente: uma introdução, uma visão geral da aplicação apresentando o escopo, os requisitos funcionais e os requisitos não funcionais que ela possui. Utilize o *template* sugerido pelo professor e disponibilizado no tópico deste trabalho.
2. Aplicar os critérios de testes de unidade, testes estáticos, testes estrutural (caixa-branca) e teste de aceitação:

- uma funcionalidade que faça persistência em banco de dados;
  - uma funcionalidade que faça algum cálculo matemático.
  - Uma funcionalidade que faça alguma verificação em um campo específico (um intervalo de valores ou validação de CPF/CNPJ etc).
3. Aplicar o critério de teste estrutural (caixa-branca). Observe que é importante selecionar uma funcionalidade na qual haja a ocorrência de comandos condicionais aninhados (if ou case) ou comandos de repetição (for ou while).
  4. Elaborar um plano de teste conforme modelo sugerido pelo professor no *template* fornecido ao tópico do trabalho.
  5. Para aplicações desenvolvidas na:
    - a) **UC Práticas em Desenvolvimento de Sistemas I**, você e sua equipe poderão utilizar a **IDE Eclipse** para criar os casos de teste de unidade por meio do framework **Junit TestCase** baseado no plano de teste elaborado anteriormente.
    - b) **UC Programação para Internet I ou II**, você e sua equipe poderão utilizar **o framework PHPUnit** para criar casos de teste de unidade.
    - c) Qualquer outra UC ou projeto pessoal de programação que você já tenha estudado ou realizado: encontre uma ferramenta ou framework apropriado para testar, como por exemplo: selenium, easyaccept, findbugs entre outros.

Critérios para entrega, apresentação e correção do trabalho:

- Entrega (1,0): deverá ser feita em formato .pdf, pelo SIGAA e em local indicado pelo professor na data limite definida.
- Completude do template (5,0): todas as seções e sub-seções do documento devem estar devidamente preenchidas com textos obedecendo a norma culta do nosso idioma.
- Apresentação (4,0): cada aluno/equipe deverá apresentar os resultados de seu trabalho. Na apresentação, deve-se contextualizar a aplicação que foi foco de teste, a ferramenta ou o framework utilizado para testar, os casos de teste utilizados e as técnicas (caixa-preta ou caixa-branca) aplicadas.

Data da entrega:

- dia 21/06/2021 – Último dia de aula da UC Teste de Software

Bom trabalho a todos e a todas!