

## Trabalho Prático - Etapa 2: Implementação e teste

### 1. Especificação

Esta seção apresenta as características fundamentais do **Trabalho Prático - Etapa 2** relativas à sua especificação.

#### 1.1 Informações gerais

A **Etapa 2** do trabalho consiste em definir funcionalidades requeridas relativas à **implementação e teste** em relação ao modelo de programa definido nas etapas anteriores a ser desenvolvido na disciplina. O trabalho deve estar organizado **obrigatoriamente** em, pelo menos, **quatro partes principais**, as quais são **descritas nas próximas seções** (1.2, 1.3, 1.4 e 1.5). Caso o grupo ache importante anexar seções adicionais (por exemplo, introdução, conclusão, desafios, etc.), estas devem estar em seções separadas.

#### 1.2 Mudanças em relação à Etapa anterior

Nesta seção deve ser indicado **quais mudanças ocorreram** em relação ao que foi especificado no documento da **etapa anterior**. É importante documentar todas as mudanças para que fiquem registradas, mesmo se já tiverem sido conversadas com o professor. Caso não tenha ocorrido mudanças, basta registrar por escrito que não houve mudanças em relação ao definido na etapa anterior.

**Atenção:** Mesmo que não tenha havido mudanças, esta seção é **obrigatória**.

#### 1.3 Implementação

Esta parte do trabalho consiste na implementação do *software*, incluindo também uma interface simples para a aplicação. O **programa** deve ser **obrigatoriamente** implementado na **linguagem de programação Java**, seguindo o **paradigma de programação Orientado a Objetos** juntamente com **convenções e boas práticas** vistas em aula. Procure utilizar a maior quantidade possível de “bibliotecas padrões” disponíveis para o propósito do trabalho. Caso seja necessário que outra biblioteca em particular seja utilizada, uma documentação deverá ser incluída no trabalho contendo seus passos de instalação e as funcionalidades utilizadas no trabalho (pode ser link de referência já existente, caso completo).

O **relatório** deve conter a **descrição textual do que foi implementado, confrontando a implementação com o projeto** feito na etapa anterior, em particular **relacionando o código com as classes definidas e seus relacionamentos, bem como suas adaptações**. É permitido (e encorajado) que haja mudanças no projeto. Neste caso especifique-as na seção anterior e faça o link correspondente, justificando-as de acordo com as decisões tomadas na implementação. Apresente na documentação **detalhes das classes com seus atributos e métodos**, incluindo informação sobre seus parâmetros, retorno, modificadores de acesso (se tomadas eventuais decisões não óbvias), interfaces, classes abstratas, métodos de classe e demais informações que o grupo julgar relevante.

Obs.: O objetivo desta etapa (e do trabalho ao todo) é focar nos aspectos estudados **relativos à implementação de classes do domínio**, sem ênfase na interface e outras partes

do conteúdo que fazem intersecção com outras disciplinas (inteligência artificial, computação gráfica, bancos de dados, redes, etc.). **Portanto, cuide ao priorizar as tarefas desenvolvidas!**

## 1.4 Teste

Esta parte do trabalho consiste em testar a aplicação a **nível de teste de desenvolvedor (testes unitários automatizados)**. Os testes unitários devem **obrigatoriamente** ser projetados e realizados **durante o desenvolvimento do código**. Caso o grupo deseje, pode utilizar os conceitos de desenvolvimento dirigido a testes (*Test-Driven Development* - TDD). Os testes devem ser feitos com a **ferramenta JUnit** e devem abranger **obrigatoriamente todas as classes do domínio**. Classes relacionadas à interface e ao armazenamento de dados não precisam ser testadas.

O relatório deve conter o **detalhamento dos testes unitários realizados, juntamente com seus casos de testes**. Além disso, deve ser apresentado um breve relato sobre a **experiência do grupo em utilizar testes unitários** durante a (ou antes da) implementação.

## 1.5 Executável (aplicação, interface)

Nesta seção, o grupo deve apresentar detalhes sobre a **aplicação (executável)**, em particular apresentando as funcionalidades desenvolvidas com imagens do software em execução, mostrando suas interfaces e interações sobre ela. Apesar de não ser a prioridade do trabalho, é fundamental que haja pelo menos algum tipo de interface gráfica (mesmo que simples), evitando que a aplicação rode apenas por linha de comando, por exemplo.

O relatório deve incluir um “**manual de uso**” simplificado, mostrando detalhes de como executar o aplicativo, acompanhado de **imagens da interface** seguida de **explicações gerais de cada funcionalidade**. Além disso, deve haver uma **discussão em linhas gerais confrontando a interface atual com a prototipada na etapa anterior**.

## 2. Apresentação

O grupo deve preparar um **conjunto de slides** para apresentar as **partes principais** do trabalho desenvolvido para o professor e os colegas. O tempo de apresentação deve ser de no máximo 20 minutos. A apresentação deve mostrar obrigatoriamente:

- Uma **breve descrição** sobre o **problema/assunto** abordado;
- O **diagrama de classes** atual (com eventuais mudanças da parte anterior);
- Uma visão geral do **código** e sua estruturação;
- Uma visão geral dos **testes unitários** realizados;
- **Demonstração em tempo real** da aplicação desenvolvida.

A ordem de apresentação será definida no início da aula, onde todos os alunos devem estar presentes. É importante prestar atenção no tempo limite e dividir a apresentação previamente entre os membros do grupo para que todos consigam participar da apresentação. Além disso, garanta que o aplicativo poderá ser executado em tempo real, isto é, verifique a adequabilidade dos computadores do laboratório para suportar o projeto (ou, alternativamente, certifique-se de que alguém do grupo tenha testado o ambiente em computador pessoal próprio a ser levado no dia da apresentação).

### 3. Entrega

Este trabalho (Etapa 2) deve ser entregue exclusivamente via commit no GitHub do grupo conforme data combinada pelo professor considerando os seguintes artefatos:

- (i) **o relatório em pdf**, devendo seguir as especificações conforme consta neste documento, nomeado como **INF01120\_grupoXX\_Etapa2\_Relatorio.pdf**
- (ii) **os slides de apresentação em pdf**, conforme as especificações deste documento, nomeado como **INF01120\_grupoXX\_Etapa2\_Slides.pdf**
- (iii) **o código fonte**, já organizado em pacotes - conforme visto em aula (pastas) - incluindo as classes de teste feitas com o JUnit;
- (iv) **o executável em zip** (contendo o .jar e outros eventuais recursos necessários para rodar a aplicação)

**É importante que todos os membros do grupo tenham commits de código no GitHub, demonstrando trabalho colaborativo.**

Como extra, fazer um README no GitHub deixa o trabalho com um aspecto bem profissional. Alguns exemplos:

<https://github.com/hashicorp/nomad>

<https://github.com/Kanaries/pygwalker>

<https://github.com/microsoft/terminal>

**Bom trabalho!**