Universidade Federal de Viçosa
Departamento de Informática
INF 112 - Programação II
2º Semestre de 2024
Professor: Julio Cesar Soares dos Reis
Estagiários de Ensino: Bruno Alencar Vieira de Rezende e Paula T. Mota Gibrim
Valor: 17 Pontos

Trabalho Prático Final

1 Introdução

Conforme já conversamos, além dos nossos exercícios práticos (aulas práticas + trabalhos individuais), a disciplina irá contar com um trabalho prático (TP) final. Para fazer o mesmo, você deve montar grupos de até 4 pessoas. Tal TP final será um projeto de software de pequeno/médio porte. O mais importante é que o mesmo aplique os conceitos vistos em aula.

O grupo deve escolher um problema de seu interesse e realizar todo o processo de desenvolvimento de um sistema de pequeno/médio porte (análise, projeto, implementação e testes), com foco na aplicação dos conceitos e técnicas vistos durante o curso (modelagem, POO, gerenciamento de memória, tratamento de exceções, etc). Além disso, **o sistema desenvolvido deverá ser acessível** considerando uma deficiência dentre as listadas em: `https://www.unoesc.edu.br/atendimento-ao-estudante/inclusao-e-acessibilidade/tipos-de-pcd/`. Uma lista de sugestões de temas é apresentada abaixo. Entretanto, o tema é aberto à negociação caso o grupo tenha outra ideia. É uma ótima oportunidade para ser criativo!

1. Batalha RPG
2. Jogo de Magic
3. Jogo de Cartas
   (a) UNO
   (b) Poker
   (c) Truco
4. Sistema de Gerência
   (a) e-commerce
   (b) Biblioteca
5. Seu tema!!!

2 Desenvolvimento e Entrega

O desenvolvimento e a entrega deverão ser feitos utilizando o sistema de controle de versão GitHub. Sugere-se que *commits/pushs* sejam feitos de maneira frequente. O calendário[1] de atividades do trabalho é mostrado na tabela abaixo:

Atividade	***Data***
Definição do tema + grupos (PVANet Moodle)	Até 14/11
Entrega parcial (User Stories/Cartões CRC)	Até 17/11
Apresentação parcial (proposta)	19/11/2024
Apresentação final	28/01/2025
Entrega final (Github)	Até 29/01/2025

Tabela 1: Calendário de atividades.

[1] Cronograma sujeito a alterações!

3 Primeira Entrega

A primeira entrega consiste de *User Stories* + cartões CRC. As *User Stories* são uma forma simples de apresentar os requisitos funcionais desejados para um determinado sistema. São artefatos de desenvolvimento utilizados principalmente em processos baseados em metodologias ágeis. As descrições são intencionalmente genéricas, dando liberdade ao grupo para decidir detalhes da implementação. O grupo deverá identificar possíveis funcionalidades interessantes de serem incorporadas ao sistema e propor pelo menos seis *User Stories*. Uma aula sobre o assunto está prevista no cronograma da disciplina.

1. A primeira entrega de definição de tema + user stories/cartões CRC vale **2pts** do projeto final;
2. Depois disso, iniciem o trabalho no tema, vamos ter uma apresentação da proposta conforme data disponível no calendário da disciplina;
3. Até a apresentação do projeto, tente criar ao menos um README.md no seu github! Se já quiser incluir o documento com as user stories/cartões CRC, ótimo! No entanto, não se preocupe. Teremos uma aula sobre este assunto!

4 Segunda Entrega

Repositório com código completo, README.md indicando como compilar e executar o código.

1. Documentação (**1pts**).
   (a) Detalhamento do projeto.
   (b) Comentários, endentação.
2. Funcionamento correto (**6 pts**).
   (a) Compila e executa, não apresenta *crash*, etc.
3. Uso correto das boas práticas e dos conceitos de OO (**8 pt**s).
   (a) Abstração, Encapsulamento, Herança e Polimorfismo.
   (b) Modularidade e componentes reusáveis.
   (c) Acessibilidade.
   (d) Tratamento de exceções.
4. Criatividade, extras (ex.: interface gráfica, banco de dados, etc) (**+Xpts**).

5 Comentários Gerais

- Comece a fazer este trabalho logo: o prazo para terminá-lo está tão longe quanto jamais poderá estar! :)
- O programa fonte deve ser claramente comentado;
- Todos os integrantes do grupo devem conhecer a totalidade do código e ter condições de explicar sua função;
- Trabalhos copiados serão penalizados;
- Deve ser fornecido com o código um arquivo Makefile com as opções "`make`"e "`make run`";
- Mais detalhes sobre a entrega poderão ser fornecidos posteriormente.

6 Livros

1. Clean Code: A Handbook of Agile Software Craftsmanship. Robert C. Martin. Prentice Hall, 2008.
2. Code Complete: A Practical Handbook of Software Construction. Steve McConnell. Microsoft Press, 2004. 2nd Edition.
3. Effective C++: 55 Specific Ways to Improve Your Programs and Designs. Scott Meyers. Addison-Wesley Professional, 2005. 3nd Edition.
4. A Tour of C++. Bjarne Stroustrup. Addison-Wesley Professional, 2013. 1st Edition.