Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Programação e Desenvolvimento de Software 2
2º Semestre de 2022
Prof. Douglas G. Macharet

Projeto Final

Valor: 30 (+2) pontos
Data de entrega: 06/12/2022

1. Introdução

O grupo deve escolher um problema de seu interesse e realizar todo o processo de desenvolvimento (análise, projeto e implementação) de uma solução. Espera-se ao final um sistema de porte médio (>2000 linhas de código) que utiliza os conceitos e técnicas vistos durante o curso (modelagem, POO, testes unitários, etc). O programa deve ser feito baseado na linguagem **C++11**.

Uma lista não exaustiva ou exclusiva de sugestões de temas é apresentada abaixo. O tema é aberto à negociação e os alunos também podem sugerir outros temas de seu interesse.

- Twitter (tweet, retweet, follow, etc)
- Jogo de cartas (distribuir cartas, fazer jogadas, alternar turnos, etc)
- Gerenciador de tarefas/compromissos (adicionar, remover, listar, histórico, etc)
- Sistema para biblioteca (busca, reserva, empréstimo, etc)
- Mini rede social (timeline, adicionar/listar amigos, post, etc)
- UFMG: Carona, Eventos, Bandejão, Gamefication, Oportunidades, ...
- Outros (Magic, RPG, Reserva de Passagens/Hotel, Netflix, Spotify, Trello/Notion, ...)

ATENÇÃO: Temas similares poderão ser escolhidos por no máximo 3 grupos!

O desenvolvimento e a entrega deverão ser feitos utilizando o sistema de controle de versão GitHub[1]. Sugere-se que commits/pushs sejam feitos de maneira frequente, sempre que houver alterações.

O calendário de atividades do trabalho é mostrado abaixo:

Atividade	**Data**
Definição do tema (Moodle)	Até 15/09
Entrega parcial (Github)	Até 13/10
Apresentação / Entrevista	01/12 e 06/12
Entrega final (Github)	06/12

Lembre-se, o objetivo não é apenas escrever um programa funcional, mas desenvolver um sistema confiável, reutilizável e de fácil manutenção e extensão! Logo, tente aplicar todos os conceitos de POO, modularidade e corretude vistos em sala de aula. Também serão avaliados critérios como criatividade na solução, assim como a possível implementação de funcionalidades extras.

[1] https://github.com/

Durante o desenvolvimento, o grupo deverá utilizar o framework doctest[2] para implementar os testes de unidade. Deve haver pelo menos uma classe de testes para cada uma das principais classes do sistema (por exemplo, as classes de entidades). Além disso, deve-se apresentar uma cobertura total do código de pelo menos **60%** (a ser verificado utilizando a ferramenta gcovr[3]).

A interface de interação com a aplicação poderá ser feita via terminal de comando. Possíveis arquivos necessários durante a etapa de inicialização do sistema deverão ser fornecidos pelo grupo.

2. Modelagem

As User Stories[4] 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.

Nesta etapa, o grupo deverá identificar possíveis funcionalidades interessantes de serem incorporadas ao sistema e propor pelo menos **SEIS** User Stories. Para cada uma, deverá ser feita uma descrição sucinta, similar ao exemplo abaixo.

- **Sistema de Gestão Acadêmica**
- **Descrição:**
  - Como professor quero gerar um relatório para verificar o desempenho dos alunos.
- **Critérios de aceitação:**
  - Mostrar a pontuação da avaliação atual de um aluno.
  - Mostrar a pontuação de avaliação passada de um aluno.
  - Mostrar a média e desvio padrão da turma para a avaliação.

Tente apresentar User Stories para **diferentes tipos de usuário do sistema** (ou considerando papeis específicos em certos contextos). Cada User Story deve apresentar entre **3 e 5 critérios de aceitação**.

Em seguida, faça pelo menos **SEIS** cartões CRC para as classes que você julga como sendo as mais importantes no sistema. Cada cartão deve apresentar entre **5 e 10 responsabilidades** (considerando conhecimento e realização) e mencionar pelo menos **2 colaborações**. Você pode usar esse site para lhe auxiliar a fazer o cartão: https://echeung.me/crcmaker/

Os Cartões CRC e as User Stories devem ser colocados no próprio **repositório do grupo** no GitHub.

3. Documentação

Todo o desenvolvimento do projeto deverá ser documentado utilizando-se Doxygen[5] e o arquivo README[6] do repositório. No código, detalhe as estruturas de dados utilizadas e o funcionamento dos métodos. Na descrição, faça uma breve apresentação do problema, visão geral da solução focando na estrutura e funcionamento do programa, e as principais dificuldades encontradas.

---

[2] https://github.com/onqtam/doctest
[3] https://gcovr.com/en/stable/
[4] https://en.wikipedia.org/wiki/User_story
[5] http://www.doxygen.org/
[6] https://docs.github.com/pt/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes

4. Comentários Gerais

- Comece a fazer o trabalho logo, enquanto o prazo está tão longe quanto jamais poderá estar.
- O trabalho deverá ser feito em grupos com 4 ou 5 alunos.
- Trabalhos copiados serão penalizados conforme anunciado.
- A entrega final será considerada o último commit na branch principal do projeto.
- Deve ser fornecido com o código um arquivo Makefile com as opções ‘make’ e ‘make run’.

5. Critérios de avaliação

5.1. Entrega Parcial (Modelagem)

- User Stories (3 pts).
- Cartões CRC (3 pts).

5.2. Entrega Final (Implementação)

- Documentação (4 pts).
    - Detalhamento do projeto (README).
    - Doxygen.
- Funcionamento correto (4 pts).
    - Compila e executa, não apresenta crash, etc.
- Uso correto das boas práticas e dos conceitos de OO (6 pts).
    - Abstração, Encapsulamento, Herança e Polimorfismo.
    - Modularização e componentes reusáveis.
- Programação defensiva / Tratamento de exceções (3 pts).
- Testes de Unidade / Cobertura (3 pts).
- Apresentação / Entrevista (4 pts).
- Participação individual ([0, 1]).
    - Baseado na proporção de commits no GitHub.
- Criatividade, extras (+2 pts).