

Projeto Final

Valor: 30 (+2) pontos

Data de entrega: 13/06/2019

1. Introdução

O grupo deve escolher um problema de seu interesse e realizar todo o processo de desenvolvimento de um sistema de médio porte (análise, projeto e implementação), aplicando 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 de sugestões de temas é apresentada abaixo. Entretanto, o tema é aberto à negociação caso o grupo tenha outra ideia.

- Twitter (tweet, retweet, follow, etc)
- Jogo de cartas (multiplayer, distribuir cartas, fazer jogadas, alternar turnos, etc)
- Gerenciador de tarefas (adicionar, remover, listar, histórico, etc)
- E-commerce (Amazon, MercadoLivre, listar itens, adicionar ao carrinho, etc)
- Sistema para biblioteca (busca, reserva, empréstimo, etc)
- Mini rede social (timeline, adicionar amigos, post, etc)

ATENÇÃO: Cada tema poderá ser escolhido por no máximo 4 grupos!

O desenvolvimento e a entrega deverão ser feitos utilizando o sistema de controle de versão GitHub¹. Sugere-se que commits/pushs sejam feitos de maneira frequente, pelo menos uma vez ao dia.

O calendário de atividades do trabalho é mostrado abaixo:

Atividade	Data
Definição do tema (Moodle)	Até 26/03
Entrega parcial (Github)	Até 30/04
Apresentação em sala de aula/Entrevista	11/06 e 13/06
Entrega final (Github)	13/06

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 utilizar os conceitos de orientação a objetos e modularidade 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 doctest² para implementar os testes de unidade. Deve haver pelo menos uma classe de testes para as principais classes do sistema (por exemplo, as classes de entidades). Deve-se apresentar uma cobertura total de pelo menos **80%** (a ser verificado utilizando a ferramenta gcov³).

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. User Stories

As User Stories⁴ 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. Deverá ser feita uma descrição sucinta, similar ao exemplo abaixo.

- **E-commerce**
 - **Descrição:** Como comprador, quero pesquisar por um item para efetuar a compra.
 - **Critérios de aceitação:**
 - A busca não pode ser feita se não for informado o nome do produto.
 - Caso o produto (ou similar) seja encontrado, o sistema deve apresentar uma lista com informações como nome e preço unitário.
 - Caso o produto não seja encontrado, o sistema deve exibir uma mensagem específica ao usuário.

3. Documentação

Um relatório deve ser entregue ao final de forma a deixar claro as decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado. Adicione também comentários gerais sobre o trabalho e as principais dificuldades encontradas.

O grupo também deverá realizar a documentação do código utilizando Doxygen⁵. Nesse caso, apenas é necessário adicionar a funcionalidade/descrição do método e utilizar pelo menos as tags @param e @return.

Entre outras coisas, o relatório deve conter:

1. Introdução: descrição do problema/jogo e visão geral sobre o funcionamento do programa.
2. Implementação: descrição sobre a implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas, o funcionamento das principais funções, bem como decisões tomadas. Possível descrição de extras implementados.

² <https://github.com/onqtam/doctest>

³ <https://gcc.gnu.org/onlinedocs/gcc/Gcov.html>

⁴ https://en.wikipedia.org/wiki/User_story

⁵ <http://www.doxygen.org/>

3. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
4. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.

4. Comentários Gerais

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- O trabalho deverá ser feito em grupos de 4 alunos.
- Trabalhos copiados serão penalizados conforme anunciado.
- Penalização por atraso: $(2^d - 1)$ pontos, onde d é o número de dias de atraso.
- A entrega final deve ser feita utilizando o conceito de Release⁶ do GitHub.
- Deve ser fornecido com o código um arquivo Makefile com as opções ‘make’ e ‘make run’.

5. Critérios de avaliação

5.1. Entrega Parcial

- Proposição e detalhamento das User Stories (4 pts).
- Cartões CRC (4 pts).

5.2. Entrega Final

- Documentação (3 pts).
 - Detalhamento do projeto.
 - Doxygen, comentários, endentação.
- Funcionamento correto (5 pts).
 - Compila e executa, não apresenta crash, etc.
- Uso correto das boas práticas e dos conceitos de OO (5 pts).
 - Abstração, Encapsulamento, Herança e Polimorfismo.
 - Modularidade e componentes reusáveis.
 - Tratamento de exceções.
- Testes de Unidade (3 pts).
- Apresentação em sala/Entrevista (3 pts).
- Participação individual (3 pts).
- Criatividade, extras (+2 pts).

⁶ <https://help.github.com/articles/creating-releases/>