

Trabalho Prático Interdisciplinar

Disciplinas: Fundamentos de Engenharia de Software
Algoritmos e Estruturas de Dados I
Curso: Engenharia de Software
Profs.: Maria Augusta Nelson, Roberto Felipe Rocha e Ivre Marjorie R. Machado
Entrega: 28/06/2020
Valor: 10 pontos (FES)

Observações:

- O trabalho poderá ser feito em **grupos de até 5 alunos**.
- Copias de trabalho receberão a nota **ZERO**
- O trabalho será avaliado em **10 pontos** na disciplina de FES
- O programa deve ser feito na linguagem de programação C
- O programa deverá ser entregue no Canvas até 28/06 às **23:59 horas**
- Deverá ser entregue o **projeto completo** do **programa** e a **documentação**
- Em caso de dúvida, envie email para seu professor

Metodologia

Este é um trabalho interdisciplinar onde você vai planejar, analisar, projetar e implementar uma solução de software para o problema apresentado utilizando o Scrum para gerenciar o seu progresso.

Inicialmente organize o seu *backlog* de produto contendo as funções básicas do sistema. Cada uma das funções será responsabilidade de um membro do grupo e será feita em *sprints* de 3-4 dias. Seguem algumas sugestões de atividades a serem feitas nas *sprints*:

- 1- Definir a assinatura da(s) função(ões). Reflita sobre os parâmetros de entrada e saída da função e comunique aos seus colegas de projeto.
- 2- Documente a sua função colocando o seu propósito, os parâmetros de entrada e saída da sua função. O nome da função deve ser escolhido sob o ponto de vista de quem usa a função ou de quem vai chamar a função e deve refletir o que a função faz.
- 3- Implemente o caso de sucesso da função.
- 4- Selecione casos de testes para verificar o funcionamento da sua função. Um caso de teste deve conter os valores de entrada para a função, a saída esperada.
- 5- Execute os casos de testes, da sua função. Inicie fazendo a execução manual de alguns poucos casos de teste. Em seguida implemente a automatização dos testes da sua função usando a biblioteca munit.

Pontifícia Universidade Católica de Minas Gerais

6- Crie um relatório de execução de testes que contenha os casos de testes, a saída retornada durante a execução e uma indicação se o teste passou ou não na função. Isso é feito comparando a saída esperada documentada no caso de testes com a saída retornada durante a execução (esperado x real).

7- Implemente os casos especiais, exceções que possam existir na sua função. Em seguida execute os casos de testes anteriores para garantir que as mudanças não quebraram o código anterior que já funcionava. Pense também nos novos casos de testes necessários para a nova versão da função.

O que deve ser entregue para os professores no Canvas

1- A evolução do *backlog* de produto a cada semana. Indique quais tarefas encontram-se no *backlog* do produto, as que foram alocadas nas *sprints* (indique em qual *sprint* cada tarefa foi alocada).

2- A documentação das funcionalidades do software

3- A documentação de casos de testes, a implementação dos casos de testes automatizados e o relatório de execução dos testes.

4- O código em C das funções e do programa principal, juntamente com o **projeto completo** do programa.