Atividade Acadêmica: Algoritmos e Programação – Estruturas Lineares
Professor: Guilherme Silva de Lacerda (guilhermeslacerda@gmail.com – gslacerda@unisinos.br)

Trabalho Final

Enunciado

Escolher um dos projetos listados a seguir como tema de trabalho. O projeto deve respeitar os pontos levantados no tópico *Organização*. Usem todos os recursos de programação que foram discutidos em aula (Boas práticas de programação, Conceitos e princípios OO, API de *Strings, Exceptions, Collections, Manipulação de Arquivos e outras*), quando necessário.
Usar testes unitários automatizados, simulando cenários e fazendo as devidas validações.

Projeto 1: Controle de Estoque

O Sistema deve apresentar as seguintes opções: Inserir produto, adicionar quantidade de produtos no estoque, retirar quantidade de produtos no estoque, listar todos os produtos, listar produtos que estão abaixo do estoque e sair.
O produto deverá ter as seguintes informações: código, nome do produto, quantidade de produtos, quantidade mínima (usada para controle).
O sistema deverá ter validações para códigos de produto, quantidades a serem adicionadas/removidas, controlar o estoque mínimo, entre outras que podem aparecer no desenvolvimento do software.

Projeto 2: Notas de Alunos

O Sistema deve apresentar as seguintes opções: Inserir aluno, Inserir disciplina, Adicionar nota na avaliação (considerar 2 notas por disciplina), Calcular média (aritmética e ponderada – se já tem as duas notas, considerar a nota1 com peso 1 e nota2 com peso 2), Listar todos os alunos com as suas disciplinas e médias (aritmética e ponderada) com resultado final (aprovado >= 7, reprovado < 7) e sair.
O aluno terá as seguintes informações: código, nome do aluno. A disciplina terá o código e nome da disciplina.
O sistema deverá ter validações para adicionar notas (menor que zero, maior que 10), códigos de disciplina e aluno na hora de adicionar nota na disciplina, entre outras que podem aparecer no desenvolvimento do software.

Projeto 3: Medida Certa

O Sistema deve apresentar as seguintes opções: Inserir pessoa, Calcular IMC, Calcular Peso Ideal, Calcular Taxa de Gordura Corporal, Listar todas as pessoas com resultado dos cálculos e para o IMC, basear-se na tabela para indicar se está obeso, peso ideal ou abaixo do peso) e sair.
A pessoa terá as seguintes informações: código, nome da pessoa, sexo, idade, peso e altura.

A fórmula para o cálculo do IMC é:

$$IMC = Peso / (Altura)^2$$

Para gerar a resposta final, utilize como referência as informações do quadro.

	Mulheres	**Homens**
Abaixo do Peso	Menor que 19,1	Menor que 20,7
Ideal	Entre 19,1 e 25,8	Entre 20,7 e 26,4
Obeso	Acima de 25,8	Acima de 26,4

A fórmula para taxa de gordura corporal é (S=0 para mulheres S=1 para homens):
TGC = (1,2 x IMC) - (10,8 x S) + (0,23 x idade) - 5,4

A fórmula para o peso ideal (Fórmula de Lorentz) é:

$$P = (h - 100) - \frac{h - 150}{K}$$

Obs: Altura (h) é dada em centímetros, K=4 para homens e K=2 para mulheres. O resultado é expresso em Kg.

Organização

1) Todo e qualquer código deve ter seu teste equivalente (classe para testar o código). Não esqueça de criar o *source folder* para o código e o *source folder test* para os testes. Respeite as convenções de pacotes
Exemplo: Classe Conta tem que ter uma ContaTeste que realize os testes nela

2) Respeite as convenções de código
(https://www.oracle.com/technetwork/java/codeconventions-150003.pdf)

3) Estruture o projeto com base nos pacotes seguindo seu agrupamento semântico, usando como base o projeto *ExemploCardapioRefatorado (ui, serviços, negocio, main, entre outros)*, disponível no EAD

4) Formar *time de 2 alunos*. Aproveite para discutir as decisões de design com o colega

5) Crie uma pasta *doc* dentro do projeto e gere um diagrama de classes do código que vocês escreveram para entender como os elementos se relacionam

6) Para a entrega, somente um do grupo precisa submetê-lo no EAD. Exporte um arquivo formato ZIP com o seguinte nome: *aluno1_aluno2_trabalhofinal.zip*

7) Usar testes unitários automatizados

8) O sistema deve gravar as informações em arquivo. *Dica:* ao iniciar a aplicação, leia os dados dos arquivos. Ao sair da aplicação, grave os dados nos arquivos.

Importante: São 3 projetos, com 2 alunos compondo cada time. Isso quer dizer que poderemos ter até 4 times (número máximo) trabalhando no mesmo projeto. Os times deverão registrar no fórum o nome do projeto e dupla. A preferência de escolha se dará pela ordem de registro no fórum.