



PROJETO INTEGRADOR

2017/2

Sumário

TEMA	3
OBJETIVO	3
DISCIPLINAS E SUAS APLICAÇÕES	3
METODOLOGIA.....	3
AVALIAÇÃO	4
ENTREGA DAS NOTAS.....	5
REGRAS E CONSIDERAÇÕES.....	5
ANEXO I -CURSO CIÊNCIA DA COMPUTAÇÃO	7
ANEXO II – ANÁLISE E DESENVOLVIMENTO DE SISTEMAS	7

TEMA

O tema se baseia em uma situação problema criada pelos professores orientadores do projeto integrador. Cada turma contém de dois a três professores orientadores do projeto, ficando os demais como professores auxiliares para confecção do resumo expandido dos grupos.

OBJETIVO

O propósito do projeto integrador é propiciar condições para que o aluno possa desenvolver de forma prática, os conhecimentos teóricos abordados em sala de aula. Através de uma ação multidisciplinar entre as disciplinas do semestre corrente do aluno. Focando em um estudo dirigido, no qual o aluno ao se deparar com uma situação problema, possa apresentar possíveis soluções.

DISCIPLINAS E SUAS APLICAÇÕES

Cabe aos professores(as), orientar e separar subtemas para as equipes participantes do projeto, além de auxiliar os alunos no desenvolvimento da pesquisa científica. As equipes devem realizar um levantamento bibliográfico a respeito do tema, apresentando um resumo expandido (conforme as normas institucionais da Iniciação Científica) e desenvolver uma atividade/protótipo/trabalho prático para as disciplinas práticas.

Em anexo encontra-se a descrição de cada período, além das especificações dos professores(as) responsáveis pelo projeto integrador do semestre.

METODOLOGIA

Os(as) alunos(as) serão separados(as) em grupos e trabalharão pelo menos 1 dia/aula com o projeto integrador em sala de aula em todas as disciplinas. Adotando a metodologia de pesquisa bibliográfica, os alunos deverão basear-se nas bibliografias indicadas pelos professores, para resolver o problema proposto em seu tema do projeto integrador.

Além da resolução técnica do problema do projeto integrador, cada grupo de aluno deverá **submeter** um resumo expandido (máximo 4 páginas) no seminário de iniciação científica da instituição. Os alunos que tiverem seus resumos aprovados, receberam 10 horas complementares pela aprovação e 15 horas pela sua apresentação em formato de Banner.

AVALIAÇÃO

Fica aberto ao professor(a) o método avaliativo (apresentação ou entrega de trabalho impresso), bem como aumentar o valor do projeto em sua disciplina. A nota valerá de acordo com a classificação dos períodos na tabela abaixo. O cálculo é formado pela média aritmética das disciplinas que o(a) aluno(a) está matriculado.

CIÊNCIA DA COMPUTAÇÃO

Período	Projeto Integrador
1º	2.0 pontos
2º	2.0 pontos
3º	2.0 pontos
4º	2.0 pontos
5º	1.0 ponto
6º	1.0 ponto
7º	1.0 ponto
8º	1.0 ponto*

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Período	Projeto Integrador
1º	2.0 pontos
2º	2.0 pontos
3º	1.0 ponto
4º	1.0 ponto
5º	1.0 ponto

Cada período seguirá as indicações e normativas contidas em anexo, não fugindo dos requisitos necessários para a conclusão do projeto. É deferido uma atribuição de nota parcial para a disciplina, se assim deseja o professor(a) orientador(a), ou seja, além da nota

do projeto integrador, o professor poderá utilizar o trabalho para compor uma nota parcial da sua disciplina.

*O 7º e 8º período do curso de Ciência da Computação, terá sua nota do projeto integrador formada por outro aspecto, uma vez que este período já realiza o Estágio Supervisionado II que abrange todos os conteúdos bases do curso e por este semestre os alunos estarem participando do projeto ENADE. Logo, fica a cargo da coordenação de curso passar sua pontuação.

ENTREGA DAS NOTAS

Do professores(as) orientador(a):

- **Data: 27/11/17.**

Cada professor(as) preencherá a nota de cada aluno(a) numa planilha e enviará até a data acima para a coordenação.

Da coordenação:

- **Data: 30/11/17.**

Fica cargo da coordenação de curso receber todas as notas e realizar o cálculo da média aritmética baseado na quantidade de disciplinas que o(a) aluno(a) está **matriculado(a)** e enviar para todos os(as) professores(as) até a data estipulada.

REGRAS E CONSIDERAÇÕES

1. Todos os(as) alunos(as) devem participar do projeto, a não participação significa na perda da nota.
2. O material final entregue ou apresentado (slides), devem ser repassados para a coordenação juntamente com as notas dos(as) alunos(as), por cada professor(a), numa planilha até a data estabelecida.
3. A atividade não entregue ou apresentada dentro do prazo estipulado pelo professor(a), como também plágios, serão desconsiderados, atribuindo **nota zero ao grupo**.
4. O(A) aluno(a) deve realizar o projeto integrador em todas as disciplinas que está matriculado(a) no semestre corrente.

5. Casos excepcionais serão analisados pelo colegiado de curso desde que o(a) aluno(a) apresente sua justificativa por escrito via e-mail para a coordenação do curso **até o dia 29/11/2017.**

Profº. Me. Edgar Cabral
Coordenador

ANEXO I -CURSO CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADOR - 1º e 2º Período

O trabalho prático consiste em desenvolver algoritmos para solucionar problemas de matemática e física. Um programa deverá ser desenvolvido para cada problema proposto. Esses programas deverão ser criados utilizando conceitos vistos nas aulas de Linguagem de Programação I, Física Aplicada e Fundamentos da Matemática.

A seguir, encontram-se a descrição do problema, a forma de submissão do trabalho e os critérios de avaliação.

- Equipes de 4 ou 5 participantes;
- Os algoritmos devem ser executados na ferramenta Visualg.

Todo o desenvolvimento do trabalho e cálculos devem ser descritos em um relatório de no máximo 4 páginas, no formato de resumo expandido da Iniciação Científica.

Regras para entrega do trabalho

Data da Entrega:

- ✓ 12/11/2017 – Entrega do trabalho por e-mail até 23h59.
- ✓ 14/11/2017 – Apresentação do trabalho na aula de Física.
- ✓ 16/11/2017 – Apresentação do trabalho na aula de Matemática.
- ✓ 17/11/2017 – Apresentação do trabalho na aula de Ling. de Prog. I.

O código-fonte dos programas e o relatório deverão ser entregues por e-mail para os endereços caroline.mazetto@gmail.com, allan_perna@hotmail.com e dinei_pcp@yahoo.com.br. O nome completo dos participantes da equipe deve ser informado no corpo do e-mail.

Especificação do Trabalho Prático

Problema 1

A empresa aérea *PhysicsAviation* transporta passageiros somente entre algumas cidades do Brasil. Essa empresa possui dois tipos de aviões:

- *VectorComponent007* que só executa trajetórias retilíneas paralelas à linha do equador e ao meridiano de Greenwich, ou seja, trajetórias horizontais e verticais, nas direções norte/sul e leste/oeste.
- *ResultantVector011* que executa qualquer trajetória retilínea entre as cidades.

Por exemplo, um dos aviões voa para o norte, de Brasília até Belém, a 1630 km de distância, levando 2h e 10 min nesse percurso. De lá, segue para oeste, chegando a

Manaus, distante 1290 km de Belém, após 1h e 50 min de voo. Qual é o vetor deslocamento total do avião? Em outras palavras, quais são as componentes vertical e horizontal do deslocamento total do avião? Qual o valor do módulo deste vetor?

Você é funcionário da *PhysicsAviation* e precisa criar uma programa que forneça algumas informações para o piloto. O piloto irá informar ao programa o tipo de avião com o qual vai viajar e o número do plano de voo que contém as informações referentes as cidades pelas quais tem que passar, origem e destino. Os planos de voo, referente a cada tipo de avião, estão apresentados nas tabelas a seguir:

Tipo de avião	Número do Plano de voo	Origem	Destino intermediário	Destino Final
VectorComponent007	PA232	Curitiba	Rio de Janeiro	Manaus
	PA457	Porto Alegre	Florianópolis	Rio de Janeiro
	PA949	Brasília	São Paulo	Curitiba
	PA1080	Belém do Pará	Manaus	Goiânia

Tipo de avião	Número do Plano de voo	Origem	Destino Final
ResultantVector011	PA360	São Paulo	Rio de Janeiro
	PA888	Brasília	Manaus

Utilize o *Google Maps* para estimar a distância entre as cidade, mas lembre-se que você deve conhecer principalmente a distância horizontal e vertical entre as cidades. Por exemplo, se o avião sai de Curitiba e vai para o Rio de Janeiro, o piloto da *PhysicsAviation* executará a seguinte trajetória aproximada com o avião Vector Component007:

1º) 280 km ao norte (linha reta paralela ao meridiano de Greenwich).

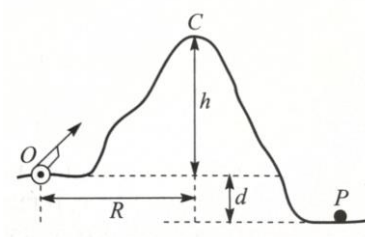
2º) 750 km ao leste (linha reta paralela ao equador).

Se o piloto estiver viajando com o avião *VectorComponent007*, o programa deve informar a componente horizontal (paralela ao equador) e vertical (paralela ao meridiano de Greenwich) do vetor deslocamento entre as cidades. Ou seja, entre a origem e o destino intermediário, entre o destino intermediário e o destino final, e entre a origem e o destino final. Além disso, o programa deve informar o módulo do vetor deslocamento entre a origem e o destino final.

Se o piloto estiver viajando com o avião *ResultantVector011* programa deve informar o módulo do vetor deslocamento entre a origem e o destino. Além disso, o programa deve informar qual o menor ângulo formado entre a linha reta que liga as cidade e uma linha paralela à linha do Equador.

Problema 2

Um canhão lança um projétil por cima de uma montanha de altura h , de forma a passar quase tangenciando o cume C no ponto mais alto de sua trajetória. A distância horizontal entre o canhão e o cume é R . Após da montanha há uma depressão de profundidade d (veja a figura abaixo). Determine a distância horizontal entre o ponto de lançamento O e o ponto P onde o projétil atinge o solo em função de R , d e h .



Uma empresa de armamento militar desenvolveu uma bomba que não pode ser rastreada se estiver próxima ao solo. O objetivo deste armamento é destruir grupos inimigos que se encontram próximos de montanhas. Assim, a base de lançamento da bomba pode ser construída do outro lado da montanha e executar um ataque surpresa, desde que a bomba não se afaste muito das rochas, por isso a bomba deve passar quase tangenciando o cume.

Os soldados do grupo de ataque têm contato com engenheiros cartográficos que conhecem as medidas da topografia da região. O geógrafo vai informar os valores de R , d e h e os soldados, com essas três informações, devem saber o local no qual a base de lançamento deve ser instalada. Crie um programa que calcule a distância horizontal entre o ponto de lançamento e o ponto onde o grupo inimigo se encontra.

Problema 3

Os terremotos são medidos através de suas magnitudes que são provocados pelo movimento de suas placas tectônicas. Durante anos estas placas apresentam movimentações menores ou maiores, em diferentes partes do mundo causando desde pequenos estragos até destruições de cidades. Terremotos podem ser classificados conforme suas intensidades e níveis de magnitude, conforme a figura abaixo:

Entenda os efeitos dos terremotos

Os sismólogos usam a escala de magnitude para representar a energia sísmica liberada por cada terremoto. Veja abaixo tabela com os efeitos típicos de cada terremoto em diversos níveis de magnitude

Escala Richter	Efeitos do terremoto
Menos de 3,5	Geralmente não é sentido, mas pode ser registrado
3,5 a 5,4	Frequentemente não se sente, mas pode causar pequenos danos
5,5 a 6,0	Ocasiona pequenos danos em edificações
6,1 a 6,9	Pode causar danos graves em regiões onde vivem muitas pessoas
7,0 a 7,9	Terremoto de grande proporção, causa danos graves
de 8 graus ou mais	Terremoto muito forte. Causa destruição total na comunidade atingida e em comunidades próximas

- Esta tabela é "aberta", portanto não é possível determinar um limite máximo de graus
- Ainda que cada terremoto tenha uma magnitude única, os efeitos de cada abalo sísmico variam bastante devido à distância, às condições do terreno, às condições das edificações e de outros fatores.

A magnitude de um terremoto na escala Richter pode ser calculada por meio da fórmula abaixo:

$$M = \frac{2}{3} \log_{10} \frac{E}{E_0}$$

onde:

- M é a Magnitude do terremoto em graus;
- E é a energia liberada no terremoto em kWh;
- E_0 é constante e vale $7 \cdot 10^{-3}$ kWh.

Crie um programa que receba como entrada a magnitude e calcule a energia liberada em Kwh. Adicionalmente, o programa deve exibir a classificação do terremoto conforme a tabela da Escala Richter, informando os seus efeitos.

Problema 4

O preço de uma corrida de TÁXI, em geral, é constituído de uma parte fixa, chamada de bandeirada, e de uma parte variável, que depende do número de quilômetros rodados. Já o preço de uma corrida de UBER, em geral é constituído de uma parte fixa, chamada de tarifa, de uma parte variável que depende do número de quilômetros rodados e mais uma taxa por minuto de viagem.

Crie um programa para calcular o valor a ser pago por uma corrida de TÁXI e UBER. Para o cálculo da corrida de TÁXI, considere que o valor da bandeirada é R\$ 4,00 e o preço do quilômetro rodado é R\$ 2,00. Para a corrida de UBER, considere que o valor da tarifa é R\$ 0,75, o preço do quilômetro rodado é R\$ 1,40 e a taxa por minuto de vigem é R\$ 0,26.

O programa deve ler uma distância em quilômetro da origem ao destino e calcular o preço da corrida de TÁXI e o preço da corrida de UBER. Por fim, o programa deve informar qual tipo de corrida é mais vantajosa (mais barata) para a distância informada.

Requisitos

- a) Para cada problema proposto, identifique os conceitos (físicos e/ou matemáticos) envolvidos;
- b) Para cada problema proposto, determine quais são as fórmulas, equações, ou funções necessárias para solucionar o problema;
- c) Crie um algoritmo (programa) para solucionar cada um dos problemas propostos;
- d) Defina a entrada, processamento e saída de cada algoritmo;
- e) Crie a representação gráfica (fluxograma) de cada algoritmo;
- f) Os programas devem exibir informações e mensagens para que o usuário realize a correta entrada de dados. Os resultados devem ser exibidos na saída de forma clara e organizada.
- g) O relatório deve ser organizado conforme a seguinte estrutura:
 - Resumo
 - Introdução
 - Desenvolvimento
 - Problema <X>
 - Descrição
 - Conceitos e Cálculos
 - Algoritmo
 - Resultados
 - Problema <X>
 - Conclusão
 - Referências

A seção Desenvolvimento deve conter uma subseção para cada problema proposto. Para cada problema devem ser apresentados uma descrição resumida do problema, os conceitos e cálculos envolvidos e o algoritmo desenvolvido.

A seção Resultados deve conter uma subseção para cada problema proposto. Para cada problema deve ser apresentado ao menos um exemplo de execução do programa correspondente. Mostre a entrada e saída produzida pelo programa.

Critérios de avaliação

Os trabalhos serão avaliados de acordo com os seguintes critérios objetivos:

Critério	Nota*
Trabalho que não apresentar o relatório no formato de artigo	De 0 a 0.3
Atendimento aos requisitos	De 0 a 0.3
Corretude dos requisitos	De 0 a 0.4

Trabalho entregue fora do prazo.	Metade da nota por dia de atraso
----------------------------------	----------------------------------

*** O total de pontos do Projeto Integrador no 2º bimestre será 2.0 pontos para o 1º e 2º período.**

Alguns alunos serão convocados durante a defesa para uma entrevista para explicar detalhes do trabalho. A nota individual do aluno dependerá desta entrevista.

Os trabalhos serão avaliados segundo critérios subjetivos definidos pelos professores, variando a nota dentro da faixa de notas definida pelo critério anterior. Alguns dos critérios subjetivos avaliados serão:

- Legibilidade (nome dos identificadores, formatação do código, uso de comentários quando necessário);
- Modularização (criação de funções bem definidas e independentes);
- Eficiência (desempenho e uso de recursos);
- Uso de estruturas de dados adequadas.

Referências

1. FORBELLONE, Andre Luiz Villar; EBERSPACHER, Nenri Frederico. Lógica de programação: A construção de algoritmos e estrutura de dados. 2.ed. São Paulo, SP: Mackron Books, 2000.
2. WIRTH, Niklaus. Algoritmos e estruturas de dados. Rio de Janeiro: LTC, 1999.
3. HALLIDAY, D.; RESNICK, R.; WALKER, J. Fundamentos de Física: mecânica. 10.ed. Rio de Janeiro, RJ: LTC, 2008.
4. GERSTING, Judith L. Fundamentos matemáticos para a ciência da computação. 4.ed. Rio de Janeiro: LTC, 2001.

Projeto Integrador – 3º e 4º período

Valor: 2.0 pontos na Disciplina de Programação
2.0 pontos na Disciplina de Estrutura de Dados
2.0 pontos na Disciplina de Banco de Dados
1.0 ponto para o Projeto Integrador

O trabalho prático da disciplina consiste em desenvolver um programa utilizando a linguagem de programação Java Orientado a Objetos, Estrutura de Dados e Banco de dados.

A seguir, encontram-se a descrição do problema, a forma de submissão do trabalho e os critérios de avaliação.

- Equipes de 3 a 5 participantes;
- O projeto deve ser criado no Eclipse;
- O programa deve compilar e executar com sucesso;
- Utilizar orientação à objetos;
- Organizar as classes em pacotes;
- Utilizar Java 7;
- Alguns alunos poderão ser chamados ao acaso durante a defesa para uma entrevista com a professora, na qual deverão explicar trechos de seus programas. A nota do aluno dependerá desta entrevista.

Todo o desenvolvimento do trabalho e resultados obtidos devem ser descritos em um relatório de no máximo 4 páginas, no formato de resumo expandido da Iniciação Científica.

Regras para entrega do trabalho

Data da Entrega:

- ✓ **19/11/2017** – Entrega do trabalho por e-mail até 23:59h
- ✓ **20/11/2017** – Apresentação no Laboratório/Defesa do código (1ª e 2ª aula) na disciplina de Programação
- ✓ **21/11/2017** – Apresentação no Laboratório/Defesa do código (1ª e 2ª aula) na disciplina de Estrutura de Dados
- ✓ **22/11/2017** – Apresentação no Laboratório/Defesa do código (1ª e 2ª aula) na disciplina de Banco de Dados

O código-fonte da solução deverá ser compactado (adicionar o projeto com o código-fonte do cliente e do servidor no mesmo arquivo zip) e entregue por e-mail (anexo ao e-mail) para o endereço mariana.silva@uniandrade.edu.br, camile.bordini@gmail.com e tiagocandido@hotmail.com.

O nome do zip deverá ser “**ProjetoIntegrador.zip**”.

Serão aceitos trabalhos entregues até as 23h59 da data limite. O assunto do e-mail deverá ser o seguinte:

Projeto Integrador:<nome1>:<nome2>:<nome3>....

O termo “<nome...>” deverá ser substituído pelos nomes dos alunos em ordem alfabética – somente um nome e um sobrenome de cada aluno.

Especificação do Trabalho Prático

1. Descrição do problema

O aluno trabalhará em um sistema de vendas que possui gerenciamento de estoque. Nele, uma das entidades (entidade: venda) terá sua interface implementada com recursos semelhantes a de um CRUD, mas com ênfase nas operações de estruturas de dados. Complementar às interfaces, por meio de recursos de banco de dados o sistema fará o controle completo do estoque, gerenciando as saídas e automatizando as reposições.

2. Requisitos

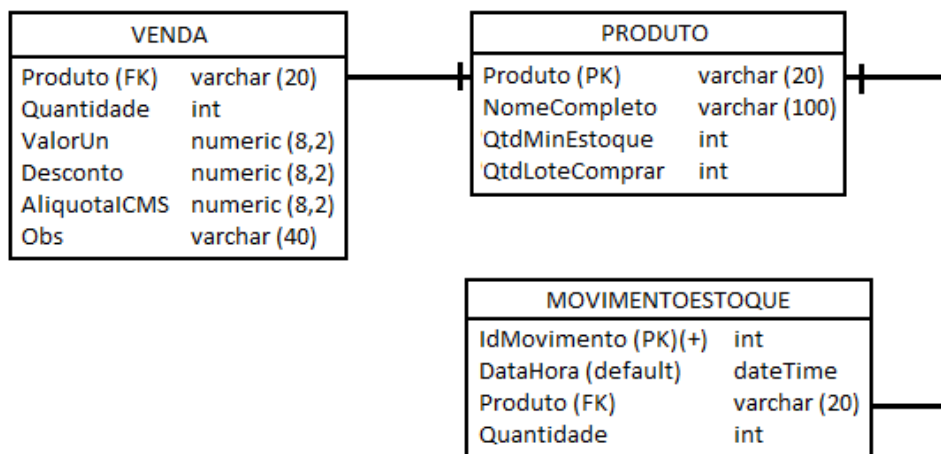
O sistema deverá possuir os requisitos abaixo:

1. Cadastrar Venda:
 - a. O cadastro de uma venda é representado pelo preenchimento dos seis atributos presentes na entidade VENDA no Diagrama de Entidades e Relacionamentos.
 - b. A entrada das informações será efetuada por leitura do teclado (utilizando classe Scanner, JOptionPane ou afins).
 - c. Os cadastros serão armazenadas em uma estrutura de Fila (enfileiradas), que deverá ser desenvolvida pelo aluno (não usar a classe Queue nativa do Java); a classe deverá se chamar Fila.java.
2. Transpor a estrutura para o banco de dados: ao utilizar a opção Desinfileirar a estrutura contendo as vendas deverá ser cadastrada no Banco de Dados via jdbc. Para isso será utilizada uma PROCEDURE chamada spInsereVenda.
3. Realizar a Leitura da Fila, bem como as operações de: Busca do Primeiro Elemento; Exibir o tamanho da Fila e Mostrar o resultado de uma consulta no banco de dados.
4. Deverá ser criado um menu no sistema que mostre as opções relacionadas aos requisitos, contendo no mínimo as opções:

```
1 - Realizar venda (enfileirar)
2 - Gravar dados na Base de Dados (desenfileirar)
3 - Mostrar Fila
4 - Buscar Primeiro Elemento da Fila
5 - Mostrar Tamanho da Fila
6 - Mostrar Itens cadastrados na Base de Dados
0 - Sair
```

2.1. Banco de Dados

O modelo lógico proposto deverá ser implementado em banco de dados Sql Server e o script (modelo físico) deverá ser entregue junto do trabalho escrito.



A tabela de PRODUTO deve ser previamente alimentada com pelo menos cinco produtos (sugestão: sal, manteiga, leite, bolacha, trigo) e as tabelas de VENDA e MOVIMENTOESTOQUE serão alimentadas pelo software durante o uso / apresentação aos professores.

A tabela MOVIMENTOESTOQUE guarda toda movimentação de venda (negativa) e de reposição de estoque (positiva). O saldo atual de um produto em estoque é representado pelo somatório das quantidades de todos os registros desta tabela.

Ao acionar uma procedure **splnserieVenda** (opção: 2 – Desinfileirar) a venda será inserida na tabela VENDA e disparará uma trigger **trMovimentaEstoque** que inserirá uma movimentação de estoque negativa. Imediatamente uma segunda trigger vinculada na tabela MOVIMENTOESTOQUE (**trVerificaNecessidadeReposicao**) será disparada para avaliar o estoque atual do produto e decidir (baseado no campo QtdMinEstoque do cadastro do produto) se uma reposição é necessária; caso o valor mínimo tenha sido atingido, uma COMPRA / REPOSIÇÃO acontecerá e será representada no sistema pela inserção pela própria trigger de uma movimentação positiva com a quantidade informada no campo QtdLoteComprar da tabela PRODUTO. Exemplo:

Produto: Pão QtdMinEstoque: 10 QtdLoteComprar: 50 Saldo Estoque: 0

Ao registrar uma venda de 5 unidades, o sistema lançará também uma movimentação de - 5 unidades. Como -5 é menor que 10 o sistema inserirá uma movimentação de reposição de +50 e o novo saldo (calculado) será +45. Se uma nova venda de 20 unidade for realizada, uma movimentação de -20 unidades será registrada e o saldo passará a ser +25, o que não desencadeia uma reposição (já que $25 > 10$). Ao realizar uma nova venda de 15 unidades, o saldo cai par 10 e gera uma reposição de 50 unidades voltando o saldo para +60, e assim sucessivamente.

Deve ser criada uma consulta que traga a relação de produtos e seus respectivos saldos de estoque. Essa consulta será utilizada para verificar o funcionamento da gestão de estoque. Opcionalmente, ela pode ser colocada no menu da aplicação, sendo o item [7].

O código fonte da procedure, das duas triggers e da consulta de saldos devem obrigatoriamente constar no trabalho.

2.2. Classe Fila

A estrutura que deverá ser usada para implementar a Fila é a de vetor, onde os elementos do vetor serão objetos de uma classe.

Além disso, para considerar a ideia de alocação dinâmica, o programa não deve limitar a quantidade de elementos a serem enfileirados. Para isso, cada vez que a fila estiver cheia, alguma providência deve ser tomada, como por exemplo, criar um novo vetor com o dobro do tamanho do original e mover os elementos da fila para o novo vetor.

a) Métodos mínimos necessários

- Enfileirar
- Desinfileirar
- Mostra Fila
- Retorna o Primeiro Elemento
- Tamanho Fila

b) Métodos auxiliares

- Cria Fila
- Fila Vazia
- Fila Cheia
- Atualiza Fila

Critérios de avaliação

Os trabalhos serão avaliados primeiramente pela execução correta, seguindo os seguintes critérios objetivos:

Critério	Nota
Trabalhos que não compilarem	De 0 a 0.5
Trabalhos que compilarem mas não gerarem resultados corretamente	De 1.0 a 1.5
Trabalhos que compilarem e gerarem as saídas corretamente	De 1.5 a 2.0
Trabalhos entregues fora do prazo	Metade da nota por dia de atraso

Em segundo lugar, os trabalhos serão avaliados segundo critérios subjetivos definidos pela professora, variando a nota dentro da faixa de notas definida pelo critério anterior. Alguns dos critérios subjetivos avaliados serão:

- Legibilidade (nomes de variáveis bem escolhidos, código bem formatado, uso de comentários quando necessário, etc.);
- Consistência (utilização de um mesmo padrão de código);
- Eficiência (sem exageros, tentar evitar grandes desperdícios de recursos);
- Uso de estruturas de dados adequadas.

Referências

1. AMADEU, Claudia Vicci. Banco de Dados. São Paulo: Pearson Education do Brasil, 2014.

2. ASCENCIO, A. F. G.; ARAÚJO, G. S. Estrutura de Dados: algoritmos, análise da complexidade e implementações em Java e C++. São Paulo: Pearson Prentice Hall, 2011.
3. CORNELL, Gary e HORSTMANN, Cay S. Core Java 2. Vol. 1: Fundamentos. Makron Books, 2000.
4. CORNELL, Gary e HORSTMANN, Cay S. Core Java 2. Vol. 2: Recursos Avançados. Makron Books, 2000.
5. DEITEL, H. M. e DEITEL, P. J. Java: Como Programar. Bookman, 2002. (Código 004.43 / D324j na Biblioteca)
6. ELMASRI, Ramez; NAVATHE, Shamkant B. Sistemas de Banco de Dados. 6. ed. São Paulo: Pearson, 2011.
7. HSQL: <http://hsqldb.org/doc/guide/ch01.html>
8. PUGA, S.; RISSETTI, G. Lógica de programação e estrutura de dados: com aplicações em JAVA. 2 ed. São Paulo: Pearson Prentice Hall, 2009.
9. Site da documentação do Java / Oracle:
<http://docs.oracle.com/javase/7/docs/api/allclasses-noframe.html>

PROJETO INTEGRADOR - 5º e 6º Período

O projeto integrador consiste na criação de uma representação gráfica para as cidades da região metropolitana de Curitiba. Para solucionar o problema proposto, devem ser utilizados conceitos e algoritmos estudados nas disciplinas de Teoria dos grafos e Computação Gráfica.

A seguir, encontram-se a descrição do problema, a forma de submissão do trabalho e os critérios de avaliação.

- Equipes de 4 ou 5 participantes;

Todo o desenvolvimento do trabalho e resultados obtidos devem ser descritos em um relatório de no máximo 4 páginas, no formato de resumo expandido da Iniciação Científica.

Regras para entrega do trabalho

Data da Entrega:

- ✓ **19/11/2017 – Entrega do trabalho por e-mail até 23h59.**
- ✓ **22/11/2017 – Apresentação do trabalho na aula Computação Gráfica.**
- ✓ **23/11/2017 – Apresentação do trabalho na aula de Teoria dos Grafos.**

O relatório e o arquivo de modelo 3D deverão ser entregues por e-mail para os endereços caroline.mazetto@gmail.com e camile.bordini@gmail.com. O nome completo dos participantes da equipe deve ser informado no corpo do e-mail.

Especificação do Trabalho Prático

Descrição do problema

A Região Metropolitana de Curitiba (RMC), originalmente com 14 municípios quando foi criada em 1973, hoje reúne 29 municípios do estado do Paraná, dentre eles: São José dos Pinhais, Araucária, Almirante Tamandaré, Campo Largo e Colombo.

Em vistas de identificar novas demandas de tráfego rodoviário entre os municípios, um instituto de pesquisa identificou que o tráfego entre alguns municípios passaram a ser mais frequentes nos últimos anos, mais especificamente dentre 26 dos 29 municípios que abrangem a RMC, são eles: Adrianópolis, Agudos do Sul, Almirante Tamandaré, Araucária, Balsa Nova, Bocaiúva do Sul, Campina Grande do Sul, Campo Largo, Campo Magro, Cerro Azul, Colombo, Contenda, Curitiba, Dr. Ulysses, Fazenda Rio Grande, Itaperuçu, Lapa, Mandirituba, Pinhais, Piraquara, Quatro Barras, Quitandinha, Rio Branco do Sul, São José dos Pinhais, Tijucas do Sul e Tunas do Paraná.

Para atender as necessidades dos municípios, o governo do estado decidiu realizar reparos nas rodovias, como alargamento de pista, duplicação ou até mesmo criação de novos trechos. Contudo, não será possível atender todas as demandas identificadas na pesquisa por questões de verbas orçamentárias. Assim, o instituto, juntamente ao órgão de trânsito

responsável, ficou responsável por classificar os municípios segundo critérios de “necessidade de ligação”. Por exemplo, se entre um município A e B quaisquer, há necessidade de realizar um reparo ou até mesmo a criação de um novo trecho de rodovia entre os municípios, isso deve estar identificado de alguma forma para que o governo possa analisar o cenário e tomar as devidas decisões quanto às prioridades de atendimento.

Sendo assim, os responsáveis pelo projeto decidiram mapear a RMC por uma representação em forma de grafo, onde os vértices representarão os municípios, e as arestas as ligações entre as cidades que deverão ser atendidas. Mas, devido às restrições orçamentárias, não será possível conectar todos os municípios entre si, e para isso, os projetistas decidiram classificá-los por cores, onde a mesma cor significa que as cidades possuem “necessidades semelhantes” quanto à demanda de tráfego. Por exemplo, identificou-se que Curitiba possui muitas demandas com diferentes municípios, logo, é necessário que pelo menos uma região de cada “cor” seja conectada com Curitiba.

Ao final do estudo, a classificação das cidades por cores e os requisitos que os projetistas identificaram foram os seguintes:

	MUNICÍPIOS
Cor 1	Curitiba
Cor 2	Almirante Tamandaré, Araucária, Campo Largo, Colombo, Piraquara, Quatro Barras, São José dos Pinhais.
Cor 3	Adrianópolis, Balsa Nova, Bocaiúva do Sul, Contenda, Dr. Ulysses, Fazenda Rio Grande, Itaperuçu, Pinhais, Tijucas do Sul, Quitandinha.
Cor 4	Agudos do Sul, Campina Grande do Sul, Campo Magro, Cerro Azul, Lapa, Mandirituba, Rio Branco do Sul, Tunas do Paraná.

Quanto à escolha dos vértices (municípios):

- a) Devem ser escolhidos exatamente 10 municípios para compor os vértices do grafo;
- b) A cidade de cor 1 (Curitiba) deve estar presente no grafo;
- c) O grafo deve ter pelo menos duas e no máximo quatro cidades de cor 2;

Quanto às escolhas de arestas (ligações entre municípios):

- a) Os municípios de cor 4 devem estar conectados a alguma cidade de cor 3;
- b) Os municípios de cor 3 devem estar conectados a alguma cidade de cor 2;
- c) A cidade de cor 1 (que é apenas Curitiba) deve estar conectada a pelo menos uma cidade de cada uma das cores restantes;
- d) Os municípios de cor 2 devem estar conectados entre si;
- e) O grafo deve ter no mínimo 13 e no máximo 17 arestas ao final;

Obs.: As regras dos itens (a) à (e) de escolha das arestas dizem respeito à quantidade mínima e máxima de ligações. As equipes podem criar novas arestas conectando duas cidades desde que não violem as regras acima (por exemplo, não há problemas em conectar cidades de cor 3 entre si), contanto que o número de arestas no grafo não ultrapasse as 17.

Requisitos

a) Crie uma representação dos municípios e suas conexões na forma de um grafo, considerando as regras definidas para a escolha dos vértices e suas ligações.

b) O grafo deve ser apresentado graficamente (desenho) e representado através de uma Lista de Adjacência, que será utilizada pelos algoritmos de busca em grafos. A ordem de escolha dos vértices para compor a lista de cada vértice fica a critério da equipe, desde que esta ordem seja respeitada pelos algoritmos.

Seja $G=(V,E)$ o grafo produzido, $V(G)$ o conjunto de vértices do grafo e $E(G)$ o conjunto de arestas. Seja também $u,v \in V(G)$ vértices quaisquer do grafo. A equipe deverá executar os seguintes procedimentos:

c) Busca em Largura (BFS)

Informações para execução do algoritmo de BFS:

(i) o vértice inicial da busca deve ser o vértice de cor 1 (Curitiba)

(ii) os atributos de cada vértice $u \in V(G)$ são:

- $u.visitado$: atributo necessário para o algoritmo de BFS;
- $u.pai$: representando o vértice que é “pai” de u na árvore de busca resultante;
- $u.d$: representando a distância de u para o vértice inicial (Curitiba) no grafo.

(iii) As informações que deverão constar no trabalho sobre cada vértice são: $u.pai$ e $u.d$ para todo $u \in V(G)$ (utilize uma tabela para isto).

(iv) Mostre a árvore de busca resultante ao final desenhando-a “de cima para baixo”, ou seja, se v é pai de u na árvore, então v deve estar mais acima de u , e com uma seta de v para u indicando a paternidade. Ao lado de cada vértice mostre também no desenho os valores de $u.d$, para todo $u \in V(G)$.

(v) Na árvore de busca resultante represente as arestas que fazem parte da árvore (setas) de uma cor e as arestas que não foram percorridas, mas fazem parte do grafo original, de outra cor.

d) Busca em Profundidade (DFS)

Informações para execução do algoritmo de BFS:

(i) o vértice inicial da busca deve ser o vértice de cor 1 (Curitiba)

(ii) os atributos de cada vértice $u \in V(G)$ são:

- *u.visitado*: atributo necessário para o algoritmo de DFS;
- *u.pai*: representando o vértice que é “pai” de u na árvore de busca resultante;
- *prenumb[u]* e *postnumb[u]*: representando os tempos de “entrada” e “saída” em cada vértice durante o algoritmo.

(iii) As informações que deverão constar no trabalho sobre cada vértice são: *u.pai*, e de *prenumb[u]* e *postnumb[u]* para todo $u \in V(G)$ (utilize uma tabela para isto).

(iv) Mostre a árvore de busca resultante ao final desenhando-a “de cima para baixo”, ou seja, se v é pai de u na árvore, então v deve estar mais acima de u , e com uma seta de v para u indicando a paternidade. Ao lado de cada vértice mostre também no desenho o valor de seus *prenumb[u]* e *postnumb[u]*, para todo $u \in V(G)$.

(v) Na árvore de busca resultante represente as arestas de árvore (setas) com uma cor e as arestas de retorno, que não foram percorridas mas fazem parte do grafo original, com outra.

e) Crie a representação tridimensional (3D) do grafo definido em (a) por meio da modelagem 3D. O modelo 3D deve apresentar as cidades e as conexões entre as cidades. Para realizar a modelagem 3D, recomenda-se o uso da ferramenta Blender.

f) Cada cidade deve ser identificada na cena. Por exemplo, o objeto 3D que representa uma cidade pode ser composto pelo nome da cidade, símbolos da cidade, brasão, bandeira, etc.

g) O modelo 3D deve ser armazenado em um único arquivo na extensão .obj.

h) O modelo 3D deve ser carregado e exibido por um visualizador 3D desenvolvido pela equipe. Esse visualizador deve permitir ao usuário realizar as operações de rotação, translação e escala. O visualizador 3D pode ser desenvolvido utilizando as tecnologias Processing, OpenGL ou WebGL.

i) O relatório deve ser organizado conforme a seguinte estrutura:

- Resumo
- Introdução
- Descrição do problema
- Desenvolvimento
 - Representação visual do grafo
 - Representação por Lista de adjacências
 - Modelagem 3D
- Resultados
 - Resultados dos algoritmos
 - Representação visual das árvores de busca resultantes
 - Representação 3D do grafo

- Conclusão
- Referências

Critérios de avaliação

Os trabalhos serão avaliados de acordo com os seguintes critérios objetivos:

Critério	Nota *
Trabalho que não apresentar o relatório no formato de artigo	De 0 a 0.3
Atendimento aos requisitos	De 0 a 0.3
Corretude dos requisitos	De 0 a 0.4
Trabalho entregue fora do prazo.	Metade da nota por dia de atraso

***O total de pontos do Projeto Integrador no 2º bimestre será 1.0 ponto para o 5º e 6º período.**

****Nas disciplinas de Computação Gráfica e Teoria dos Grafos o valor do Projeto Integrador será de 3.0 pontos, ou seja, será adicionado proporcionalmente 2.0 pontos na média bimestral.**

Alguns alunos serão convocados durante a defesa para uma entrevista, na qual deverão explicar o trabalho desenvolvido. A nota individual do aluno dependerá desta entrevista.

Os trabalhos serão avaliados segundo critérios subjetivos definidos pelos professores, variando a nota dentro da faixa de notas definida pelo critério anterior. Alguns dos critérios subjetivos avaliados serão:

- Qualidade da escrita do relatório;
- Criatividade da modelagem 3D.

Referências

1. CONCI, Aura; AZEVEDO, Eduardo; LETA, Fabiana R. Computação gráfica. Rio de Janeiro: Elsevier, 2007.
2. Blender: <https://www.blender.org/>
3. BOAVENTURA P. O. N. Grafos: Teoria, Modelos, Algoritmos. 3. ed. São Paulo: Blucher, 2003.
4. GERSTING, J. L. Fundamentos Matemáticos para a Ciência da Computação, 5ª edição, 2004.

ANEXO II – ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

PROJETO INTEGRADOR - 1º e 2º Período

O trabalho prático consiste em realizar um estudo de caso, demonstrando através de uma pesquisa o comportamento de usuários no acesso à internet e como isso reflete na segurança da informação. Esses programas deverão ser criados utilizando conceitos vistos nas aulas de Redes de Computadores e Segurança e Sistemas de Informação.

A seguir, encontram-se a descrição do problema de pesquisa e a forma de submissão do trabalho e os critérios de avaliação.

- Equipes com 4 ou 5 participantes;

Todo o desenvolvimento da pesquisa cálculos devem ser descritos em um relatório de no máximo 4 páginas, no formato de resumo expandido da Iniciação Científica.

Regras para entrega do trabalho

Data da Entrega:

- 10/10/2017 - Entrega 01: referencial teórico, objetivos, metodologia de pesquisa, e protocolo de pesquisa de metodologia.
- 14/11/2017 - Entrega 02: coleta de dados parametrizada, análise e correções.
- 21/11/2017 - Apresentação do trabalho na aula de Segurança.
- 22/11/2017 - Apresentação do trabalho na aula de Redes.
- 22/11/2017 - Entrega 03: artigo no formato de Iniciação Científica.

Obs.: Recomenda-se que a equipe inicie os trabalhos em campo após a validação do protocolo de pesquisa.

Especificação do Trabalho Prático

O objetivo deste trabalho é realizar uma pesquisa em campo, visando traçar um perfil do comportamento dos usuários da Internet na comunidade acadêmica da Uniandrade. A relevância deste estudo deve refletir na dimensão humana no cenário da segurança da informação e por consequência de sua interação nos ambientes computacionais nos quais está inserido (inclusive nos laboratórios e redes sem-fio na própria instituição). A dimensão humana ainda é o elo mais fraco, responsável por grande parte das vulnerabilidades exploradas por ameaças aos sistemas dos mais diversos tipos. Por esta razão, o trabalho procura abordar o comportamento das pessoas no universo tecnológico descrevendo as ameaças a que estas pessoas se expõem e quais medidas poderiam ser aplicadas,

utilizando conceitos estudados nas disciplinas de redes de computadores e segurança, no sentido de minimizar a exposição à estas ameaças.

O sucesso de um programa de segurança da informação quanto ao uso seguro dos dispositivos computacionais em uma instituição não depende somente de seus gestores e do corpo técnico que proveem o ambiente de rede mas também do comprometimento dos usuários da rede.

Metodologia

A importância da metodologia utilizada em uma pesquisa é justificada devido à necessidade de um embasamento científico adequado, buscando a melhor abordagem para esclarecer as questões da pesquisa.

O objetivo dessa abordagem é descrever os comportamentos de alunos atitudes em relação à utilização dos computadores no laboratório e a segurança de suas informações.

A metodologia será de abordagem qualitativa, que tem como fonte seu ambiente natural e seu principal instrumento é o pesquisador. Segundo Ludke e André (1986, p.11): “a pesquisa qualitativa supõe o contato direto e prolongado do pesquisador com o ambiente e a situação que está sendo investigada, via de regra através do trabalho intensivo de campo”.

O instrumento a ser utilizado nesta pesquisa é o questionário, que é um método de coleta de dados. Ele visa recolher informações se baseando, geralmente, na inquisição de um grupo representativo da população em estudo. Para tal, coloca-se uma série de questões que abrangem um tema de interesse para os investigadores. Segundo Gil (2008, p.121):

pode-se definir questionário como a técnica de investigação composta por um conjunto de questões que são submetidas a pessoas com o propósito de obter informações sobre conhecimentos, crenças, sentimentos, valores, interesses, expectativas, aspirações, temores, comportamento presente ou passado etc

Protocolo de Pesquisa

A pergunta básica da pesquisa é: "o comportamento da comunidade acadêmica em ambientes de Internet é seguro?" Neste sentido, deve-se elaborar 10 perguntas auxiliares que, baseadas em hipóteses, auxiliem o pesquisador a responder a pergunta central. O objetivo é coletar dados de modo a evidenciar uma vulnerabilidade humana, no que se refere a segurança da informação.

Coleta de Dados:

A Coleta de Dados deve ser realizada através da aplicação do questionário. Recomenda-se de que no universo pesquisado incluam-se alunos e docentes na proporção de 80-20. Para que se busque dados que se aproximem da realidade, devem ser aplicados ao menos 20 questionários com pessoas de cursos diferentes, diferentes faixas de idade, etc. Pedimos que não sejam alvos de pesquisa alunos e professores dos cursos de Análise e Desenvolvimento de Sistemas ou Ciência da Computação, pois o conhecimento prévio da causa, pode comprometer o resultado da pesquisa. Os dados brutos devem ser apresentados em formatos de gráficos no trabalho e devem dar sustento a análise a ser desenvolvida na etapa posterior.

Análise de Dados:

Aqui, tanto as respostas quanto as perguntas auxiliares devem ser analisadas e com base nos referências teóricas e nos conteúdo das disciplinas envolvidas servir de subsídio a resposta da pergunta central, propondo se for o caso, comportamentos mais adequados aos objetivos da segurança da informação.

Conclusão:

Conclusão do trabalho propondo, se for o caso, estudos futuros e ações que visem mitigar os problemas encontrados na pesquisa.

Critérios de avaliação

Os trabalhos serão avaliados de acordo com os seguintes critérios objetivos:

Critério	Nota*
Entrega 1 de acordo com o proposto aderente ao tema	De 0 a 0.5
Entrega 2 de acordo com o proposto aderente o tema	De 0 a 0.5
Entrega 3 de acordo com o prosto aderente ao tema e formato	De 0 a 0.5
Apresentação, Defesa do trabalho e arguição	De 0 a 0.5
Trabalho entregue fora do prazo.	Metade da nota por dia de atraso

*** O total de pontos do Projeto Integrador no 2º bimestre será 2.0 pontos para o 1º e 2º período.**

Alguns alunos poderão ser convocados durante a defesa para uma entrevista para explicar detalhes do trabalho. A nota individual do aluno dependerá desta entrevista.

Os trabalhos serão avaliados segundo critérios subjetivos definidos pelos professores relacionados a qualidade do trabalho científico (referencial teórico bibliografias, apresentação, erros de comunicação e expressão, erros de análise relacionada aos temas das disciplinas, correção, etc...), variando a nota dentro da faixa de notas definida pelo critério anterior.

Referências

- GIL, Antonio de Loureiro. Segurança em informática: ambientes mainframe e de microinformática segurança empresarial e patrimonial. 2.ed. São Paulo: Atlas, 1998.
- GALVÃO, Michele Costa. Fundamentos de Segurança da Informação. 1 ed. São Paulo, Pearson Education do Brasil, 2015
- HERSENT, Olive. Telefonia IP. 1ed. São Paulo. Pretince Hall,2002.
- KORUSE, James F. ROSS, Keitch. Redes de Computadores e a Internet- Uma Nova Abordagem. 1ed. São Paulo Editora Addison Wesley.2003
- LIMA FILHO, Eduardo Correia. Fundamentos de Redes de Cabeamento Estruturado. 1 ed. São Paulo, Pearson Education do Brasil, 2014
- LÜDKE, Menga; ANDRÉ, Marli. Pesquisa em educação: abordagens qualitativas. São Paulo: EPU, 1986
- SOARES, Luiz Fernando Gomes; LEMOS, Guido; COLCHER, Sérgio. Redes de computadores: das lans, mans e wans às redes atm. 2.ed. Rio de Janeiro: Campus, 1995.
- STALLING, Wiliam. Criptografia e Segurança de Redes. 4 ed. São Paulo. Editora Pearson Pretice Hall, 2008
- ROQUE, Catia Aparecida. O Projeto HoneyNet. 1 ed. São Paulo. Editora Pearson Pretice Hall, 2008
- TANENBAUM, Andrew S. Redes de computadores. Rio de Janeiro: Campus, 1999.

Projeto Integrador – 3º e 4º período

Valor: 2.0 pontos na Disciplina de Programação
2.0 pontos na Disciplina de Estrutura de Dados
2.0 pontos na Disciplina de Banco de Dados
1.0 ponto para o Projeto Integrador

O trabalho prático da disciplina consiste em desenvolver um programa utilizando a linguagem de programação Java Orientado a Objetos, Estrutura de Dados e Banco de dados.

A seguir, encontram-se a descrição do problema, a forma de submissão do trabalho e os critérios de avaliação.

- Equipes de 3 a 5 participantes;
- O projeto deve ser criado no Eclipse;
- O programa deve compilar e executar com sucesso;
- Utilizar orientação à objetos;
- Organizar as classes em pacotes;
- Utilizar Java 7;
- Alguns alunos poderão ser chamados ao acaso durante a defesa para uma entrevista com a professora, na qual deverão explicar trechos de seus programas. A nota do aluno dependerá desta entrevista.

Todo o desenvolvimento do trabalho e resultados obtidos devem ser descritos em um relatório de no máximo 4 páginas, no formato de resumo expandido da Iniciação Científica.

Regras para entrega do trabalho

Data da Entrega:

- ✓ **19/11/2017** – Entrega do trabalho por e-mail até 23:59h
- ✓ **20/11/2017** – Apresentação no Laboratório/Defesa do código (1ª e 2ª aula) na disciplina de Programação
- ✓ **21/11/2017** – Apresentação no Laboratório/Defesa do código (1ª e 2ª aula) na disciplina de Estrutura de Dados
- ✓ **22/11/2017** – Apresentação no Laboratório/Defesa do código (1ª e 2ª aula) na disciplina de Banco de Dados

O código-fonte da solução deverá ser compactado (adicionar o projeto com o código-fonte do cliente e do servidor no mesmo arquivo zip) e entregue por e-mail (anexo ao e-mail) para o endereço mariana.silva@uniandrade.edu.br, camile.bordini@gmail.com e tiagocandido@hotmail.com.

O nome do zip deverá ser “**ProjetoIntegrador.zip**”.

Serão aceitos trabalhos entregues até as 23h59 da data limite. O assunto do e-mail deverá ser o seguinte:

Projeto Integrador:<nome1>:<nome2>:<nome3>....

O termo “<nome...>” deverá ser substituído pelos nomes dos alunos em ordem alfabética – somente um nome e um sobrenome de cada aluno.

Especificação do Trabalho Prático

1. Descrição do problema

O aluno trabalhará em um sistema de vendas que possui gerenciamento de estoque. Nele, uma das entidades (entidade: venda) terá sua interface implementada com recursos semelhantes a de um CRUD, mas com ênfase nas operações de estruturas de dados. Complementar às interfaces, por meio de recursos de banco de dados o sistema fará o controle completo do estoque, gerenciando as saídas e automatizando as reposições.

2. Requisitos

O sistema deverá possuir os requisitos abaixo:

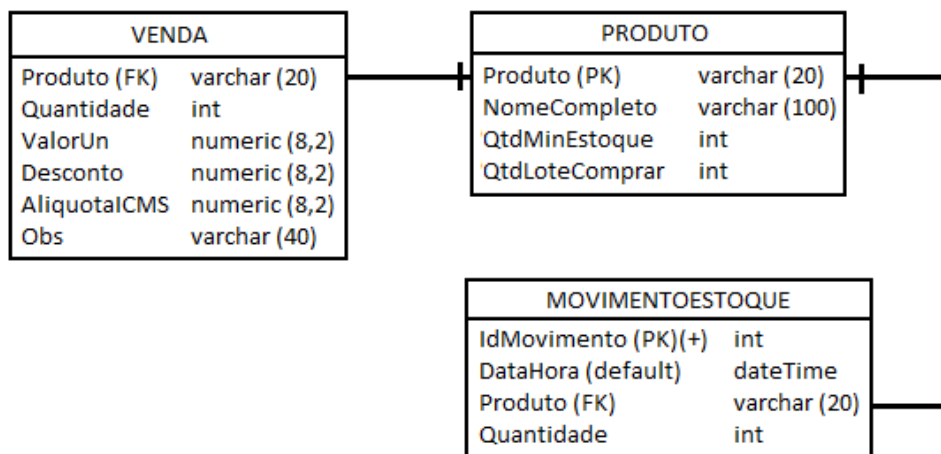
5. Cadastrar Venda:

- a. O cadastro de uma venda é representado pelo preenchimento dos seis atributos presentes na entidade VENDA no Diagrama de Entidades e Relacionamentos.
 - b. A entrada das informações será efetuada por leitura do teclado (utilizando classe Scanner, JOptionPane ou afins).
 - c. Os cadastros serão armazenadas em uma estrutura de Fila (enfileiradas), que deverá ser desenvolvida pelo aluno (não usar a classe Queue nativa do Java); a classe deverá se chamar Fila.java.
6. Transpor a estrutura para o banco de dados: ao utilizar a opção Desinfileirar a estrutura contendo as vendas deverá ser cadastrada no Banco de Dados via jdbc. Para isso será utilizada uma PROCEDURE chamada spInsereVenda.
7. Realizar a Leitura da Fila, bem como as operações de: Busca do Primeiro Elemento; Exibir o tamanho da Fila e Mostrar o resultado de uma consulta no banco de dados.
8. Deverá ser criado um menu no sistema que mostre as opções relacionadas aos requisitos, contendo no mínimo as opções:

```
1 - Realizar venda (enfileirar)
2 - Gravar dados na Base de Dados (desenfileirar)
3 - Mostrar Fila
4 - Buscar Primeiro Elemento da Fila
5 - Mostrar Tamanho da Fila
6 - Mostrar Itens cadastrados na Base de Dados
0 - Sair
```

2.1. Banco de Dados

O modelo lógico proposto deverá ser implementado em banco de dados Sql Server e o script (modelo físico) deverá ser entregue junto do trabalho escrito.



A tabela de PRODUTO deve ser previamente alimentada com pelo menos cinco produtos (sugestão: sal, manteiga, leite, bolacha, trigo) e as tabelas de VENDA e MOVIMENTOESTOQUE serão alimentadas pelo software durante o uso / apresentação aos professores.

A tabela MOVIMENTOESTOQUE guarda toda movimentação de venda (negativa) e de reposição de estoque (positiva). O saldo atual de um produto em estoque é representado pelo somatório das quantidades de todos os registros desta tabela.

Ao acionar uma procedure **spInsereVenda** (opção: 2 – Desinfileirar) a venda será inserida na tabela VENDA e disparará uma trigger **trMovimentaEstoque** que inserirá uma movimentação de estoque negativa. Imediatamente uma segunda trigger vinculada na tabela MOVIMENTOESTOQUE (**trVerificaNecessidadeReposicao**) será disparada para avaliar o estoque atual do produto e decidir (baseado no campo QtdMinEstoque do cadastro do produto) se uma reposição é necessária; caso o valor mínimo tenha sido atingido, uma COMPRA / REPOSIÇÃO acontecerá e será representada no sistema pela inserção pela própria trigger de uma movimentação positiva com a quantidade informada no campo QtdLoteComprar da tabela PRODUTO. Exemplo:

Produto: Pão QtdMinEstoque: 10 QtdLoteComprar: 50 Saldo Estoque: 0

Ao registrar uma venda de 5 unidades, o sistema lançará também uma movimentação de - 5 unidades. Como -5 é menor que 10 o sistema inserirá uma movimentação de reposição de +50 e o novo saldo (calculado) será +45. Se uma nova venda de 20 unidade for realizada, uma movimentação de -20 unidades será registrada e o saldo passará a ser +25, o que não desencadeia uma reposição (já que 25 > 10). Ao realizar uma nova venda de 15 unidades, o saldo cai par 10 e gera uma reposição de 50 unidades voltando o saldo para +60, e assim sucessivamente.

Deve ser criada uma consulta que traga a relação de produtos e seus respectivos saldos de estoque. Essa consulta será utilizada para verificar o funcionamento da gestão de estoque. Opcionalmente, ela pode ser colocada no menu da aplicação, sendo o item [7].

O código fonte da procedure, das duas triggers e da consulta de saldos devem obrigatoriamente constar no trabalho.

2.2. Classe Fila

A estrutura que deverá ser usada para implementar a Fila é a de vetor, onde os elementos do vetor serão objetos de uma classe.

Além disso, para considerar a ideia de alocação dinâmica, o programa não deve limitar a quantidade de elementos a serem enfileirados. Para isso, cada vez que a fila estiver cheia, alguma providência deve ser tomada, como por exemplo, criar um novo vetor com o dobro do tamanho do original e mover os elementos da fila para o novo vetor.

a) Métodos mínimos necessários

- Enfileirar
- Desinfileirar
- Mostra Fila
- Retorna o Primeiro Elemento
- Tamanho Fila

b) Métodos auxiliares

- Cria Fila
- Fila Vazia
- Fila Cheia
- Atualiza Fila

Critérios de avaliação

Os trabalhos serão avaliados primeiramente pela execução correta, seguindo os seguintes critérios objetivos:

Critério	Nota
Trabalhos que não compilarem	De 0 a 0.5
Trabalhos que compilarem mas não gerarem resultados corretamente	De 1.0 a 1.5
Trabalhos que compilarem e gerarem as saídas corretamente	De 1.5 a 2.0
Trabalhos entregues fora do prazo	Metade da nota por dia de atraso

Em segundo lugar, os trabalhos serão avaliados segundo critérios subjetivos definidos pela professora, variando a nota dentro da faixa de notas definida pelo critério anterior. Alguns dos critérios subjetivos avaliados serão:

- Legibilidade (nomes de variáveis bem escolhidos, código bem formatado, uso de comentários quando necessário, etc.);
- Consistência (utilização de um mesmo padrão de código);
- Eficiência (sem exageros, tentar evitar grandes desperdícios de recursos);
- Uso de estruturas de dados adequadas.

Referências

1. AMADEU, Claudia Vicci. Banco de Dados. São Paulo: Pearson Education do Brasil, 2014.
2. ASCENCIO, A. F. G.; ARAÚJO, G. S. Estrutura de Dados: algoritmos, análise da complexidade e implementações em Java e C++. São Paulo: Pearson Prentice Hall, 2011.

3. CORNELL, Gary e HORSTMANN, Cay S. Core Java 2. Vol. 1: Fundamentos. Makron Books, 2000.
4. CORNELL, Gary e HORSTMANN, Cay S. Core Java 2. Vol. 2: Recursos Avançados. Makron Books, 2000.
5. DEITEL, H. M. e DEITEL, P. J. Java: Como Programar. Bookman, 2002. (Código 004.43 / D324j na Biblioteca)
6. ELMASRI, Ramez; NAVATHE, Shamkant B. Sistemas de Banco de Dados. 6. ed. São Paulo: Pearson, 2011.
7. HSQL: <http://hsqldb.org/doc/guide/ch01.html>
8. PUGA, S.; RISSETTI, G. Lógica de programação e estrutura de dados: com aplicações em JAVA. 2 ed. São Paulo: Pearson Prentice Hall, 2009.
9. Site da documentação do Java / Oracle:
<http://docs.oracle.com/javase/7/docs/api/allclasses-noframe.html>