

Universidade do Estado de Santa Catarina - UDESC

Curso: **Sistemas de Informação**

Disciplinas:

Banco de Dados II - Prof. Osmar de Oliveira Braz Júnior

Estrutura de Dados - Prof. Geraldo Menegazo Varela

Programação II - Prof. Fernando dos Santos

Nome: _____ Data: ____/____/____

Trabalho Interdisciplinar 4 – Back Office e Front End

1. Tema

Desenvolver um sistema em Java para *desktop* para atender a necessidade de um comércio. Um sistema é um conjunto de aplicações que dão suporte ao negócio de uma empresa. O sistema deve possuir uma aplicação de *Back Office* (Retaguarda) e uma aplicação de *Front End* (Frente de Caixa). A retaguarda serve para alimentar o sistema com dados para atender a frente de caixa. A frente de caixa tem a função de abrir os pedidos e lançar todos os produtos comprados por um cliente no comércio.

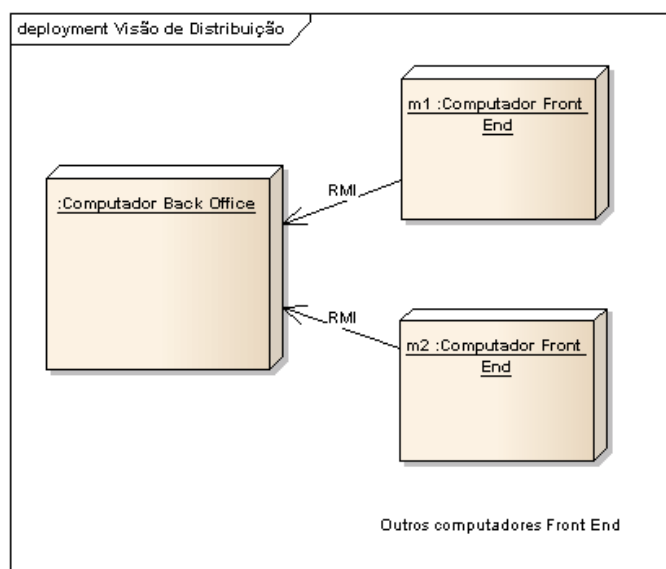


Figura 1 – Diagrama de Distribuição do Sistema

A aplicação deve ser baseada em *JFrame* e atender um ramo de comércio com Supermercado, Vestuário, Farmácia, Eletrônicos, Móveis e Material de Construção. Outras áreas podem ser sugeridas aos professores. O sistema irá trabalhar com diversas fontes de dados. Cada uma é especificada de acordo com a necessidade e ambiente.

Negócio:	Tabelas

A aplicação de retaguarda será responsável pelo cadastro dos itens do negócio a serem comercializados. Todos os relatórios serão extraídos do sistema de retaguarda bem como o cadastro dos usuários, preços dos itens e promoções que venham a existir. A aplicação de retaguarda deve ser capaz de trabalhar com duas fontes de dados de tecnologias diferentes, portanto é necessário utilizar um padrão de

projeto para isolar as fontes. Toda a configuração da escolha da fonte e configuração da persistência deve ser feita em arquivo texto para não ser necessário compilar a aplicação.

Na frente de caixa será realizado login do operador responsável lançar os itens a serem vendidos. Após o operador se identificar, o sistema apresenta a opção de abrir o caixa. Na abertura do caixa este deve se conectar com a retaguarda para importar os dados de produtos e promoções. A conexão com a retaguarda para importação deve ser feita via invocação de métodos remotos (RMI). Com os dados importados, a frente de caixa fica disponível para a abertura de pedidos e lançamentos de itens. Os dados da frente de caixa de produto e pedido ficam armazenados em arquivos binários encadeados, para que não haja necessidade de um SGBD. Quando o período de trabalho do operador termina este deve fechar o caixa e enviar à retaguarda os pedidos lançados por ele. A conexão com a retaguarda para envio dos pedidos deve ser feita também através de RMI. E estes pedidos após enviar são removidos da frente de caixa. Deve ser possível pela retaguarda identificar qual caixa lançou um pedido. Para tal deve existir um cadastro dos caixas.

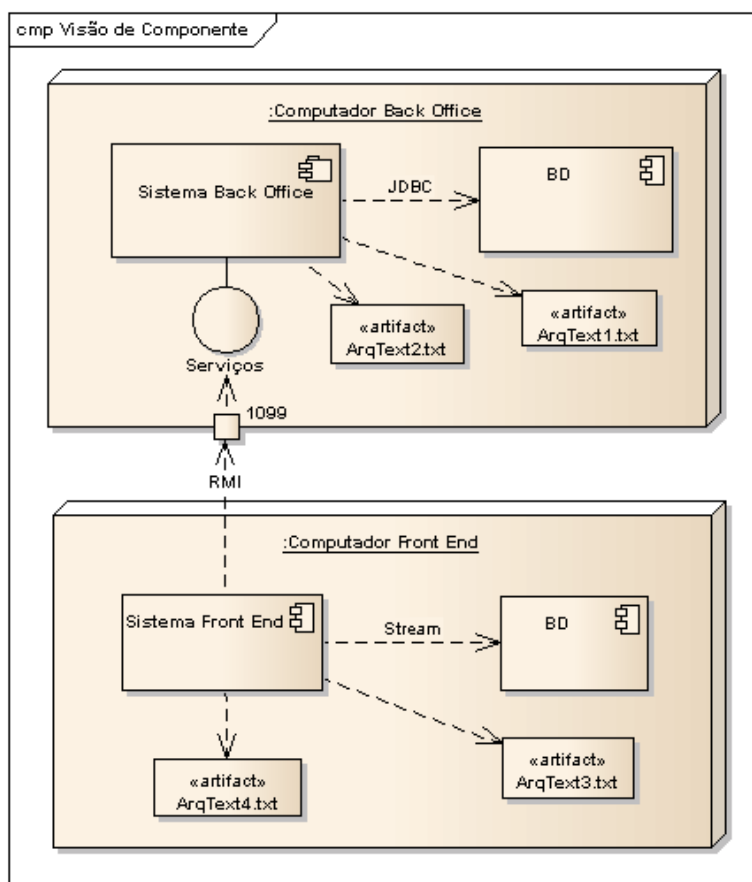


Figura 2 – Diagrama de Componentes do Sistema

2. Escrito

- Devem seguir o modelo de projeto interdisciplinar.

3. Objetivos

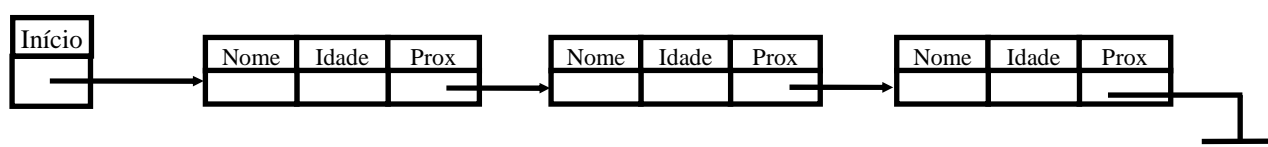
- Desenvolver a habilidade de modelar e implementar sistemas de software utilizando Orientação a Objetos e Banco de Dados.
- Desenvolver a habilidade de modelar e implementar sistemas em camadas.
- Introduzir conceitos de engenharia de software no projeto de sistemas de software.
- Desenvolver a habilidade de construir aplicações utilizar interface gráfica.
- Conhecer e utilizar padrões de projeto para trabalhar com diversas fontes de dados.
- Conhecer e utilizar frameworks de persistência de dados.
- Aplicar Linguagem SQL DDL e DML no desenvolvimento de software.
- Analisar as vantagens e desvantagens de se utilizar Banco de Dados.

- Servir de comparação entre o armazenamento em Banco de Dados Relacional, Frameworks e com Arquivos.
- Integrar os conteúdos das disciplinas de Banco de Dados II, Estrutura de Dados e Programação II.

4. Considerações sobre o a implementação:

- Usar Orientação Objetos.
- Não gerar o campo chave de forma automática. Para poder testar a ordem e a integridade dos campos chaves e relacionados no arquivo binário encadeado.
- Utilizar duas formas de armazenamento na retaguarda.
- Utilizar o padrão de projeto *Abstract Factory* para separar as implementações das duas formas de armazenamento (<http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>).
- Desenvolver em camadas as duas aplicações
 - Interface deve conter somente código para da interface com o usuário,
 - Controle deve conter somente código de relação entre interface e entidade,
 - Entidade somente código referente ao negócio,
 - Objetos de Acesso a Dados somente código de recuperação de dados de mecanismo persistente.
- Os relatórios devem baseados em *JasperReport*
- Organizar os arquivos da aplicação em pacotes.
- Separar os arquivos de extensão .java dos .class.
- A execução da aplicação deve ser feita sem a IDE.
- A aplicação deve estar empacotada para facilitar a instalação e execução.
- Serão utilizado arquivos textos para configurar as aplicações.
 - Retaguarda
 - Para a retaguarda utilizar arquivo texto 1 para especificar a fonte de dados do **Abstract Factory**.
 - As informações de configuração do banco de dados como, usuário, senha, driver, url devem ser armazenadas e lidas do arquivo texto 2.
 - Frente de Caixa
 - As informações de configuração do arquivo binário como nome do arquivo e endereço devem ser armazenadas e lidas do arquivo texto 3.
 - O endereço da retaguarda deve ser armazenada em lidas do arquivo texto 4.
- A comunicação entre frente de caixa e retaguarda deve ser através da tecnologia RMI (*Remote Method Invocation*). Para aprender como funciona o RMI, os alunos devem consultar o capítulo 18 do livro “Use a cabeça! Java”, de Kathy SIERRA e Bert BATES. Este livro está disponível na biblioteca do CEAVI.
- Para a frente de caixa utilizar encadeamento em Arquivo Binário
 - É necessário registro de tamanho fixo para a implementação dos objetos.
 - O início precisa estar definido em um arquivo específico que guarda o valor do registro inicial da lista.
 - Uma vez lido o início a utilização do programa é igual a uma lista simplesmente encadeada, os dados do arquivo não devem ser carregados totalmente para a memória no máximo 2 registros estarão disponíveis de cada vez. Isto para evitar que todos os objetos sejam alocados em memória e posteriormente descarregados em disco.

Representação da estrutura



- Não esquecer de adicionar aos dados a serem armazenados o campo próximo que irá indicar onde está o próximo registro no arquivo, encerre a lista em arquivo com o valor -1 no campo próximo.

Representação Exemplo dos Arquivos Binários

AlunoInicio.dat

inicio		id	descritivo	cidade	prox
1	0	30	João	40	4
	1	10	Maria	40	3
	2	50	Fernando	30	-1
	3	20	Antônio	10	0
	4	40	Guilherme	30	2

Aluno.dat(Arquivo Binário 1)

CidadeInicio.dat

inicio		id	descritivo	prox
2	0	40	Itajaí	-1
	1	20	Blumenau	3
	2	10	Içara	1
	3	30	Jaguaruna	0

Cidade.dat (Arquivo Binário 2)

- Para os arquivos binário implementar as seguintes regras:
 - Aplicar os conceitos de manipulação de listas encadeadas ao manipular e manter os arquivos binários,
 - Realizar Inclusão ordenada utilizando o campo chave,
 - Exclusão pode ser feita de forma lógica,
 - Impedir a duplicidade do campo chave,
 - Validar campos de preenchimento obrigatório,
 - Validar domínios,
 - Manter a integridade referencial entre os objetos.

5. Artefatos a serem Entregues

- **Trabalho escrito impresso.**
- Mídia com o trabalho escrito e código fonte.

6. Data de Entrega

- O trabalho será entregue e apresentado no dia 12 e 14/11, a ordem será através de sorteio.
- No dia 17/10 será realizada uma prévia do trabalho. Neste dia o trabalho deve ser mostrado ao professor a fim de avaliar o desenvolvimento.
- A entrega somente será aceita se atender os itens anteriores.

7. Avaliação

A avaliação se dará pela prévia do trabalho(10%), apresentação (30%) e escrito e implementação (60%). Cada aplicação vale 50% na nota da implementação, onde:

Na prévia do trabalho serão analisados: Análise do Tema e Contextualização

Na apresentação: Clareza de idéias e argumentação, uso dos recursos e tempo.

No trabalho escrito/implementação: Atender os itens solicitados no modelo do trabalho, consistência e integridade dos modelos.

Na implementação: Requisitos devidamente modelados e implementados e funcionais.

Somente a implementação, escrito ou a apresentação não será considerado.

O trabalho deve implementar o máximo dos requisitos do projeto do software.

7. Observações

Os alunos que fazem somente a disciplina de **Banco de Dados II** devem implementar a forma de armazenamento com SGBDs tanto para a frente de caixa como para a retaguarda. Para esta situação a interface pode ser feita utilizando *JOptionPane*.

Os alunos que fazem somente a disciplina de **Estrutura de Dados** devem implementar a forma de armazenamento com Encadeamento em Arquivo Binário para a Frente de Caixa e XML para retaguarda. Para XML pode utilizar alguma biblioteca. Nesta situação a interface pode ser feita utilizando *JOptionPane*.

E quem faz somente **Programação II** deve implementar a forma de armazenamento com JDBC e JPA para a frente de caixa e retaguarda.

Os alunos que fazem somente duas das disciplinas devem utilizar a persistência especificada anteriormente.

Para os alunos quem fazem todas as disciplinas do **Banco de Dados II**, **Estruturas de Dados** e **Programação II** devem implementar persistência para retaguarda via JDBC e JPA e para a frente de caixa em Arquivo Binário encadeado. Neste caso a interface deve ser baseada em *JFrame* ou similar.

Para todas as situações o padrão de projeto *Abstract Factory* deve ser seguido e implementado tanto para a frente de caixa como retaguarda segundo a referência.