

Universidade do Sul de Santa Catarina - UNISUL  
Unidade Curricular: **Sistemas Distribuídos e Mobile – Turma A**  
Professores: Osmar de Oliveira Braz Júnior

### A3 – Desempenho de compreensão

Meta de compreensão relacionada: **Projetar e desenvolver sistemas com arquiteturas baseadas em serviços.**

O trabalho consiste **em projetar e desenvolver um sistema utilizando serviços**. O trabalho será realizado em grupo de 4 alunos (não será aceito trabalho individual ou dupla). A escolha dos grupos é de responsabilidade dos alunos. Você pode utilizar a aplicação desenvolvida em Java da UC de Programação de Soluções Computacionais ou desenvolver a aplicação abaixo.

#### 1. Contextualização

Toda empresa de comércio, possui um estoque, no qual armazena os seus produtos. O controle eficiente do estoque permite que a empresa tenha um planejamento melhor de suas compras, na medida em que tem condições de saber quanto tem de cada produto, sabendo assim, quanto deverá comprar para repor suas necessidades e atender bem a demanda de seus clientes.

O controle do estoque é feito através da atualização das quantidades disponíveis de cada um dos itens do estoque. Isso é feito através de um levantamento de todos os produtos com os quais a empresa trabalha. Após o levantamento, todos os produtos serão cadastrados no sistema (operações de inclusão, alteração, consulta e exclusão de produtos), para que a partir desse momento, o sistema reflita exatamente o que existe no estoque da empresa. Os produtos são classificados em categorias, por exemplo limpeza, enlatados, vegetais, óleos, etc... Uma funcionalidade importante é permitir reajustar os preços de todos os produtos em um determinado percentual.

A cada nova entrada ou saída de produtos do estoque (movimentação) ocorre em uma data, o saldo atual do produto é atualizado (i.e., saída subtrair e entrada adicionar). No momento da saída é importante avisar que a quantidade do produto está abaixo da quantidade mínima para que se providencie a compra do produto para o estoque. E se estiver dando uma entrada avise se a quantidade está acima da quantidade máxima para que não se compre mais deste produto. Não é necessário cadastrar previamente os clientes ou fornecedores.

Para o comerciante, existem cinco **relatórios básicos**:

- Lista de Preços: uma relação de todos os produtos do estoque em ordem alfabética, constando do preço de cada um dos deles, unidade medida e categoria.
- Balanço Físico/Financeiro: uma relação de todos os produtos do estoque, em ordem alfabética, constando da quantidade disponível de cada um dos produtos, o valor total de cada um dos produtos (valor unitário \* a quantidade em estoque) e o valor total do estoque (somatório dos totais de cada produto).
- Relatório dos produtos que estão abaixo da quantidade mínima contendo o nome do produto, a quantidade mínima e a quantidade em estoque.
- Relatório da quantidade de produtos por categoria contendo o nome da categoria e quantidade de produtos distintos.
- Relatório do produto que mais teve saída e mais teve entrada.

As informações de um produto são:

- Nome
- Preço unitário
- Unidade
- Quantidade em estoque
- Quantidade mínima em estoque
- Quantidade máxima em estoque

- Categoria

As informações da categoria são:

- Nome
- Tamanho (Pequeno, Médio, Grande)
- Embalagem (Lata, Vidro, Plástico)

As informações da movimentação são:

- Nome do produto
- Data da movimentação
- Quantidade movimentada
- Tipo da movimentação (Entrada ou Saída)

O trabalho é um desenvolvimento contínuo ao longo do semestre, portanto os professores podem lançar *issues* nos repositórios do trabalho. Estes *issues* precisam ser sanados até a entrega do trabalho. Estes *issues* se não resolvidos podem representar descontos no trabalho.

## **2. Tema**

O sistema deve ser desenvolvido em duas camadas de forma distribuída utilizando java. Portanto deve ser elaborado dois projetos na IDE e dois repositórios no github, um para a camada que fornece o serviço o *back-end* e outra para a camada que consome o serviço o *front-end*. Desta forma a camada de serviço deve ficar em um computador e a camada que consome o serviço em outro.

A aplicação de *front-end* envia mensagens para o servidor de *back-end* que envia de volta os resultados. A aplicação de *front-end* exibe os resultados. A aplicação de *back-end* localizado no servidor, recebe as mensagens do *front-end*, realiza o processamento e retorna ao *front-end*. O servidor de banco de dados pode ser instalado no servidor de *back-end* ou outro servidor.

A conexão do *front-end* com o *back-end* pode ser qualquer de através de qualquer tecnologia de distribuição apresentada em aula, ex. RMI, Socket ou WebServices.

## **2. Objetivos**

- Desenvolver a habilidade de projetar e desenvolver sistemas utilizando serviços.

## **3. Documentação (7,0 pontos)**

Devem constar do trabalho escrito pelo menos os seguintes tópicos: (0,5 pontos)

- 1) Atender as normas do caderno de metodologia da UNISUL modelo TCC, disponível em: <https://www.unisul.br/sou-estudante/biblioteca/>
- 2) Devendo incluir no **mínimo**: Capa, Sobrecapa, Sumário, Introdução, Desenvolvimento, Conclusão e Referências.
- 3) Devem constar no escrito:
  - Introdução
  - Desenvolvimento
    - Descrição do ambiente de desenvolvimento
    - Requisitos funcionais e não funcionais
    - Arquitetura do sistema
    - Padrões de projeto utilizado no sistema
    - Distribuição das tarefas entre os membros da equipe. Criar uma tabela com 3 colunas. A primeira coluna com o nome do aluno, a segunda coluna com a funcionalidade e a terceira com link(s) do(s) arquivo(s) criados ou modificado(s) para atender a funcionalidade.
    - Links dos projetos no Github
  - Conclusão

Avaliação da **implementação** irá considerar os seguintes requisitos e sua execução:

- 1) Implementação da funcionalidade que produza e consuma o serviço CRUD **produto** (1,5 ponto)
- 2) Implementação da funcionalidade que produza e consuma o do serviço CRUD **categoria** (1,5 ponto)
- 3) Implementação da funcionalidade que produza e consuma o do serviço CRUD **movimentação(entrada/saída)** (1,5 pontos)
- 4) Implementação da funcionalidade que produza e consuma o serviço dos **Relatórios** (1,0 pontos)
- 5) Organização e padronização do código fonte (Padronização de pacotes, tabulação, atributos, métodos, comentários) (1,0 ponto)
- 6) Desenvolvimento Colaborativo (Github) (0,5 pontos)
  - Links dos projetos no Github
  - Usar Branch main.
  - Estrutura do repositório (licença, readme, src).
  - Convenção das mensagens de commits.

**Atenção:** A avaliação da distribuição das tarefas de desenvolvimento (código fonte) de forma igualitária entre os membros do grupo será realizada através dos commits dos integrantes do grupo no seu repositório. Commits de arquivos que não sejam código fonte (ex.: Readme, pom.xml, e outros) não serão considerados. Cada integrante deve utilizar sua conta pessoal no GitHub para realizar as alterações. Somente será considerado o código fontes da última versão do programa no repositório. Commits em arquivos inexistentes na última versão do software não serão considerados. Pontos serão atribuídos ou removidos de cada aluno de acordo com os itens da implementação de 1 a 5. É de responsabilidade do grupo acompanhar e registrar todas as alterações no repositório através de commit. Um histórico de commits saudável deve ser organizado, compreensível, atômico e rastreável, permitindo que qualquer pessoa — inclusive o próprio desenvolvedor no futuro — entenda claramente o que mudou, por que mudou e quando mudou, sem precisar adivinhar. Para isso, é essencial que apresente alterações frequentes e granulares, evitando “megacommits” que concentrem diversas modificações não relacionadas em um único registro. Cada commit deve ter uma finalidade bem definida, como corrigir um bug no login ou adicionar um teste unitário para uma função específica, pois essa abordagem mantém o repositório organizado, facilita a compreensão da evolução do código e possibilita a reversão de mudanças problemáticas de forma simples e segura.

#### **4. Apresentação (3,0 pontos)**

**Todas as equipes** devem **gravar um vídeo da apresentação do trabalho**. **Todos** os integrantes devem aparecer no vídeo e participar da apresentação do trabalho. **Deve** ser utilizado algum software de apresentação tipo PowerPoint. Todos os grupos têm o tempo médio é de 10 minutos (variação de 2 minutos) para a apresentação.

No mínimo devem constar na apresentação os seguintes tópicos:

- Capa
- Sumário
- Descrição do sistema/arquitetura/padrões
- Descrição dos serviços *backend/frontend*
- Demonstração das funcionalidades com o software em execução
  - Cadastro/Gerenciamento de produtos
  - Cadastro/Gerenciamento de categoria
  - Movimentação (Entrada/Saída)
  - Relatórios
- Conclusão

**Importante:** O vídeo deve ter qualidades mínimas de luz e som, possibilitando a identificação visual do(s) aluno(s), a compreensão das falas e práticas. O vídeo deve **iniciar** pela apresentação de todos os alunos com sua identificação(nome), semestre e curso (0,5 pontos). Como é um trabalho em grupo **todos** os integrantes devem apresentar parte do conteúdo, portanto dividam o tempo de forma igualitária entre

os membros. Cada integrante deve **demonstrar parte da execução do software** (1,5 pontos). A apresentação deve seguir uma sequência lógica e coerente no tempo estabelecido (1,0 pontos). O vídeo pode ser editado para unir as falas dos integrantes do grupo.

**Socialização:** O vídeo deve ser postado no Youtube como **não listado** e o link enviado ao professor juntamente com a postagem do trabalho para sua socialização.

## **5. Artefatos a serem entregues**

**Todos** os alunos devem fazer a entrega dos artefatos do trabalho. Será descontando **0,5 pontos** se o aluno não fizer a postagem no prazo e local especificado(ulife). Postagem de arquivos não solicitados equivale a não postar o trabalho. Os artefatos a serem entregues são.

- Arquivo do trabalho (PDF)
- Arquivo com slides da Apresentação (PDF),
- Arquivo texto com o link do vídeo da Apresentação.

**Compacte** os arquivos a serem enviados em um arquivo .zip ou .rar para realizar uma única postagem. Os alunos e o grupo são **responsáveis** pela correta postagem do arquivo da A3 no sistema. Após o prazo final não é possível fazer o envio ou correções.

## **6. Datas**

- Dia **01/09/2025** a A3 é apresentada a turma.
- Até o dia **15/09/2025** o tema do trabalho, os links dos repositórios e o nome dos integrantes da equipe deve ser postado Ulife na seção conteúdo no item “A3 – Grupo”
- Até o dia **10/11/2025** o trabalho deve ser postado no Ulife na seção conteúdo no item “A3 - Postagem do trabalho”.
- Socialização das apresentações dia **17 a 21/11/2025**

## **7. Acompanhamento**

Os professores definirão, em momento oportuno durante as aulas, períodos destinados à orientação e ao acompanhamento do desenvolvimento do trabalho. Para tratar especificamente da A3, o agendamento deverá ser solicitado por e-mail com, no mínimo, 24 horas de antecedência. A presença de todos os integrantes do grupo é obrigatória

## **8. Defesa de Trabalho**

Se necessário o professor pode convocar a defesa do trabalho de forma individual para comprovação da elaboração do trabalho.

**Boa Atividade!**