

AVALIAÇÃO N2 — GERÊNCIA DE CONFIGURAÇÃO DE SOFTWARE

Disciplina: Gerência de Configuração de Software

Sistema: Gestão de Estoque e Caixa

Formato: Documentação técnica (PDF)

Aluno:

1. CONTEXTO DA AVALIAÇÃO

Esta avaliação tem como objetivo aplicar os principais conceitos de **Gerência de Configuração de Software (GCS)** ao Sistema de Gestão de Estoque e Caixa, desenvolvido na disciplina de Construção de Software.

2. O QUE DEVE SER ENTREGUE

Você deve entregar **1 arquivo PDF** contendo **3 documentos**, totalizando **3 a 5 páginas**.

Estrutura da Entrega:

```
N2_GCS_SeuNome.pdf
|
|- DOCUMENTO 1: Itens de Configuração e Estrutura do Projeto
|- DOCUMENTO 2: Baselines e Versionamento
|- DOCUMENTO 3: Rastreabilidade + RFC
```

DOCUMENTO 1 — ITENS DE CONFIGURAÇÃO E ESTRUTURA DO PROJETO

1.1. Itens de Configuração (ICs)

Liste os principais itens do Sistema de Gestão de Estoque e Caixa, organizados nas seguintes categorias:

Backend (Java Spring Boot)

Identifique e liste:

- **Controllers:** Classes que recebem requisições HTTP
- **Services:** Classes com lógica de negócio
- **Entities:** Classes que representam tabelas do banco
- **Repositórios:** Interfaces de acesso a dados
- **Arquivos de configuração:** application.yml, SecurityConfig.java, pom.xml

Frontend (Angular)

Identifique e liste:

- **Componentes:** Telas do sistema
- **Services:** Classes que comunicam com o backend
- **Rotas:** Arquivo de configuração de navegação
- **Guards e Interceptors:** Controle de acesso e requisições

Banco de Dados

Identifique e liste:

- **Scripts SQL ou Migrations:** Arquivos de criação/alteração de estrutura
- **Tabelas principais:** Listar as tabelas do sistema

Documentação

Liste os documentos produzidos:

- Diagramas (Modelo ER, Arquitetura)
- Documentos de requisitos
- Artefatos de GCS

Configurações

- **Backend:** application.properties ou application.yml
- **Frontend:** environment.ts, environment.prod.ts
- **Build:** package.json, pom.xml

1.2. Estrutura do Projeto (Conceitual)

Descreva **como o repositório deveria ser organizado** de forma profissional e lógica.

Você não precisa criar o repositório, apenas descrever a estrutura ideal.

Exemplo de Estrutura:

```
sistema-estoque-caixa/
|
|   -- backend/
|   |   -- src/
|   |   |   -- main/
|   |   |   |   -- java/com/sistema/estoque/
|   |   |   |   |   -- controllers/
|   |   |   |   |   -- services/
|   |   |   |   |   -- repositories/
|   |   |   |   |   -- entities/
|   |   |   |   |   -- config/
|   |   |   |   |   -- resources/
|   |   |   |   |   -- application.yml
|   |   |   |   -- test/
|   |   |   -- pom.xml
|
|   -- frontend/
|   |   -- src/
|   |   |   -- app/
|   |   |   |   -- components/
|   |   |   |   -- services/
|   |   |   |   -- guards/
|   |   |   |   -- interceptors/
|   |   |   |   -- environments/
|   |   |   -- package.json
|
|   -- database/
|   |   -- migrations/
|   |   -- seed/
|
|   -- docs/
|   |   -- requirements/
```

```
design/  
└── architecture/  
└── README.md
```

Descrição Narrativa:

Explique brevemente a organização proposta (2 a 3 parágrafos). Descreva o propósito de cada pasta principal e como ela contribui para a organização do projeto.

DOCUMENTO 2 — BASELINES E VERSIONAMENTO

2.1. Baseline BLO – Inicial

Uma **baseline** é um marco formal onde itens de configuração são aprovados e congelados.

Descreva a **Baseline BLO** (estrutura inicial do projeto).

Deve incluir:

- **Data:** Início do desenvolvimento
- **Estrutura inicial:** Pastas principais criadas
- **Arquivos essenciais:** Arquivos de configuração básicos
- **Primeiro conjunto de ICs:** Estrutura de diretórios, configurações de ambiente, schema inicial
- **Critério de estabilidade:** Como você sabe que a BLO está pronta?

2.2. Baseline BL1 – Funcional

Descreva a **Baseline BL1** (sistema completo com funcionalidades principais).

Deve incluir:

- **Data:** Final do desenvolvimento
- **Funcionalidades concluídas:** Liste as funcionalidades principais implementadas
- **Requisitos entregues:** Quais requisitos funcionais foram atendidos?
- **Critérios de estabilidade:** O que foi verificado para considerar a baseline aprovada?

2.3. Política de Versionamento

Defina uma **política simples de versionamento** para o projeto.

Formato sugerido: Semantic Versioning (MAJOR.MINOR.PATCH)

Regras:

- **MAJOR (X.0.0):** Mudanças grandes que quebram compatibilidade
- **MINOR (0.X.0):** Novas funcionalidades mantendo compatibilidade
- **PATCH (0.0.X):** Correções de bugs e ajustes menores

Exemplos de Versões:

Versão	Descrição	Tipo
v0.1.0	Estrutura inicial (BLO)	Inicial
v1.0.0	Sistema funcional completo (BL1)	MAJOR
v1.1.0	Cadastro de produtos com melhorias	MINOR
v1.1.1	Correção no cálculo de estoque	PATCH

DOCUMENTO 3 — RASTREABILIDADE + RFC

3.1. Matriz de Rastreabilidade

A **matriz de rastreabilidade** conecta requisitos funcionais aos componentes de código.

Crie uma matriz com **3 a 5 requisitos** do sistema.

Modelo de Tabela:

Requisito	Arquivo(s) Relacionado(s)	Observação
Autenticação do usuário	AuthController, AuthService	Requisito principal
Movimentar estoque	EstoqueService	Atualiza quantidade
Registrar venda	VendaService	Baixa estoque

Instruções:

- Escolha **3 a 5 requisitos funcionais** relevantes do sistema
- Para cada requisito, liste os **principais arquivos** relacionados (backend e frontend)
- Adicione **observações** sobre a implementação

3.2. RFC – Request for Change

Uma **RFC (Request for Change)** é um documento formal que registra uma solicitação de mudança no sistema.

Você deve elaborar 1 RFC para o sistema e implementá-la.

A RFC deve conter:

1. **Descrição da mudança:** O que será alterado?
2. **Justificativa:** Por que esta mudança é necessária?
3. **Impacto:** Em quais itens de configuração (1 ou 2 arquivos)?
4. **Riscos:** Quais problemas podem surgir e como mitigar?
5. **Status:** Aprovada

Exemplo de RFC:

RFC-001: Adicionar campo "Categoria" no produto

1. Descrição da mudança:

Adicionar um campo "categoria" (texto) na entidade Produto.

2. Justificativa:

Melhorar a filtragem de estoque e permitir relatórios segmentados por tipo de produto.

3. Impacto nos Itens de Configuração:

- Product.java - adicionar atributo categoria
- ProductController.java - adicionar parâmetro no DTO
- Tabela PRODUCTS - adicionar coluna categoria

4. Possíveis Riscos:

- Produtos antigos ficarão sem categoria (NULL)
- Mitigação: definir categoria padrão "Geral" para registros existentes

5. Status: APROVADA

FORMATO DA ENTREGA

Especificações:

- **Formato:** PDF único
- **Nome do arquivo:** N2_GCS_SeuNome.pdf

- **Extensão:** 3 a 5 páginas
- **Plataforma:** Google Classroom
- **Data limite:** Conforme informado em sala

Dicas para uma boa entrega:

- Use formatação clara (títulos, tabelas, listas)
 - Seja objetivo e direto
 - Contextualize tudo ao Sistema de Estoque e Caixa
 - Evite textos genéricos
 - Revise antes de enviar
-