

DDD Java – Sprint 1 e 2  
Sistema de Gestão de Amostras DASA  
Engenharia de Software – 2º Ano

Feito por

Bruno Carneiro Leão – RM 555563

Paulo Akira Okama – RM 556840

Guilherme da Cunha Melo – RM 555310

Matheus Gushi Morioka – RM 556935

Victor Mattenhauer Lopes Capp – RM 555753

Maio 2025

## 1. Sumário

1. Descritivo do Projeto
  - 2.1. Descrição Geral
  - 2.2. Objetivo do Projeto
  - 2.3. Arquitetura e Estrutura do Sistema
  - 2.4. Funcionalidades Principais
  - 2.5. Principais Classes e suas Funções
  - 2.6. Conclusão
2. Modelagem de Classes UML

## 2. Descritivo do Projeto

### Descrição Geral

O projeto consiste no desenvolvimento de um sistema desktop, utilizando a linguagem Java, que simula um software para gerenciamento de amostras laboratoriais, pacientes, médicos e sedes da empresa DASA.

O sistema permite realizar cadastros de pacientes, médicos, endereços, bem como efetuar o controle e a análise de amostras laboratoriais, incluindo cálculos de volume e verificação se estão dentro dos padrões aceitáveis.

Além disso, o sistema possibilita atualizar informações dos pacientes e endereços, e verificar se determinada amostra é considerada recente com base na data de entrada e no prazo especificado.

### Objetivo do Projeto

Desenvolver um software funcional em Java que atenda aos seguintes requisitos:

- Gerenciar dados de pacientes, médicos e sedes.
- Realizar o cadastro completo dos pacientes e seus respectivos endereços.
- Cadastrar médicos responsáveis pela coleta e análise.
- Registrar amostras laboratoriais e calcular seu volume.
- Verificar se uma amostra ultrapassa um limite de volume definido.
- Verificar se uma amostra é considerada recente (comparando a data de entrada com a data atual).
- Permitir a atualização de informações dos pacientes e dos endereços.

### Arquitetura e Estrutura do Sistema

O sistema foi estruturado com Programação Orientada a Objetos (POO), aplicando conceitos como:

- Herança:
  - Pessoa é classe pai de Paciente e Medico.
  - Logradouro é classe pai de Endereco e Sede.
- Interfaces:
  - AmostraVolume define métodos obrigatórios como calcularVolume() e ultrapassaLimiteVolume().
  - PacienteIdade define a obrigatoriedade de implementar setData\_nasc().
- Encapsulamento:
 

Todas as classes possuem atributos privados e métodos getters e setters para controle de acesso.
- Classe Principal:
 

A classe MainSoftware gerencia toda a interação com o usuário através de interface gráfica simples utilizando JOptionPane.

## Funcionalidades Principais

### Opção Funcionalidade

- 1      Cadastrar um novo paciente, seu endereço e o médico responsável
- 2      Analisar amostras (cálculo do volume e verificação de limite)
- 3      Verificar se uma amostra é considerada recente
- 4      Atualizar informações do paciente (nome, CPF, email e telefone)
- 5      Atualizar informações do endereço do paciente

## Principais Classes e suas Funções

- MainSoftware – Classe principal com o menu e controle do fluxo do programa.
- Paciente – Armazena dados dos pacientes e implementa controle de data de nascimento.
- Medico – Armazena informações dos médicos responsáveis.
- Amostra – Armazena dados da amostra, calcula seu volume, verifica limite e validade (data recente).
- Endereco e Sede – Representam localizações, sendo Endereco para paciente e Sede para unidade de coleta.
- Pessoa e Logradouro – Classes genéricas que servem de base para Paciente/Medico e Endereco/Sede, respectivamente.

- Interfaces:
  - AmostraVolume – Obrigatoriedade de cálculo de volume e verificação de limite.
  - Pacientelidade – Controle da data de nascimento.

## Conclusão

O sistema foi projetado para atender de forma simples e objetiva as demandas de cadastro e análise de amostras laboratoriais. Com uma estrutura bem definida de Programação Orientada a Objetos, o projeto permite expansão futura, podendo ser integrado a banco de dados ou interfaces gráficas mais robustas.

## 3. Modelagem de Classes UML

