

# Especificação de Requisitos: Gerador de Dados Sintéticos Flexível

Cristhian Eduardo Kapelinski de Avilla      Rafael da Silva Moral  
Lucas Correa Rodrigues

14 de outubro de 2025

## 1 Introdução

### 1.1 Objetivo do Documento

Este documento especifica os requisitos funcionais e não funcionais para a primeira versão (MVP - Mínimo Produto Viável) da ferramenta de geração de dados sintéticos. Seu propósito é guiar o desenvolvimento, garantindo que a equipe se concentre nos recursos essenciais para um lançamento rápido e funcional.

### 1.2 Escopo do Projeto

O software proposto é uma ferramenta para gerar arquivos CSV sintéticos com alta flexibilidade. O foco principal é permitir que o usuário defina a estrutura do arquivo e gere dados baseados em padrões de expressões regulares (regex) e distribuições estatísticas. Destina-se a desenvolvedores, testadores, analistas de dados e pesquisadores que necessitam de dados sintéticos para testes de software, análise de dados ou simulações.

## 2 Requisitos Funcionais

**RF01:** O sistema deve permitir que o usuário defina o número de linhas, o número de colunas e os nomes das colunas para o arquivo CSV a ser gerado.

**RF02:** O sistema deve permitir que o usuário especifique uma expressão regular (regex) para gerar dados textuais em uma coluna.

**RF03:** O sistema deve ser capaz de gerar dados numéricos em uma coluna seguindo uma distribuição estatística gaussiana (normal), com média e desvio padrão configuráveis pelo usuário.

**RF04:** O sistema deve exportar os dados gerados para um arquivo no formato CSV em um diretório local.

**RF05:** O sistema deve validar as configurações de entrada antes da geração dos dados para evitar erros, como uma expressão regular com sintaxe inválida.

**RF06:** O sistema deve ser capaz de gerar dados numéricos que sigam uma tendência linear (ex:  $y = mx + b$ ).

**RF07:** O sistema deve permitir a combinação de diferentes tipos de geradores em colunas distintas no mesmo arquivo CSV (ex: uma coluna com regex e outra com dados gaussianos).

**RF08:** O sistema deve ser extensível para suportar outras distribuições estatísticas, como uniforme ou exponencial.

**RF09:** O sistema deve permitir a configuração de aspectos estruturais do arquivo de saída, como o caractere delimitador de campo e o separador decimal.

**RF10:** O sistema deve oferecer uma funcionalidade de visualização básica dos dados gerados, como um histograma para uma distribuição gaussiana.

**RF11:** O sistema deve permitir a importação de um arquivo CSV existente para análise e detecção de padrões.

**RF12:** O sistema deve ter a capacidade de simular imperfeições de dados do mundo real, como a injeção controlada de outliers e a geração de valores ausentes.

**RF13:** O sistema deve suportar a exportação dos dados para múltiplos formatos além de CSV, como XLSX e JSON.

### 3 Requisitos Não Funcionais

**RNF01:** (**Usabilidade**) O sistema deve possuir uma interface amigável e intuitiva, seja ela via linha de comando ou GUI, com documentação clara.

**RNF02:** (**Desempenho**) O sistema deve ser capaz de gerar arquivos CSV com até 1 milhão de linhas em menos de 5 minutos em um hardware padrão.

**RNF03:** (**Portabilidade**) O sistema deve ser executável em múltiplas plataformas, como Windows, Linux e macOS.

**RNF04:** (**Confiabilidade**) A taxa de erro na geração de dados deve ser inferior a 1%, garantindo que todos os dados gerados correspondam perfeitamente aos padrões de regex ou parâmetros estatísticos definidos.

**RNF05:** (**Segurança**) O sistema não deve armazenar dados sensíveis e deve garantir que o uso de expressões regulares não permita injeções maliciosas.

**RNF06:** (**Manutenibilidade**) O código-fonte deve ser modular e acompanhado de testes unitários para as principais funções de geração de dados.

**RNF07:** (**Escalabilidade**) A arquitetura do software deve ser projetada para suportar extensões futuras, como a adição de novas distribuições estatísticas.

### 4 Priorização de Requisitos (MoSCoW)

#### 4.1 Must Have (Essencial para o Lançamento)

- RF01: Definição da estrutura do CSV (linhas, colunas, nomes)
- RF02: Geração de dados com Regex
- RF03: Geração de dados com distribuição Gaussiana
- RF04: Exportação para arquivo CSV
- RF05: Validação de configurações de entrada
- RNF01: Usabilidade mínima para operação
- RNF03: Portabilidade entre sistemas operacionais
- RNF04: Confiabilidade na geração dos dados

## **4.2 Should Have (Importante, se houver tempo)**

- RF06: Geração de dados com tendência linear
- RF07: Combinação de colunas com tipos de dados diferentes
- RNF02: Meta de desempenho para 1 milhão de linhas
- RNF05: Segurança (não armazenamento de dados)

## **4.3 Could Have (Desejável, para o futuro)**

- RF08: Suporte a distribuições estatísticas adicionais
- RF09: Configuração de delimitadores e separadores do CSV
- RNF06: Manutenibilidade (código modular e testes unitários)
- RNF07: Escalabilidade (arquitetura extensível)

## **4.4 Won't Have (Fora do Escopo para esta versão)**

- RF10: Visualização de dados com gráficos
- RF11: Importação e análise de arquivos CSV existentes
- RF12: Simulação de imperfeições (outliers, valores ausentes)
- RF13: Suporte a outros formatos de exportação (XLSX, JSON)

# **5 Modelagem UML**

## **5.1 Diagrama de Casos de Uso**

A Figura 1 apresenta os casos de uso essenciais para o MVP (Mínimo Produto Viável) do sistema. O diagrama foca nas funcionalidades ‘Must Have’, ilustrando as interações diretas do usuário e as dependências internas (‘«include»’) entre os casos de uso.

Estereótipo	Significado
«Must»	Essencial para o MVP

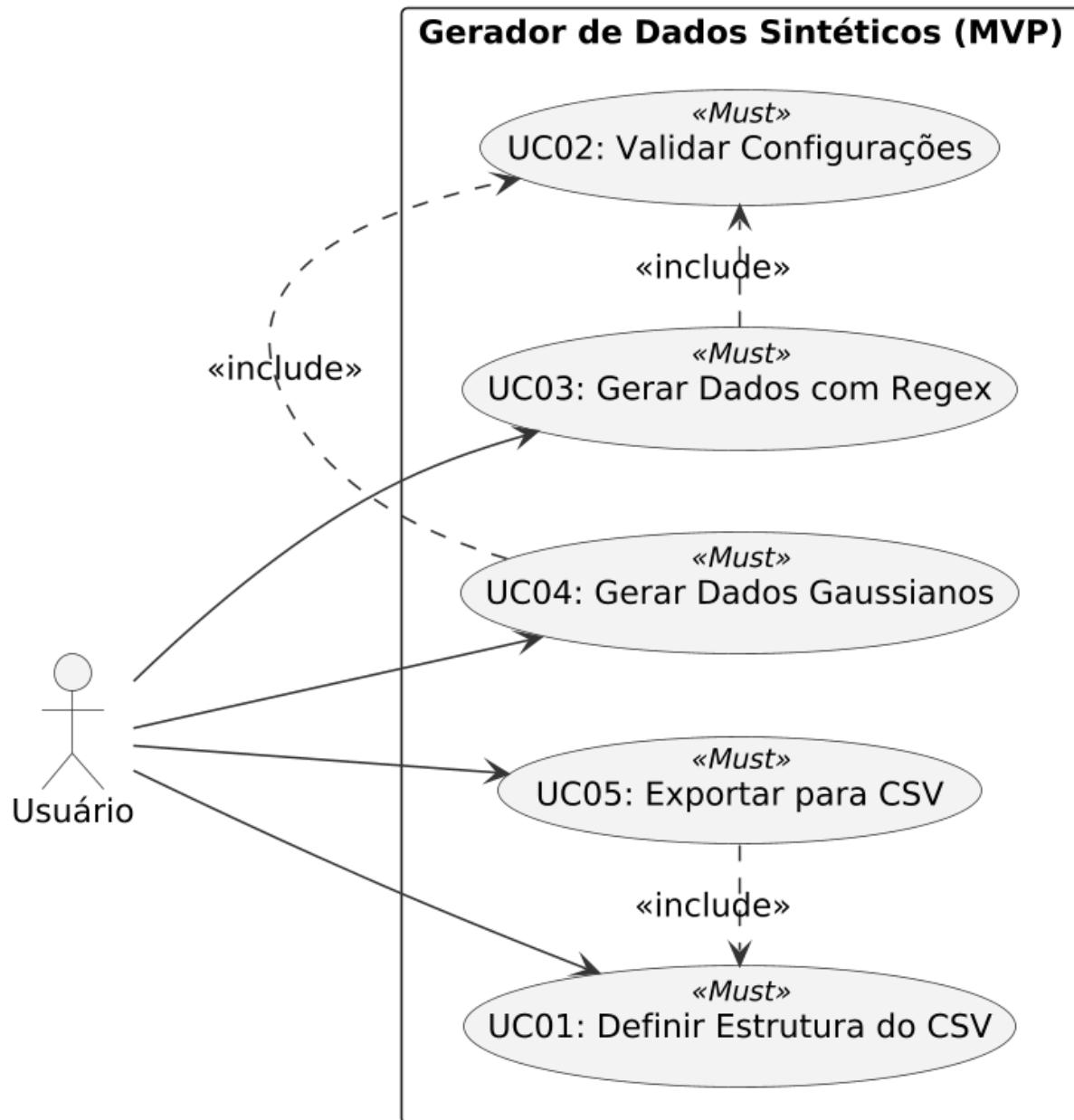


Figura 1: Diagrama de Casos de Uso focado no MVP do sistema.

## 5.2 Documentação dos Casos de Uso

A seguir, são detalhados os casos de uso do MVP, com suas respectivas relações de inclusão conforme o diagrama UML.

### 5.2.1 UC01: Definir Estrutura do CSV

- Identificador único do caso de uso: UC01

- **Autor primário:** Usuário
- **Descrição:** O caso de uso permite que o usuário especifique os parâmetros fundamentais do arquivo CSV a ser gerado, incluindo o número total de linhas de dados, o número de colunas e os respectivos nomes para o cabeçalho.
- **Pré-condições:** O software deve estar em execução.
- **Fluxo principal:**
  1. O usuário inicia a ferramenta.
  2. O usuário informa o número de linhas desejado.
  3. O usuário informa o número de colunas.
  4. Para cada coluna, o usuário fornece um nome.
  5. O sistema armazena temporariamente essa configuração estrutural.
- **Fluxo alternativo/exceção:**
  - **5a. Entrada inválida:** Se o usuário fornecer um valor não numérico ou negativo para linhas/colunas, o sistema exibe uma mensagem de erro e solicita a correção.
- **Pós-condições:** A estrutura do CSV (metadados) está definida e pronta para a configuração dos geradores de dados.
- **Relações:** Este caso de uso é incluído pelo **UC05: Exportar para CSV**.
- **Requisitos/restricções:** RF01.

### 5.2.2 UC02: Validar Configurações

- **Identificador único do caso de uso:** UC02
- **Autor primário:** Usuário (indiretamente, através de outros casos de uso)
- **Descrição:** Valida as entradas de configuração fornecidas pelo usuário para um gerador de dados específico (ex: sintaxe de Regex, parâmetros numéricos de uma distribuição) para garantir que são válidas antes de prosseguir com a geração.
- **Pré-condições:** O usuário forneceu uma entrada de configuração para um gerador de dados.
- **Fluxo principal:**
  1. O sistema recebe um parâmetro de configuração (ex: um padrão Regex, uma média e desvio padrão).
  2. O sistema verifica se o parâmetro atende às regras de sintaxe e/ou de negócio.
  3. O sistema confirma que a configuração é válida.
- **Fluxo alternativo/exceção:**
  - **3a. Configuração inválida:** Se o parâmetro for inválido (ex: Regex com sintaxe errada), o sistema rejeita a entrada, informa o usuário sobre o erro específico e aguarda a correção. O fluxo do caso de uso que o incluiu é interrompido até a correção.
- **Pós-condições:** A configuração foi validada com sucesso ou uma mensagem de erro foi exibida ao usuário.
- **Requisitos/restricções:** RF05.

### 5.2.3 UC03: Gerar Dados com Expressão Regular (Regex)

- **Identificador único do caso de uso:** UC03
- **Autor primário:** Usuário
- **Descrição:** Permite ao usuário definir que os dados de uma coluna sejam gerados com base em um padrão de expressão regular (regex).
- **Pré-condições:** A estrutura do CSV (UC01) já foi definida.
- **Fluxo principal:**
  1. O usuário seleciona uma coluna e escolhe o tipo de gerador "Regex".
  2. O usuário insere a expressão regular desejada.
  3. O sistema executa o caso de uso **UC02 (Validar Configurações)** para a expressão fornecida.
  4. Com a validação bem-sucedida, o sistema associa o gerador Regex à coluna especificada.
- **Pós-condições:** Uma coluna está configurada para gerar dados que correspondam ao padrão regex validado.
- **Relações:** «include» **UC02: Validar Configurações**.
- **Requisitos/restricções:** RF02.

### 5.2.4 UC04: Gerar Dados Gaussianos

- **Identificador único do caso de uso:** UC04
- **Autor primário:** Usuário
- **Descrição:** Permite ao usuário definir que os dados de uma coluna sejam gerados seguindo uma distribuição estatística Gaussiana (normal).
- **Pré-condições:** A estrutura do CSV (UC01) já foi definida.
- **Fluxo principal:**
  1. O usuário seleciona uma coluna e escolhe o tipo de gerador "Gaussiano".
  2. O usuário insere os parâmetros numéricos: média e desvio padrão.
  3. O sistema executa o caso de uso **UC02 (Validar Configurações)** para os parâmetros fornecidos.
  4. Com a validação bem-sucedida, o sistema associa o gerador Gaussiano à coluna especificada.
- **Pós-condições:** Uma coluna está configurada para gerar dados numéricos seguindo a distribuição Gaussiana especificada.
- **Relações:** «include» **UC02: Validar Configurações**.
- **Requisitos/restricções:** RF03.

### 5.2.5 UC05: Exportar para CSV

- **Identificador único do caso de uso:** UC05
- **Autor primário:** Usuário
- **Descrição:** Gera os dados conforme todas as configurações definidas e os salva em um arquivo no formato CSV.
- **Pré-condições:** Todas as colunas da estrutura foram associadas a um tipo de gerador de dados válido.
- **Fluxo principal:**

1. O caso de uso se inicia garantindo que a funcionalidade de **UC01 (Definir Estrutura do CSV)** foi concluída e é válida.
2. O usuário seleciona a opção para "Gerar e Exportar".
3. O sistema solicita o diretório e o nome do arquivo de saída.
4. O usuário confirma a localização e o nome.
5. O sistema processa as configurações, gera os dados e os escreve no arquivo .csv.
6. O sistema exibe uma mensagem de sucesso.

- **Fluxo alternativo/exceção:**

– **5a. Erro de permissão de escrita:** Se o sistema não tiver permissão para escrever no diretório escolhido, uma mensagem de erro é exibida.

- **Pós-condições:** Um arquivo .csv com os dados sintéticos é salvo no diretório local especificado.
- **Relações:** «include» **UC01: Definir Estrutura do CSV**.
- **Requisitos/restricções:** RF04.

## 5.3 Diagrama de Atividades

A Figura 2 ilustra o fluxo de trabalho principal do usuário ao utilizar o sistema, desde a definição inicial da estrutura do arquivo até a exportação final dos dados sintéticos.

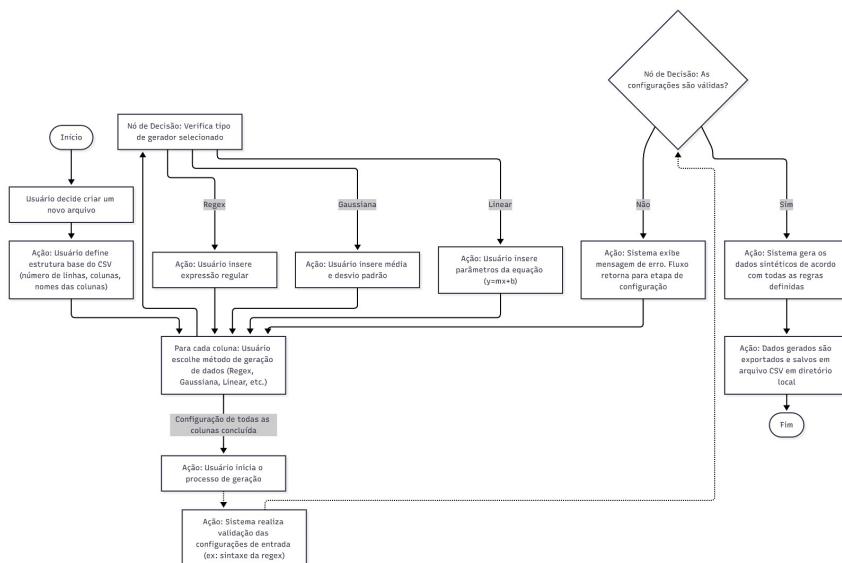


Figura 2: Diagrama de Atividades do fluxo principal de geração de dados.

## 6 Diagrama de Classes

A Figura 3 apresenta o diagrama de classes do sistema **Gerador de Dados Sintéticos Flexível**, representando as principais entidades, atributos, operações e relacionamentos. O modelo foi desenvolvido com base nos requisitos funcionais RF01 a RF07 e nos não funcionais RNF01 a RNF04, garantindo modularidade e extensibilidade da arquitetura.

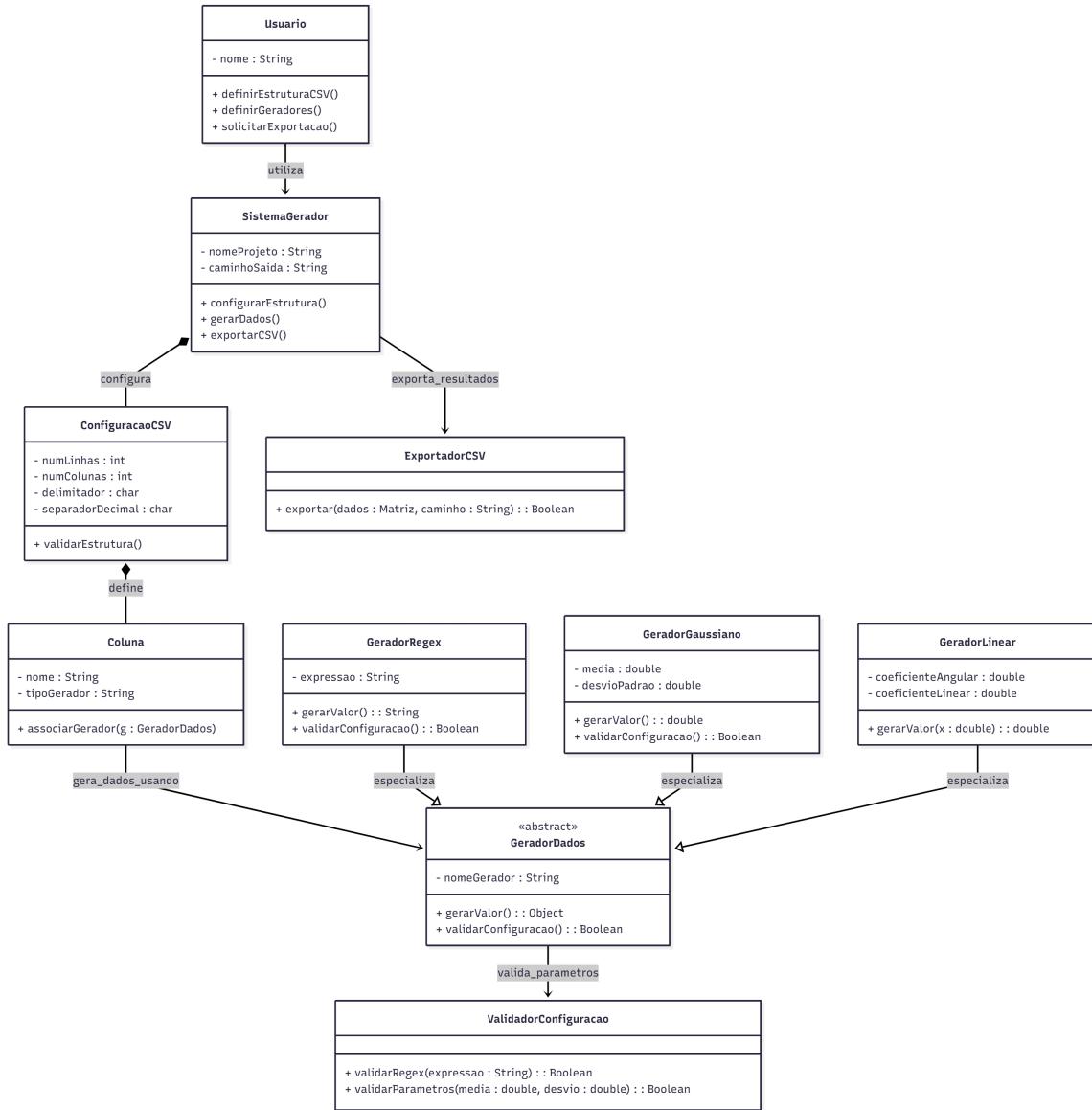


Figura 3: Diagrama de Classes do sistema Gerador de Dados Sintéticos Flexível.

### 6.1 Descrição das Relações Entre Classes

- **Usuario → SistemaGerador (utiliza)**: Representa a interação direta do usuário com o sistema, iniciando a definição de estrutura, configuração dos geradores e exportação dos dados.
- **SistemaGerador \*– ConfiguracaoCSV (configura)**: Indica que o sistema contém e gerencia uma configuração de arquivo CSV, responsável por parâmetros estruturais como número de linhas, colunas e delimitadores.

- **ConfiguracaoCSV \*– Coluna (define):** Demonstra que a configuração é composta por múltiplas colunas, cada uma representando um campo do arquivo gerado.
- **Coluna → GeradorDados (gera\_dados\_usando):** Cada coluna utiliza um gerador de dados específico (regex, gaussiano, linear etc.) para produzir seus valores.
- **SistemaGerador → ExportadorCSV (exporta\_resultados):** O sistema invoca o componente de exportação para gravar os dados gerados em um arquivo CSV.
- **GeradorDados → ValidadorConfiguracao (valida\_parametros):** Todo gerador de dados utiliza o módulo de validação para verificar a consistência de suas configurações antes da execução.
- **GeradorRegex –|> GeradorDados (especializa):** Representa a herança, onde o gerador de regex estende a classe abstrata de gerador genérico para dados textuais.
- **GeradorGaussiano –|> GeradorDados (especializa):** Indica que o gerador gaussiano herda e implementa o comportamento para dados numéricos com distribuição normal.
- **GeradorLinear –|> GeradorDados (especializa):** Define uma especialização para geração de dados com tendência linear ( $y = mx + b$ ).

## 6.2 O que Foi Feito

Durante esta etapa, foi elaborado o **diagrama de classes UML** completo do sistema, detalhando:

- As principais classes e seus atributos e métodos;
- Os tipos de relacionamento entre os componentes (associação, composição e herança);
- A hierarquia de especialização dos geradores de dados;
- A integração entre os módulos de geração, validação e exportação de dados.

O diagrama foi modelado de forma a garantir a **modularidade** do sistema, permitindo a inclusão futura de novos tipos de geradores (como uniformes ou exponenciais), atendendo assim aos requisitos de extensibilidade (RF08 e RNF07).

## 7 Contribuições Individuais

- **Cristhian Eduardo Kapelinski de Avilla:** Responsável pela pesquisa e definição dos requisitos funcionais essenciais (RF01 a RF04), focando na usabilidade do sistema (RNF01). Liderou a aplicação da metodologia MoSCoW para a priorização dos requisitos, garantindo que o escopo do MVP fosse claro e alcançável. Também contribuiu na documentação da seção de Introdução.
- **Rafael da Silva Moral:** Encarregado da especificação dos requisitos não funcionais, com ênfase no desempenho (RNF02), portabilidade (RNF03) e confiabilidade (RNF04) da aplicação. Além disso, detalhou os requisitos funcionais mais complexos, como a geração de dados com tendência linear (RF06) e a combinação de diferentes tipos de geradores (RF07).
- **Lucas Correa Rodrigues:** Focado na definição dos requisitos de extensibilidade e futuro do projeto, como o suporte a novas distribuições estatísticas (RF08) e a simulação de imperfeições nos dados (RF12). Foi o principal responsável por delinear o que estaria fora do escopo inicial (seção "Won't Have"), ajudando a manter o foco da equipe na entrega do produto mínimo viável.