

A Importância de Requisitos Definidos para a Equipe de Desenvolvimento Utilizando UML e PDCA



Roberta F. Schmidt
2024

Capítulo

01

INTRODUÇÃO



PREFÁCIO

Este e-book foi criado com a ajuda da inteligência artificial GPT-4 da OpenAI, garantindo uma escrita clara e concisa, adaptada por mim para ser facilmente compreendida.

Além disso, todas as imagens foram gerados utilizando o GitHub Copilot, uma ferramenta de criação visuais para geração de imagens.

Esperamos que este e-book seja uma ferramenta valiosa para você, oferecendo insights práticos e exemplos reais que podem ser aplicados diretamente em seus projetos de desenvolvimento de software.

Aproveite a leitura e utilize esses conhecimentos para aprimorar suas práticas de engenharia de software, garantindo projetos mais sucedidos e eficientes.



INTRODUÇÃO

Neste e-book, exploraremos como requisitos bem definidos são essenciais para o sucesso de qualquer projeto de desenvolvimento de software. Usaremos a metodologia PDCA (Plan-Do-Check-Act) e diagramas UML (Unified Modeling Language) para estruturar e ilustrar a importância de requisitos claros e bem documentados. Utilizaremos um exemplo prático ao longo do e-book para mostrar como esses conceitos podem ser aplicados no mundo real.



Capítulo

02

PLANEJAR

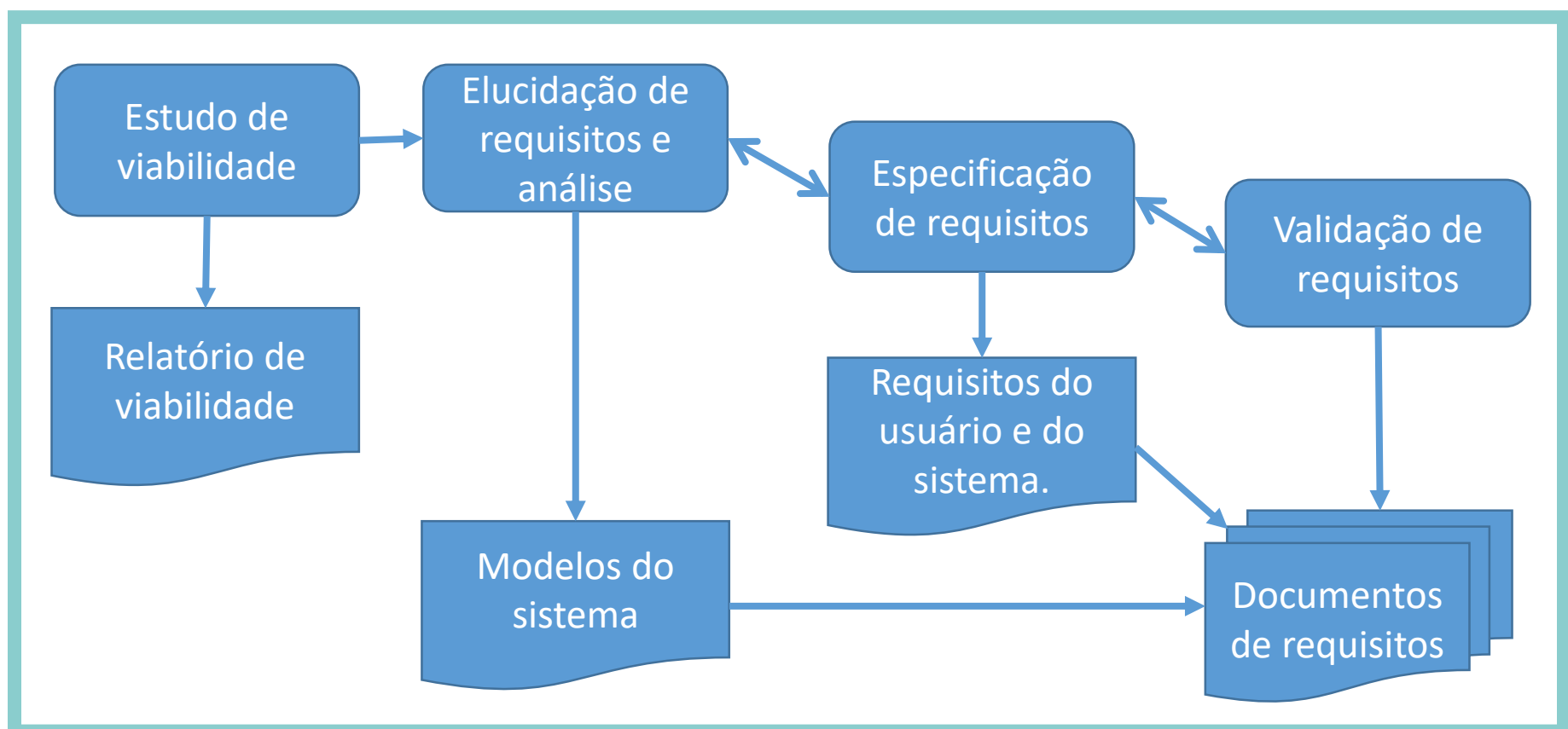
PLAN

1.1 Definindo Requisitos

Antes de iniciar qualquer projeto, é crucial entender exatamente o que é necessário. Imagine que uma empresa de tecnologia, a TechSolutions, foi contratada para desenvolver um sistema de gerenciamento de bibliotecas.

1.2 Entrevista com o Cliente

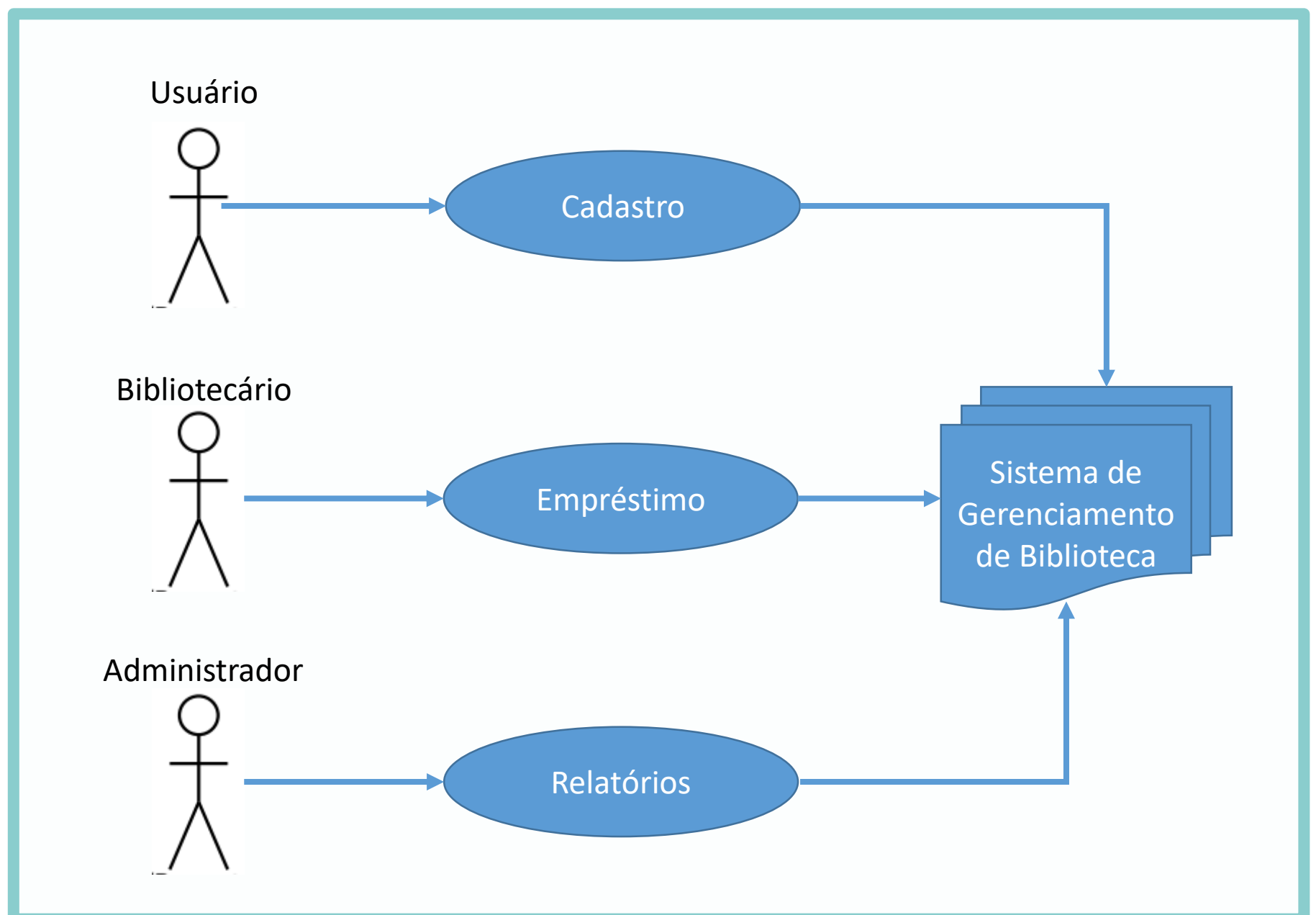
Para definir os requisitos, a equipe da TechSolutions realiza entrevistas com bibliotecários e administradores de bibliotecas para entender suas necessidades. Eles descobrem que o sistema deve permitir a gestão de empréstimos de livros, cadastro de novos usuários e o acompanhamento de reservas.



1.3 Uso de Diagramas UML

Para visualizar esses requisitos, a equipe usa um Diagrama de Casos de Uso. Este diagrama ajuda a identificar os principais atores (como bibliotecários e usuários) e as interações que eles terão com o sistema.

Exemplo de Diagrama de Casos de Uso:



Capítulo

03

EXECUTAR

DO

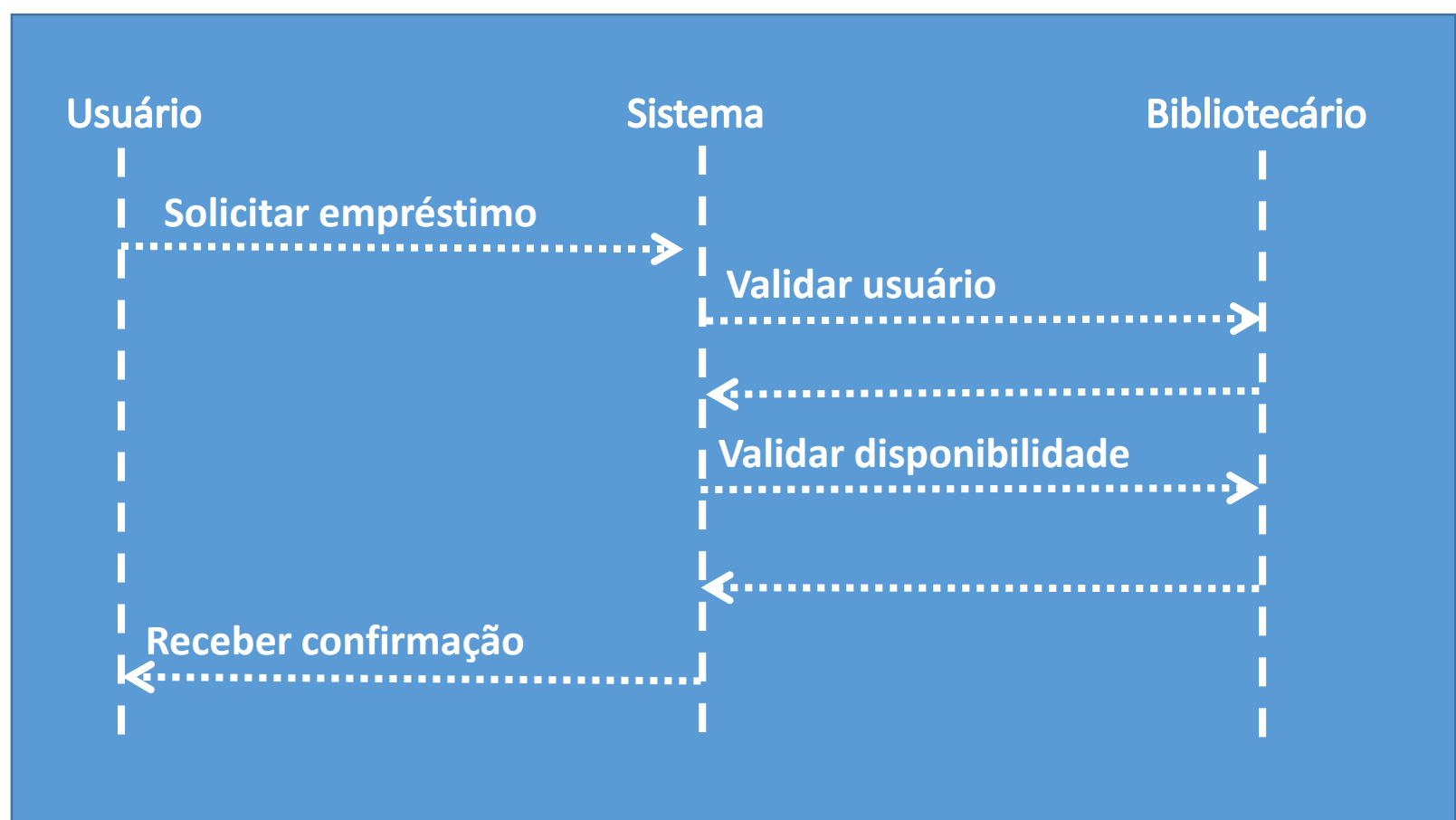
2.1 Desenvolvimento Inicial

Com os requisitos bem definidos e documentados, a equipe começa a desenvolver o sistema. Eles dividem o trabalho em sprints semanais, garantindo que cada funcionalidade seja implementada conforme planejado.

2.2 Uso de Diagramas UML

Durante a fase de desenvolvimento, a equipe utiliza Diagramas de Sequência para detalhar como os diferentes componentes do sistema interagem entre si. Isso ajuda a garantir que todos os membros da equipe compreendam como as funcionalidades devem ser implementadas.

Exemplo de Diagrama de Sequência:



Capítulo

04

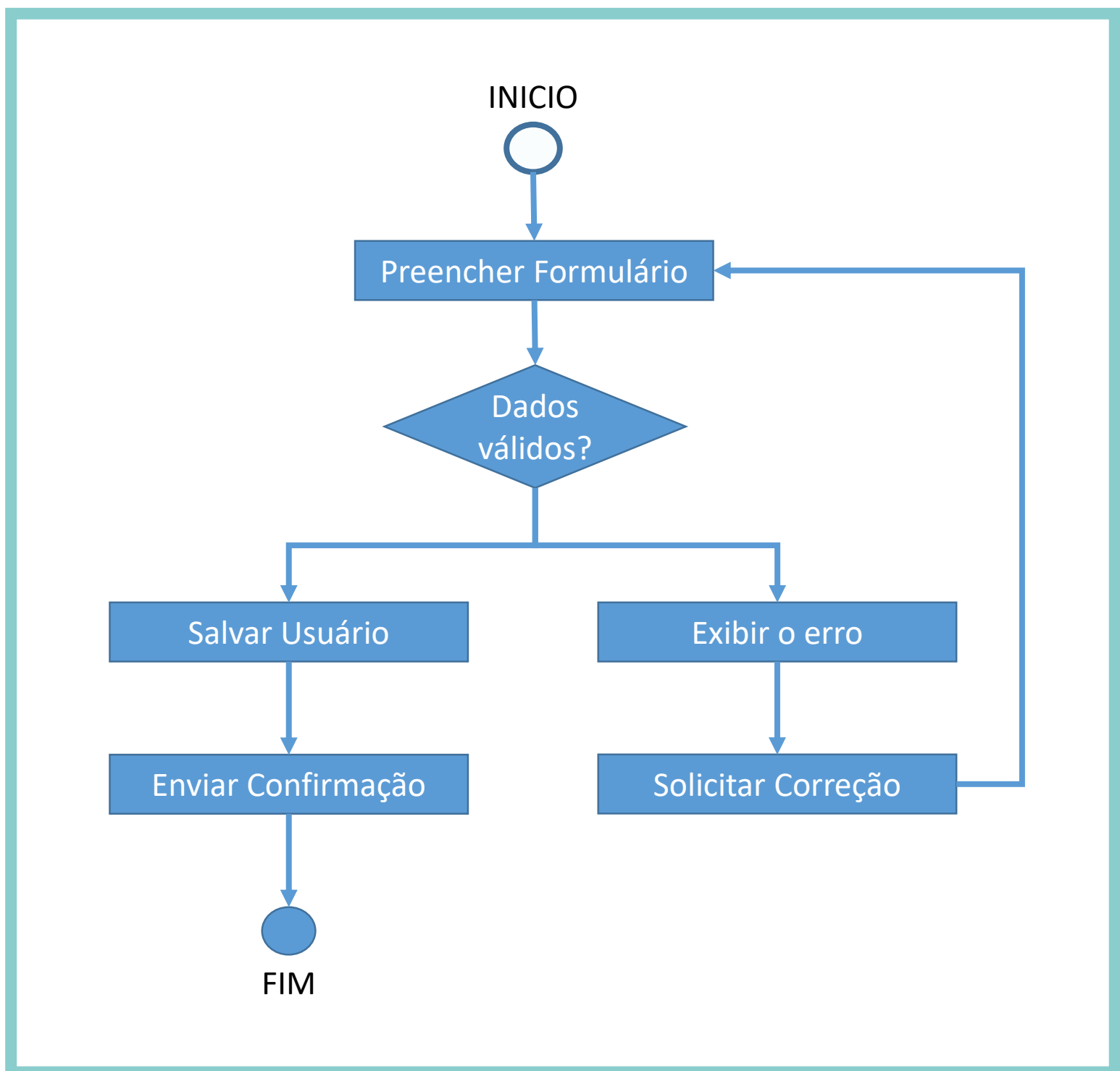
VERIFICAR

CHECK

Verificação das Regras de Negócio (Check)

Testes e Validação: Realize testes para garantir que as regras de negócio estão sendo aplicadas corretamente.

Revisão por Pares: Peça para que outros membros da equipe revisem a implementação das regras.



Capítulo

05

AGIR

ACT

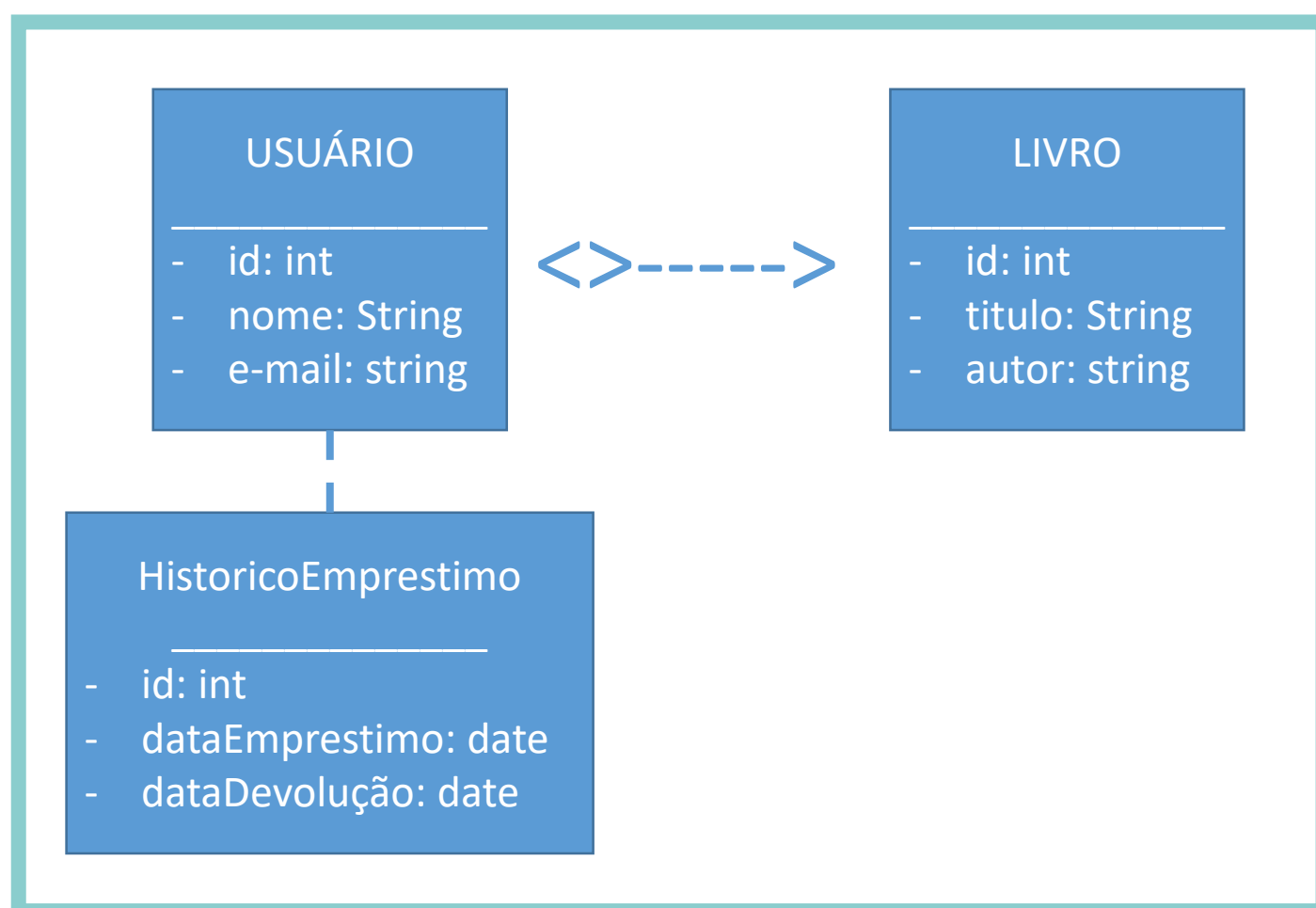
4.1 Implementação de Melhorias

Após os testes, a equipe identifica áreas de melhoria. Talvez alguns usuários tenham dificuldades com o processo de empréstimo ou cadastro. A equipe então revisa os requisitos e ajusta o sistema conforme necessário.

4.2 Uso de Diagramas UML

Para documentar e planejar essas melhorias, a equipe utiliza Diagramas de Classe, que detalham a estrutura do sistema e como as diferentes classes interagem. Isso ajuda a identificar facilmente onde as alterações precisam ser feitas.

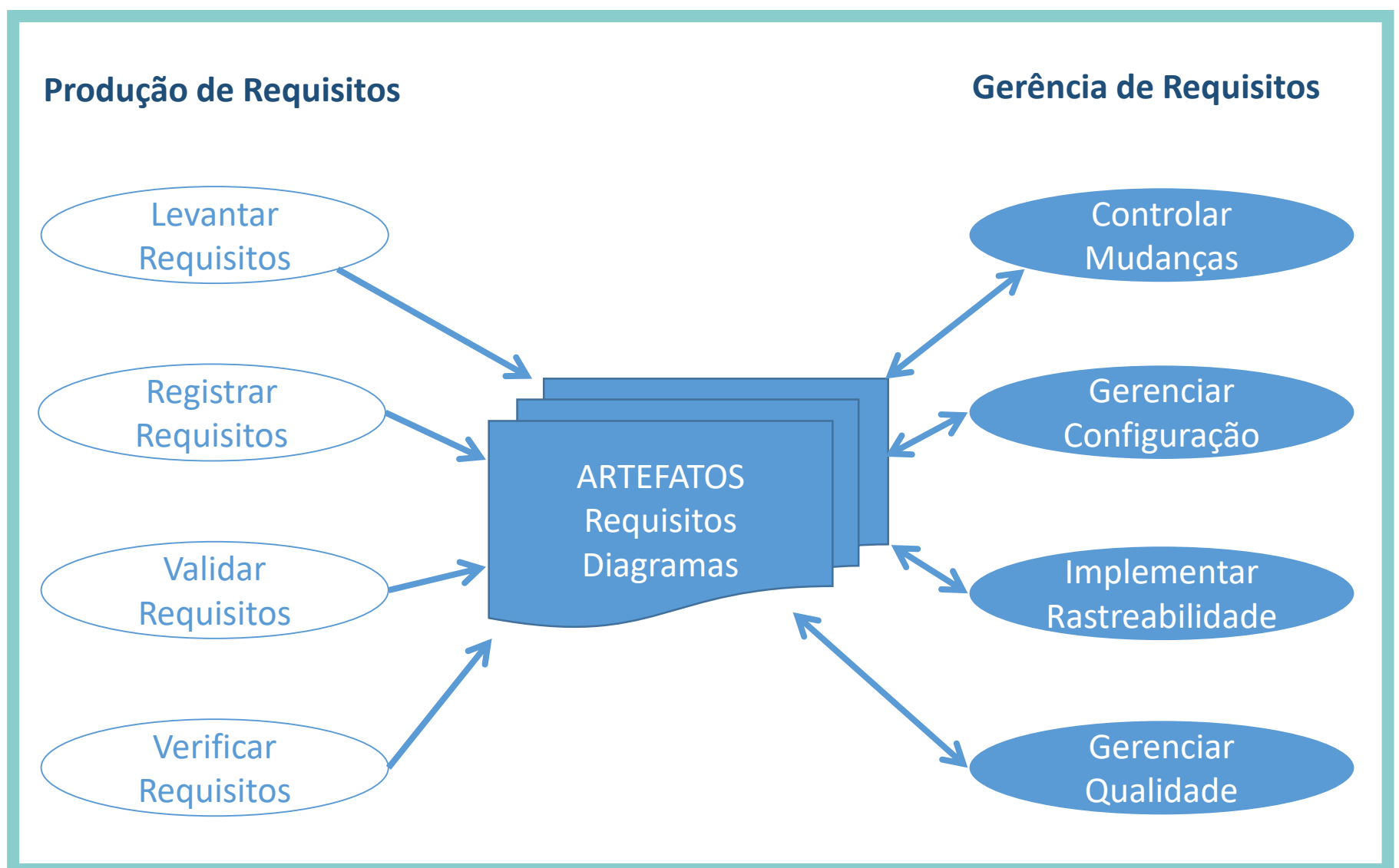
Exemplo de Diagrama de Classe



Ação com Diagramas UML (Act)

Refinamento de Diagramas: Ajuste os diagramas UML conforme as melhorias são implementadas.

Atualização Contínua: Mantenha os diagramas UML atualizados para refletir as mudanças no sistema.



CONCLUSÃO



Integrar regras de negócio claras, diagramas UML e a metodologia PDCA pode melhorar significativamente a qualidade das entregas no desenvolvimento de software. Essa abordagem não apenas agiliza o processo de desenvolvimento, mas também aumenta a produtividade da equipe, garantindo que os requisitos dos usuários e as regulamentações legais sejam atendidos de forma eficiente e eficaz. Continue explorando para aprofundar seu conhecimento e aplicar essas técnicas em seus próprios projetos, assegurando um desenvolvimento mais organizado e de alta qualidade.

