



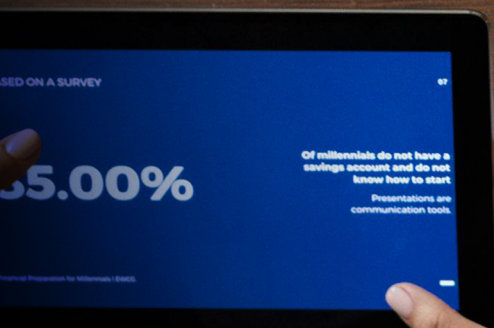
**CITHA**

Inovação e  
Sustentabilidade  
na Amazônia

2025

# METODOLOGIAS ÁGEIS:

## DA EMPATIA À ENTREGA DE VALOR COM DESIGN THINKING E SCRUM





**Dados Internacionais de Catalogação na Publicação (CIP)**  
**(Câmara Brasileira do Livro, SP, Brasil)**

Metodologias ágeis [livro eletrônico] : da empatia  
à entrega de valor com design thinking e scrum /  
Wenndisson da Silva Souza ... [et al.] ;  
prefácio por Nivaldo Rodrigues e Silva ; revisão  
Alyson de Jesus dos Santos. -- 2. ed. --  
Manaus, AM : Ed. dos Autores, 2025.  
PDF

Outros autores: Alexandre Lopes Martiniano,  
Eduardo Palhares Júnior, Lucélia Cunha da Rocha  
Santos, Nivaldo Rodrigues e Silva  
Bibliografia  
ISBN 978-65-01-80808-6

1. Administração de empresa - Metodologia  
2. Criatividade nos negócios 3. Design thinking  
4. Scrum (Desenvolvimento de software) I. Souza,  
Wenndisson da Silva. II. Martiniano, Alexandre Lopes.  
III. Palhares Júnior, Eduardo. IV. Santos, Lucélia  
Cunha da Rocha. V. Silva, Nivaldo Rodrigues e.  
VI. Santos, Alyson de Jesus dos. VII. Título.

25-317628.0

CDD-658.4063

**Índices para catálogo sistemático:**

1. Design thinking : Criatividade nos negócios :  
Administração 658.4063

Maria Alice Ferreira - Bibliotecária - CRB-8/7964

DOI: 10.5281/zenodo.17674820





# Expediente do IFAM

**Reitor**

Jaime Cavalcante Alves

**Pró-Reitor de Administração**

Fábio Teixeira Lima

**Pró-Reitor de Gestão de Pessoas**

Leandro Amorim Damasceno

**Pró-Reitora de Ensino**

Rosângela Santos da Silva

**Pró-Reitora de Extensão**

Maria Francisca Moraes de Lima

**Pró-Reitor de Pesquisa, Pós-Graduação e Inovação**

Paulo Henrique Rocha Aride

**Diretor Geral do Campus Manaus Distrito Industrial**

Nivaldo Rodrigues e Silva





# Expediente do Projeto CITHA

## **Gestores**

Nivaldo Rodrigues e Silva  
Samirames da Silva Fleury  
Alyson de Jesus dos Santos  
Maria Cassiana Andrade Braga  
Adanilton Rabelo de Andrade

## **Coordenadores**

Tiago Francisco Andrade Diocesano  
Jaidson Brandão da Costa  
Elcivan dos Santos Silva  
Martinho Correia Barros  
Adelino Maia Galvão Filho

---

## Expediente de Produção

---

## **Autores**

Wenndisson da Silva Souza  
Alexandre Lopes Martiniano  
Eduardo Palhares Júnior  
Lucélia Cunha da Rocha Santos  
Nivaldo Rodrigues e Silva

## **Avaliação Pedagógica**

Samirames da Silva Fleury

## **Diagramadores**

Eduardo Palhares Jr.  
Wenndisson da Silva Souza  
Fabio Serra Ribeiro Couto

## **Revisores de Texto**

Alyson de Jesus dos Santos

# **Metodologias Ágeis:** *da empatia à entrega de valor com Design Thinking e Scrum*

## **Autores**

Wenndisson da Silva Souza  
Alexandre Lopes Martiniano  
Eduardo Palhares Júnior  
Lucélia Cunha da Rocha Santos  
Nivaldo Rodrigues e Silva

Prefácio por Nivaldo Rodrigues e Silva

## **Revisão**

Alyson de Jesus dos Santos

**2ª Edição**

Manaus - AM  
2025

# Prefácio

Este e-book é um guia completo sobre metodologias ágeis, com destaque especial para as práticas de Design Thinking e Scrum. Estas ferramentas são apresentadas como instrumentos indispensáveis para solucionar desafios complexos, especialmente em contextos como o da Amazônia, onde sustentabilidade e eficiência são prioridades. A abordagem prática e dinâmica deste material o torna essencial para aqueles que buscam aplicar metodologias modernas em cenários reais e transformadores.

Estruturado de forma clara e interativa, o e-book convida o leitor a explorar, experimentar e implementar essas metodologias em projetos diversos. O módulo 4, em especial, exemplifica essa proposta ao detalhar a proposta de gestão sustentável de uma fazenda na Amazônia. A integração entre tecnologia, colaboração e inovação destaca o potencial das metodologias ágeis em fomentar soluções criativas para problemas relacionados à gestão de recursos naturais e sustentabilidade.

O projeto vai além de uma proposta acadêmica, ilustrando como práticas ágeis podem ser catalisadoras de mudanças reais e significativas. Desde a aplicação de sensores para monitoramento ambiental até sistemas de manejo integrado, o experimento prático evidencia como o Design Thinking e o Scrum podem transformar ideias em resultados concretos, sempre com foco nas necessidades locais e no impacto ambiental positivo.

Ao longo das páginas, os leitores serão incentivados a refletir sobre as implicações de cada metodologia e a explorar novas formas de resolver problemas. Este e-book não apenas ensina, mas inspira a adoção de uma mentalidade ágil e colaborativa, essencial para enfrentar os desafios do século XXI. Que este material seja um ponto de partida para a inovação e a transformação sustentável.



# **Projeto de Capacitação e Interiorização em Tecnologias Habilitadoras na Amazônia - CITHA**

O projeto CITHA surge com o objetivo de fortalecer a economia da Amazônia por meio do incentivo ao empreendedorismo local e do desenvolvimento sustentável. Sua proposta é capacitar profissionais e impulsionar a criação de startups voltadas para a bioeconomia, além de apoiar cooperativas locais na melhoria de seus processos produtivos. A implementação de tecnologias inovadoras é uma das estratégias centrais do projeto, visando oferecer soluções eficientes que atendam às necessidades regionais, como a otimização dos recursos naturais e a melhoria da infraestrutura local.

Ao longo de sua execução, o projeto se compromete a integrar os diversos stakeholders, como governos, empresas, ONGs e comunidades, por meio da capacitação da mão de obra local. O objetivo é formar um capital intelectual qualificado, capaz de apoiar uma governança eficiente, promover a inovação e assegurar a sustentabilidade. O CITHA dedica-se à criação de processos internos que incentivem o desenvolvimento de novos métodos e tecnologias, adaptáveis às particularidades do território amazônico.

Em síntese, o projeto CITHA visa criar um ciclo de desenvolvimento que não só incentive o empreendedorismo, mas também promova a modernização das estruturas locais, elevando a qualidade de vida das populações da Amazônia. Focado em áreas como bioeconomia, inovação e transferência de tecnologia, o projeto busca estabelecer um ecossistema mais forte e autossustentável, capaz de responder eficientemente às demandas do mercado e da sociedade.

# Lista de Expressões para Enriquecimento de Conteúdo

Este material foi cuidadosamente estruturado para apoiar sua jornada de aprendizado. Ao longo dos capítulos, você encontrará diversas chamadas sinalizadas por ícones especiais, que ajudarão a destacar pontos-chave e enriquecer sua compreensão. Durante a diagramação, esses ícones serão inseridos conforme as indicações dos autores, guiando você para diferentes tipos de conteúdo e atividades que potencializam seu estudo.

## **Fique Alerta!**

Destaque para conceitos, expressões e trechos fundamentais que merecem sua atenção especial para a compreensão do conteúdo.

## **Iniciando o diálogo...**

Espaço para reflexão crítica. Aqui você será convidado(a) a problematizar os temas abordados, relacionando-os com sua experiência e buscando conexões relevantes para aprofundar seu aprendizado.

## **Conhecendo um pouco mais!**

Indicação de fontes complementares, como livros, entrevistas, vídeos, aplicativos, links e outros recursos para ampliar seu conhecimento sobre o tema.

## **Caso Prático**

Aplicação direta do conteúdo em exemplos concretos, para facilitar a fixação e demonstrar a utilidade do que foi aprendido.

## **Copie e Teste!**

Trechos de código prontos para serem copiados e executados, para que você possa experimentar, validar e explorar na prática os conceitos estudados.

## **🎯 Objetivos do Capítulo**

Apresenta de forma clara as competências e habilidades que você irá desenvolver. Funciona como um 'contrato de aprendizado', mostrando exatamente o que você será capaz de fazer ou explicar após a conclusão do capítulo.

# Sumário

<b>1</b>	<b>Fundamentos do Design Thinking</b>	<b>10</b>
1.1	O que é Design Thinking? . . . . .	11
1.2	As Fases do Design Thinking . . . . .	12
1.3	Aplicando seus conhecimentos . . . . .	19
1.4	Considerações do módulo . . . . .	20
<b>2</b>	<b>Introdução às Metodologias Ágeis</b>	<b>21</b>
2.1	Fundamentos do Pensamento Ágil . . . . .	22
2.2	Abordagem Ágil vs. Tradicionais: Um Comparativo . . . . .	25
2.3	Aplicando seus conhecimentos . . . . .	30
2.4	Considerações do módulo . . . . .	30
<b>3</b>	<b>Introdução ao Scrum</b>	<b>31</b>
3.1	O que é SCRUM? . . . . .	32
3.2	Ciclo de desenvolvimento . . . . .	34
3.3	Aplicando seus conhecimentos . . . . .	39
3.4	Considerações do módulo . . . . .	41
<b>4</b>	<b>Experimento Prático</b>	<b>42</b>
4.1	Gestão sustentável de uma fazenda . . . . .	43
4.2	Considerações do módulo . . . . .	45
<b>5</b>	<b>Estudo de Caso Aplicado: Gestão de uma Fazenda de Piscicultura</b>	<b>46</b>
5.1	Cenário: Fazenda de Piscicultura . . . . .	46
5.2	Etapa 1: Design Thinking . . . . .	47
5.3	Etapa 2: Scrum . . . . .	48
5.3.1	Grupo de Trabalho . . . . .	49
5.3.2	Visão . . . . .	50
5.3.3	Histórias do Usuário . . . . .	50
5.3.4	Classificação das Histórias . . . . .	52
5.3.5	Backlog Priorizado do Produto . . . . .	52
5.3.6	Backlog do Produto (com Épicas Detalhados) . . . . .	53
5.3.7	Backlog da Sprint . . . . .	53
5.3.8	Sprint Planning . . . . .	54
5.3.9	Review da Sprint . . . . .	57
5.3.10	Retrospectiva da Sprint . . . . .	57
5.3.11	Ata da Reunião . . . . .	59
5.4	Considerações Finais do Estudo de Caso . . . . .	60



# Capítulo 1

## Fundamentos do Design Thinking

### Iniciando o diálogo...

Neste módulo, você vai aprender o que é Design Thinking, como ele funciona, e por que ele é uma abordagem tão importante para resolver problemas de forma criativa e colaborativa. O foco será em entender cada uma das fases do processo e aplicar essas técnicas em atividades práticas, ajudando você a desenvolver soluções inovadoras em qualquer área.

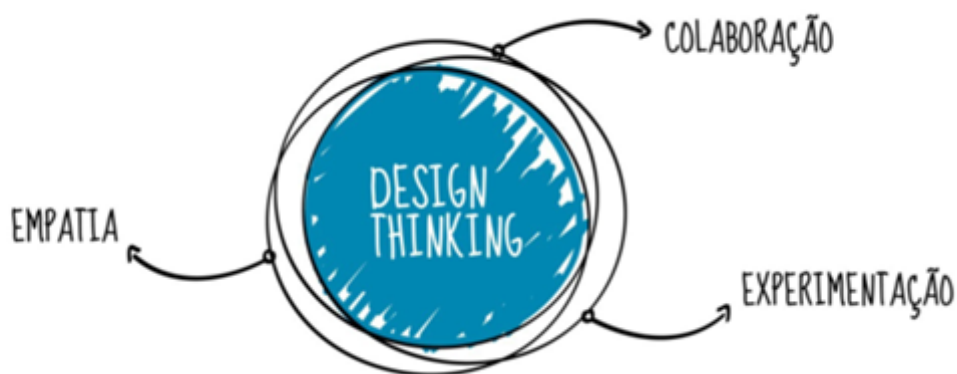


Figura 1.1: Diagrama de design thinking

Fonte: <https://educapes.capes.gov.br/handle/capes/572344>

### 🎯 Objetivos do Capítulo

Ao final deste capítulo, você será capaz de:

- Definir o que é Design Thinking e seus pilares fundamentais.
- Descrever as cinco fases do processo de Design Thinking: Empatia, Definição, Ideação, Prototipação e Teste.
- Identificar e aplicar ferramentas práticas para cada fase, como o Mapa de Empatia e os 5 Porquês.
- Compreender como o Design Thinking pode ser usado para resolver problemas complexos de forma criativa e colaborativa.

## 1.1 O que é Design Thinking?

Design Thinking é uma abordagem metodológica focada na inovação e na solução de problemas complexos a partir de uma perspectiva profundamente humana. O conceito, embora com raízes em discussões acadêmicas dos anos 60, foi consolidado e popularizado para o mundo dos negócios por Tim Brown, da renomada consultoria de design IDEO. Em sua obra de referência, ele o descreve não como um processo estritamente linear, mas como um sistema de espaços sobrepostos — inspiração, ideação e implementação — que se equilibram para gerar soluções inovadoras [4].



Figura 1.2: Conceito de design thinking aplicado em uma reunião  
Fonte: Gerada por Canva Magic Studio, 2025

A força desta abordagem reside no equilíbrio sistêmico entre três pilares essenciais, que devem ser considerados de forma integrada:

- **Desejabilidade (Desirability):** O ponto de partida é o ser humano. A solução proposta atende a uma necessidade real, resolve uma dor latente ou é genuinamente desejada pelas pessoas a quem se destina?
- **Fezabilidade (Feasibility):** A solução pode ser construída do ponto de vista técnico e organizacional? Possuímos os recursos, a tecnologia e a capacidade para torná-la realidade?
- **Viabilidade (Viability):** A solução é sustentável como um modelo de negócio? Ela se integra à estratégia da organização e possui um caminho para a sustentabilidade financeira?

Diferente de abordagens tradicionais que podem partir de uma restrição tecnológica ou de uma análise de mercado puramente quantitativa, o Design Thinking se inicia com a **em-patia**. O objetivo é imergir no contexto do usuário para gerar *insights* que guiarão todo o processo. Este método iterativo e não-linear valoriza a colaboração entre equipes multidisciplinares e a prototipação rápida como formas de aprender, testar hipóteses e reduzir os riscos inerentes à inovação.

### Conhecendo um pouco mais!

Vídeo

O que é design thinking? (e quais são suas etapas fundamentais?)

<https://www.youtube.com/watch?v=7hZMGSamsYA>

ou aponte a câmera do seu smartphone para o Qr Code ao lado



## 1.2 As Fases do Design Thinking

O processo de Design Thinking é frequentemente apresentado em cinco fases distintas para facilitar a compreensão. No entanto, é crucial entender que, na prática, este não é um caminho linear. As equipes frequentemente circulam entre as fases, retornando à empatia após um teste, ou gerando novas ideias durante a prototipação. O valor do modelo está em fornecer uma estrutura para navegar a complexidade da inovação, e não em ser uma receita rígida a ser seguida.



Figura 1.3: As 5 fases do design thinking

Fonte: Elaborada pelos autores



## Passo 1: EMPATIA (ENTENDENDO AS PESSOAS)

A primeira fase do Design Thinking é Empatia. Aqui, o objetivo é se colocar no lugar das pessoas que enfrentarão o problema que você quer resolver. Isso significa que você precisa conversar com elas, observar o que elas fazem no dia a dia e tentar entender como elas se sentem em relação ao problema.

**Atividade:** Você está trabalhando na **gestão sustentável de uma fazenda**, na fase de Empatia, pode começar conversando com trabalhadores, familiares, fornecedores ou moradores locais para entender as dificuldades enfrentadas no dia a dia. Por exemplo, pergunte:

- Quais os principais desafios de infraestrutura, como sistemas de energia e automação?
- De que forma ocorre a captação e o uso da água?
- Existem problemas de desperdício ou escassez?
- Quais dificuldades se destacam no cultivo de alimentos e na criação de animais?
- Há interesse ou conhecimento sobre sistemas integrados de produção?

Além das entrevistas, faça observações em campo. Por exemplo, veja se há ineficiência no manejo da água, dificuldade em organizar os cultivos ou problemas na alimentação dos animais. Essas informações vão ajudá-lo a mapear as necessidades reais e criar soluções inovadoras.

### Fique Alerta!

Use a ferramenta abaixo do Design Thinking, o Mapa de Empatia, que serve para conhecer as pessoas e coletar informações para resolver o problema.

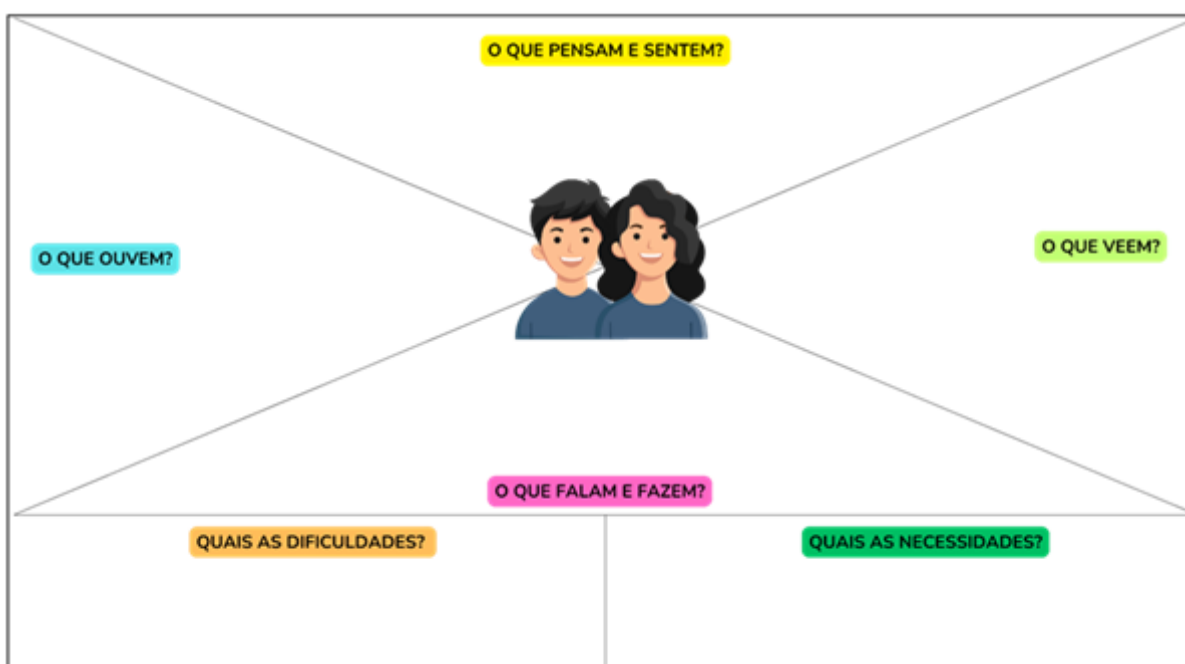


Figura 1.4: Estrutura do mapa de empatia  
Fonte: Elaborada pelos autores

### Como usar o mapa de empatia?

1. **Desenhe o Mapa de Empatia com quatro quadrantes:** O que pensam e sentem, O que ouvem, O que veem e O que falam e fazem.
2. **Converse com as pessoas envolvidas no problema** e registre suas respostas em cada quadrante:
  - **O que pensam e sentem:** Pergunte sobre os sentimentos e preocupações que elas têm em relação ao problema.
  - **O que ouvem:** Pergunte sobre o que amigos, colegas ou familiares dizem sobre o problema.
  - **O que veem:** Observe o ambiente e peça que descrevam o que elas notam ao seu redor que contribui para o problema.
  - **O que falam e fazem:** Observe como elas lidam com o problema e o que falam sobre ele.
3. Após preencher o mapa, revise as informações para entender melhor **as dificuldades** e **as necessidades** das pessoas. Isso ajudará você a definir o problema de forma clara e empática antes de buscar soluções.

#### Fique Alerta!

Agora use a ferramenta do Design Thinking, os 5 Porquês, para entender a causa raiz do problema e identificar soluções mais eficazes.

O diagrama mostra a estrutura dos 5 Porquês em um caderno espiralado. No topo, há uma linha de espiral verde. Abaixo dela, há uma caixa laranja com o texto "Meu problema é:". Seguem cinco caixas brancas, cada uma com o texto "Por quê?". Abaixo dessas caixas, há uma caixa branca com o texto "A causa raiz do problema é:". À esquerda da caixa final, há um ícone de um caderno e uma caneta.

Figura 1.5: Estrutura dos 5 Porquês  
Fonte: Elaborada pelos autores

### Como Usar a Ferramenta dos 5 Porquês?

1. **Defina o Problema** de forma clara. Escreva uma frase curta sobre o problema que você identificou.
2. **Pergunte "Por quê?"** em relação à situação do problema. A resposta vai ajudá-lo a identificar a causa mais imediata.
3. **Pergunte novamente "Por quê?"** para a resposta anterior. Continue perguntando "Por quê?" até chegar à causa raiz do problema. Repita o processo até cinco vezes ou até que as respostas não revelem mais causas subjacentes.
4. **Analise a causa raiz** encontrada na última resposta. Esse é o motivo mais profundo do problema e, provavelmente, o ponto ideal para começar a pensar em soluções.
5. **O que podemos concluir?** A causa raiz pode ser a falta de organização e comunicação no início do trabalho em grupo. Com isso, você pode propor soluções focadas na organização inicial e na divisão de tarefas.

### Passo 2: DEFINIÇÃO DO PROBLEMA

Na fase de **Definição**, você organiza tudo o que aprendeu na fase de Empatia. Aqui, é importante deixar o problema bem claro e específico. Quando o problema é bem definido, é mais fácil criar uma solução realmente útil.

**Atividade:** Depois de coletar informações na fase de Empatia, organize-as para identificar o problema central. Por exemplo:

- Você pode descobrir que a falta de sensoriamento e automação na estrutura da fazenda reduz a produtividade.
- Ou que os sistemas de captação e monitoramento de água são insuficientes, causando desperdícios ou dificuldades no manejo.

Uma definição clara do problema pode ser: "A fazenda enfrenta dificuldades em monitorar o consumo de água e energia, afetando a eficiência das atividades agrícolas e de criação de animais."

#### Fique Alerta!

Utilize a ferramenta de Design Thinking, o Diagrama de Afinidades, para organizar as informações e definir com precisão o problema central.

### Como Usar o Diagrama de Afinidades?

1. **Revise todas as informações coletadas** durante a fase de empatia. Escreva cada insight, observação ou comentário em um cartão, uma folha pequena recortada ou post-it.
2. **Organize os cartões** em grupos de ideias semelhantes. Por exemplo, se você tem várias observações sobre o desconforto físico dos alunos, agrupe essas respostas.



3. **Identifique padrões e temas principais** entre os grupos. Dê um nome a cada grupo, como “Infraestrutura”, “Água”, “Agricultura” ou “Manejo”, dependendo das questões mais mencionadas.
4. **Priorize os temas** para ajudar a identificar o problema mais importante a ser resolvido.

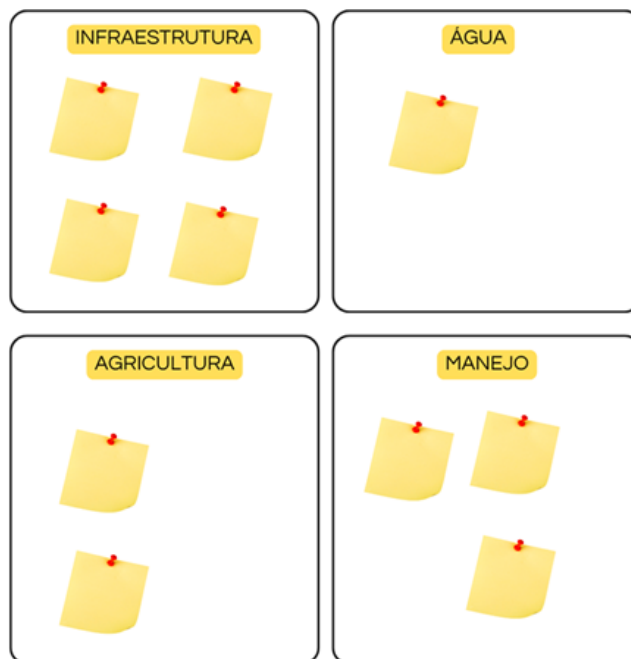


Figura 1.6: Diagrama de afinidades  
Fonte: Elaborada pelos autores

### Passo 3: IDEIAÇÃO

Agora que o problema está claro, é hora de ser criativo! Na fase de **Ideação**, você vai gerar o maior número possível de ideias para resolver o problema. Não se preocupe se algumas ideias parecerem malucas – quanto mais criativo, melhor!

**Atividade:** Para resolver problemas identificados, pense em ideias criativas e viáveis. Por exemplo:

- **Estrutura Física:** Projetar um sistema de energia renovável com painéis solares para alimentar dispositivos de automação.
- **Captação de Água:** Criar reservatórios inteligentes com sensores que monitoram o nível e a qualidade da água.
- **Agricultura e Criação Integradas:** Implementar um sistema de aquaponia onde a água dos tanques de peixes seja reutilizada como fertilizante natural para os vegetais.

Reúna colegas e faça uma sessão de Brainstorming (tempestade de ideias). Peça a cada um que sugira soluções para o problema que você definiu. Anote todas as ideias, mesmo as mais simples ou ousadas. Em seguida, utilize a Matriz SWOT para avaliar e organizar as melhores ideias, identificando as forças, fraquezas, oportunidades e ameaças associadas a cada solução proposta

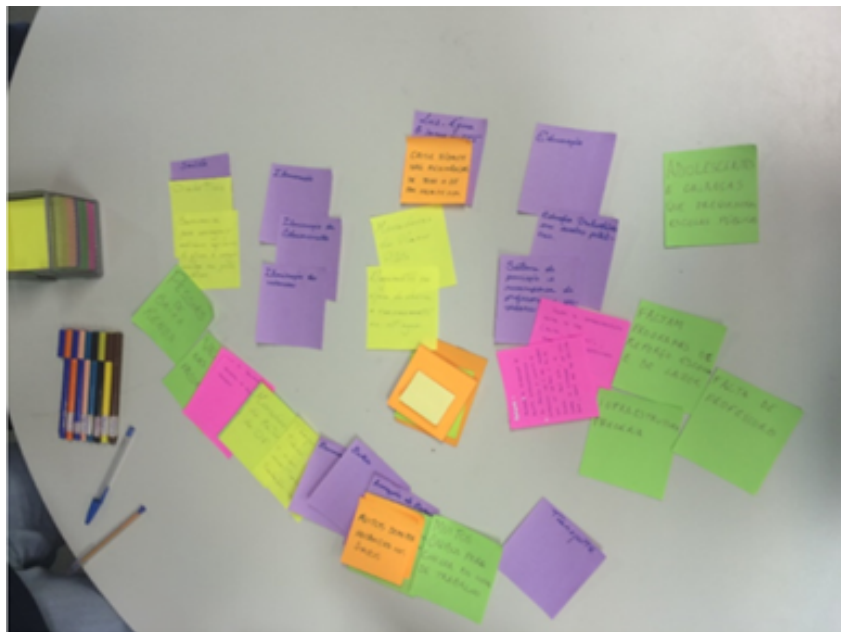


Figura 1.7: Exemplo de Brainstorm  
Fonte: Yasminkalume, 2017

### Como conduzir o Brainstorming?

1. **Defina o problema claramente** para o grupo. Certifique-se de que todos entendam a questão central que desejam resolver.
2. **Estabeleça um ambiente aberto e sem julgamentos.** Deixe claro que qualquer ideia é válida, incentivando o grupo a pensar de forma criativa e sem restrições.
3. **Registre todas as ideias** em post-its, papéis recortados ou em um quadro visível para todos. Isso facilita a visualização e organização das propostas.
4. **Priorize as ideias** que mais se destacam com base na viabilidade e no impacto que podem ter para resolver o problema.

### Como usar a matriz SWOT para avaliar as ideias?

1. Desenhe a matriz SWOT com quatro quadrantes: Forças (Strengths), Fraquezas (Weaknesses), Oportunidades (Opportunities) e Ameaças (Threats).
2. Analise cada ideia prioritária com a Matriz SWOT, preenchendo os quadrantes:
  - **Forças:** Quais são os pontos fortes dessa ideia?
  - **Fraquezas:** Quais são as limitações ou pontos fracos?
  - **Oportunidades:** Que oportunidades essa ideia cria?
  - **Ameaças:** Que obstáculos ou riscos podem surgir?
3. Compare as ideias com a Matriz SWOT para decidir qual delas apresenta o melhor equilíbrio entre pontos fortes, oportunidades e menor quantidade de ameaças e fraquezas.
4. Escolha as soluções com o melhor perfil SWOT para implementar ou para o próximo estágio de desenvolvimento.

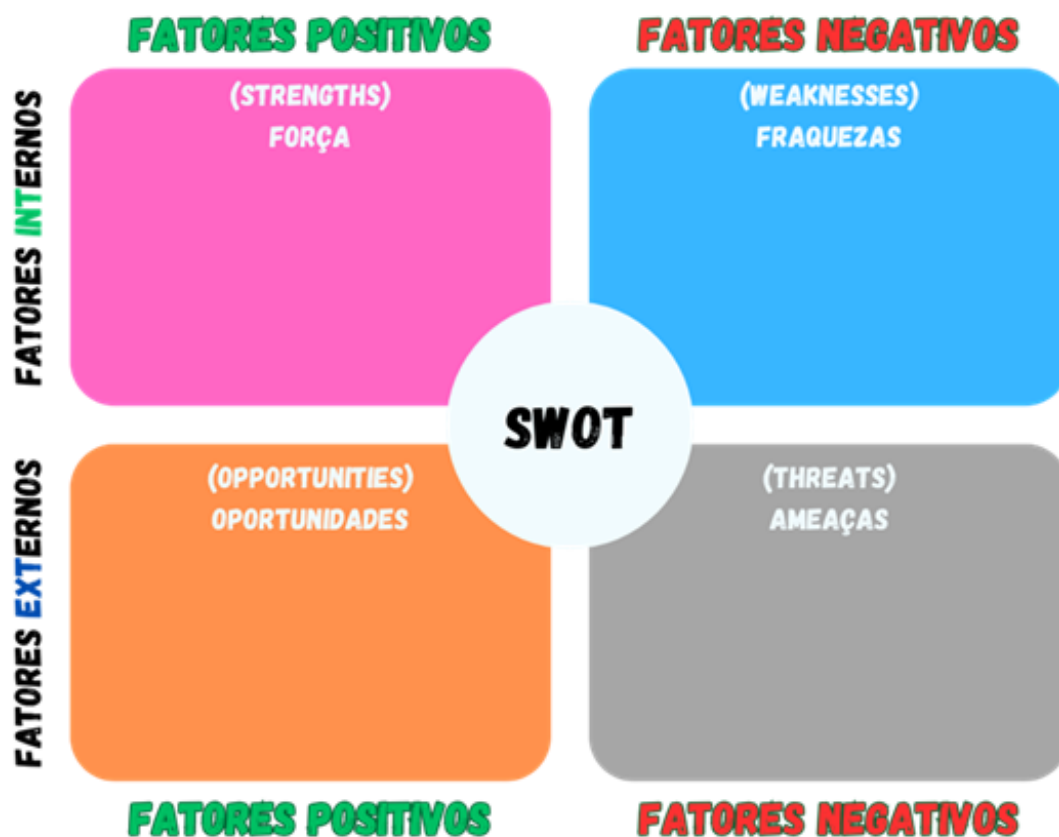


Figura 1.8: Representação gráfica da Matriz SWOT e seus quatro quadrantes  
Fonte: Elaborada pelos autores

#### Passo 4: PROTOTIPAÇÃO (CRIANDO MODELOS)

A fase de **Prototipação** é onde você escolhe as melhores ideias e começa a transformá-las em algo concreto. Aqui, você vai criar uma versão simples da solução para que as pessoas possam experimentá-la.

**Atividade:** Para testar uma das ideias, você pode criar um protótipo simples, por exemplo:

- Monte uma maquete da fazenda com tanques para aquaponia e um sistema básico de irrigação conectados a sensores.
- Desenhe esquemas para o posicionamento de painéis solares e dispositivos automáticos que melhorem a gestão de energia.
- Simule o funcionamento de um aplicativo que monitore os reservatórios de água e os cultivos em tempo real.

Selecione uma das ideias desenvolvidas na fase de Ideação e construa um protótipo inicial. Utilize materiais simples como papel, canetas, objetos recicláveis ou qualquer item acessível para criar uma versão preliminar da sua solução. Lembre-se: o protótipo não precisa ser perfeito nem finalizado. O objetivo é produzir uma representação funcional da ideia, algo que possa ser testado, ajustado e melhorado com base no feedback dos usuários. Concentre-se em mostrar como a solução funcionaria na prática e esteja aberto a modificá-la conforme as sugestões recebidas.

## Passo 5: TESTES (COLHENDO FEEDBACK)

Na fase de **Testes**, você vai apresentar seu protótipo para as pessoas que enfrentarão o problema e pedir para que elas o experimentem. Com base no feedback delas, você poderá fazer ajustes e melhorias na solução.

**Atividade:** Implemente o protótipo em pequena escala e colete feedback:

- **Estrutura Física:** Teste o impacto do uso de sensores automáticos no desempenho dos trabalhadores.
- **Aquaponia:** Observe como o sistema integrado de peixes e vegetais atende às necessidades de subsistência e comercialização.
- **Captação de Água:** Avalie se o monitoramento digital reduz o desperdício e melhora a distribuição para os cultivos e animais.

### Fique Alerta!

O Design Thinking, não é uma fórmula mágica. Seu sucesso depende de um verdadeiro engajamento com as necessidades e realidades dos usuários. Isso significa que as empresas e equipes que utilizam essa metodologia precisam estar dispostas a **escutar ativamente**, **observar comportamentos** e **adaptar suas ideias** conforme novos aprendizados surgem. Não basta seguir as etapas do Design Thinking de forma mecânica – é essencial que a equipe tenha empatia, abertura para mudanças e disposição para experimentar novas soluções constantemente.

Por exemplo, gigantes da tecnologia usaram o Design Thinking não apenas para criar produtos inovadores, mas também para refinar suas ofertas continuamente. Eles estão sempre atentos ao feedback dos usuários, fazendo ajustes e melhorias com base no uso real e nas necessidades que surgem com o tempo. A chave do sucesso dessas empresas é que elas nunca param de **aprender com seus usuários**. Elas sabem que **um produto nunca está** "pronto" – ele pode sempre ser melhorado, e isso só acontece quando o foco permanece em quem vai utilizá-lo.

Ao entender o que os usuários precisam e quais são suas dores, essas empresas desenvolvem soluções que não só resolvem problemas, mas também melhoram a experiência do dia a dia. Seja um celular fácil de usar, uma plataforma que ajuda pessoas a encontrar lugares ou uma ferramenta de ensino inovadora, o Design Thinking está por trás de muitas das inovações que vemos hoje.

## 1.3 Aplicando seus conhecimentos

1. Escreva um texto refletindo sobre como o Design Thinking pode transformar a maneira como você lida com desafios cotidianos.
  - (a) Pense em problemas que enfrenta no dia a dia, seja em casa, na escola ou na comunidade, e considere como as etapas dessa metodologia como empatia, definição do problema e prototipagem podem ajudá-lo a compreender melhor as necessidades das pessoas envolvidas, gerar ideias criativas e testar soluções práticas.
  - (b) Avalie também como essa abordagem colaborativa pode incentivar novas formas de resolver problemas, promovendo inovação e impacto positivo em sua rotina

## 1.4 Considerações do módulo

O Design Thinking nos ensina a colocar as pessoas no centro de qualquer solução, focando em entender suas necessidades e criar algo que faça diferença em suas vidas. Ao seguir as cinco fases – **Empatia, Definição, Ideação, Prototipação e Testes** – você é capaz de gerar ideias inovadoras e aplicáveis, ajustando as soluções conforme recebe feedback.

Esse processo é colaborativo e criativo, permitindo que equipes de diferentes áreas trabalhem juntas para resolver problemas complexos de forma eficaz. No próximo módulo, vamos explorar como o **Scrum**, uma metodologia ágil, ajuda a organizar o trabalho em equipe para que essas soluções possam ser desenvolvidas de maneira eficiente e entregues rapidamente.

### Conhecendo um pouco mais!

Produto Técnico e Tecnológico  
GUIA DIDÁTICO DO DESIGN THINKING

<https://educapes.capes.gov.br/handle/capes/572344>

ou aponte a câmera do seu smartphone para o Qr Code ao lado



# Capítulo 2

## Introdução às Metodologias Ágeis

### Iniciando o diálogo...

Bem-vindo(a) estudante, neste módulo, você vai aprender o que são metodologias ágeis, por que elas são importantes e como elas diferem dos métodos tradicionais, como são aplicadas em projetos, possibilitando a entrega de resultados de forma mais rápida e eficiente.

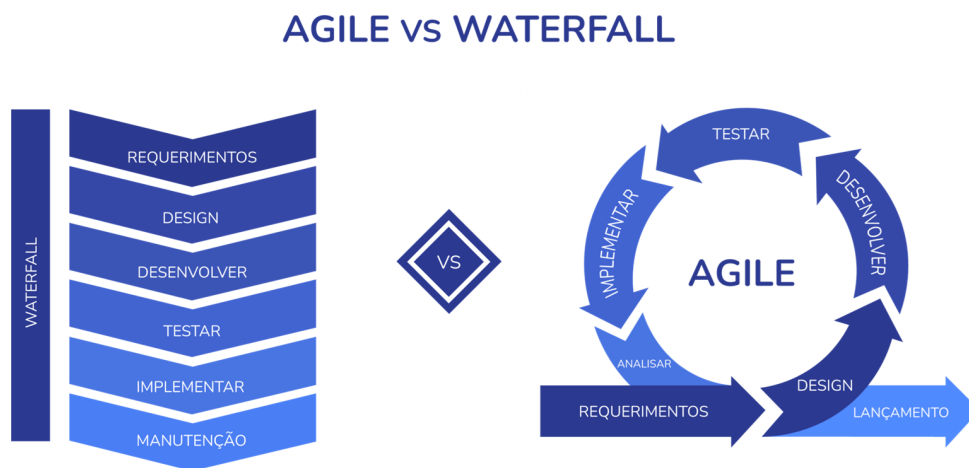


Figura 2.1: Arquitetura do modelo tradicional vs o modelo ágil

Fonte: XPEducação, 2022

### 🎯 Objetivos do Capítulo

Ao final desta leitura, você será capaz de:

- Diferenciar o conceito de **agilidade** (capacidade de adaptação) de simplesmente **velocidade**.
- Compreender os quatro valores fundamentais e os doze princípios do Manifesto Ágil, a base filosófica do movimento.
- Analisar as principais diferenças práticas entre as metodologias ágeis e as abordagens tradicionais de gerenciamento de projetos.
- Identificar os contextos em que cada abordagem é mais adequada.

## 2.1 Fundamentos do Pensamento Ágil

Antes de mergulhar nas práticas e cerimônias de uma metodologia específica, é fundamental compreender a filosofia que a sustenta. O termo "Ágil" (do inglês, *Agile*) no contexto de desenvolvimento de projetos refere-se não a uma única metodologia, mas a uma mentalidade, um conjunto de valores e princípios focados na adaptação, colaboração e entrega de valor de forma contínua.

### O Que é Agilidade? (Além da Velocidade)

É um erro comum associar agilidade exclusivamente com velocidade. Embora equipes ágeis frequentemente entreguem resultados de forma mais rápida, o cerne do conceito é a **capacidade de responder a mudanças** de forma eficaz.

Uma analogia útil é comparar o gerenciamento de projetos a uma viagem de carro:

- Uma **abordagem tradicional** é como planejar a viagem inteira usando um mapa de papel impresso. O caminho é definido do início ao fim. Se houver um bloqueio na estrada, o plano se torna obsoleto e o custo para mudar a rota é alto.
- Uma **abordagem ágil** é como usar um aplicativo de GPS. Você sabe o destino, mas o caminho é recalculado em tempo real com base nas condições do trânsito, permitindo desvios e otimizações para chegar ao destino da forma mais eficiente possível.

Portanto, a agilidade está na flexibilidade e na adaptação, e não apenas na rapidez da execução.

### O Guarda-Chuva Ágil e o Manifesto

O pensamento ágil não nasceu em um vácuo. No final dos anos 90, diversas metodologias "leves" (em oposição aos processos pesados e burocráticos da época) ganhavam força, como o **Extreme Programming (XP)**, focado na excelência técnica e na colaboração [2], e o **Scrum**, um framework para gerenciar trabalhos complexos de forma iterativa [8].

Em 2001, dezessete praticantes desses métodos se reuniram e consolidaram os valores comuns que guiavam seus trabalhos, redigindo o **Manifesto para o Desenvolvimento Ágil de Software** [1]. Este documento estabeleceu uma mentalidade baseada em valores fundamentais que abriga diversas metodologias e frameworks, conforme ilustrado na figura 2.2,

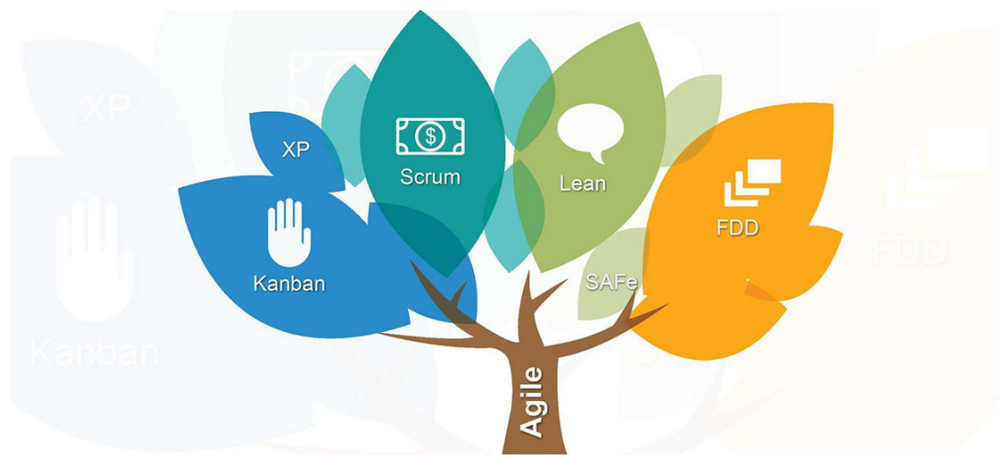


Figura 2.2: O "Guarda-Chuva Ágil" e suas diversas metodologias e frameworks.

Fonte: Cy Mastrodomenico, 2020



Os quatro valores do manifesto priorizam:

- **Indivíduos e interações** mais que processos e ferramentas;
- **Software em funcionamento** mais que documentação abrangente;
- **Colaboração com o cliente** mais que negociação de contratos;
- **Responder a mudanças** mais que seguir um plano.

Isso não significa que os itens à direita não tenham valor, mas que os itens à esquerda são *mais* valorizados.

### Fique Alerta!

A equipe deve estar preparada para mudanças no escopo, tempo, custo, tecnologia, arquitetura, entre outros. Iterações curtas de desenvolvimento permitem que mudanças possam ser rapidamente inseridas no projeto, de forma que atendam às novas necessidades.

## Os Doze Princípios de Suporte

Para dar suporte prático aos quatro valores, o Manifesto é acompanhado por doze princípios. Em vez de uma simples lista, podemos compreendê-los melhor ao agrupá-los por afinidade temática:

### Foco no Cliente e no Valor

Estes princípios garantem que o trabalho realizado esteja sempre alinhado às necessidades do cliente, entregando o máximo de valor possível.

- **1. Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.**

A ênfase está na entrega contínua de valor, e não em uma única entrega massiva no final do projeto. Isso permite que o cliente comece a ter retorno sobre seu investimento mais cedo.

- **2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se aproveitam de mudanças, para a vantagem competitiva do cliente.**

A mudança não é vista como um problema, mas como uma oportunidade. O objetivo é construir o produto certo, e o ágil permite que a definição de "certo" evolua conforme o aprendizado do mercado.

### Processo, Ritmo e Entrega

Estes princípios governam como o trabalho flui, como o progresso é medido e como a sustentabilidade da equipe é mantida.

- **3. Entregar frequentemente software funcionando, de duas em duas semanas a dois em dois meses, com preferência à menor escala de tempo.**

Ciclos curtos de entrega (iterações) reduzem o risco, aumentam a previsibilidade e criam ciclos de feedback rápidos com o cliente.

- **7. Software funcionando é a medida primária de progresso.**

O progresso não é medido por documentos aprovados ou fases concluídas, mas sim por incrementos de produto real, testado e funcional. É a medida mais honesta de avanço.

- **8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.**

Este princípio é contra a cultura de "heróis" e de longas horas de trabalho ("crunch time"). Uma equipe ágil deve encontrar um ritmo de trabalho que possa ser mantido a longo prazo sem esgotamento.

- **10. Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial.**

Não se trata de fazer menos, mas de evitar o desperdício. O foco é em construir apenas o que é estritamente necessário para atender às necessidades atuais, evitando a superengenharia e funcionalidades que "talvez um dia sejam úteis".

### Colaboração e Excelência da Equipe

O sucesso do Ágil depende fundamentalmente das pessoas e da forma como elas interagem.

- **4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.**

Este princípio busca quebrar os silos de comunicação. A colaboração diária garante que a equipe de desenvolvimento esteja sempre alinhada com os objetivos de negócio.

- **5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário, e confie neles para fazer o trabalho.**

É um princípio de empoderamento. Equipes motivadas e com autonomia têm maior probabilidade de encontrar as melhores soluções.

- **6. O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é a conversa face a face.**

Embora a tecnologia tenha evoluído, o princípio da comunicação de alta fidelidade permanece. A comunicação direta (seja presencial ou por vídeo) é muito mais rica e menos suscetível a mal-entendidos do que longas trocas de e-mails ou documentos.

- **11. As melhores arquiteturas, requisitos e designs emergem de equipes que são auto-organizáveis.**

A inteligência coletiva da equipe, que está mais próxima do trabalho, tende a gerar soluções mais eficazes e elegantes do que aquelas ditadas de cima para baixo.

### Qualidade Técnica e Melhoria Contínua

Agilidade não é uma desculpa para o caos ou para a falta de qualidade técnica.

- **9. Contínua atenção à excelência técnica e bom design aumenta a agilidade.**

Um código limpo e bem projetado é mais fácil de manter e modificar. Investir em qualidade técnica não atrasa a equipe; pelo contrário, permite que ela continue a se mover rapidamente no futuro, reduzindo o "débito técnico".

- **12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento.**

Este é o motor da melhoria contínua. É a base para cerimônias como a "Retrospectiva" do Scrum. A equipe deve parar, inspecionar seu próprio processo e se adaptar para melhorar.

#### Conhecendo um pouco mais!

Vídeo

Entenda o que são os métodos ágeis em 5 minutos

[https://www.youtube.com/watch?v=ds\\_Fydzsu08](https://www.youtube.com/watch?v=ds_Fydzsu08)

ou aponte a câmera do seu smartphone para o Qr Code ao lado



## 2.2 Abordagem Ágil vs. Tradicionais: Um Comparativo

Com os valores e princípios do pensamento ágil estabelecidos, uma pergunta natural surge: o Ágil é sempre a melhor opção? A resposta é não. A eficácia de uma metodologia depende inteiramente do tipo de problema que se pretende resolver. Para entender essa distinção, podemos recorrer a uma poderosa ferramenta de tomada de decisão: o Framework Cynefin.



### Ferramenta-Chave: O Framework Cynefin

Desenvolvido por Dave Snowden, o Cynefin não é uma matriz de classificação, mas um framework de "sense-making" que nos ajuda a entender o ambiente em que estamos operando. Para a nossa discussão, dois de seus domínios são essenciais [10]:

**O Domínio Complicado:** Aqui, a relação entre causa e efeito existe e é previsível, mas requer análise de especialistas. Construir um motor de avião é *complicado*. Não há incerteza sobre as leis da física, mas é preciso um conhecimento profundo para planejar e executar corretamente. **É o terreno ideal para as metodologias tradicionais (Cascata)**, que se baseiam em análise, planejamento detalhado e execução precisa.

**O Domínio Complexo:** Aqui, a relação entre causa e efeito só pode ser entendida em retrospecto. Não há uma "resposta certa" a ser planejada. Desenvolver um novo aplicativo para um mercado incerto é *complexo*. É impossível saber de antemão o que os usuários irão querer. **É o terreno ideal para as metodologias ágeis**, que se baseiam em experimentar (construir um incremento), sentir a resposta do ambiente (feedback do cliente) e adaptar o próximo passo.

Portanto, a pergunta não é "qual é melhor?", mas sim "o meu problema é complicado ou complexo?".

Essa distinção fica clara com exemplos práticos. A construção de uma ponte é um projeto **complicado**; os requisitos são fixos e as mudanças são extremamente custosas, favorecendo

a abordagem tradicional. O desenvolvimento do Spotify, por outro lado, é um projeto **complexo**; a equipe precisa constantemente lançar novas funcionalidades, medir o engajamento dos usuários e adaptar o produto, o que exige uma abordagem ágil.

A tabela a seguir resume os principais pontos de contraste entre as duas abordagens.

Tabela 2.1: Principais Diferenças: Ágil vs. Tradicional (Cascata)

<b>Critério</b>	<b>Abordagem Tradicional</b>	<b>Abordagem Ágil</b>
<b>Planejamento</b>	Detalhado e completo no início (prescritivo).	De alto nível no início, com detalhes surgindo a cada ciclo (adaptativo).
<b>Mudanças</b>	Vistas como um problema a ser evitado; processo de controle de mudanças é formal e custoso.	Bem-vindas e esperadas; vistas como uma oportunidade para agregar valor.
<b>Entrega de Valor</b>	Tipicamente uma única entrega ao final do projeto.	Entregas frequentes e incrementais de produto funcional.
<b>Papel do Cliente</b>	Envolvimento concentrado no início (requisitos) e no final (aceitação).	Colaboração contínua e diária durante todo o projeto.

Com este framework mental e as principais diferenças em mente, podemos agora aprofundar a análise, detalhando como as abordagens Ágil e Tradicional se contrastam em aspectos práticos do dia a dia de um projeto, desde a forma como lidam com o planejamento e a documentação até o papel do cliente e a estrutura da equipe.

As metodologias ágeis são uma alternativa às abordagens tradicionais, por serem menos burocráticas e flexíveis, permitindo ajustes de direcionamento e escopo durante o processo de desenvolvimento. De uma maneira geral, as metodologias tradicionais são mais “pesadas” e devem ser utilizadas em situações em que os requisitos do sistema são estáveis. Por outro lado, em projetos nos quais os requisitos são passíveis de alterações e com equipes reduzidas, utilizar propostas mais ágeis tem se mostrado fundamental.

O processo de desenvolvimento das metodologias tradicionais baseia-se em um conjunto de atividades predefinidas, que invariavelmente se iniciam com o levantamento completo de um conjunto de requisitos, seguido por um projeto de alto nível e, consequentemente, de uma implementação, validação, até chegar à manutenção. Outro aspecto está relacionado com as atividades existentes nas metodologias tradicionais, ligadas a processos orientados a documentação, os quais, de certa forma, são considerados fatores limitadores aos desenvolvedores, pois muitas organizações não possuem recursos ou inclinação para processos burocráticos. Esta é uma das principais razões que levam organizações de pequeno porte a não usar nenhum tipo de processo, o que pode proporcionar resultados indesejados na qualidade do produto desenvolvido, dificultando a entrega nos prazos acordados e adequação aos custos predefinidos.

Existem vários aspectos responsáveis por diferenciar as metodologias tradicionais das ágeis, os quais podem ser observados em regras que explicam como produzir e conduzir processos, na forma peculiar de desenvolver software existente no ambiente de desenvolvimento e na maneira de pensar dos integrantes da equipe. Talvez esta última seja, de fato, a principal diferença que podemos observar, pois pensar diferente implica a adoção de princípios de desenvolvimento diferentes.

Outra característica importante das metodologias ágeis é que elas são preparadas para aceitar mudanças durante o processo de desenvolvimento de software, ao passo que as metodologias tradicionais são resistentes a elas. Metodologias ágeis são baseadas em dados estatísticos obtidos de históricos referentes à implementação do código. Já os métodos tradicionais utilizam como base normas que definem padrões a serem seguidos. A estratégia de trabalho adotada é imposta pela própria equipe de desenvolvimento, não exigindo um grande controle dos processos, o que é bem diferente do observado nas metodologias tradicionais, uma vez que percebemos claramente a imposição mais acirrada, por meio de inúmeras normas e políticas que devem ser respeitadas.

Quando pensamos em aspectos contratuais, diferentemente do que ocorre com as metodologias tradicionais, que são baseadas em rígidos contratos, as metodologias ágeis não pregam o uso de contratos, pois essa postura reforça o princípio de que as mudanças são bem-vindas. Se o contrato tiver de ser estabelecido, que seja o mais flexível possível, de forma a resguardar os interesses da equipe de desenvolvimento e do cliente.

A participação dos clientes também acontece de forma diferenciada. Nas metodologias tradicionais, mesmo podendo participar de reuniões, não possuem poder de decisão no que diz respeito à forma de desenvolvimento do projeto. Nas metodologias ágeis, o cliente faz parte da equipe de desenvolvimento, podendo auxiliar e indicar o rumo que a implementação pode tomar.

Finalmente, podemos citar o reduzido tamanho das equipes de desenvolvimento de software que utilizam paradigmas ágeis, em torno de dez indivíduos que trabalham em um mesmo local. Projetos que usam metodologias tradicionais geralmente possuem uma grande quantidade de integrantes na equipe, muitas vezes geograficamente dispersos.

Percebemos também diferenças quanto ao custo gerado com mudanças ao longo do desenvolvimento do software. Considerando seus princípios, as metodologias ágeis já estão preparadas para aceitar mudanças no projeto, pois estão focadas nas pessoas e não nos processos, e por natureza denotam um controle rígido deles. Nas metodologias tradicionais, à medida que alterações são necessárias na fase próxima de seu encerramento, seu custo tende a crescer exponencialmente. Nas metodologias ágeis o custo pode crescer, mas não nas mesmas proporções que nas metodologias tradicionais.

Um fator que contribui para que o custo se mantenha estável nas metodologias ágeis é que elas não estão amarradas a contratos ou documentações, além de sua iteratividade, que possibilita desenvolver o sistema incrementalmente. É importante lembrar que, mesmo considerando as vantagens das metodologias ágeis no desenvolvimento de software, seu uso sempre deve estar vinculado ao contexto do software que será desenvolvido. Certamente softwares cuja precisão e confiabilidade são fatores de decisão e risco, como ocorre com sistemas cirúrgicos ou aeroviários, não podem dispensar as recomendações implícitas características das metodologias tradicionais, principalmente quanto à necessidade de documentar o que deve ser feito. A Tabela 1 apresenta algumas diferenças básicas entre as metodologias tradicionais e ágeis.

Tabela 2.2: Comparativo detalhado entre Metodologias Tradicional e Ágil.

<b>Critério</b>	<b>Metdologia Tradicional</b>	<b>Metdologia Ágil</b>
Planejamento	Comumente realizado em detalhe para todo o projeto em sua fase inicial.	Planejamento de alto nível no início do projeto e os detalhes são realizados durante o projeto. Não é necessário possuir um planejamento detalhado de todo o projeto. A restrição se dá apenas em possuir os detalhes do trabalho para a próxima iteração.
Riscos	Pode exigir um grande esforço e equipe para atuar com os riscos de todo o projeto.	Prioriza os riscos gerais do projeto, mas foca principalmente nos riscos das próximas iterações, atuando assim em um escopo bem reduzido. A própria equipe atua com os riscos e pode obter apoio externo.
Equipe	Possui profissionais com papéis bem definidos, quantificada e mobilizada conforme o planejamento do projeto. A equipe executa o projeto guiado pelo Gerente de Projetos conforme o plano estabelecido.	Equipe multidisciplinar, multifuncional e auto-organizada. Ela decide como fazer e atua de forma colaborativa.
Tempo de entrega	É realizado conforme o plano estabelecido e pode durar semanas, meses ou até mesmo anos.	Fixo e é conforme a definição de duração das iterações que comumente varia entre 1 e 4 semanas.
Aceitação de mudanças	Gerenciamento formal de mudanças, pois exige alteração do plano já realizado e geralmente precisa passar por aprovações formais de um ou mais níveis hierárquicos.	Mudanças são bem-vindas. Evita-se mudar o escopo da iteração em andamento, mas o escopo das futuras iterações podem ser replanejado conforme a necessidade do cliente.
Previsibilidade	Depende do intervalo de monitoramento e controle do projeto. Quanto mais curto, maior a chance de prever as ocorrências futuras. Quanto maior o intervalo, menor a chance de prever as ocorrências futuras.	Tende a ter uma grande previsibilidade futura devido à constante análise e feedback através das oportunidades de inspeção e adaptação providas pelo método.
Resultados ao longo do tempo	Tende a demorar a dar resultados a curto prazo, pois as entregas são geralmente realizadas ao fim do projeto. Melhores resultados são visíveis em projetos de maior duração.	Gera resultados a curto, médio e longo prazo, pois atua com entregas antecipadas e de valor agregado e contínuo ao cliente.

*Continua na próxima página...*

Tabela 2.2: (Continuação)

<b>Critério</b>	<b>Metdologia Tradicional</b>	<b>Metdologia Ágil</b>
Apresentação de informações	Geralmente é formal e previamente agendada com os stakeholders em intervalos de tempo. As informações podem ser detalhadas ou não conforme a necessidade do público envolvido.	Geralmente informal e utiliza radiadores de informação no ambiente de trabalho durante todo o projeto, de modo que as informações do projeto fiquem visíveis e transparentes a toda equipe e envolvidos.
Prazo de entrega	Conforme estabelecido no planejamento do projeto. No caso de mudanças aprovadas, varia conforme os impactos das solicitações e podem ser traumáticas aos envolvidos quanto às suas expectativas.	Conforme o tamanho da iteração e o planejamento das releases para as entregas significativas.
Documentação	Detalhada desde o início do projeto.	Abrangente no início e detalhada somente o necessário durante o projeto conforme os objetivos das iterações e releases.
Atuação do cliente	Nas fases iniciais e nas principais validações do produto.	Durante todo o projeto, o cliente faz parte da equipe.
Discussão e melhorias	Geralmente em prazos longos através da realização de reuniões após uma etapa ou grande entrega do projeto.	Em prazos curtos, sempre ao final das iterações.
Comandante	Gerente de Projetos.	Equipe do Projeto.
Papéis	Claros e definidos.	Conforme a confiança na equipe e ambiente colaborativo.
Processo	Guiado conforme o plano do projeto e nos processos estabelecidos no plano.	Empírico e guiado ao produto e às pessoas. Orientado à geração de valor e conforme priorização dos riscos.
Resultado	Melhor resultado em projetos com escopo muito bem definido e orientado a planejamento.	Melhor resultado em projetos cujo escopo é dinâmico e construído durante a execução do projeto.



Fica claro, portanto, que a escolha da metodologia não é uma questão de preferência, mas de adequação ao contexto. Não se trata de uma batalha entre "velho" e "novo", mas de um entendimento maduro sobre a natureza do problema a ser resolvido. Compreender essa distinção é o primeiro passo para liderar projetos de sucesso. No próximo capítulo, mergulharemos fundo no mais popular dos frameworks ágeis, o Scrum, para entender como ele coloca em prática os valores e princípios que discutimos aqui.

### 2.3 Aplicando seus conhecimentos

1. O Manifesto Ágil é uma base conceitual para metodologias ágeis, composto por 12 princípios que orientam equipes a desenvolverem produtos e projetos de forma colaborativa, adaptável e centrada no cliente, para esta atividade.
  - (a) Leia os 12 princípios do Manifesto Ágil, em grupos, discuta e responda.
  - (b) Qual princípio você considera mais importante? Por quê?
  - (c) Dê um exemplo prático de como esse princípio pode ser aplicado em um projeto da vida real.
2. Imagine que você faz parte de uma equipe responsável pelo projeto de gestão de uma fazenda. Divida-se em dois grupos, onde um grupo deverá representar as metodologias ágeis, enquanto o outro representará as metodologias tradicionais. Cada grupo deve responder às seguintes perguntas com base na abordagem que representam:
  - (a) Como vocês lidariam com mudanças no projeto durante a execução?
  - (b) De que forma garantiriam que o produto seja entregue dentro do prazo e da qualidade esperada?
  - (c) Como organizariam a equipe para alcançar os objetivos do projeto?
  - (d) Após a discussão, compare as respostas dos dois grupos, destacando quais foram as principais diferenças entre as duas abordagens? E quais benefícios as metodologias ágeis apresentam em relação às tradicionais em cenários que exigem maior flexibilidade e adaptação?

### 2.4 Considerações do módulo

Neste segundo módulo, você foi apresentado ao conceito de metodologias ágeis e às suas principais características, com foco na flexibilidade e adaptabilidade durante o desenvolvimento de projetos. Vimos que o Manifesto Ágil é um marco importante para essas metodologias, trazendo à tona valores como a valorização das pessoas e interações, a entrega contínua de produtos funcionais, e a importância de colaborar com o cliente.

As metodologias ágeis oferecem uma abordagem diferente dos métodos tradicionais, permitindo ajustes constantes ao longo do projeto e facilitando a resposta a mudanças de forma mais eficiente. Essa flexibilidade é essencial em um ambiente de desenvolvimento onde as demandas e tecnologias estão sempre mudando.

# Capítulo 3

## Introdução ao Scrum

### Iniciando o diálogo...

Neste capítulo, vamos mergulhar no framework ágil mais utilizado no mundo: o Scrum. Longe de ser uma metodologia prescritiva, o Scrum é um framework simples e leve que ajuda pessoas, equipes e organizações a gerar valor através de soluções adaptativas para problemas complexos.



Figura 3.1: Os Três Pilares do Empirismo: a base do Scrum.

Fonte: Thiago Anastácio, 2020

### 🎯 Objetivos do Capítulo

Ao final deste capítulo, você será capaz de:

- Compreender o Scrum como um framework fundamentado no **empirismo**.
- Identificar e descrever as responsabilidades dos três papéis do Time Scrum: **Product Owner**, **Scrum Master** e **Desenvolvedores**.
- Detalhar o propósito e a dinâmica dos cinco eventos do Scrum: a **Sprint**, o **Planejamento**, a **Reunião Diária**, a **Revisão** e a **Retrospectiva**.
- Descrever os três artefatos do Scrum: o **Product Backlog**, o **Sprint Backlog** e o **Incremento**.

### 3.1 O que é SCRUM?

Para entender o Scrum, imagine que ele não é um manual de instruções detalhado, mas sim uma caixa de LEGOs com algumas regras simples. Ele não diz o que construir, mas oferece as peças (papéis, eventos, artefatos) e as regras de encaixe (o framework) para que uma equipe possa construir, inspecionar e adaptar qualquer produto complexo de forma eficaz. Oficialmente, segundo seus criadores, Ken Schwaber e Jeff Sutherland, o Scrum é um "framework leve que ajuda pessoas, equipes e organizações a gerar valor através de soluções adaptativas para problemas complexos". A palavra framework é fundamental aqui, pois ele é intencionalmente incompleto, servindo como um contêiner onde diversas outras técnicas e práticas podem ser aplicadas.

As raízes do Scrum são mais antigas do que o Manifesto Ágil e remontam a um influente artigo de 1986 da Harvard Business Review, intitulado "The New New Product Development Game", de Hirotaka Takeuchi e Ikujiro Nonaka [11]. No artigo, eles descreveram uma abordagem de desenvolvimento de produtos mais holística, rápida e flexível, comparando as equipes de alta performance a um time unido e coeso de Rugby – daí a origem do termo "Scrum". Essa abordagem surgiu como uma resposta direta às falhas dos modelos sequenciais e rígidos, como o Waterfall, que se mostravam ineficazes para lidar com a incerteza e a complexidade dos projetos modernos.

A base filosófica do Scrum é o empirismo, a teoria de que o conhecimento provém da experiência e da tomada de decisões com base no que é observado. Para que o empirismo funcione na prática, ele se apoia em três pilares fundamentais: Transparência, Inspeção e Adaptação.

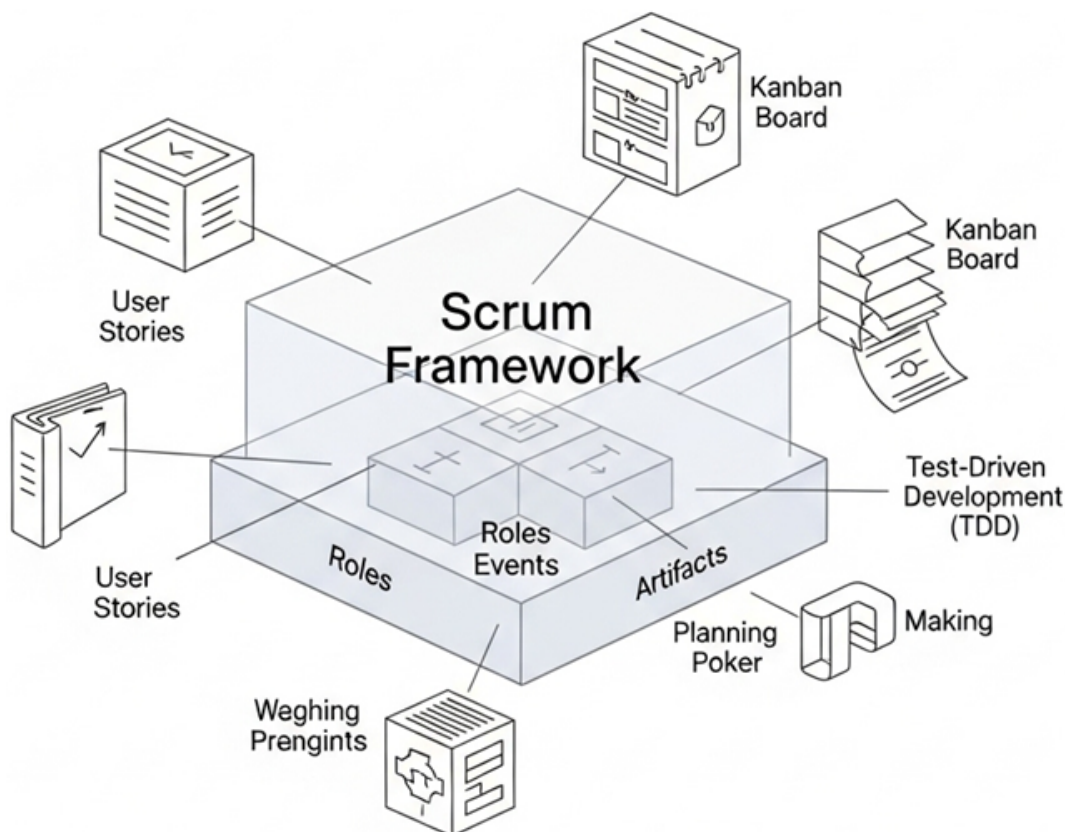


Figura 3.2: Scrum como um Framework Contêiner

Fonte: Gerada por Google Gemini 2.5 Pro, 2025

**1. Transparência:** O estado real do trabalho, os desafios e o progresso devem ser visíveis e compreendidos por todos os envolvidos – tanto a equipe que realiza o trabalho quanto as pessoas que o recebem. A transparência vai além de um quadro de tarefas; ela exige uma linguagem comum e padrões claros. No projeto da fazenda, a transparência seria garantida se toda a equipe, do agrônomo ao desenvolvedor do aplicativo, compartilhasse o mesmo Product Backlog priorizado e entendesse que a tarefa “implementar sensor de umidade” só está “Pronta” quando o sensor está instalado, coletando dados e exibindo-os de forma funcional no aplicativo, conforme um critério predefinido.

### Fique Alerta!

#### A Definição de Pronto (Definition of Done - DoD)

Um dos conceitos mais críticos para garantir a transparência é a “Definição de Pronto”. Trata-se de um acordo formal e compartilhado por todo o Time Scrum, que descreve todos os critérios de qualidade e trabalho que uma funcionalidade deve atender para ser considerada “concluída” e parte do Incremento. Sem um DoD claro, a equipe corre o risco de gerar mal-entendidos e entregar um trabalho que não possui a qualidade necessária, comprometendo o valor do produto.

**2. Inspeção:** Os artefatos do Scrum e o progresso em direção às metas acordadas devem ser inspecionados frequentemente e diligentemente para detectar variações ou problemas indesejados. A inspeção não deve ser vista como microgerenciamento, mas como uma ferramenta para a equipe avaliar sua própria trajetória. Essa inspeção ocorre em eventos específicos do Scrum. Por exemplo, a equipe da fazenda inspeciona seu progresso em direção à meta da Sprint todos os dias durante a Daily Scrum, e o produto em si é inspecionado junto aos stakeholders durante a Sprint Review.

**3. Adaptação:** Se a inspeção revelar que um ou mais aspectos do processo ou do produto se desviaram dos limites aceitáveis, um ajuste deve ser feito o mais rápido possível para minimizar desvios futuros. A adaptação é a ação corretiva. Se, durante a Sprint Review, o agricultor (stakeholder) percebe que o protótipo do sistema de irrigação não é prático, a equipe adapta o plano para a próxima Sprint, ajustando os itens e as prioridades do Product Backlog com base nesse valioso feedback. O ciclo de Transparência, Inspeção e Adaptação é o motor que impulsiona a melhoria contínua e a agilidade no Scrum.

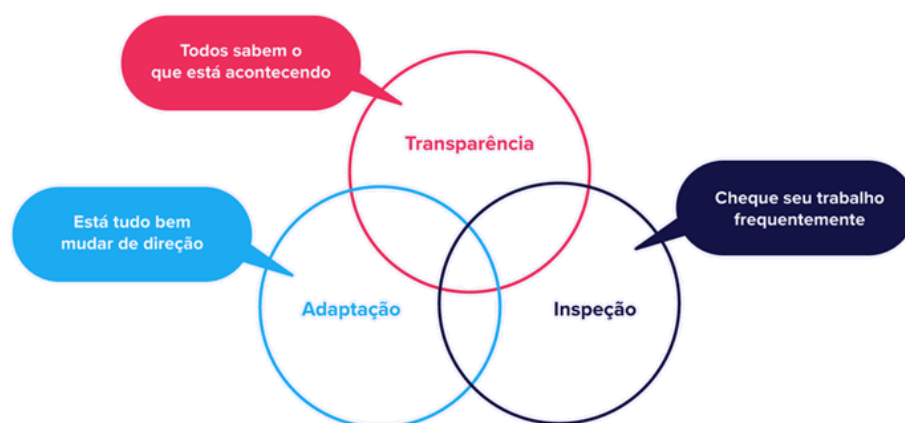


Figura 3.3: O Ciclo do Empirismo no Scrum.  
Fonte: Adaptado de Felipe Guimaraes e Equipe Aela, 2022

## 3.2 Ciclo de desenvolvimento

Agora que entendemos a filosofia por trás do Scrum, vamos conhecer os elementos que compõem seu framework. Estes elementos são projetados para dar vida ao ciclo de transparência, inspeção e adaptação.

### O Time Scrum

**Product Owner :** ("dono" do produto): representa o cliente e é responsável por garantir que a equipe SCRUM agregue valor ao negócio. Portanto, desempenha o papel de moderador entre os interesses do cliente e do Team, tendo como responsabilidade manter a equipe funcional e produtiva. O Product Owner é responsável por:

- Definir a visão e as funcionalidades do produto;
- Definir as prioridades;
- Elaborar e manter o Product Backlog;
- Definir as prioridades e o ROI (Return of Investment);
- Decidir sobre as datas de lançamento do produto;
- Representar o cliente (quando este não está presente);
- Aceitar ou rejeitar os resultados dos trabalhos.

**SCRUM Master:** (mestre SCRUM): representante do cliente no projeto e desempenha um papel importante de facilitador, responsável pela remoção de impedimentos (problemas técnicos, administração de conflitos, itens não planejados) que eventualmente surjam durante o desenrolar do desenvolvimento. Atua na definição de funcionalidades de acordo com seu valor para o cliente, planejando e elaborando em conjunto com o Product Owner uma lista de prioridades. Portanto, o SCRUM Master desempenha um papel de responsabilidade técnica na condução do projeto, mantendo a equipe focada nas suas tarefas. O SCRUM Master é responsável por:

- Desempenhar o papel de líder, representando a gerência do projeto;
- Remover impedimentos;
- Proteger a equipe SCRUM;
- Ajudar o Product Owner com o Product Backlog;
- Ser o facilitador da equipe SCRUM, garantindo sua plena produtividade;
- Garantir a colaboração entre os diversos papéis e funções;
- Atuar como escudo para interferências externas;
- Aplicar os valores e as práticas SCRUM.

**Desenvolvedores:** São as pessoas do Time Scrum que se comprometem a criar qualquer aspecto de um Incremento utilizável a cada Sprint. As habilidades específicas dos Desenvolvedores são frequentemente amplas e variam com o domínio do trabalho. No entanto, todos os Desenvolvedores são responsáveis por:

- Criar um plano para a Sprint, o Sprint Backlog;
- Injetar qualidade ao aderir a uma Definição de Pronto;
- Adaptar seu plano a cada dia em direção à Meta da Sprint; e,
- Manterem-se mutuamente como profissionais responsáveis.

No caso da fazenda, a equipe de Desenvolvedores seria multifuncional, incluindo engenheiros de software, especialistas em hardware para os sensores e talvez um designer de UX para o aplicativo.

### Os Eventos do Scrum

**Sprints:** corresponde às iterações. O objetivo é gerar um produto "entregável" de valor para o cliente, que foi previamente combinado com ele. Cada sprint deve ocorrer em um período de duas a quatro semanas. O produto é projetado, codificado e testado durante a sprint. As tarefas escolhidas para fazerem parte de uma sprint devem ser retiradas de outro documento, denominado Product Backlog, que contém um conjunto de requisitos que representam o trabalho que deve ser feito. No final de cada sprint, outra reunião deve ser realizada, para revisar o que foi feito, avaliar o progresso e identificar lições aprendidas para serem usadas na próxima sprint.

**Planejamento da sprint ou Sprint Planning Meeting:** é a primeira reunião do projeto e todos precisam participar; deve ter uma duração de no máximo oito horas. Durante o planejamento da sprint podemos utilizar uma técnica que estima o tamanho do trabalho a ser realizado. Esta é a reunião em que o Product Owner planeja e elabora a lista de prioridades que devem ser cumpridas pelo projeto, dividindo a reunião em duas partes:

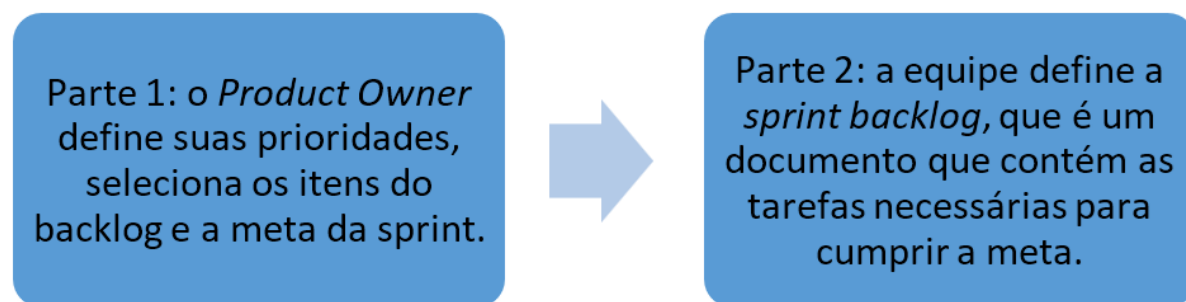


Figura 3.4: Etapas do planejamento da sprint

Fonte: Elaborada pelos autores

**Reunião diária ou Daily SCRUM:** é uma reunião que cada membro do time deve responder sobre o que já fez, sobre o que pretende fazer e se há algum impedimento para a conclusão da(s) tarefa(s) sob sua responsabilidade. A reunião diária deve ter uma duração muito rápida, preferencialmente de no máximo quinze minutos, tendo como participantes apenas o time e o SCRUM Master. Cada membro do time deve responder a três perguntas:

1. O que eu fiz desde a última reunião?
2. O que vou fazer até a próxima?
3. Tive ou estou tendo algum impedimento? Quais?

**Revisão da sprint ou Sprint Review:** é uma reunião de balanço sobre tudo o que foi feito durante uma sprint. Nessa reunião o time deve mostrar os resultados da sprint para o Product Owner e seus convidados. Observe que devem ser apresentados somente os itens que estiverem 100% prontos, ou seja, se faltou uma única atividade, o item não deve ser apresentado. Essa reunião deve ter uma duração estimada de quatro horas e precisa contar com os seguintes participantes: Product Owner, SCRUM Master, Equipe SCRUM e outros convidados. Deve-se marcar essa reunião sempre no final da sprint. Os objetivos esperados após a reunião:

- Apresentar o que a equipe fez durante a sprint;
- Entregar o produto (software funcionando) ao Product Owner (geralmente uma demo da parte implementada).

Após a apresentação, o Product Owner conversa com seus convidados e tem o direito de aceitar ou rejeitar a sprint com base no que foi apresentado. Qualquer necessidade de mudança ou inserção de novas features (funcionalidades) será incorporada ao Product Backlog em um momento oportuno, além de priorizada novamente.

**Retrospectiva da sprint ou Sprint Retrospective:** o objetivo dessa reunião é verificar o que houve de bom e o que pode ser melhorado em uma sprint. São avaliados aspectos relacionados ao trabalho em equipe, pontos positivos e negativos, e feitas reflexões sobre estratégias de melhoria que podem ser adotadas. Devem participar da reunião, obrigatoriamente, o time e o SCRUM Master. O Product Owner também pode participar, sempre que convidado. Todos os membros do time devem responder basicamente a duas perguntas: o que foi bom durante a sprint e o que se pode fazer para melhorar a próxima.

O SCRUM Master deve tomar nota de tudo e o time deve priorizar os itens apontados em uma ordem ideal de mudança. A retrospectiva é uma excelente forma de garantir a melhoria contínua do processo. Essa reunião deve acontecer logo após a revisão da sprint, com duração aproximada de três horas.

### Os Artefatos do Scrum

**Product Backlog:** é um documento que representa a visão do produto de forma modular, contendo todos os itens que devem ser desenvolvidos durante o projeto. Basicamente é uma lista de prioridades feitas logo no início do projeto, com o objetivo de esclarecer e elencar o que deve ser entregue para o cliente. Esses itens devem ser escritos de forma clara e simples, de fácil entendimento tanto para o time de desenvolvimento quanto para o cliente. O Product Backlog deve ser criado e mantido pelo Product Owner, que tem a liberdade de alterar esse documento quando quiser, desde que os itens alterados não estejam na sprint que estiver sendo desenvolvida no momento.

Os itens do Product Backlog devem ser priorizados em função do ROI (Return of Investment), ou seja, os que apresentarem maior valor para o negócio devem ser desenvolvidos primeiro. O tamanho de cada item deve ser estimado pelo time de desenvolvimento. O Product Owner deve manter e priorizar o Product Backlog constantemente.



Até este ponto, apresentamos todas as 'peças de LEGO' que compõem o Scrum: o Time (quem executa), os Eventos (quando o trabalho acontece) e os Artefatos (o que guia e o que é produzido). Agora que conhecemos cada componente individualmente, podemos finalmente montar o quebra-cabeça. O texto e a figura a seguir ilustram como todos esses elementos se conectam e interagem para formar o ciclo de desenvolvimento completo, transformando as necessidades do cliente em um Incremento de valor a cada Sprint.

Tabela 3.1: Exemplo de um Product Backlog priorizado.

Prioridade (ROI)	Ator	Req. Func.	Descrição do Item	Tamanho	Release	Sprint	Status
Essencial	Usuário	RF001	Cadastro de usuário				To Do
Essencial	Usuário	RF002	Cadastro de currículo				To Do
Essencial	Usuário	RF003	Cadastro de interesse				To Do
Essencial	Usuário	RF004	Busca de oportunidades de emprego				To Do
Essencial	Empresas	RF005	Cadastro de empresas				To Do
Essencial	Empresas	RF006	Busca de candidato a emprego				To Do

Uma prática essencial para manter o Product Backlog organizado e eficiente, também conhecido como "refinamento do backlog". Essa atividade ocorre regularmente e tem como objetivo revisar, ajustar e priorizar os itens do backlog, garantindo que estejam claros e prontos para serem desenvolvidos nos próximos sprints, alguns pontos a serem considerados.

- **Dividir tarefas grandes:** Quebrar itens complexos em partes menores e mais gerenciáveis;
- **Estimar esforço:** Avaliar o tamanho ou esforço necessário para concluir cada item;
- **Ajustar prioridades:** Reordenar os itens do backlog com base no feedback do cliente ou em mudanças no projeto.

Essa prática é conduzida de forma colaborativa, envolvendo o **Product Owner**, o **Scrum Master** e a equipe de desenvolvimento, e ajuda a alinhar todos os membros sobre o trabalho que será realizado. Embora não seja uma cerimônia formal do Scrum, o **Backlog Refinement** é fundamental para preparar a equipe para os sprints, aumentando a produtividade e a clareza sobre os próximos passos.

**Sprint Backlog:** é um artefato oriundo da Sprint Planning Meeting e representa todas as tarefas que devem ser desenvolvidas durante uma sprint ou iteração. Cada item deve ser detalhado em tarefas e cada uma dessas tarefas deve ter uma estimativa de esforço, neste caso, em horas.

Tabela 3.2: Exemplo de Sprint Backlog com detalhamento de tarefas e horas.

Item	Tarefa	1	2	3	4	5	6	7	8	9	10	Total
ES006	Criar Base de Dados	4	0	0	0	0	0	0	0	0	0	4
	Desenvolver Modelo	8	8	0	0	0	0	0	0	0	0	16
	Desenvolver Controle	0	0	4	4	0	0	0	0	0	0	8
	Desenvolver View	0	0	0	4	4	0	0	0	0	0	8
	Teste de Unidade	0	0	0	0	4	2	2	0	0	0	8
	Teste Funcional	0	0	0	0	0	4	4	2	0	0	10
	Documentação Técnica	0	0	0	0	0	0	4	4	2	2	12

**Task Board** é um quadro utilizado para o acompanhamento das sprints, durante as reuniões diárias. Por meio das informações contidas neste registro, aliadas ao seu posicionamento no task board, torna-se possível a qualquer um observar o andamento do projeto, de maneira clara e intuitiva.

Tabela 3.3: Estrutura de um Quadro de Tarefas (Task Board).

Product Backlog	To Do	Doing	Done
Sprint Backlog	Unplanned Items	Impediments	To Discuss

**Incremento:** O Incremento é um passo concreto em direção à Meta do Produto. Cada Incremento é aditivo a todos os Incrementos anteriores e verificado, garantindo que todos funcionem juntos. Para fornecer valor, o Incremento deve ser utilizável. Múltiplos Incrementos podem ser criados dentro de uma Sprint. A soma dos Incrementos é apresentada na Sprint Review, apoiando assim o empirismo. No entanto, um Incremento pode ser entregue aos stakeholders antes do final da Sprint. A Sprint Review nunca deve ser considerada um portão para liberar valor. O momento em que o trabalho se torna um Incremento é quando ele atende à Definição de Pronto (DoD).

## Resumo do ciclo de desenvolvimento SCRUM

Até este ponto, apresentamos todas as 'peças de LEGO' que compõem o Scrum: o Time (quem executa), os Eventos (quando o trabalho acontece) e os Artefatos (o que guia e o que é produzido). Agora que conhecemos cada componente individualmente, podemos finalmente montar o quebra-cabeça. O texto e a figura a seguir ilustram como todos esses elementos se conectam e interagem para formar o ciclo de desenvolvimento completo, transformando as necessidades do cliente em um Incremento de valor a cada Sprint.



Figura 3.5: Ciclo de desenvolvimento da metodologia Scrum

Fonte: Cohn, 2005

O ciclo de desenvolvimento da metodologia SCRUM pode ser observado na figura 3.5, e ilustra o fato de que no início de cada projeto, clientes e desenvolvedores se reúnem com o objetivo de definir o **Backlog** do produto (que representa a **lista de requisitos**). Nesse momento também são estimados os custos do projeto e é feita uma análise dos riscos, bem como as ferramentas de trabalho e os integrantes da equipe são escolhidos e são definidas as datas para entrega de resultados a partir de prioridades sinalizadas pelo cliente. Esse período também é utilizado para escolher o SCRUM Master, eleito entre a equipe alocada para o projeto.

Depois que o Product Backlog é definido, a equipe deve se dedicar à definição da Sprint Backlog, que contém uma lista de atividades que serão realizadas na próxima sprint, momento em que também são definidas as responsabilidades de cada membro do time. Após os desenvolvedores discutirem quais padrões serão adotados, as atividades de análise, codificação e testes devem se iniciar. Ao final de cada sprint um incremento do produto deve ser apresentado ao cliente para que o time obtenha uma retroalimentação. Caso algum defeito seja encontrado, deve ser adicionado ao backlog do produto. Na medida em que o ciclo de desenvolvimento ocorre, são aplicados diversos mecanismos de controle do SCRUM, como, por exemplo, ações sobre as funcionalidades não entregues, a necessidade de mudanças para correção de defeitos, problemas técnicos encontrados, bem como os riscos e as estratégias para evitá-los.

### 3.3 Aplicando seus conhecimentos

1. Qual é o principal objetivo do Scrum?
  - (a) Criar documentação extensa
  - (b) Entregar produtos de forma rápida e incremental
  - (c) Estabelecer um cronograma rígido de trabalho
  - (d) Aumentar o número de reuniões
2. Quais são os três papéis principais na metodologia Scrum?
  - (a) Gerente de Projetos, Analista de Negócios, Desenvolvedor

- (b) Product Owner, Scrum Master, Equipe de Desenvolvimento
  - (c) Cliente, Fornecedor, Consultor
  - (d) Coordenador, Supervisor, Executor
3. Qual é a duração típica de uma Sprint no Scrum?
- (a) 1 semana
  - (b) 2 a 4 semanas
  - (c) 1 mês
  - (d) 3 meses
4. O que é o Product Backlog?
- (a) Um documento que lista as funcionalidades do produto em ordem de prioridade
  - (b) Um cronograma de entregas
  - (c) Um registro de problemas encontrados durante o projeto
  - (d) Um relatório de desempenho da equipe
5. Qual das seguintes cerimônias do Scrum ocorre no início de cada Sprint?
- (a) Sprint Review
  - (b) Daily Scrum
  - (c) Sprint Retrospective
  - (d) Sprint Planning Meeting
6. Descreva as principais responsabilidades do Product Owner e do Scrum Master.
7. Explique a importância das reuniões diárias (Daily Scrum) e como elas contribuem para a eficiência da equipe.
8. Quais são os principais benefícios de utilizar a metodologia Scrum em projetos de desenvolvimento?
9. Como o feedback dos clientes é incorporado no processo Scrum?
10. Agora você faz parte de uma equipe ágil, novamente fica como sugestão o projeto da gestão de uma fazenda. Divida-se em grupos e atribua papéis específicos para cada integrante: Product Owner, Scrum Master e Desenvolvedores.
- (i) Durante uma simulação de reunião de planejamento, o Product Owner deverá apresentar o problema e definir as prioridades do projeto;
  - (ii) O Scrum Master será responsável por garantir que o processo seja seguido corretamente e por remover quaisquer impedimentos que surjam;
  - (iii) Já os Desenvolvedores deverão discutir as possíveis soluções e planejar as tarefas necessárias para alcançar os objetivos definidos.
  - (iv) Após a simulação, cada grupo deve compartilhar suas experiências, destacando os desafios enfrentados ao desempenhar cada papel e como as funções colaboraram para o sucesso do projeto.

## 3.4 Considerações do módulo

Ao longo deste módulo, você teve a oportunidade de explorar o Scrum, uma metodologia ágil que transforma a gestão de projetos, promovendo eficiência, colaboração e inovação. Aprendemos sobre os papéis fundamentais dentro da equipe Scrum—o Product Owner, o Scrum Master e a equipe de desenvolvimento—e como cada um deles contribui para o sucesso do projeto.

As cerimônias, como o Sprint Planning, Daily Scrum, Sprint Review e Sprint Retrospective, foram abordadas, mostrando como cada uma delas é essencial para manter a equipe alinhada, focada e em constante melhoria. A importância de um Product Backlog bem definido e priorizado, junto com um Sprint Backlog claro, garante que a equipe esteja sempre trabalhando nas tarefas mais valiosas e relevantes para o cliente.

Compreender o ciclo de desenvolvimento, as iterações (sprints) e a aplicação prática dos artefatos Scrum, como o Burndown Chart e o Task Board, é crucial para implementar essa metodologia de forma eficaz em sua equipe. Ao aplicar essas práticas, você estará apto a criar um ambiente de trabalho mais dinâmico e responsivo, capaz de se adaptar rapidamente às mudanças e feedbacks.

Por fim, lembre-se de que o Scrum é mais do que um conjunto de regras e práticas; é uma filosofia que coloca as pessoas e a colaboração no centro do processo. Ao incorporar esses princípios em sua abordagem de gestão de projetos, você não apenas alcançará melhores resultados, mas também contribuirá para um ambiente de trabalho mais motivador e inovador. Agora, com esse conhecimento, você está pronto para aplicar o Scrum e transformar a forma como sua equipe desenvolve produtos e soluções.

### Conhecendo um pouco mais!

E-book

O Guia Definitivo para o Scrum: As Regras do Jogo

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-European.pdf>

ou aponte a câmera do seu smartphone para o Qr Code ao lado



### Conhecendo um pouco mais!

E-book

SCRUM para projetos ágeis

[https://sebrae.com.br/Sebrae/Portal%20Sebrae/Arquivos/ebook\\_sebrae\\_scrum-para-projetos-aggais.pdf](https://sebrae.com.br/Sebrae/Portal%20Sebrae/Arquivos/ebook_sebrae_scrum-para-projetos-aggais.pdf)

ou aponte a câmera do seu smartphone para o Qr Code ao lado



# Capítulo 4

## Experimento Prático

### Iniciando o diálogo...

Olá, estudante. Aqui, você terá a oportunidade de aplicar **Design Thinking** para entender e solucionar desafios na gestão de uma fazenda, utilizando a experiência dos trabalhadores como base para melhorias. Além disso, usaremos o framework ágil **Scrum** para gerenciar e implementar essas mudanças de forma colaborativa e organizada.



Figura 4.1: Ilustração de reunião para gestão da fazenda  
Fonte: Gerada por Google Gemini 2.5 Pro, 2025

### 🎯 Objetivos do Capítulo

Ao final deste capítulo, você será capaz de:

- Explorar desafios reais relacionados à sustentabilidade e eficiência na gestão rural.
- Incentivar o trabalho em equipe e a aplicação prática de Design Thinking.
- Desenvolver soluções criativas que conectem diferentes aspectos da fazenda de forma integrada.
- Preparar você estudante para enfrentar problemas do mundo real com inovação e colaboração.

## 4.1 Gestão sustentável de uma fazenda

Neste exercício, você será guiado(a) pelas etapas de um projeto real de reestruturação, desde a compreensão das necessidades dos trabalhadores até a implementação de novas práticas. Esta será uma excelente oportunidade para experimentar duas metodologias amplamente usadas no mercado, integrando criatividade e eficiência no processo de gestão.

O foco é criar um **Produto Mínimo Viável (MVP)**, ou seja, uma versão simplificada e inicial das melhorias que podem ser implementadas em uma fazenda, com base nas ideias desenvolvidas em equipe. Vamos lá?

Esta atividade visa integrar o aprendizado prático do Design Thinking com temas reais relacionados à sustentabilidade e à gestão de uma fazenda. Como sugestão ao professor, os estudantes podem ser divididos em quatro grupos, cada um com um tema central, e serão responsáveis por explorar e desenvolver soluções criativas para um aspecto específico da fazenda.

### Passo a Passo da Atividade

#### 1. Divisão de Grupos e Temas

- **Grupo 1:** Estrutura Física da Fazenda (Automação, Sensoriamento, Sistemas de Energia Renovável).
- **Grupo 2:** Captação e Tratamento da Água (Reservatórios, Filtros, Sensores, Monitoramento).
- **Grupo 3:** Agricultura Familiar (Cultivo de Vegetais e Legumes para Subsistência e Comercialização).
- **Grupo 4:** Criação de Animais (Bovinos, Suínos, Peixes e Integração com Agricultura).

#### 2. Direcionamento sobre os temas aos grupos: Cada grupo deverá identificar os principais desafios do seu tema e propor soluções criativas e viáveis, alinhadas ao contexto da fazenda e à sustentabilidade.

##### • Grupo 1: Estrutura Física da Fazenda

- **Desafios:** Falta de automação, alto consumo de energia e ausência de sistemas renováveis.
- **Exemplo de Solução:**
  - \* Implementar sensores alimentados por energia solar para monitorar temperatura e umidade nas plantações;
  - \* Desenvolver um aplicativo que permita o controle remoto dos sistemas automatizados.

##### • Grupo 2: Captação e Tratamento de Água

- **Desafios:** Desperdício de água e dificuldade em monitorar sua qualidade e distribuição.
- **Exemplo de Solução:**
  - \* Criar reservatórios inteligentes com sensores para medir níveis de água e alertar sobre a necessidade de manutenção;



- \* Projetar um sistema automatizado de irrigação gota-a-gota conectado a um aplicativo.
  - **Grupo 3: Agricultura Familiar**
    - **Desafios:** Baixa produtividade devido à falta de infraestrutura e dificuldades no manejo.
    - **Exemplo de Solução:**
      - \* Implantar estufas automatizadas que otimizem o uso de água e nutrientes;
      - \* Desenvolver um sistema de cultivo integrado com aquaponia, aproveitando os resíduos dos peixes como fertilizante
  - **Grupo 4: Criação de Animais**
    - **Desafios:** Manejo ineficiente e integração limitada entre agricultura e criação.
    - **Exemplo de Solução:**
      - \* Projetar tanques para peixes conectados a sistemas de irrigação que utilizem a água enriquecida com nutrientes;
      - \* Desenvolver um plano de manejo sustentável para animais, aproveitando resíduos para adubação.
3. **Fases do Design Thinking e Gerenciamento com Scrum:** Os grupos trabalharão nas cinco fases do Design Thinking, enquanto organizam e executam suas tarefas utilizando o Scrum:
- **Empatia:** Entrevistas e observações para identificar as necessidades dos trabalhadores e desafios do ambiente. Durante esta etapa, a equipe fará a primeira reunião de planejamento (Sprint Planning), dividindo as tarefas em histórias de usuário e criando um Product Backlog para listar tudo o que será desenvolvido.
  - **Definição:** Organização das informações coletadas para formular um problema claro e relevante. Use o Backlog Refinement para ajustar e priorizar as histórias de usuário com base no problema definido.
  - **Ideação:** Sessões de brainstorming para gerar ideias inovadoras e viáveis. Durante as Daily Scrums, os grupos discutirão o andamento das ideias, removerão bloqueios e ajustarão tarefas conforme necessário.
  - **Prototipagem:** Criação de modelos práticos que representem as soluções propostas. Execute as tarefas definidas no sprint, mantendo o foco na entrega do Mínimo Produto Viável (MVP).
  - **Teste:** Apresentação dos protótipos para coletar feedback e realizar ajustes. Conclua o ciclo com uma Sprint Review, onde cada grupo apresenta o protótipo para a equipe e avalia os resultados.
4. **Apresentação Final onde cada grupo deverá apresentar:**
- O problema identificado;
  - As ideias desenvolvidas na fase de Ideação;
  - O protótipo criado, explicando seu funcionamento e benefícios;
  - Os ajustes realizados com base no feedback obtido durante os testes.

## 4.2 Considerações do módulo

Ao concluir este experimento prático, é importante refletir sobre o impacto de cada fase do processo, tanto no desenvolvimento de soluções quanto na implementação de novas práticas de gestão na fazenda. O uso do Design Thinking permitiu uma abordagem focada nas necessidades reais dos trabalhadores, garantindo que cada solução proposta estivesse enraizada na experiência diária daqueles que são diretamente afetados pelas mudanças. Isso não só aumenta a probabilidade de sucesso, mas também promove um ambiente de colaboração e inovação.

Por outro lado, o framework **Scrum** trouxe agilidade e organização para o processo de implementação das mudanças. O uso de **Sprints** e reuniões diárias facilitou a visualização do progresso, o ajuste rápido de planos e a entrega contínua de valor. Assim, as soluções puderam ser testadas, ajustadas e refinadas de forma iterativa, o que é crucial em ambientes dinâmicos e em constante mudança, como uma fazenda.

Essa experiência também destaca a importância da **empatia** e do **feedback contínuo** como elementos-chave no desenvolvimento de soluções sustentáveis e eficientes. Ao ouvir os trabalhadores e incorporar suas sugestões no processo, você não apenas melhora a eficiência operacional, mas também cria um ambiente de trabalho mais participativo e motivador, o que pode ter efeitos positivos a longo prazo.

Por fim, é essencial considerar que a **integração de metodologias ágeis com abordagens criativas como o Design Thinking** pode ser aplicada em diversas áreas, não apenas na agricultura. Este módulo fornece uma base sólida para que você explore esses conceitos em outras áreas de atuação, sempre com o foco na solução de problemas reais e na criação de valor para as pessoas envolvidas.

### Fique Alerta!

#### O que fazer a seguir?

1. Continue utilizando as ferramentas de Design Thinking para identificar problemas e criar soluções inovadoras em outros contextos.
2. Explore o Scrum e outras metodologias ágeis em diferentes tipos de projetos, garantindo que a colaboração, a entrega contínua e a adaptação rápida sejam partes integrantes do seu processo de trabalho.
3. Mantenha sempre o feedback dos usuários como uma bússola para direcionar e ajustar suas soluções.

# Capítulo 5

## Estudo de Caso Aplicado: Gestão de uma Fazenda de Piscicultura

### Iniciando o diálogo...

Neste capítulo, aplicaremos os conceitos de Design Thinking e Scrum apresentados nos capítulos anteriores a um cenário prático. O objetivo é demonstrar, passo a passo, como essas metodologias podem ser utilizadas para analisar um problema, propor soluções centradas no usuário e gerenciar o desenvolvimento de forma ágil. Acompanharemos o desenvolvimento de um sistema para automatizar processos em uma fazenda de piscicultura. Serão apresentadas algumas propostas de soluções e aplicações dentro do contexto no cenário apresentado. Destacamos que não foram esgotadas as ideias e tão pouco as possibilidades de soluções para a problematização apresentada. Assim, o objetivo desse material é oferecer subsídios práticos para novas experiências que acontecerão no decorrer do curso.

### 5.1 Cenário: Fazenda de Piscicultura

Antes de aplicarmos qualquer metodologia, é fundamental entender o contexto e os desafios do problema. O cenário a seguir descreve as operações e necessidades de uma fazenda de piscicultura que busca modernização.

A Fazenda de Piscicultura trabalha com espécies reofílicas de peixes. Essas espécies precisam migrar e formar cardumes para se reproduzirem. Como os peixes são criados em cativeiro, é realizado o método de reprodução induzida dos peixes com o uso do extrato de hipófise, processo que envolve a aferição de hora-grau, conceito para estabelecer o tempo aproximado em que os peixes estarão aptos para desovar em função da temperatura do ambiente de confinamento. Para cada espécie de peixe, há um valor específico de hora-grau, por exemplo, para o peixe Tambaqui, são necessárias aproximadamente 200 horas/graus.

Ao se aproximar desta medida de horas/graus, são analisados os aspectos comportamentais do peixe, momentos que antecedem a desova dos peixes, pois a extrusão e fertilização dos ovos precisam ser realizados a seco. A aferição das horas-gradus, são realizadas de maneira manual, ou seja, o técnico faz os registros das horas utilizando o marcador no relógio, mede a temperatura com um termômetro e anota as informações coletadas para fazer as contagens das horas-gradus.

A fazenda realiza a limpeza dos tanques periodicamente, com renovação periódica da água dos tanques a partir do seu sistema de recirculação de água, onde a água suja que sai dos

tanques é utilizada na irrigação do plantio hidropônico de verduras e hortaliças da fazenda, após o tratamento, a água tratada é reutilizada, para os próprios tanques dos peixes.

A fazenda solicita o desenvolvimento de um sistema, de forma que as atividades da fazenda, sejam realizadas de forma automatizadas como: verificação de horas-graus no processo de reprodução de peixe, comportamento do peixe após atingir o valor hora-grau, verificação da oxigenação da água, medição do nível da água no tanque, a irrigação e adubação do plantio.

Segue as informações da aferição de hora-grau = soma do número de horas e temperatura da água:

- 1ª aferição: Hora 1 = 30 graus  $\rightarrow 1 + 30 = 31$  horas/graus
- 2ª aferição: Hora 2 = 31 graus  $\rightarrow 2 + 31 = 33$  horas/graus
- Nª aferição: Hora M = K graus  $\rightarrow M + K = Z$  horas/graus
- Total = 1ª aferição + 2ª aferição + ... + Nª aferição.

## 5.2 Etapa 1: Design Thinking

Com o cenário compreendido, iniciamos a aplicação do Design Thinking (como visto no Capítulo 1) para explorar o problema sob a perspectiva humana, identificar as reais necessidades e gerar ideias de soluções.

Nessa etapa do projeto, é importante estar sensível à necessidade das pessoas, focar nas ideias que possibilitam uma solução e que haja boas transformações, ainda que sejam pequenas. A equipe não pode ter medo ou vergonha em apresentar as ideias, ainda que possam parecer “malucas”, não devem sentir “medo de errar”, o foco é apresentar as ideias de forma criativa, ou seja, o foco nessa etapa é apresentar a “Tempestade de ideias”, os filtros e os ajustes deverão ser feitos em um segundo momento.

É muito importante que todos os membros da equipe tenham um espírito colaborativo, todos deverão ter uma visão colaborativa entre as ideias individuais e não desmerecer a ideia alheia, o absurdo para um pode ser a solução para o outro. Não há ideias absolutas, mas há ideias que poderão ser recebidas e posteriormente serem otimizadas.

### Fique Alerta!

Buscar soluções que sejam desejáveis (atendem a uma necessidade humana), praticáveis (tecnicamente viáveis) e viáveis (economicamente sustentáveis).

A tabela a seguir resume a aplicação das cinco fases do Design Thinking ao cenário da fazenda, mostrando as ferramentas utilizadas e os resultados obtidos em cada etapa.

Tabela 5.1: Apresentação do Design Thinking do Projeto do Sistema.

Etapa	Ferramenta/Conceito	Aplicação no Cenário
Stakeholders	Partes interessadas	Proprietário da fazenda; Piscicultores; Técnicos Agrícola; Funcionários da Fazenda.

*Continua na próxima página...*

Tabela 5.1: (Continuação)

Etapa	Ferramenta/Conceito	Aplicação no Cenário
<b>Empatia</b>	Mapa da Empatia: o que pensam e sentem, o que ouvem, o que veem, o que falam e fazem.	<i>Dores identificadas:</i> Aferição manual da hora e temperatura; Cálculo manual das horas-temperaturas; Inspeção visual do comportamento dos peixes; Verificação da oxigenação da água; Medir o nível da água; Irrigação manual do plantio.
<b>Definição</b>	Diagrama de afinidades: Revisar, organizar, identificar padrões e priorizar temas .	<i>Problemas definidos:</i> Necessidade de automatizar as aferições de horas-graus; Automatizar a observação dos comportamentos dos peixes; Automatizar a coleta das informações do nível, temperatura e oxigenação da água; Automatizar a irrigação e o uso de resíduos.
<b>Ideação</b>	Brainstorming (tempestade de ideias) e Matriz SWOT.	<i>Ideias geradas:</i> Sistematizar as atividades das aferições de horas-graus, medição de temperatura, nível e oxigenação e comportamento dos peixes de forma automatizada; Sistematizar a irrigação e adubação do plantio; Criar um Aplicativo do sistema automatizado.
<b>Prototipação</b>	Criação de Modelos (ex: maquetes, wireframes).	<i>Protótipo proposto:</i> Sistema de Automação de Aferições, comportamento e Irrigação.
<b>Teste</b>	Colhendo Feedback dos stakeholders.	<i>Validação:</i> Testar o funcionamento do protótipo do sistema no cenário apresentado e validar as funcionalidades com os usuários.

### Fique Alerta!

As ideias e soluções apresentadas são um ponto de partida. O processo de ideação no Design Thinking incentiva a geração de múltiplas alternativas.

## 5.3 Etapa 2: Scrum

Após a fase de Design Thinking, onde exploramos o problema e idealizamos uma solução (um sistema automatizado), passamos a usar o framework Scrum (detalhado no Capítulo 3) para gerenciar o desenvolvimento dessa solução de forma ágil e iterativa.

Na metodologia SCRUM, as tarefas são adaptáveis, interativas, rápidas, flexíveis e eficientes, com o intuito de fornecer resultados ágeis em todas as etapas do projeto, de forma transparente, responsivo e contínuo progresso. Visando as características do SCRUM, iremos empregá-las na próxima fase do projeto.

### Fique Alerta!

A equipe SCRUM deverá estar focada e comprometida com o projeto em sua totalidade, ou seja, desde a concepção, desenvolvimento, testes, entrega e feedback pós-entrega. Lembrar sempre: O Time SCRUM é colaborativo e não apenas participativo, esse entendimento fará toda a diferença no desenvolvimento, progresso e resultados.

### 5.3.1 Grupo de Trabalho

O primeiro passo no Scrum é definir o Time Scrum. A tabela a seguir mostra uma possível composição, separando os stakeholders (grupo "Central") dos papéis que compõem o Time Scrum (grupo "Não-Central"), embora lembremos que o Product Owner atua como ponte entre eles e é parte do Time Scrum.

Tabela 5.2: Apresentação do Grupo de Trabalho.

Central (Stakeholders)	Não – Central (Time SCRUM)
<ul style="list-style-type: none"><li>• Proprietário da fazenda;</li><li>• Piscicultores;</li><li>• Técnicos Agrícola;</li><li>• Funcionários da fazenda.</li></ul>	Product Owner
	Scrum Master
	Scrum Desenvolvedores

A figura abaixo ilustra o fluxo de trabalho no Scrum, desde a priorização do Backlog pelo Product Owner até a entrega de incrementos (Entregáveis) pelo Time Scrum e a validação na Sprint Review.

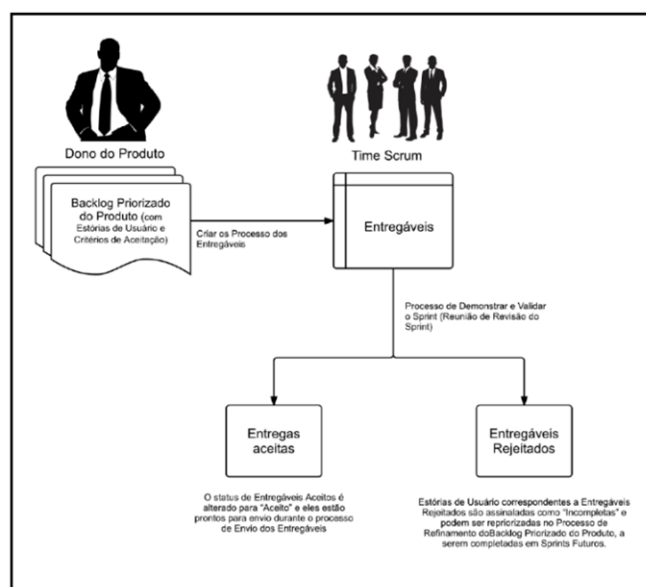


Figura 5.1: Diagrama do fluxo de incremento do projeto  
Fonte: Scrumstudy.

### 5.3.2 Visão

A "Visão" do produto define o objetivo geral e o propósito do que será construído. Ela serve como guia para o Product Owner e o Time Scrum.

Tabela 5.3: Apresentação da Visão do Projeto do Sistema.

Conceito	Descrição
<b>Visão (apresentação da demanda)</b>	"A fazenda solicita o desenvolvimento de um sistema, de forma que essas atividades de verificação de horas-graus no processo de reprodução de peixe, comportamento do peixe após atingir o valor hora-grau, verificação da oxigenação da água, medição do nível da água no tanque, a irrigação e adubação do plantio, sejam realizados de forma automatizadas."

#### Fique Alerta!

Após o recebimento das demandas (Visão), o Product Owner deverá se reunir com as partes envolvidas (stakeholders) para coletar o levantamento das necessidades específicas, que serão traduzidas em Histórias de Usuário.

### 5.3.3 Histórias do Usuário

As Histórias de Usuário descrevem funcionalidades sob a perspectiva de quem as utilizará ("Como um [ator], eu quero [ação] para que [benefício]"). Elas formam a base do Product Backlog. Veja exemplos coletados para o sistema da fazenda:

Tabela 5.4: Apresentação das Histórias da Visão do Projeto do Sistema.

ID-H	Descrição da História do Usuário
1	Como <i>proprietário da fazenda</i> , eu gostaria de <b>ter um sistema que automatizasse as atividades</b> de piscicultura e plantio da fazenda.
2	Como <i>proprietário</i> , eu gostaria que <b>cada funcionário criasse seu próprio cadastro no sistema</b> , seja piscicultor, técnico agrônomo ou funcionário da fazenda e que <b>haja classificação de acesso</b> .
3	Como <i>proprietário</i> , eu gostaria que o sistema <b>gerasse relatório com gráficos</b> das atividades da piscicultura e plantio da fazenda.
4	Como <i>piscicultor</i> eu preciso <b>acompanhar as informações aferidas da temperatura e oxigenação</b> da água dos tanques dos peixes.
5	Como <i>piscicultor</i> eu preciso de um sistema que <b>realize a medição da hora e temperatura</b> no período da reprodução dos peixes e que <b>gere o cálculo hora-grau</b> .
6	Como <i>piscicultor</i> , desejo que o sistema <b>permita o cadastro de hora-grau</b> para cada espécie de peixe.

Continua na próxima página...

Tabela 5.4: (Continuação)

ID-H	Descrição da História do Usuário
7	Como <i>piscicultor</i> eu quero <b>realizar consultas das informações coletadas</b> dos tanques dos peixes.
8	Como <i>piscicultor</i> eu quero que o sistema <b>realize relatórios com gráficos</b> com as informações hora-grau do período da reprodução dos peixes.
9	Como <i>piscicultor</i> eu quero que o sistema <b>realize relatórios com gráficos</b> do período das temperaturas e oxigenação da água, por dia, semana, mês e ano para analisar a influência das estações climáticas.
10	Como <i>piscicultor</i> , quero um sistema que <b>realize amostras visuais do comportamento dos peixes</b> no período da reprodução e que <b>gere um relatório</b> .
11	Como <i>piscicultor</i> , quero um sistema que <b>realize amostras visuais do comportamento dos peixes</b> em cada estação climática e que <b>gere um relatório</b> .
12	Como <i>técnico agrícola</i> , quero que o <b>sistema de irrigação do plantio seja automatizado</b> .
13	Como <i>técnico agrícola</i> , quero que o <b>aproveitamento dos resíduos dos peixes</b> , utilizados para adubar o plantio, <b>seja realizado de forma automatizada</b> .
14	Como <i>piscicultor</i> , desejo que o sistema <b>emita um sinal de alerta</b> quando a hora-grau tiver próxima da hora-grau apropriada para a reprodução por cada espécie de peixe.
15	Como <i>técnico agrícola</i> , quero que o sistema <b>emita um sinal de alerta</b> , caso o sistema automatizado de irrigação não esteja operando de forma correta.
16	Como <i>técnico agrícola</i> , quero que o sistema <b>emita um sinal de alerta</b> , caso o sistema automatizado de adubação não esteja operando de forma correta.

Note que nem toda História de Usuário precisa seguir rigidamente o formato "Como um..., eu quero..., para que...". O importante é capturar a necessidade do usuário.

### Fique Alerta!

Esta lista inicial pode (e deve) evoluir. Novas histórias podem ser adicionadas e as existentes podem ser refinadas ao longo do projeto.

### Fique Alerta!

"Os Épicas são escritos nas fases iniciais do projeto, quando a maioria das Histórias de Usuário são funcionalidades de alto nível ou quando as descrições de produtos e requisitos são amplamente definidas..." (ScrumStudy). Muitas das histórias acima podem ser consideradas Épicas iniciais.



### 5.3.4 Classificação das Histórias

Após coletar as histórias, o próximo passo é agrupá-las por funcionalidades ou temas comuns. Isso ajuda a organizar o trabalho e a identificar os principais requisitos do sistema.

#### Fique Alerta!

Este processo de agrupar e refinar histórias faz parte do *Backlog Refinement* (Refinamento do Backlog), uma atividade contínua no Scrum.

Tabela 5.5: Apresentação da classificação das Histórias do Projeto do Sistema.

ID-Func.	Funcionalidade Agrupada	Histórias de Usuário (ID-H) Relacionadas
R001	Cadastrar Funcionário	2
R002	Gerar e Emitir Relatórios	3, 8, 9, 10, 11, 12
R003	Consultar aferições	4, 5, 7
R004	Consultar cálculo hora-grau	5
R005	Cadastrar hora-grau	6
R006	Emissão de sinal de alerta	5, 6, 14, 15, 16

### 5.3.5 Backlog Priorizado do Produto

Com as funcionalidades identificadas, o Product Owner cria o Product Backlog Priorizado. Esta é a lista ordenada de tudo que é desejado no produto, com os itens de maior valor para o negócio no topo.

#### Fique Alerta!

“O Dono do Produto desenvolve um Backlog Priorizado do Produto, que contém uma lista de prioridades de negócios e de requisitos dos projetos, escritos na forma de Épico(s), que são as Histórias de Usuário de alto nível.” (ScrumStudy).

Tabela 5.6: Apresentação do Product Backlog Priorizado do Projeto do Sistema.

ID	Funcionalidade (Item do Backlog)	Prioridade
R001	Cadastrar Funcionário	Alta
R004	Consultar cálculo hora-grau	Alta
R005	Cadastrar hora-grau	Alta
R006	Emissão de sinal de alerta	Alta
R002	Gerar e Emitir Relatórios	Média
R003	Consultar aferições	Média

Note que a ordem na tabela acima foi ajustada para refletir a priorização (Alta antes de Média). A priorização é dinâmica e pode mudar ao longo do projeto.

### 5.3.6 Backlog do Produto (com Épicas Detalhados)

Itens de maior prioridade no Product Backlog (especialmente os Épicas) são detalhados em Histórias de Usuário menores e mais gerenciáveis, prontas para serem consideradas em uma Sprint. Abaixo, vemos o detalhamento do item R001 ("Cadastrar Usuário").

Tabela 5.7: Apresentação do Detalhamento de Épico (R001) no Backlog do Produto.

ID-Fun.	Épico	ID-E	História de Usuário Detalhada	Prior.
R001	Cadastrar Usuário	E001	Como usuário, quero poder fazer login usando meu usuário/CPF e senha para acessar o sistema.	Alta
		E002	Como administrador, quero poder gerenciar (criar, editar, desativar) contas de usuários.	Alta
		E003	Como administrador, quero poder definir perfis de acesso (piscicultor, técnico, etc.).	Alta
		E004	Como novo funcionário, quero poder realizar um auto-cadastro inicial no sistema.	Alta
		E005	Como usuário logado, quero poder visualizar e editar meus dados cadastrais.	Alta
		E006	Como usuário logado, quero ver um menu principal (Dashboard) com as opções disponíveis para meu perfil.	Alta
R002	Gerar e Emitir Relatórios	...	(Detalhar futuramente)	...
R003	Consultar aferições	...	(Detalhar futuramente)	...
R004	Consultar cálculo hora-grau	...	(Detalhar futuramente)	...
R005	Cadastrar hora-grau	...	(Detalhar futuramente)	...
R006	Emissão de sinal de alerta	...	(Detalhar futuramente)	...

#### Fique Alerta!

O detalhamento dos Épicas R002 a R006 seria feito em sessões de Refinamento do Backlog quando sua prioridade os trouxer para o topo da lista.

### 5.3.7 Backlog da Sprint

Durante a Sprint Planning (Planejamento da Sprint), o Time Scrum seleciona os itens de maior prioridade do Product Backlog que acredita poder concluir dentro da Sprint. Esses itens formam o Sprint Backlog.

### Fique Alerta!

**Sprint Backlog:** é a lista de trabalho (itens do Product Backlog e tarefas para alcançá-los) que o Time Scrum se compromete a realizar durante a Sprint.

A tabela abaixo mostra um exemplo de planejamento de Sprints para entregar as funcionalidades de Cadastro de Usuário (R001), divididas em grupos menores (Épicos E001 a E006) ao longo de três Sprints. A Release 1 agrupa a entrega dessas funcionalidades.

Tabela 5.8: Apresentação do Backlog da Sprint do Projeto do Sistema.

ID da Sprint	ID-E	Itens do Product Backlog Selecionados	Complex.	Início	Fim
Sprint 1	E001	Cadastrar Usuário – Área de Login	Média	06/01/2025	20/01/2025
	E004	Cadastrar Usuário – Área de Login – Auto Cadastro	Média		
	E005	Cadastrar Usuário – Meus Dados	Média		
Sprint 2	E002	Cadastrar Usuário – Administrador	Baixa	21/01/2025	27/01/2025
	E003	Cadastrar Usuário – Gerenciar Perfil	Baixa		
Sprint 3	E006	DashBoard Menu	Média	28/01/2025	11/02/2025
<b>Entregar Release 1:</b> 12/02/2025					

### 5.3.8 Sprint Planning

A reunião de Sprint Planning é onde o Time Scrum detalha o \*como\* o trabalho selecionado para o Sprint Backlog será realizado. Define-se a Meta da Sprint e criam-se as tarefas necessárias para transformar os itens do backlog em um Incremento "Pronto". Abaixo, um exemplo do planejamento detalhado para a Sprint 1. Após a definição de cada Sprint, cada Sprint deverá ser documentada de acordo com as especificações técnicas e abordagens de desenvolvimento. Utilizando as informações da tabela anterior, a documentação de cada Sprint é a próxima atividade a ser realizada, atividade conhecida como Sprint Planning.

### Fique Alerta!

O planejamento da Sprint é muito importante, pois são documentados os procedimentos fundamentais para a execução das atividades, as ações, o tempo de execução e tempo de entrega e quais as tarefas de cada pessoa do Time Scrum. Todos os membros do Time Scrum deverão participar do planejamento da Sprint.

Tabela 5.9: Apresentação da Sprint 1 - Modulo 1 do Projeto do Sistema.

Sprint Planning	
Data	06/01/2025
Sprint	1
Participantes	Membro 1, Membro 2, Membro 3, ..., Membro N
Meta da Sprint	(Exemplo) Disponibilizar as funcionalidades básicas de login (usuário/senha, recuperação) e auto-cadastro para os usuários da fazenda.
Objetivos da Sprint	A Sprint 1 tem como objetivo o desenvolvimento da tela de Login para o Módulo #1 do sistema. O usuário terá a opção de realizar o login utilizando o nome do usuário, CPF ou a sua conta do Google, por meio de uma API integrada.
Time-box	2 semanas
Itens do Sprint Backlog	<b>Itens do Product Backlog:</b> E001, E004, E005 (Funcionalidade não listada no Backlog, mas mencionada nos objetivos: Recuperar Senha)  <b>Tarefas Planejadas:</b> (Exemplo) <ul style="list-style-type: none"><li>- Criar modelo de dados do usuário</li><li>- Desenvolver API de autenticação</li><li>- Desenvolver API de recuperação de senha</li><li>- Desenvolver tela de login (UI)</li><li>- Desenvolver tela de auto-cadastro (UI)</li><li>- Desenvolver tela de "Meus Dados"(UI)</li><li>- Integrar UI com APIs</li><li>- Escrever testes unitários/integração</li><li>- Configurar pipeline de CI/CD</li></ul>
<b>Detalhes das Funcionalidades e Testes</b>	
<i>Funcionalidade 1: Autenticação com Login e Senha (E001)</i>	
<b>Descrição</b>	A funcionalidade permite que os usuários façam login no sistema utilizando nome de usuário ou CPF e senha, cadastrado anteriormente no sistema.
<b>Testes (Critérios de Aceite)</b>	<i>Cenário 1 (Sucesso):</i> Login com credenciais válidas. <b>Dado</b> que o usuário está na tela de login <b>Quando</b> ele insere nome de usuário/CPF e senha corretos E clica em "Entrar" <b>Então</b> ele é redirecionado para a página principal do sistema.  <i>Cenário 2 (Falha):</i> Usuário/CPF inválido. <b>Dado</b> que o usuário está na tela de login <b>Quando</b> ele insere nome de usuário ou CPF inexistente E clica em "Entrar" <b>Então</b> o sistema exibe a mensagem de erro: "O nome de usuário ou CPF está incorreto. Tente novamente."

Tabela 5.9: (Continuação)

## Sprint Planning

*Cenário 3 (Falha):* Senha inválida.

**Dado** que o usuário está na tela de login

**Quando** ele insere usuário/CPF válido e senha incorreta  
E clica em "Entrar"

**Então** o sistema exibe a mensagem de erro: "A senha está incorreta. Tente novamente."

*Cenário 4 (Falha):* Conta bloqueada/desativada.

**Dado** que o usuário está na tela de login

**Quando** ele tenta logar com uma conta  
bloqueada/desativada

E clica em "Entrar"

**Então** o sistema exibe a mensagem de erro: "Sua conta está bloqueada ou desativada. Entre em contato com o suporte."

### Funcionalidade 2: Recuperar Senha

#### Descrição

A funcionalidade permite a recuperação da senha através do e-mail caso o usuário esqueça sua senha.

#### Testes (Critérios de Aceite)

*Cenário 1 (Sucesso):* Recuperação bem-sucedida.

**Dado** que o usuário está na tela de login

**Quando** ele clica em "Recuperar senha"

E insere seu e-mail cadastrado

E clica no link recebido por e-mail

E define e confirma uma nova senha

**Então** o sistema exibe a mensagem "Senha alterada com sucesso"

E ele consegue logar com a nova senha.

*Cenário 2 (Falha):* E-mail não cadastrado.

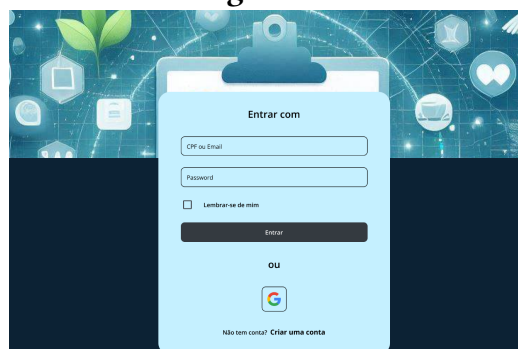
**Dado** que o usuário está na tela de recuperação de senha

**Quando** ele insere um e-mail não cadastrado

**Então** o sistema exibe uma mensagem informativa (sem confirmar se o e-mail existe ou não por segurança).

### UI/UX - Sprint 1

#### Tela de Login:



### Necessidades da Sprint

Essencial.

**Fique Alerta!**

As documentações das demais Sprints (Sprint 2, Sprint 3, etc.) seguiriam um formato similar, detalhando os itens selecionados para cada uma.

”Notem, que na documentação acima, não consta o Módulo “Efetuar login com uma conta social (Gmail)”, portanto, ao realizar o Review da Sprint realizar as alterações pertinentes.”Isso exemplifica a natureza adaptativa do Scrum: o plano pode mudar.

**5.3.9 Review da Sprint**

Ao final da Sprint, ocorre a Sprint Review, uma reunião onde o Time Scrum demonstra o Incremento “Pronto” para os stakeholders e coleta feedback.

Tabela 5.10: Apresentação da Review Sprint 1 - Modulo 1 do Projeto de Sistemas.

Review da Sprint 1			
Data	20/01/2025 (Exemplo - Fim da Sprint 1)		
Sprint	1		
Participantes	Jaidson Schwarzenegger, Walter Stalone, Hillton Smith, Rogerio Craig, Alexandre Hanks, Eduardo Van Damme, Luciana Stone, Lucélia Bullock (e outros stakeholders relevantes)		
Resultados da Revisão do Backlog da Sprint			
Item Demonstrado	Status (Aceito?)	Feedback / Observações	
Autenticação com Login e Senha	ACEITO	Funciona conforme esperado.	
Auto-Cadastro	ACEITO	OK.	
Recuperar Senha	NÃO ACEITO	Feedback: Cenário onde usuário não tem acesso ao e-mail precisa de alternativa (ex: pergunta de segurança?). <i>Criar nova história no Product Backlog.</i>	
Meus Dados	PARCIAL	Tela criada, mas falta integração com backend (será feito na Sprint 2?).	
Bugs Identificados Durante a Review			
Bug ID	Descrição	Item Relacionado	Prioridade
BUG-001	Mensagem de erro de login genérica	Autenticação	Baixa

**5.3.10 Retrospectiva da Sprint**

Após a Sprint Review, o Time Scrum realiza a Sprint Retrospective. O objetivo é inspecionar a si mesmo e criar um plano para melhorias a serem aplicadas na próxima Sprint, focando em pessoas, relacionamentos, processos e ferramentas.

Tabela 5.11: Apresentação da Retrospective Sprint 1 - Modulo 1 do Projeto do Sistema.

Retrospectiva da Sprint 1	
<b>Data</b>	20/01/2025 (Exemplo - Fim da Sprint 1)
<b>Sprint</b>	1
<b>Participantes</b>	<i>Time Scrum:</i> Jaidson Schwarzenegger, Walter Stalone, Hillton Smith, Rogerio Craig, Alexandre Hanks, Eduardo Van Damme, Luciana Stone, Lucélia Bullock
Discussão (Exemplo: O que funcionou? O que não funcionou? O que melhorar?)	
<b>O que fizemos bem?</b>	<ul style="list-style-type: none"> <li>- Concluímos a maioria das tarefas planejadas para a Sprint.</li> <li>- Boa comunicação e colaboração dentro da equipe de desenvolvimento.</li> <li>- Eficiência na integração inicial de Frontend e Backend.</li> </ul>
<b>O que pode melhorar?</b>	<ul style="list-style-type: none"> <li>- Documentação do código pode ser mais detalhada para facilitar futuras manutenções.</li> <li>- Tempo gasto em debugging durante a integração pode ser reduzido (talvez com mais testes automatizados?).</li> <li>- Estimativas de complexidade/esforço podem ser refinadas.</li> <li>- Falha na comunicação sobre a necessidade de alternativa para recuperação de senha (não discutido na Planning).</li> </ul>
Plano de Ação para a Próxima Sprint (Sprint 2)	
Ação	Responsável(eis)
1. Revisar e melhorar os padrões de documentação de código.	Todos os Desenvolvedores
2. Pesquisar e implementar mais testes de integração automatizados.	Desenvolvedor(es) específico(s)
3. Discutir o feedback da Review sobre "Recuperar Senha" com o PO e refinar/criar história no Backlog.	Scrum Master / Time Scrum
4. Incluir revisão de Critérios de Aceite mais detalhada na próxima Sprint Planning.	Scrum Master / Time Scrum
5. Alocar tempo na Sprint 2 para refatoração técnica e pagamento de débitos (ex: documentação).	Time de Desenvolvimento

### 5.3.11 Ata da Reunião

Embora não seja um artefato formal do Scrum, registrar decisões importantes ou feedbacks de reuniões como a Sprint Review em uma ata pode ser útil para comunicação e referência futura, especialmente com stakeholders externos.

Tabela 5.12: Apresentação da Ata da Reunião (Exemplo pós-Sprint Review 1).

Ata da Reunião	
<b>Data:</b> 20/01/2025	<b>Horário Início:</b> 14:00 <b>Horário Término:</b> 15:00
<b>Local:</b>	Sala de Reunião Ifam – Campus Manacapuru
<b>Participantes:</b>	Jaidson Schwarzenegger, Walter Stalone, Hillton Smith, Rogerio Craig, Alexandre Hanks, Eduardo Van Damme, Luciana Stone, Lucélia Bullock, + <i>Stakeholders (Proprietário, etc.)</i>
<b>Objetivo da Reunião</b>	Sprint Review da Sprint 1 - Módulo 1. Apresentar o Incremento e coletar feedback.
<b>Itens Apresentados</b>	<ul style="list-style-type: none"> <li>- Demonstração da Tela de Login (e-mail/senha).</li> <li>- Demonstração da Tela de Auto-Cadastro.</li> <li>- Demonstração (parcial) da Tela "Meus Dados".</li> <li>- Demonstração da Funcionalidade de Recuperação de Senha.</li> </ul>
<b>Decisões e Feedbacks</b>	<ul style="list-style-type: none"> <li>- Login e Auto-Cadastro: <b>Aceitos</b>.</li> <li>- Meus Dados: <b>Aceito parcialmente</b> (aguardando integração backend).</li> <li>- Recuperar Senha: <b>Não Aceito</b>. Necessário discutir alternativa para usuários sem acesso ao e-mail. Product Owner irá priorizar nova história/ajuste no Backlog.</li> <li>- <b>Questionamentos:</b> Como foi a integração...? Feedback dos usuários...? Desafios...? (Respostas discutidas na reunião).</li> </ul>
<b>Próximos Passos (Ações Externas ao Time Scrum, se houver)</b>	
<b>Nº</b>	<b>Ação / Responsável / Prazo</b>
<b>1</b>	Product Owner priorizar história/ajuste para "Recuperar Senha" no Product Backlog / PO / Antes da Sprint Planning 2
<b>Encerramento</b>	A reunião foi encerrada às 15:00 com agradecimentos pelo feedback e confirmação dos próximos passos.
<b>Assinaturas (Opcional)</b>	
Nome: _____	Assinatura: _____
Nome: _____	Assinatura: _____



## 5.4 Considerações Finais do Estudo de Caso

Este estudo de caso demonstrou como o Design Thinking ajuda a entender o problema e definir o "o quê" construir, enquanto o Scrum organiza o "como" construir de forma ágil e adaptativa.

Ao longo deste material, foi apresentado uma amostra de um cenário de uma fazenda sustentável e suas necessidades de sistematizar alguns dos seus serviços aplicados à piscicultura e plantio. Foi realizado o levantamento das Histórias do proprietário da fazenda e realizada as suas classificações. Assim, foram encontradas possíveis funcionalidades de soluções para que alguns serviços fossem realizados de forma sistemática.

Esperamos que este exemplo prático inspire você estudante a aplicar essas poderosas metodologias em seus próprios projetos e desafios. Use SCRUM e transforme a forma como sua equipe desenvolve produtos e soluções. Mãos à obra!

# Referências Bibliográficas

- [1] BECK, Kent, et al. **Manifesto para o Desenvolvimento Ágil de Software**. 2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acessado em: 03 de julho de 2025.
- [2] BECK, Kent. **Extreme Programming Explained: Embrace Change**. Addison-Wesley Professional, 2000.
- [3] BRISSOS, Fábio; BRESSANE, Thaís. **Scrum: Gestão Ágil para Projetos de Sucesso**. São Paulo: Brasport, 2020.
- [4] BROWN, Tim. **Design thinking: uma metodologia poderosa para decretar o fim das velhas ideias**. 10. ed. comemorativa. Rio de Janeiro: Alta Books, 2020.
- [5] COHN, Mike. **User Stories Applied: For Agile Software Development**. Boston: Addison-Wesley Professional, 2020.
- [6] FILATRO, Andrea; CAVALCANTI, Carolina Costa. **Design Thinking na educação presencial, a distância e corporativa**. São Paulo: Editora Senac São Paulo, 2018.
- [7] PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. Porto Alegre: AMGH, 2016.
- [8] SCHWABER, Ken; SUTHERLAND, Jeff. **O Guia do Scrum: As Regras do Jogo**. Disponível em: <https://scrumguides.org>. Acessado em: 03 de julho de 2025.
- [9] SILVA, Mauro. **Design Thinking: Inovação em Negócios**. São Paulo: Casa do Código, 2021.
- [10] SNOWDEN, David J.; BOONE, Mary E. **A Leader's Framework for Decision Making**. *Harvard Business Review*, nov. 2007.
- [11] TAKEUCHI, Hirotaka; NONAKA, Ikujiro. **The New New Product Development Game**. *Harvard Business Review*, jan. 1986.
- [12] SEBRAE. Guia sobre o que é Design Thinking e Como Aplicá-lo!
- [13] Sbrocco, José Henrique T. de C. Macedo, Paulo Cesar. **Metodologias Ágeis Engenharia de Software sob Medida**. 1ª Edição. Editora Érica. 2012.
- [14] Scrumstudy: Um Guia para o CONHECIMENTO EM SCRUM. 3ª Edição, 2017.
- [15] Schwaber, Ken. Sutherland, Jeff. **The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game**. 2020. <https://scrumguides.org>. Acessado em 11 de janeiro de 2025.

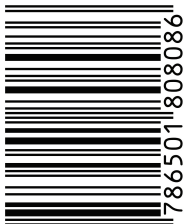




# CITHA

Capacitação e Interiorização em  
Tecnologias Habilitadoras na Amazônia

ISBN: 978-65-01-80808-6



9 786501 808086

**CBL**