

## SUA JORNADA EM PRODUTO

# O Roadmap Essencial do Product Manager / Owner Moderno

*Do zero ao primeiro emprego em produto*

Baseado em: Marty Cagan, Teresa Torres, Dan Olsen, Melissa Perri, Roman Pichler, Nir Eyal, Jeff Gothelf, John Cutler + SAFe 6.0 + fontes do projeto

Elaborado por: **Marcos Ruiz**  
Product Manager · [linkedin.com/in/zenduardo](https://www.linkedin.com/in/zenduardo)

### Objetivo

Conquistar uma vaga como PM ou PO com base sólida, portfólio real e capacidade de se destacar em processos seletivos.

### Duração estimada

21 semanas (5 meses) | 9 fases progressivas |  
Estudo de 1 a 2h por dia — Atualizado: 2026

## Visão Geral do Curso

Este curso foi desenhado com um princípio central: aprender só o que gera valor agora. Sem teoria desnecessária. Sem jargões vazios. Cada módulo foi construído sobre o que os maiores especialistas do mundo em produto ensinam, filtrado pelo que você realmente precisa para entrar no mercado.

Esta versão foi revisada e atualizada em 2026, incorporando o SAFe 6.0 (Scaled Agile Framework), os frameworks modernos de métricas de fluxo (Lead Time, Throughput, DORA), e as práticas de roadmap adaptável que estão sendo adotadas pelas maiores organizações de tecnologia do mundo.

## Mapa de Fases

#	Fase	Tópico Central	Semanas	Prioridade
0	Fundamentos	O que é PM/PO, mentalidade de produto	2	Essencial
1	Entender o usuário	Discovery, pesquisa, JTBD e timeboxes	3	Essencial
2	Backlog & Priorização	User stories, OKRs, roadmap adaptável	3	Essencial
3	Ágil na prática	Scrum, Kanban, timeboxes e cerimônias	3	Essencial

4	Estratégia de produto	Visão, MVP, go-to-market, Lean Canvas	2	Alta
5	Métricas & Dados	North Star, Lead Time, Throughput, DORA	2	Alta
6	Ágil Escalado	SAFe 6.0, ART, tribos, RTE, PO no scale	2	Alta
7	Ferramentas	Jira, Notion, Miro, Figma, SQL básico	1	Alta
8	Portfólio & Entrevistas	Cases, frameworks, simulações	3	Crítico

### Como usar este curso

1. Siga as fases em ordem. Cada uma constrói sobre a anterior.
2. Faça os exercícios práticos. Aprender PM sem praticar é como ler sobre natação sem entrar na água.
3. Documente tudo no Notion. Cada exercício se tornará parte do seu portfólio.
4. Use este projeto como mentor: pergunte, simule entrevistas, refine seus artefatos.
5. Não pule para a Fase 8 antes de ter material real para mostrar. Portfólio vazio não emprega.

# Os Grandes Nomes que Você Precisa Conhecer

Antes de entrar no conteúdo técnico, você precisa saber de quem estamos aprendendo. Esses são os autores que aparecem em toda entrevista de PM, em toda conversa de comunidade, e cujas ideias moldaram a disciplina como ela é hoje.

## Os 8 Pensadores Fundamentais

Nome	Por que importa para você
<b>Marty Cagan (SVPG)</b>	O pai do PM moderno. O livro "Inspired" é leitura obrigatória. Define o que é um time de produto de verdade versus um time de execução. Quase toda entrevista de PM cita Cagan.
<b>Teresa Torres</b>	Criou o framework de Continuous Discovery. Ensina como falar com usuários de forma sistemática e contínua. Essencial para Discovery.
<b>Nir Eyal</b>	Autor do "Hooked" e do Hook Model. Explica por que usuários ficam viciados em produtos. Fundamental para pensar retenção e engajamento.
<b>Roman Pichler</b>	Especialista em estratégia de produto e Ágil. Seu livro "Strategize" é o mais votado pela comunidade para roadmaps. Muito útil no dia a dia de PO.
<b>Dan Olsen</b>	"Lean Product Playbook" é um manual prático para validação rápida de hipóteses. Perfeito para quem quer aprender MVP e Lean na prática.
<b>Melissa Perri</b>	Autora de "Escaping the Build Trap". Desmonta a cultura de apenas construir features sem gerar valor. Linguagem direta e muito aplicada.
<b>Jeff Gothelf</b>	Co-autor de "Lean UX". Une produto, design e estratégia. Essencial para entender como times modernos funcionam.
<b>John Cutler</b>	Filósofo do produto. Posts profundos sobre cultura organizacional e sistemas de produto. Leitura para quando você quiser ir além do básico.

### Ordem de leitura recomendada (lean)

Prioridade 1 (leia agora): Inspired, de Marty Cagan | The Lean Product Playbook, de Dan Olsen

Prioridade 2 (durante o curso): Escaping the Build Trap, de Melissa Perri | Continuous Discovery Habits, de Teresa Torres

Prioridade 3 (após a primeira vaga): Hooked, de Nir Eyal | Strategize, de Roman Pichler | Lean UX, de Jeff Gothelf

Nota: os livros Inspired, Lean Product Playbook e os guias de Scrum e OKR já estão nas fontes deste projeto.

FASE 0 2 semanas

## FUNDAMENTOS DE PRODUTO

Entender o campo antes de entrar nele

Antes de aprender qualquer framework, você precisa entender o que é produto de verdade, e o que separa um PM/PO de um gerente de projetos ou de um analista de negócios. Essa confusão é a raiz de muitas frustrações no início de carreira.

### O que você vai aprender

- O que é um produto digital e o que é um serviço
- PM vs. PO vs. Project Manager: as três funções que o mercado confunde o tempo todo
- Como empresas de tecnologia organizam times de produto (squads, tribos, chapters)
- A mentalidade de produto: pensar em outcomes, não em output
- O ciclo de vida de um produto: discovery, delivery, crescimento, maturidade
- Por que a maioria das empresas ainda opera no modo "fábrica de features"

### Conceitos-chave desta fase

<b>Output vs. Outcome</b>	Output é o que você entrega (uma feature). Outcome é o resultado que ela gera (mais retenção, mais receita). Empresas de produto pensam em outcome. Fábrica de features pensa em output.
<b>Product Manager</b>	Responsável por descobrir qual produto construir e garantir que o time construa o produto certo. Atua na interseção entre negócio, tecnologia e usuário.
<b>Product Owner</b>	Papel definido pelo Scrum. Responsável por maximizar o valor do produto gerenciando o backlog e priorizando o trabalho do time de desenvolvimento.
<b>Feature Team vs. Product Team</b>	Feature team executa o que é pedido. Product team tem autonomia para resolver problemas. Marty Cagan é obsessivo nessa distinção.
<b>Squad</b>	Time multifuncional pequeno (5 a 9 pessoas) com PM/PO, designer e engenheiros, autônomo para resolver um problema específico do usuário.

### Exercícios práticos

1. Escolha um produto digital que você usa todo dia (por exemplo: seu banco, um app de música). Descreva em um parágrafo: qual problema ele resolve, para quem, e como ganha dinheiro.
2. Pesquise uma vaga de PM e uma de PO no LinkedIn. Liste as três principais diferenças nas responsabilidades descritas.
3. Leia os capítulos 1 a 6 do livro Inspired (disponível nas fontes do projeto). Anote os três insights que mais te surpreenderam.

### Referências desta fase

Inspired, de Marty Cagan (caps. 1 a 6): o que é um bom time de produto

SVPG Blog (svpg.com): artigos gratuitos de Cagan sobre Product vs. Feature Teams

Scrum Guide (scrum.org): definição oficial do papel do Product Owner — versão 2020, revisada

FASE 1 3 semanas

## ENTENDER O USUÁRIO

*Discovery, pesquisa, Jobs-to-be-Done e timeboxes de discovery*

Nenhum produto sobrevive sem entender profundamente quem é o usuário e qual problema real ele tem. Essa é a fase mais subestimada por quem está aprendendo produto e a mais valorizada por quem contrata. Discovery não é uma etapa, é um hábito contínuo.

### O que você vai aprender

- Como conduzir entrevistas com usuários de forma eficiente (Teresa Torres e Marty Cagan)
- Jobs-to-be-Done (JTBD): o framework que muda como você pensa sobre necessidade do usuário
- Como criar personas e mapas de empatia que realmente guiam decisões
- Continuous Discovery: o hábito de falar com usuários toda semana
- Como e quanto tempo dedicar ao discovery sem paralisar a entrega
- Como transformar insights de pesquisa em hipóteses de produto
- Como distinguir o que o usuário diz do que o usuário realmente precisa

### Quanto tempo dura o Discovery?

Uma das dúvidas mais comuns de quem está aprendendo produto: "discovery tem prazo?". A resposta é: depende do modelo adotado. Existem dois modelos principais em uso no mercado em 2026:

<b>Discovery por sprint (Dual Track)</b>	O time mantém uma faixa de 20 a 30% da capacidade dedicada a discovery dentro de cada sprint de 2 semanas. Não para de entregar, mas nunca para de aprender. Modelo mais comum em times maduros.
<b>Discovery contínuo (Teresa Torres)</b>	Uma entrevista por semana com um usuário real, toda semana, indefinidamente. Não é um projeto, é uma prática de time. A meta é acumular 50 a 100 entrevistas por trimestre no time.
<b>Sprint de Discovery dedicado</b>	Alguns times separam 1 sprint inteiro (2 semanas) para discovery antes de um novo ciclo de desenvolvimento. Útil para projetos grandes ou novos domínios com muita incerteza.
<b>Spike de pesquisa</b>	Bloco de tempo fixo (normalmente 3 a 5 dias) para investigar uma hipótese específica antes de estimar ou desenvolver. Muito usado no Scrum para reduzir incerteza técnica ou de usuário.

### Referência de tempo por atividade de discovery

Entrevista de usuário: 30 a 60 minutos por sessão. Ideal: 5 a 8 entrevistas por ciclo.

Análise e síntese de entrevistas: 1 a 2 horas por entrevista realizada.

Mapa de empatia ou jornada do usuário: 2 a 4 horas com o time, em workshop.

Definição de persona: 2 a 3 horas, idealmente baseada em pelo menos 5 entrevistas reais.

Opportunity Solution Tree: 1 a 2 horas para montar o mapa inicial após síntese das entrevistas.

Teste de protótipo ou validação de hipótese: 1 a 3 dias dependendo da complexidade do experimento.

## Conceitos-chave desta fase

<b>Jobs-to-be-Done</b>	As pessoas não compram produtos. Elas "contratam" produtos para realizar um trabalho em suas vidas. Ninguém quer uma furadeira, elas querem um furo na parede. Esse pensamento muda tudo na descoberta de oportunidades.
<b>Entrevista de usuário</b>	Conversa estruturada (não survey) para entender comportamentos, motivações e frustrações reais. A regra de ouro: pergunte sobre o passado, nunca sobre o futuro. "O que você fez quando..." é melhor do que "O que você faria se..."
<b>Mapa de Empatia</b>	Ferramenta visual que captura o que o usuário pensa, sente, vê, ouve, fala e faz. Ajuda o time a desenvolver empatia real pelo usuário.
<b>Persona</b>	Representação semifictícia do seu usuário ideal baseada em dados reais de pesquisa. Diferente do "usuário hipotético" criado em sala de reunião.
<b>Opportunity Solution Tree</b>	Framework de Teresa Torres que conecta o outcome desejado às oportunidades descobertas com usuários e às soluções possíveis a serem testadas.

## Exercícios práticos

4. Escolha um produto (pode ser o mesmo da Fase 0). Identifique o Job-to-be-Done principal que ele resolve. Escreva no formato: "Quando [situação], eu quero [motivação], para que [resultado esperado]."
5. Conduza uma entrevista de usuário de 30 minutos com alguém que usa o produto que você escolheu. Use as técnicas de Teresa Torres: sem perguntas hipotéticas, foco no comportamento passado. Documente no Notion.
6. Crie uma persona baseada nos dados da sua entrevista. Use o template de mapa de empatia.
7. Leia o capítulo sobre Discovery do Lean Product Playbook (disponível nas fontes do projeto).

## Referências desta fase

Continuous Discovery Habits, de Teresa Torres (referência externa)  
The Lean Product Playbook, de Dan Olsen, caps. sobre Discovery (fontes do projeto)  
Inspired, de Marty Cagan, caps. sobre pesquisa de usuários (fontes do projeto)  
Ferramenta: Miro ou FigJam para mapas de empatia

## BACKLOG, PRIORIZAÇÃO E ROADMAP

*User stories, OKRs e como decidir o que construir*

Saber o que o usuário quer é só metade do trabalho. A outra metade é decidir o que construir, quando e por quê. Essa é a essência da priorização. É aqui que o PM/PO passa boa parte do seu tempo e também onde é mais cobrado em entrevistas.

### O que você vai aprender

- Como escrever User Stories que o time realmente entende e consegue executar
- Critérios de aceitação: o contrato entre PO e time de desenvolvimento
- Definition of Ready (DoR) e Definition of Done (DoD)
- Frameworks de priorização: MoSCoW, RICE, ICE, WSJF
- Como construir, manter e comunicar um Product Roadmap adaptável
- OKRs: o framework de metas que o Google, o Spotify e a Amazon usam
- Épicos vs. Features vs. User Stories vs. Tasks: a hierarquia do backlog

### A hierarquia do backlog

<b>Épico</b>	Grande bloco de trabalho que pode durar semanas ou meses. Exemplo: "Módulo de pagamentos". Subdivide-se em features e user stories.
<b>Feature</b>	Funcionalidade específica que entrega valor ao usuário. Exemplo: "Pagamento via Pix". Parte de um épico.
<b>User Story</b>	Descrição de uma funcionalidade do ponto de vista do usuário. Formato: "Como [persona], quero [ação], para [benefício]." Exemplo: "Como cliente, quero pagar via Pix, para não precisar inserir os dados do cartão."
<b>Task</b>	Tarefa técnica para implementar uma user story. Domínio do time de engenharia.
<b>Bug</b>	Defeito em uma funcionalidade existente. Tem prioridade própria no backlog.

### Os frameworks de priorização que você precisa dominar

<b>MoSCoW</b>	Must have, Should have, Could have, Won't have. Simples e direto. Ideal para alinhar expectativas com stakeholders em sprints curtos.
<b>RICE</b>	Reach x Impact x Confidence / Effort. Gera um score numérico para comparar itens. Muito usado em times de produto com cultura de dados.
<b>ICE Score</b>	Impact x Confidence x Ease. Versão simplificada do RICE. Bom para priorizações rápidas em times pequenos.
<b>WSJF</b>	Weighted Shortest Job First. Usado no SAFe 6.0. Prioriza o que tem maior valor de negócio com menor duração de trabalho.

## Roadmap: a bússola que pode mudar de direção

O roadmap é muito mais do que uma lista de datas e features. Ele é a representação visual da estratégia do produto ao longo do tempo, e precisa ser tratado como um instrumento vivo, não como um contrato gravado em pedra.

<b>Para que serve um roadmap</b>	Comunicar prioridades estratégicas ao time, a stakeholders e à liderança. Alinhar expectativas sobre o que será construído e por quê. Ele não é um cronograma de features, é uma declaração de apostas estratégicas.
<b>Tipos de roadmap</b>	Roadmap por tema (agrupado por objetivo estratégico), por tempo (horizonte de curto, médio e longo prazo), por outcome (o que queremos mover, não o que vamos construir). O modelo mais moderno em 2026 é o roadmap orientado a outcomes.
<b>Horizonte de confiança</b>	Curto prazo (próximas 4 a 8 semanas): alta confiança, nível de detalhe alto. Médio prazo (2 a 4 meses): confiança média, itens ainda podem mudar. Longo prazo (acima de 4 meses): baixa confiança, direção estratégica apenas.
<b>Por que o roadmap muda</b>	Dados de uso revelam comportamentos inesperados. Feedback de usuários invalida hipóteses. Mercado muda. Concorrentes lançam novos produtos. Estratégia da empresa é revisada. Um bom PM atualiza o roadmap com frequência e comunica as mudanças com clareza.
<b>O erro mais comum</b>	Tratar o roadmap como compromisso de entrega. Isso gera ansiedade no time, promessas impossíveis para stakeholders e resistência a mudanças necessárias. O roadmap é uma direção, não uma promessa.

## Como acompanhar e priorizar o roadmap

Um roadmap sem revisão periódica vira um documento morto. A cadência recomendada por Roman Pichler e pela maioria das referências atuais é:

- Revisão semanal do backlog: o PO verifica se as prioridades do topo ainda fazem sentido à luz dos dados mais recentes.
- Revisão mensal do roadmap de médio prazo: o PM alinha com stakeholders, incorpora novos dados de produto e ajusta as apostas do próximo ciclo.
- Revisão trimestral do roadmap de longo prazo: conecta a evolução do produto à estratégia de negócio. Pode coincidir com o ciclo de OKRs ou com o PI Planning do SAFe.
- Evento de comunicação de mudanças: sempre que o roadmap sofrer alterações relevantes, o PM comunica ativamente o porquê, não apenas o quê. Isso constrói confiança com o time e com stakeholders.

## OKRs: o framework de metas

OKR significa Objectives and Key Results. É o sistema de metas usado pelas maiores empresas de tecnologia do mundo. Entender OKRs é obrigatório para qualquer PM/PO moderno.

<b>Objective</b>	O QUE você quer atingir. Deve ser inspirador, qualitativo e com prazo definido. Exemplo: "Tornar nossa plataforma a mais confiável do mercado financeiro."
<b>Key Results</b>	COMO você vai medir o progresso. Deve ser mensurável, com baseline e meta. Exemplo: "Reduzir o tempo de indisponibilidade de 4h/mês para 30min/mês até dezembro."
<b>Regra de ouro</b>	O Objective responde "para onde?". O Key Result responde "como sabemos que chegamos lá?". Se o KR não tiver número, não é um Key Result.

### Exercícios práticos

8. Escreva 5 User Stories para o produto que você escolheu, com critérios de aceitação no formato Gherkin (Given/When/Then).
9. Crie um backlog com 15 itens e priorize usando o framework RICE. Documente o raciocínio de cada score.
10. Defina 1 OKR para o produto: 1 Objective + 3 Key Results mensuráveis com baseline real ou estimado.
11. Construa um roadmap de 3 meses no Notion ou no Miro. Marque o nível de confiança de cada item: alto, médio ou baixo. Simule uma mudança de prioridade e escreva como você comunicaria isso a um stakeholder.

### Referências desta fase

OKR Practical Guide (português), disponível nas fontes do projeto  
 Inspired, de Marty Cagan, caps. sobre backlog e priorização (fontes do projeto)  
 Product Management Essentials, caps. sobre user stories (fontes do projeto)  
 Strategize, de Roman Pichler: referência para roadmaps orientados a outcome  
 Ferramenta: Jira ou Notion para gestão de backlog | Miro para roadmap visual

## ÁGIL NA PRÁTICA

Scrum, Kanban, timeboxes e como o PO opera no dia a dia

Ágil não é um processo. É uma mentalidade. O Scrum e o Kanban são os frameworks mais usados no mercado, e você vai trabalhar com um deles, ou os dois, em praticamente todo ambiente de produto. Mais importante do que decorar cerimônias: entender por que elas existem e quanto tempo elas devem durar.

### O que você vai aprender

- Os valores e princípios do Manifesto Ágil: o fundamento de tudo
- Scrum: papéis (PO, SM, Dev Team), eventos e artefatos
- Timeboxes de cada evento: quanto tempo cada cerimônia deve durar na prática
- Como o PO opera dentro do Scrum: refinamento, planning, review, retrospectiva
- Kanban: fluxo contínuo, limites de WIP, métricas de fluxo
- Diferença prática entre Scrum e Kanban: quando usar cada um
- Dual Track Agile: discovery e delivery acontecendo em paralelo

### Os 3 papéis do Scrum

<b>Product Owner (PO)</b>	Máximo responsável pelo valor do produto. Gerencia o Product Backlog, prioriza itens, representa os stakeholders e valida o que foi entregue. É o "o quê" e o "por quê".
<b>Scrum Master (SM)</b>	Guardião do processo Scrum. Remove impedimentos, facilita cerimônias e protege o time de interferências externas. Não é gerente.
<b>Dev Team</b>	Time multifuncional de desenvolvedores, designers e QAs. Auto-organizável. Responsável pelo "como".

### Os eventos do Scrum e seus timeboxes

Uma das perguntas mais práticas de quem está entrando na área: "Quanto tempo dura cada cerimônia?" O Scrum Guide 2020 define os timeboxes máximos. Na prática, times maduros costumam ser mais curtos do que o máximo definido.

Evento	Timebox	Participantes	Propósito
<b>Sprint</b>	1 a 4 semanas (mais comum: 2 semanas)	PO, SM, Dev Team	Iteração de trabalho. Ao final, um incremento potencialmente entregável.
<b>Sprint Planning</b>	Até 8h para sprint de 1 mês (4h para sprint de 2 semanas)	PO, SM, Dev Team	Definir o Sprint Goal e selecionar os itens do backlog para o sprint.
<b>Daily Scrum</b>	15 minutos (fixo, todos os dias)	Dev Team (SM e PO podem observar)	Sincronizar progresso e identificar impedimentos. Não é relatório.

<b>Sprint Review</b>	Até 4h para sprint de 1 mês (2h para sprint de 2 semanas)	PO, SM, Dev Team, Stakeholders	Inspeccionar o incremento e adaptar o backlog com feedback real.
<b>Sprint Retrospectiva</b>	Até 3h para sprint de 1 mês (1h30 para sprint de 2 semanas)	PO, SM, Dev Team	Inspeccionar o processo e definir melhorias para o próximo sprint.
<b>Backlog Refinement</b>	Até 10% da capacidade do sprint (4 a 8h por sprint de 2 semanas)	PO + Dev Team (SM facilita)	Detalhar, estimar e priorizar itens do backlog para os próximos sprints.

### Dica prática sobre timeboxes

Timebox é um limite, não uma meta. Se o Planning terminou em 2h e o objetivo foi atingido, encerre. Não preencha o tempo só porque ele está disponível.

Times que estouram o timebox frequentemente estão com o backlog mal refinado. Investir em refinamento reduz o tempo de planning.

Daily de 15 minutos é inegociável. Se está demorando mais, o problema não é o timebox, é a cultura de reunião do time.

O PO é obrigatório na Review e no Planning. Na Retrospectiva, é recomendado. Na Daily, pode participar como observador.

### Kanban em produto

Kanban é um sistema de gerenciamento de fluxo. Não tem sprints, não tem papéis fixos. O trabalho flui de forma contínua. Ideal para times de suporte, manutenção e operações de produto.

- Colunas típicas: Backlog > Em análise > Em desenvolvimento > Em teste > Entregue
- WIP Limit: limite de itens em cada coluna. Previne sobrecarga e identifica gargalos.
- Lead Time: tempo total desde a solicitação até a entrega. Métrica central do Kanban.
- Cycle Time: tempo de desenvolvimento ativo. Diferente do Lead Time.
- Throughput: quantidade de itens entregues por período. Indica a capacidade real do time.

### Exercícios práticos

12. Configure um board Scrum no Jira ou no Notion para o produto do seu exercício anterior. Crie um Sprint com 5 User Stories e simule um Planning de 2 horas seguindo o timebox correto.
13. Escreva um Sprint Goal para o seu primeiro sprint. Formato sugerido: "Para [usuário], [benefício], por meio de [funcionalidade]."
14. Simule uma Sprint Review: documente o que foi entregue e o que seria o feedback esperado de um stakeholder.
15. Leia o Scrum Guide 2020 completo (gratuito em [scrum.org](https://www.scrum.org)). É curto, direto e obrigatório. A versão 2020 removeu o conceito de "comprometimento" e mudou linguagem importante sobre papéis.

### **Referências desta fase**

Conhecimento em Scrum, disponível nas fontes do projeto

Scrum 360, de Jorge Audy (fontes do projeto)

Product Owner Guide 2024, disponível nas fontes do projeto

Scrum Guide 2020 oficial (scrum.org), gratuito — versão mais atual

Ferramenta: Jira (padrão do mercado) ou Notion para boards

## ESTRATÉGIA DE PRODUTO

Visão, MVP, Lean Canvas e go-to-market

Sem estratégia, o backlog é só uma lista de tarefas. Estratégia de produto é a ponte entre a visão de longo prazo e o que o time faz na próxima semana. É aqui que o PM começa a se diferenciar do PO: o PM pensa no produto como negócio, não apenas como funcionalidade.

### O que você vai aprender

- Como criar uma Product Vision convincente e inspiradora
- Lean Canvas: o canvas de negócios que substituiu o Business Model Canvas para startups
- MVP (Minimum Viable Product): o conceito mais mal interpretado em produto
- Product-Market Fit: como reconhecer e como medir
- Go-to-Market (GTM): como lançar uma funcionalidade ou produto
- Estratégia vs. Roadmap: a diferença que a maioria dos PMs confunde

### MVP: o conceito mais mal entendido em produto

MVP não é uma versão inferior do produto. É a versão mais rápida e barata de testar sua hipótese mais arriscada. O objetivo do MVP não é entregar valor, é aprender. Dan Olsen e Eric Ries explicam isso melhor do que qualquer um.

<b>Hipótese</b>	O que você acredita que é verdadeiro sobre o usuário e o problema. Exemplo: "Acreditamos que investidores de varejo não sabem interpretar seu extrato mensal."
<b>Risco mais alto</b>	A suposição que, se errada, invalida o produto inteiro. Teste essa primeiro.
<b>Experimento</b>	A forma mais rápida e barata de validar ou invalidar a hipótese. Pode ser uma landing page, um protótipo em papel, uma pesquisa ou uma entrevista.
<b>Aprendizado validado</b>	O resultado do experimento: a hipótese foi confirmada ou refutada? O que mudamos no produto com base nesse aprendizado?

### Lean Canvas: o mapa do seu produto

O Lean Canvas é uma ferramenta de uma página que captura as principais hipóteses do seu produto. Criado por Ash Maurya, é ideal para PMs que precisam comunicar a estratégia rapidamente para stakeholders.

- Problema: os 3 principais problemas que você resolve
- Segmentos de clientes: quem sente esses problemas com mais intensidade
- Proposta de valor única: por que você, e não a alternativa existente
- Solução: as 3 principais funcionalidades que resolvem os problemas
- Canais: como você alcança seus clientes

- Fluxo de receita e estrutura de custos: como o negócio sustenta o produto
- Métricas-chave: os números que indicam a saúde do produto
- Vantagem injusta: o que você tem que não pode ser facilmente copiado

### Exercícios práticos

16. Escreva uma Product Vision para o produto que você escolheu. Use o formato de Roman Pichler: "Para [cliente-alvo], que [necessidade/problema], o [produto] é um [categoria] que [benefício-chave]."
17. Preencha um Lean Canvas completo para o seu produto. Use o Miro ou o site leancanvas.com.
18. Defina o MVP de uma feature do seu produto: qual é a hipótese? Qual é o maior risco? Como testaria em 1 semana com o menor esforço possível?
19. Leia os capítulos sobre MVP e Product-Market Fit do Lean Product Playbook (fontes do projeto).

### Referências desta fase

The Lean Product Playbook, de Dan Olsen, caps. sobre MVP e Product-Market Fit (fontes do projeto)

Running Lean, de Ash Maurya (preview disponível nas fontes do projeto)

Inspired, de Marty Cagan, caps. sobre Product Vision (fontes do projeto)

## MÉTRICAS E DADOS

North Star, Lead Time, Throughput, DORA e cultura de dados em produto

Um PM que não fala de dados não passa de um gerente de opinião. Você não precisa ser cientista de dados, mas precisa saber quais números importam, como lê-los e como usar dados para tomar decisões e defender prioridades. Em 2026, o domínio de métricas de fluxo, como Lead Time e Throughput, está se tornando diferencial em entrevistas e também está entrando nos OKRs das maiores empresas de tecnologia.

### O que você vai aprender

- North Star Metric: a métrica que une toda a empresa em torno de um objetivo
- Métricas de produto vs. métricas de vaidade
- O funil de produto: aquisição, ativação, retenção, receita, indicação (AARRR)
- NPS, CSAT e CES: como medir satisfação do usuário
- Lead Time e Cycle Time: previsibilidade e saúde do fluxo de entrega
- Throughput: capacidade real de entrega do time por período
- DORA Metrics: os indicadores de DevOps que viraram padrão de mercado
- A/B Testing: o básico que todo PM precisa entender
- Churn e LTV: as métricas mais importantes de negócios SaaS

### O framework AARRR (Pirate Metrics)

<b>Acquisition</b>	Como os usuários chegam ao seu produto. Canais: orgânico, pago, indicação. Métrica: CAC (Custo de Aquisição por Cliente).
<b>Activation</b>	O momento em que o usuário experimenta o valor central do produto pela primeira vez. "Aha moment". Métrica: taxa de ativação.
<b>Retention</b>	Com que frequência os usuários voltam. É a métrica mais importante para produtos digitais. Churn alto mata qualquer negócio.
<b>Revenue</b>	Como o produto gera dinheiro. Métricas: MRR, ARR, LTV, ARPU.
<b>Referral</b>	Usuários que recomendam o produto para outros. NPS alto indica alto potencial de indicação.

### North Star Metric

A North Star Metric é a única métrica que melhor captura o valor que você entrega para o usuário. Ela alinha produto, growth e negócio em torno do mesmo objetivo. Exemplos reais:

- Spotify: minutos ouvidos por usuário por mês
- Airbnb: noites reservadas
- WhatsApp: mensagens enviadas por dia
- Nubank: transações mensais por cliente ativo

Cuidado com métricas de vaidade: número de downloads, pageviews, usuários cadastrados. São números que parecem importantes, mas não medem se o usuário está tendo sucesso com o produto.

### Lead Time, Throughput e Previsibilidade

As métricas de fluxo são hoje um diferencial real para PMs e POs que trabalham com times de desenvolvimento. Elas respondem à pergunta que todo stakeholder faz: "Quando vai ficar pronto?" — mas de forma honesta e baseada em dados históricos, não em estimativas.

<b>Lead Time</b>	Tempo total desde a criação de um item no backlog até a sua entrega ao usuário. Inclui tempo de espera, análise, desenvolvimento, testes e deploy. É a métrica mais diretamente ligada à percepção do cliente sobre velocidade.
<b>Cycle Time</b>	Tempo de trabalho ativo, ou seja, do momento em que o time começa a trabalhar em um item até ele ser entregue. Sempre menor ou igual ao Lead Time. Foco na eficiência interna do time.
<b>Throughput</b>	Quantidade de itens entregues por unidade de tempo (por semana ou por sprint). Indica a capacidade real e sustentável do time. Mais confiável do que velocity para planejamento de médio prazo.
<b>Previsibilidade com Lead Time</b>	Ao analisar o histórico de Lead Time de dezenas de itens, o PM consegue responder com confiança: "90% dos nossos itens são entregues em até X dias." Essa afirmação baseada em dados é muito mais valiosa do que uma estimativa pontual.
<b>Lead Time em OKRs</b>	Grandes empresas como Amazon, Spotify e Nubank já incluem Lead Time como Key Result em OKRs de time. Exemplo real: "Reduzir o Lead Time médio de 21 para 10 dias até o final do trimestre." Isso conecta o ritmo de entrega a um objetivo estratégico mensurável.

### DORA Metrics: o padrão moderno de DevOps e produto

As DORA Metrics (DevOps Research and Assessment) são quatro indicadores que medem a saúde do processo de entrega de software. Em 2026, elas são amplamente adotadas em empresas de tecnologia e frequentemente aparecem em entrevistas de PM sênior.

<b>Deployment Frequency</b>	Com que frequência o time faz deploy em produção. Times de elite fazem deploy múltiplas vezes por dia. Times de baixo desempenho fazem deploy a cada semana ou menos.
<b>Lead Time for Changes</b>	Tempo desde o commit do código até ele estar em produção. Mede a eficiência do pipeline de entrega. Meta de elite: menos de 1 hora.
<b>Change Failure Rate</b>	Percentual de deploys que causam incidentes em produção. Meta de elite: abaixo de 5%. Alta taxa indica problemas com testes ou qualidade.
<b>Time to Restore Service</b>	Quanto tempo o time leva para recuperar o serviço após um incidente. Meta de elite: menos de 1 hora. Mede resiliência operacional.

*O PM não é responsável pelas DORA Metrics, mas precisa entendê-las para ter conversas inteligentes com engenharia e para interpretar a saúde da entrega. Times com DORA Metrics saudáveis entregam mais valor com menos risco.*

### **Exercícios práticos**

20. Defina a North Star Metric do produto que você escolheu. Justifique por que essa métrica representa valor real entregue ao usuário.
21. Monte o funil AARRR do produto com métricas hipotéticas, mas realistas para cada etapa.
22. Escolha uma feature do seu roadmap e defina: qual métrica essa feature vai mover? Em quanto? Em qual prazo?
23. Crie uma tabela com 10 itens fictícios de backlog, com Lead Time hipotético para cada um. Calcule o percentil 85 (P85) e use esse número para responder: "Em quanto tempo o time entregaria um novo item?"
24. Pesquise as DORA Metrics do seu produto ou empresa atual. Se não tiver acesso, simule com dados hipotéticos e classifique o time como Elite, High, Medium ou Low performer.

## ÁGIL ESCALADO E SAFE

Tribos, ART, RTE e o papel do PO em grandes organizações

A maioria dos cursos de PM ignora o que acontece quando o Scrum já não é suficiente. Quando uma empresa tem 10, 20 ou 50 times de produto trabalhando em paralelo, surgem novos desafios de coordenação, dependências e alinhamento estratégico. É aqui que entram os frameworks de ágil escalado, sendo o SAFe 6.0 (Scaled Agile Framework) o mais adotado no mercado corporativo brasileiro e mundial em 2026.

### O que você vai aprender

- O que é ágil escalado e por que ele existe
- Tribos, squads, chapters e guilds: o modelo organizacional que o Spotify popularizou
- SAFe 6.0: os quatro níveis do framework e seus papéis principais
- ART (Agile Release Train): o coração operacional do SAFe
- PI Planning: o evento trimestral que alinha dezenas de times
- RTE (Release Train Engineer): o papel que mais cresce no mercado corporativo
- O papel do PO dentro do SAFe: diferenças em relação ao Scrum clássico

### Tribos, Squads, Chapters e Guilds

Antes de entrar no SAFe, é importante entender o modelo organizacional que o Spotify popularizou e que influenciou profundamente como empresas de tecnologia se organizam globalmente. Esse modelo não é exclusivo do Spotify, foi amplamente adotado (com variações) por Nubank, iFood, Mercado Livre e outras grandes empresas brasileiras.

<b>Squad</b>	Menor unidade autônoma. Time multifuncional (PM/PO, designer, engenheiros) com autonomia para resolver um problema específico do usuário. Geralmente entre 5 e 10 pessoas.
<b>Tribo (Tribe)</b>	Conjunto de squads que trabalham em problemas relacionados ao mesmo domínio de produto. Exemplo: "Tribo de Pagamentos" ou "Tribo de Onboarding". Geralmente de 50 a 150 pessoas.
<b>Chapter</b>	Grupo de pessoas com a mesma especialidade técnica que pertencem a squads diferentes dentro de uma tribo. Exemplo: "Chapter de UX" ou "Chapter de Backend". O líder do chapter cuida do desenvolvimento técnico das pessoas.
<b>Guild</b>	Comunidade de prática voluntária que cruza todas as tribos. Exemplo: "Guild de Dados" ou "Guild de Acessibilidade". Foco em compartilhar conhecimento entre squads.

### SAFe 6.0: o framework de ágil escalado mais adotado no mundo

O SAFe (Scaled Agile Framework) foi atualizado para a versão 6.0 em 2023, com foco em Business Agility, fluxo de valor e integração com IA. Em 2026, é o framework de escala mais utilizado por empresas de médio e grande porte, especialmente em setores como financeiro, saúde e governo.

<b>Team Level</b>	O nível mais básico. Times individuais trabalhando em Scrum ou Kanban. Sprints de 2 semanas. O PO e o SM atuam aqui.
<b>Program Level (ART)</b>	O coração do SAFe. O Agile Release Train (ART) é um grupo de 50 a 125 pessoas (5 a 12 times) alinhadas em torno de uma missão comum. Operam em Planning Intervals (PI) de 8 a 12 semanas.
<b>Large Solution Level</b>	Para soluções que exigem múltiplos ARTs trabalhando juntos. Usado em produtos extremamente complexos como sistemas de defesa, aeronaves ou grandes plataformas bancárias.
<b>Portfolio Level</b>	Gestão de múltiplos ARTs. Alinha investimentos estratégicos com OKRs e Value Streams da empresa. Foco em Lean Portfolio Management (LPM).

### ART: Agile Release Train

O Agile Release Train (ART) é o motor de entrega no SAFe. Pense nele como um "time de times" que compartilha uma visão, um backlog de programa e uma cadência de entrega. É a unidade organizacional central no SAFe 6.0.

- Tamanho: 50 a 125 pessoas, organizadas em 5 a 12 squads
- Cadência: Planning Intervals (PI) de 8 a 12 semanas (mais comum: 3 meses)
- Cada PI contém 4 a 5 sprints de desenvolvimento + 1 sprint de Inovação e Planejamento (IP Iteration)
- Ao final de cada PI, o ART entrega um System Demo integrado de todos os times
- O PI Planning é o evento central: 2 dias onde todos os times do ART se reúnem para planejar o próximo trimestre

### PI Planning: o evento que alinha dezenas de times

O PI Planning é considerado o evento mais importante do SAFe. É quando toda a ART para por 2 dias para planejar o próximo Planning Interval. Facilitado pelo RTE, o evento reúne POs, PMs, engenheiros, designers, liderança e stakeholders.

<b>Duração</b>	2 dias completos (event timebox fixo). Para ARTs distribuídos em múltiplos fusos horários, pode ser estendido para 3 dias.
<b>Frequência</b>	A cada Planning Interval, ou seja, a cada 8 a 12 semanas. Geralmente trimestral.
<b>Dia 1</b>	Apresentação do contexto de negócio, visão de produto e visão de arquitetura. Times fazem breakout para planejar suas iterações e identificar dependências.
<b>Dia 2</b>	Times refinam os planos, resolvem dependências entre squads, apresentam seus PI Objectives e a ART vota sua confiança no plano (Fist of Five).

<b>Outputs do PI Planning</b>	PI Objectives de cada time, Programa Board com dependências mapeadas, riscos ROAMed (Resolved, Owned, Accepted, Mitigated) e o Confidence Vote da ART.
-------------------------------	--

## RTE: Release Train Engineer

O Release Train Engineer (RTE) é o papel de liderança servidora da ART. Em termos simples, é o "Scrum Master da ART". É um dos papéis que mais cresce no mercado corporativo brasileiro, especialmente em bancos, fintechs e grandes empresas de tecnologia.

<b>Responsabilidades centrais</b>	Facilitar o PI Planning, o Scrum of Scrums, o PO Sync e o Inspect & Adapt. Remover impedimentos que transcendem o nível do squad. Gerenciar riscos e dependências entre times.
<b>Coach e líder servente</b>	O RTE não manda nos times. Ele habilita, remove obstáculos, facilita comunicação e garante que a ART siga os princípios do SAFe e do Lean-Agile.
<b>Diferença do Scrum Master</b>	O SM cuida de 1 time. O RTE cuida da ART inteira (50 a 125 pessoas). O RTE facilita eventos de programa, não de time.
<b>Métricas de responsabilidade</b>	Predictability (PI Objectives alcançados vs. planejados), Throughput da ART, DORA Metrics, Flow efficiency. O RTE usa dados para identificar gargalos sistemáticos.
<b>Certificação</b>	O SAFe 6.0 oferece a certificação SAFe RTE. Em 2026, essa certificação tem alta demanda em empresas que já adotaram SAFe.

## O papel do PO no SAFe

No SAFe, o PO não é exatamente igual ao PO do Scrum clássico. Existem nuances importantes que aparecem em entrevistas e no dia a dia de quem trabalha em empresas com SAFe implementado.

<b>PO no Scrum clássico</b>	Responsável pelo product backlog inteiro, pela visão do produto e pela priorização estratégica. Único ponto de contato com stakeholders.
<b>PO no SAFe</b>	Responsável pelo team backlog de um squad específico. Trabalha sob a direção do Product Manager (PM), que cuida do Program Backlog e da visão de produto da ART.
<b>PM no SAFe</b>	No SAFe, o Product Manager é responsável pelo backlog de programa (features), pela visão do produto na ART e pela conexão com o Portfolio. É um nível acima do PO.
<b>PO Sync</b>	Evento recorrente semanal ou quinzenal onde os POs de todos os times do ART se reúnem com o Product Manager para alinhar prioridades, dependências e ajustes de escopo. Facilitado pelo RTE.
<b>Dica prática</b>	Se você é PO em uma empresa com SAFe, seu backlog vem do PM da ART, não diretamente dos stakeholders. Entender essa hierarquia evita muita confusão e conflito.

## Exercícios práticos

25. Desenhe no Miro ou no papel a estrutura organizacional de uma empresa fictícia com 3 tribos, cada uma com 3 squads. Nomeie cada squad com um domínio de produto claro.
26. Simule um PI Planning simplificado: escreva os PI Objectives de 2 squads fictícios para o próximo trimestre, identifique 2 dependências entre eles e sugira como resolvê-las.
27. Pesquise 3 vagas de RTE ou Agile Coach com SAFe no LinkedIn. Liste as competências e certificações mais exigidas.
28. Compare os papéis de PO no Scrum Guide 2020 e no SAFe 6.0. Escreva as 3 principais diferenças que encontrar.

### Referências desta fase

SAFe 6.0 Framework ([scaledagileframework.com](https://scaledagileframework.com)) — fonte primária e atualizada  
SAFe 6.0 Implementation Roadmap — disponível nas fontes do projeto  
Scaling Software Agility — disponível nas fontes do projeto  
Scaling Agile with Atlassian and SAFe — disponível nas fontes do projeto  
SAFe PI Planning eBook — disponível nas fontes do projeto  
Certificações relevantes: SAFe PO/PM, SAFe RTE, SAFe Agilist (SA)

**FERRAMENTAS DO PM/PO***O arsenal que você usa todo dia*

Ferramentas não fazem o PM, mas saber usá-las demonstra maturidade profissional e acelera muito o seu trabalho. O mercado brasileiro usa principalmente o stack abaixo. Você não precisa dominar todas desde o início, mas precisa conhecer cada uma.

Ferramenta	Para que serve	Nível de prioridade
<b>Jira</b>	Gestão de backlog, sprints, bugs e roadmap. Padrão da indústria.	Essencial. Aprenda primeiro.
<b>Notion</b>	Documentação de produto, PRD, notas de entrevista, portfólio pessoal.	Essencial. Use no seu portfólio.
<b>Miro / FigJam</b>	Workshops, mapas de empatia, jornadas de usuário, PI Planning remoto.	Essencial para discovery.
<b>Figma (básico)</b>	Ver e comentar protótipos do designer. Você não precisa criar layouts.	Alta prioridade. Aprenda a navegar.
<b>Confluence</b>	Wiki corporativa, documentação de times. Muito usado com Jira.	Alta prioridade.
<b>SQL básico</b>	Consultar dados diretamente no banco. Diferencial real em entrevistas.	Alta prioridade. Invista 1 semana.
<b>Google Analytics / Mixpanel / Amplitude</b>	Analisar comportamento do usuário no produto.	Média prioridade. Conhecer conceitualmente.
<b>Hotjar / FullStory</b>	Mapas de calor e gravações de sessão para entender comportamento.	Média prioridade.
<b>Jira Align / Rally</b>	Ferramentas de gestão de portfólio e SAFe em escala.	Média prioridade. Relevante para empresas com SAFe.
<b>Productboard / Aha!</b>	Ferramentas dedicadas de gestão de produto. Menos comuns no Brasil.	Baixa prioridade agora.

**SQL Básico para PM/PO**

Você não precisa ser DBA. Mas saber escrever uma query básica é um diferencial enorme em entrevistas e no dia a dia. Foque nesses comandos:

- SELECT, FROM, WHERE: buscar dados com filtros

- GROUP BY, COUNT, SUM, AVG: agregar dados para análise
- JOIN: cruzar tabelas (por exemplo: usuários + transações)
- ORDER BY, LIMIT: organizar e limitar resultados

*Plataforma recomendada para aprender SQL: Mode Analytics SQL Tutorial (gratuito) ou SQLZoo.net.*

## PORTFÓLIO E ENTREVISTAS

Como transformar aprendizado em emprego

Esta é a fase mais crítica do curso. De nada adianta todo o aprendizado se você não consegue comunicar o que sabe e demonstrar na prática. Portfólio e entrevista são habilidades separadas. Ambas precisam de treino consciente.

### O que você vai aprender

- Como montar um portfólio de produto no Notion que conta uma história
- O que é um case study de produto e como escrever o seu
- Como responder as perguntas mais comuns em entrevistas de PM/PO
- Frameworks de resposta: STAR e CIRCLES
- Como resolver cases de produto ao vivo
- O que os recrutadores de produto realmente estão avaliando

### Seu portfólio de produto: o que incluir

<b>Case Study Principal</b>	Um projeto (real ou fictício) que demonstra discovery, priorização, roadmap, métricas e resultado. Este é o item mais importante do portfólio.
<b>Artefatos produzidos</b>	User Stories, OKRs, Lean Canvas, Roadmap, Personas, Mapa de empatia. Documentados e explicados no Notion.
<b>Análises de produto</b>	Escolha um produto famoso (Nubank, iFood, Inter) e analise: o que faria diferente? Que feature proporia? Qual métrica monitoraria?
<b>Presença no LinkedIn</b>	Conteúdo sobre produto, comentários em posts de referência, conexões com PMs do mercado. Visibilidade conta.

### Os frameworks de entrevista que você precisa dominar

<b>STAR</b>	Situation, Task, Action, Result. Para perguntas comportamentais: "Conta uma vez que você priorizou com dados escassos." Estruture sua resposta nesses 4 blocos.
<b>CIRCLES</b>	Comprehend, Identify customers, Report needs, Cut through prioritization, List solutions, Evaluate tradeoffs, Summarize. Para perguntas de design de produto: "Como você melhoraria o app do Itaú?"

### As 10 perguntas mais comuns em entrevistas de PM/PO

29. Por que você quer trabalhar com produto?
30. Me fale de um produto que você admira. O que mudaria nele?
31. Como você priorizaria um backlog com 50 itens e stakeholders disputando espaço?
32. Como você tomaria uma decisão de produto sem dados suficientes?

33. Qual é o produto que você mais usa? O que o torna bom?
34. Me diga um produto ruim. O que você faria de diferente?
35. Como você mediria o sucesso de uma nova feature?
36. Já teve conflito com um engenheiro ou designer? Como resolveu?
37. O que é Product-Market Fit e como você saberia se o produto o atingiu?
38. Como você trabalharia com stakeholders que sempre querem colocar tudo como prioridade máxima?

### **Plano de ação para entrevistas**

1. Crie seu portfólio no Notion antes de aplicar para qualquer vaga.
2. Treine pelo menos 5 respostas no framework STAR para perguntas comportamentais.
3. Pratique um case ao vivo com alguém que te dê feedback honesto. Use este projeto para simular.
4. Pesquise cada empresa antes da entrevista: qual produto eles têm? Qual é o problema central deles? O que você mudaria?
5. No final de toda entrevista, pergunte: "Qual é o maior desafio de produto que o time está enfrentando agora?" Isso diferencia o candidato do resto.

## Seu Diferencial: Background em Tecnologia

A maioria dos candidatos a PM/PO vem de negócios, marketing ou design. Seu background técnico é um diferencial competitivo real, e você precisa usá-lo estrategicamente durante todo o processo seletivo.

Situação	Como usar seu background técnico como vantagem
Entrevistas técnicas	Você entende APIs, sistemas, banco de dados. Pode falar de viabilidade técnica com propriedade. Isso impressiona recrutadores e times de engenharia.
Grooming e refinamento	Você já entendeu o que o engenheiro precisa saber antes de codar. Suas user stories serão mais precisas e completas do que as de 90% dos candidatos.
Conversas com o time	Você não vai sobrecarregar o time de dúvidas básicas. Isso cria confiança rapidamente.
Estimativas e viabilidade	Você pode desafiar estimativas de forma informada, sem ser arrogante. Isso é raro.
Dados e ferramentas	SQL, analytics, logs de sistema. Você vai conseguir explorar dados sem depender 100% de um analista.
SAFe e ágil escalado	Quem vem de engenharia entende o impacto de dependências técnicas entre times. Isso é ouro no nível de ART.

## Como Usar Este Projeto Como Seu Mentor

Você tem acesso a um ambiente de aprendizado ativo, com os principais livros de produto carregados como fontes. Aqui está como extrair o máximo:

- Faça perguntas conceituais: "O que é JTBD? Me dá um exemplo no contexto financeiro."
- Peça exercícios: "Me dá um exercício prático de priorização com RICE para praticar."
- Simule entrevistas: "Me entreviste para uma vaga de PO em um banco digital."
- Revise artefatos: "Aqui está minha User Story. O que está errado? Como melhorar?"
- Aprofunde conceitos dos livros: "O que Marty Cagan diz sobre product vision no Inspired?"
- Crie conteúdo: "Me ajuda a transformar esse aprendizado em um post para o LinkedIn."

### Próximos passos imediatos

1. Comece hoje pela Fase 0. Leia os capítulos 1 a 6 do Inspired (fontes do projeto).
2. Abra um espaço no Notion para documentar todos os exercícios. Isso já é o embrião do seu portfólio.
3. Escolha UM produto para usar como fio condutor de todos os exercícios do curso.
4. Siga no LinkedIn: Marty Cagan, Teresa Torres, Melissa Perri e Roman Pichler. Conteúdo gratuito de alto nível todo dia.
5. Volte aqui sempre que travar. Cada fase tem exercícios que podem ser refinados e discutidos.

# GLOSSÁRIO

Termos e siglas essenciais do universo de produto — atualizado 2026

Termo / Sigla	Definição
<b>A/B Testing</b>	Experimento controlado que divide usuários em dois grupos (A e B) para comparar qual versão de uma feature gera melhor resultado. Decisão baseada em dados, não em opinião.
<b>AARRR (Pirate Metrics)</b>	Framework de métricas criado por Dave McClure: Acquisition, Activation, Retention, Revenue, Referral. Mapeia o ciclo de vida do usuário dentro do produto.
<b>Acceptance Criteria</b>	Critérios de aceitação. Condições que uma user story precisa satisfazer para ser considerada concluída. Escritos em formato Gherkin (Given/When/Then).
<b>ART (Agile Release Train)</b>	Grupo de 50 a 125 pessoas (5 a 12 squads) que trabalham juntos em torno de uma missão comum no SAFe. Operam em Planning Intervals de 8 a 12 semanas.
<b>ARR (Annual Recurring Revenue)</b>	Receita Recorrente Anual. Indica a receita contratada que se renova a cada ano. Métrica central de negócios SaaS.
<b>Backlog</b>	Lista priorizada de tudo que precisa ser feito em um produto: features, melhorias, correções de bugs e débitos técnicos. Gerenciado pelo PO.
<b>Build Trap</b>	Armadilha de construção. Conceito de Melissa Perri: quando uma empresa mede sucesso pela quantidade de features entregues, em vez do valor gerado ao usuário.
<b>CAC (Customer Acquisition Cost)</b>	Custo de Aquisição por Cliente. Quanto a empresa gasta, em média, para conquistar um novo cliente.
<b>Change Failure Rate</b>	Percentual de deploys que causam incidentes em produção. Métrica DORA. Meta de elite: abaixo de 5%.
<b>Chapter</b>	No modelo de tribos, grupo de pessoas com a mesma especialidade técnica dentro de uma tribo. O líder do chapter cuida do desenvolvimento técnico das pessoas.
<b>CIRCLES Method</b>	Framework de entrevista para design de produto: Comprehend, Identify customers, Report needs, Cut through prioritization, List solutions, Evaluate tradeoffs, Summarize.
<b>Continuous Discovery</b>	Prática de descoberta contínua criada por Teresa Torres: manter contato semanal com usuários para gerar insights e hipóteses de produto de forma sistemática.
<b>CES (Customer Effort Score)</b>	Métrica que mede o esforço do usuário para resolver um problema com o produto. Quanto menor o esforço, melhor a experiência.
<b>CSAT (Customer Satisfaction Score)</b>	Pontuação de satisfação do cliente. Medido com uma pergunta direta após uma interação: "Quão satisfeito você ficou?"
<b>Cycle Time</b>	Tempo de desenvolvimento ativo de um item, do início do trabalho até a entrega. Sempre menor ou igual ao Lead Time. Mede eficiência interna do time.

<b>Daily Scrum</b>	Cerimônia diária de 15 minutos (timebox fixo) onde o time sincroniza o progresso e identifica impedimentos. Não é relatório de status.
<b>Definition of Done (DoD)</b>	Conjunto de critérios acordados que define quando um item do backlog está realmente concluído e pronto para ser entregue.
<b>Definition of Ready (DoR)</b>	Conjunto de critérios que um item do backlog precisa atender antes de entrar em um sprint. Garante que o time tenha informação suficiente para trabalhar.
<b>Deployment Frequency</b>	Com que frequência o time faz deploy em produção. Métrica DORA. Times de elite fazem deploy múltiplas vezes ao dia.
<b>Discovery</b>	Fase de pesquisa e validação. Processo de entender o problema do usuário e identificar a melhor solução antes de construir.
<b>DORA Metrics</b>	Deployment Frequency, Lead Time for Changes, Change Failure Rate e Time to Restore Service. Quatro indicadores de saúde do processo de entrega de software, amplamente adotados em 2026.
<b>Dual Track Agile</b>	Modelo onde Discovery e Delivery acontecem em paralelo e de forma contínua, alimentando-se mutuamente dentro do mesmo ciclo de sprints.
<b>Epic</b>	Épico. Grande bloco de trabalho que representa uma funcionalidade ampla ou objetivo de negócio. Subdivide-se em features e user stories.
<b>Feature</b>	Funcionalidade específica e entregável de um produto que resolve uma necessidade do usuário. Parte de um épico.
<b>Feature Team</b>	Time que executa o que lhe é pedido, sem autonomia para questionar ou propor soluções. Contrário do Product Team.
<b>Go-to-Market (GTM)</b>	Estratégia de lançamento de um produto ou funcionalidade: público-alvo, canais, mensagem, preço e plano de execução.
<b>Gherkin</b>	Linguagem estruturada para critérios de aceitação: Given (dado que), When (quando), Then (então).
<b>Guild</b>	Comunidade de prática voluntária que cruza todas as tribos. Foco em compartilhar conhecimento entre squads de especialidades semelhantes.
<b>Hook Model</b>	Framework de Nir Eyal: Trigger, Action, Variable Reward, Investment. Explica como produtos criam hábitos em seus usuários.
<b>ICE Score</b>	Framework de priorização: Impact, Confidence, Ease. Gera um score rápido para comparar itens do backlog.
<b>IP Iteration (Innovation &amp; Planning)</b>	No SAFe, o último sprint de cada PI. Dedicado a inovação, treinamento, hackathons e ao próprio evento de PI Planning. Não carrega entregáveis de produto.
<b>Jobs-to-be-Done (JTBD)</b>	Framework que orienta o foco do produto para "qual trabalho o usuário precisa realizar em sua vida". Criado por Clayton Christensen.
<b>KPI (Key Performance Indicator)</b>	Indicador-chave de desempenho. Métrica usada para avaliar o sucesso de uma ação, feature ou estratégia.
<b>Kanban</b>	Sistema de gestão de fluxo com colunas, cartões e WIP Limits. Foco em fluxo contínuo. Não tem sprints nem papéis fixos.
<b>Lead Time</b>	Tempo total desde a criação de um item no backlog até a sua entrega ao usuário. Métrica central de previsibilidade. Usado em OKRs de times de elite em 2026.

<b>Lead Time for Changes</b>	Tempo desde o commit do código até o deploy em produção. Métrica DORA. Meta de elite: menos de 1 hora.
<b>Lean Canvas</b>	Ferramenta de uma página para mapear hipóteses de negócio: problema, solução, métricas, canais e estrutura financeira. Criado por Ash Maurya.
<b>LTV (Lifetime Value)</b>	Receita total que um cliente gera para o negócio durante todo o seu relacionamento com o produto.
<b>MoSCoW</b>	Framework de priorização: Must have, Should have, Could have, Won't have.
<b>MRR (Monthly Recurring Revenue)</b>	Receita Recorrente Mensal. Métrica de negócios SaaS que indica a receita contratada que se renova todo mês.
<b>MVP (Minimum Viable Product)</b>	A versão mais simples de um produto ou feature que permite testar a hipótese mais arriscada com o menor esforço. O objetivo é aprender, não entregar.
<b>North Star Metric (NSM)</b>	A métrica única que melhor representa o valor central que o produto entrega ao usuário. Alinha produto, negócio e growth.
<b>NPS (Net Promoter Score)</b>	Métrica de lealdade: "Em uma escala de 0 a 10, o quanto você recomendaria este produto?" Resultado: Promotores menos Detratores.
<b>OKR (Objectives and Key Results)</b>	Framework de metas: Objective inspirador + Key Results mensuráveis. Popularizado pelo Google. Conecta estratégia a execução.
<b>Opportunity Solution Tree</b>	Ferramenta de Teresa Torres que conecta o outcome desejado às oportunidades com usuários e às soluções a serem testadas.
<b>Outcome</b>	Resultado. O impacto gerado no comportamento do usuário ou no negócio. Foco do PM. Oposto de output.
<b>Output</b>	Entrega. O que foi construído e lançado. Não mede valor, apenas volume de trabalho.
<b>PI (Planning Interval)</b>	No SAFe 6.0, timebox de 8 a 12 semanas durante o qual o ART entrega valor contínuo. Composto por 4 a 5 sprints de desenvolvimento + 1 IP Iteration.
<b>PI Planning</b>	Evento de 2 dias que abre cada Planning Interval no SAFe. Toda a ART se reúne para planejar o próximo trimestre, mapear dependências e votar a confiança no plano.
<b>PI Objectives</b>	Objetivos de curto prazo definidos por cada squad durante o PI Planning. Representam o comprometimento do time para o próximo Planning Interval.
<b>PM (Product Manager)</b>	Profissional responsável por descobrir qual produto construir, definir a estratégia e garantir que o time entregue valor ao usuário e ao negócio.
<b>PO (Product Owner)</b>	Papel definido pelo Scrum. Responsável por gerenciar o backlog, priorizar e maximizar o valor do trabalho do time de desenvolvimento.
<b>PO Sync</b>	Evento recorrente no SAFe onde os POs de todos os times do ART se reúnem com o Product Manager para alinhar prioridades e dependências.
<b>PRD (Product Requirements Document)</b>	Documento que descreve o que o produto deve fazer, para quem, por quê e como o sucesso será medido.
<b>Product-Market Fit (PMF)</b>	Estado em que o produto resolve um problema real para um segmento específico com intensidade suficiente para gerar crescimento orgânico.

<b>Product Vision</b>	Declaração de longo prazo que descreve o estado futuro que o produto deseja alcançar. Bússola estratégica do time.
<b>Refinement (Backlog Refinement)</b>	Cerimônia contínua onde PO e time revisam, detalham, estimam e priorizam itens do backlog. Timebox: até 10% da capacidade do sprint.
<b>RICE</b>	Framework de priorização: Reach × Impact × Confidence ÷ Effort. Gera score numérico para comparar itens.
<b>Roadmap</b>	Representação visual da direção estratégica do produto ao longo do tempo. Instrumento vivo, não contrato. Revisado com cadência definida.
<b>RTE (Release Train Engineer)</b>	Papel de liderança servidora da ART no SAFe. É o "Scrum Master da ART": facilita o PI Planning, remove impedimentos que transcendem o squad e gerencia riscos e dependências entre times.
<b>SAFe (Scaled Agile Framework)</b>	Framework ágil para grandes organizações. Versão atual: SAFe 6.0 (2023). Estruturado em quatro níveis: Team, Program (ART), Large Solution e Portfolio.
<b>Scrum</b>	Framework ágil baseado em sprints de tempo fixo. Três papéis (PO, SM, Dev Team), cinco eventos e três artefatos. Guiado pelo Scrum Guide 2020.
<b>Scrum Master (SM)</b>	Guardião do processo Scrum. Remove impedimentos, facilita cerimônias e protege o time de interferências. Não é gerente.
<b>Spike</b>	Tarefa de pesquisa técnica ou funcional de tempo fixo para reduzir incertezas antes de estimar ou desenvolver uma user story complexa.
<b>Sprint</b>	Iteração de tempo fixo no Scrum (1 a 4 semanas, mais comum: 2 semanas). Ao final, um incremento potencialmente utilizável.
<b>Sprint Goal</b>	Objetivo do sprint. Frase curta que descreve o valor a ser entregue. Orienta as decisões do time durante a iteração.
<b>Sprint Planning</b>	Cerimônia de abertura do sprint. Timebox: 4h para sprint de 2 semanas. PO apresenta itens, time seleciona e define o Sprint Goal.
<b>Sprint Review</b>	Cerimônia de encerramento. Timebox: 2h para sprint de 2 semanas. Time apresenta o incremento, PO e stakeholders dão feedback.
<b>Sprint Retrospective</b>	Cerimônia de melhoria contínua. Timebox: 1h30 para sprint de 2 semanas. Time inspeciona o processo e define ações de melhoria.
<b>Squad</b>	Time multifuncional pequeno e autônomo, responsável por um problema específico do usuário ou domínio do produto.
<b>Stakeholder</b>	Parte interessada. Qualquer pessoa ou grupo afetado pelo produto ou com interesse em suas decisões.
<b>STAR Method</b>	Framework para entrevistas comportamentais: Situation, Task, Action, Result.
<b>Story Points</b>	Unidade relativa de esforço no Scrum. Representa complexidade e incerteza, não horas de trabalho.
<b>Throughput</b>	Quantidade de itens entregues por unidade de tempo. Indica a capacidade real e sustentável do time. Mais confiável que velocity para planejamento.
<b>Time to Restore Service</b>	Tempo para recuperar o serviço após incidente em produção. Métrica DORA. Meta de elite: menos de 1 hora.
<b>Tribo (Tribe)</b>	Conjunto de squads que trabalham em domínios de produto relacionados. Geralmente de 50 a 150 pessoas.

<b>User Story</b>	Descrição de funcionalidade do ponto de vista do usuário. Formato: "Como [persona], quero [ação], para [benefício]."
<b>Velocity</b>	Quantidade de story points entregues por sprint. Usada para planejar sprints futuros. Métrica interna do time, não de produtividade.
<b>WIP Limit (Work In Progress)</b>	Limite de itens em andamento em cada etapa do fluxo. Previne sobrecarga e identifica gargalos no Kanban.
<b>WSJF (Weighted Shortest Job First)</b>	Framework de priorização do SAFe: prioriza o trabalho que entrega o maior valor de negócio no menor tempo possível.

---

*Curso desenvolvido com base em: Inspired (Cagan), The Lean Product Playbook (Olsen), Scrum 360 (Audy), Conhecimento em Scrum, Product Owner Guide 2024, OKR Practical Guide, Continuous Discovery Habits (Torres), Escaping the Build Trap (Perri), Hooked (Eyal), Lean UX (Gothelf), SAFe 6.0 Framework (scaledagile.com) e referências do SVPG. Atualizado em 2026.*