

10 Passos para Criar um Software que Vende

Guia prático do diagnóstico ao lançamento — para empreendedores que querem transformar uma ideia em um produto digital de sucesso.

Por que este checklist existe?

Mais de 1.000 projetos de software nos ensinaram uma lição: o sucesso de um produto digital não depende apenas de código bem escrito — depende de um processo estruturado que começa muito antes da primeira linha. Este guia reúne os 10 passos essenciais que seguimos na Fábrica de Software para transformar ideias em software que realmente gera resultado.

Diagnóstico do Problema

Antes de pensar em solução, entenda profundamente o problema. Converse com clientes reais, mapeie dores e identifique o que realmente precisa ser resolvido. Um software só vende se resolve um problema real.

Liste as 3 maiores dores do seu cliente ideal

Valide com pelo menos 5 pessoas do seu público-alvo

Dica: Se você não consegue explicar o problema em 1 frase, ainda não entendeu ele.

Definição do MVP (Produto Mínimo Viável)

Não tente construir tudo de uma vez. Defina a versão mais enxuta que entrega valor real. Liste funcionalidades essenciais, importantes e supérfluas. Construa só o essencial primeiro.

Liste 3 funcionalidades indispensáveis para o MVP

Separe o que é "desejável" do que é "essencial"

Dica: Seu MVP deve resolver UM problema bem, não vários problemas mal.

Pesquisa de Concorrência

Conheça seus concorrentes diretos e indiretos. Analise o que eles fazem bem, onde falham e como você pode se diferenciar. Use ferramentas como Google, SimilarWeb e redes sociais.

Identifique 5 concorrentes diretos

Analise avaliações negativas deles (é ouro puro)

Dica: Onde seus concorrentes falham é sua maior oportunidade.

Modelo de Negócio e Monetização

Defina como seu software vai gerar receita: assinatura mensal, taxa única, freemium, comissão? Quanto seus clientes estão dispostos a pagar? Qual o valor que seu software entrega?

Defina o modelo de precificação

Calcule o custo de aquisição de cliente (CAC)

Dica: Um software que não gera receita é um hobby, não um negócio.

Prototipação das Telas

Crie wireframes e protótipos navegáveis antes de escrever código. Ferramentas como Figma permitem testar a experiência do usuário sem gastar horas de desenvolvimento. Mostre para clientes reais.

Crie o fluxo principal do usuário (jornada)

Teste o protótipo com 3 pessoas do público-alvo

Dica: Um protótipo bem testado economiza semanas de retrabalho.

Escolha da Stack Tecnológica

A tecnologia certa faz toda a diferença. Escolha linguagens, frameworks e bancos de dados que combinem com seu projeto: escalabilidade necessária, time disponível no mercado e custo de manutenção.

Defina frontend, backend, banco de dados e infraestrutura

Considere o custo de manutenção a longo prazo

Dica: Prefira tecnologias maduras e com comunidade ativa.

Desenvolvimento Ágil com Entregas Semanais

Trabalhe em ciclos curtos (sprints) com entregas reais a cada 1-2 semanas. Isso permite validar funcionalidades rapidamente, ajustar o rumo e evitar surpresas no final do projeto.

Defina sprints de 1 a 2 semanas

Estabeleça uma ferramenta de acompanhamento (Trello, Notion)

Dica: Se não está visível, não está acontecendo.

Testes e Qualidade

Nada mata mais a credibilidade de um software do que bugs. Invista em testes automatizados, testes manuais e validação com usuários reais. Um software com bugs perde clientes.

Crie testes para as funcionalidades críticas

Convide usuários reais para testar antes do lançamento

Dica: Um bug encontrado pelo cliente custa 10x mais que um encontrado nos testes.

SEO e Estratégia de Lançamento

Antes de lançar, prepare o terreno: landing page, SEO, redes sociais, lista de espera. Um lançamento bem planejado gera tração desde o primeiro dia. Não lance no silêncio.

Crie uma landing page de pré-lançamento

Monte uma lista de espera com pelo menos 50 potenciais clientes

Dica: O lançamento começa semanas antes do software ficar pronto.

Deploy, Monitoramento e Iteração

Lançar é só o começo. Monitore métricas (usuários ativos, retenção, NPS), colete feedback e itere continuamente. Software de sucesso não é construído — é evoluído.

Configure ferramentas de monitoramento (erros, performance)

Crie um canal de feedback direto com usuários

Dica: Seu software nunca está pronto. E é assim que deve ser.

