

10 Passos para Criar um Sistema Sob Medida

Guia prático para transformar suas regras de negócio em um sistema web eficiente e escalável.

Por que este checklist existe?

Mais de 1.000 projetos de software nos ensinaram uma lição: sistemas personalizados só entregam resultados quando são construídos com base em processos bem mapeados e regras de negócio claras. Este guia reúne os 10 passos essenciais que seguimos na Fábrica de Software para transformar suas necessidades em um sistema web que realmente resolve seus desafios.

Mapeamento de Processos e Regras de Negócio

Antes de escrever qualquer linha de código, documente todos os fluxos, exceções e regras do negócio. Um sistema sem esse mapeamento corre o risco de não atender às necessidades reais da operação.

Mapeie todos os fluxos de trabalho atuais da empresa

Documente exceções e regras específicas de cada processo

Dica: Se você não consegue desenhar o processo, não consegue automatizá-lo.

Definição do Escopo e Funcionalidades Prioritárias

Separe o essencial do desejável. Defina um MVP realista com as funcionalidades que realmente resolvem os problemas mais críticos do negócio.

Liste todas as funcionalidades desejadas e priorize por impacto

Defina o MVP com as funcionalidades indispensáveis

Dica: Um sistema que faz 3 coisas bem é melhor que um que faz 10 coisas mal.

Modelagem de Dados e Arquitetura

Projete o banco de dados, APIs e integrações futuras com visão de longo prazo. Uma arquitetura bem planejada hoje evita dores de cabeça enormes amanhã.

Modele o banco de dados com todas as entidades e relacionamentos

Defina as APIs e integrações necessárias

Dica: [Invista tempo na modelagem — ela é a fundação do sistema.](#)

Escolha da Stack Tecnológica

Selecione linguagens, frameworks e infraestrutura que se adequem ao porte do projeto, à escalabilidade esperada e à disponibilidade de profissionais no mercado.

Defina frontend, backend, banco de dados e infraestrutura

Avalie custo de manutenção e escalabilidade da stack

Dica: [Prefira tecnologias consolidadas com comunidade ativa e bom suporte.](#)

Prototipação das Telas e Fluxos

Crie protótipos navegáveis das telas do sistema para validar a experiência do usuário antes do desenvolvimento. Isso reduz drasticamente o retrabalho.

Crie protótipos de todas as telas principais do sistema

Valide os fluxos com usuários reais antes de programar

Dica: [Um protótipo testado economiza semanas de desenvolvimento.](#)

Desenvolvimento Ágil com Sprints

Trabalhe com entregas quinzenais, código limpo, testes constantes e revisão contínua. A agilidade permite ajustar o rumo rapidamente e entregar valor mais cedo.

Defina sprints de 1 a 2 semanas com entregas reais

Realize code reviews e testes a cada sprint

Dica: [O melhor código é aquele que resolve o problema e é fácil de manter.](#)

Testes Automatizados e de Integração

Garanta a qualidade do sistema com testes unitários, testes de API e validação de fluxos completos. Um sistema testado é um sistema confiável.

Crie testes unitários para as funcionalidades críticas

Teste todas as integrações com APIs e sistemas externos

Dica: [Teste automatizado é seguro — teste manual é esquecível.](#)

Segurança e Conformidade (LGPD)

Implemente criptografia, controle de acesso, backups automáticos e proteção de dados em conformidade com a LGPD. Segurança não é opcional — é requisito.

Implemente autenticação e autorização por níveis de acesso

Configure criptografia de dados e backups automáticos

Dica: [Um vazamento de dados pode custar muito mais que o sistema inteiro.](#)

Homologação com Usuários Reais

Valide o sistema com usuários reais antes do lançamento oficial. Eles encontrarão problemas que nenhum teste automatizado descobre.

Selecione usuários reais para testar o sistema em produção controlada

Colete feedback e corrija os ajustes antes do lançamento oficial

Dica: [Seus usuários são seus melhores testadores.](#)

Deploy, Treinamento e Suporte

Publique o sistema em produção, treine a equipe para utilizar corretamente e ofereça suporte contínuo. Um sistema sem suporte se degrada com o tempo.

Realize treinamento presencial ou remoto com a equipe

Estabeleça canal de suporte e SLA de atendimento

Dica: [Um sistema bem implementado com suporte bom dura anos.](#)

