O que é uma **Fábrica de Software?**

- Modelo organizado para criar software com previsibilidade
- Trabalho em etapas: entender, construir, testar e entregar (processo bem definido)
- Padrões de qualidade, documentação e governança definidos (engenharia de software)
- Rastreabilidade de requisito até a entrega final do produto
- Melhoria contínua de processos e resultados alcançados

As 3 Posições-Chave na Fábrica de Software

Cada papel entrega uma parte essencial do resultado final.

ANALISTA DE SISTEMAS

- Traduz a necessidade do negócio em requisitos claros
- Define e documenta o “o quê” e o “por quê” do sistema

PROGRAMADOR

- Implementa os componentes e integra tudo em funcionamento
- Entrega o sistema aderente aos requisitos e com qualidade técnica

TESTER / QA

- Valida se o sistema funciona como descrito na documentação
- Garante qualidade, evidências e confiança antes da entrega

Requisitos → **Implementação** → **Validação**

Analista de Sistemas: Clareza, Requisitos e Documentação

A documentação é a base para construir e testar com precisão.

- Entende o negócio e traduz a necessidade em requisitos claros
- Define escopo, regras de negócio, fluxos e integrações
- Documenta cada componente do sistema (visão e detalhe)
- Requisitos Funcionais: o que o sistema faz (casos, telas, regras)
- Requisitos Não Funcionais: segurança, performance, disponibilidade, auditoria, usabilidade
- Define critérios de aceite: como saber que “está pronto” e correto

Entregáveis do Analista

User Stories / Casos de Uso

Diagramas (UML/BPMN) + Fluxos

Protótipos / Wireframes

Critérios de Aceite + Regras

Sem documentação detalhada, o time constrói no escuro.

Arquiteto de Software: Visão, Estrutura e Decisões Técnicas

Define como o sistema será construído para ser escalável, seguro e sustentável.

- Define a arquitetura: componentes, camadas, integrações e padrões
- Traduz requisitos não funcionais em decisões técnicas (performance, segurança, disponibilidade)
- Seleciona tecnologias e estabelece diretrizes (frameworks, serviços, padrões de código)
- Garante coesão técnica e evita “arquitetura Frankenstein”
- Antecipação de riscos: escalabilidade, custo, manutenção, observabilidade
- Apoia o time (Analista, Dev e QA) com referência técnica e governança

Entregáveis do Arquiteto

Diagramas de arquitetura (C4/UML) + visão de componentes

Padrões e guidelines (boas práticas, convenções, code review)

Decisões registradas (ADR) + padrões de integração

Requisitos NFR operacionalizados (SLA, segurança, observabilidade)

“Arquitetura bem feita reduz custo futuro e aumenta confiabilidade.”

Programador: Construção, Integração e Execução

Quem transforma a documentação em sistema real e funcional.

- Implementa cada componente do sistema (front, back, integrações, dados)
- "Dá vida" às regras de negócio com código correto e consistente
- Mantém o sistema coeso: arquitetura, padrões e organização
- Garante aderência total aos requisitos e critérios de aceite
- Trata exceções, validações e cenários de erro com robustez
- Entrega com controle de versão e rastreabilidade (mudanças e releases)

Entregáveis do Programador

- Código versionado (Git)
- APIs / Integrações
- Banco de dados + migrações
- Documentação técnica + testes unitários

"Código bom não é só 'funcionar': é ser sustentável e confiável."

Tester (QA): Validação, Evidências e Qualidade

Garante que o sistema responde exatamente ao que a documentação exige.

- Valida cada funcionalidade com base na documentação e critérios de aceite
- Testa fluxos, integrações e cenários de erro (casos positivos e negativos)
- Garante rastreabilidade: requisito → teste → evidência → aprovação
- Registra defeitos com clareza (passos, impacto, prioridade)
- Testes não funcionais (quando aplicável): performance, segurança básica, usabilidade
- Apoia a homologação e libera a entrega com confiança

Entregáveis do Tester

Plano de Testes

Casos de Teste + Checklist

Relatórios de Defeitos

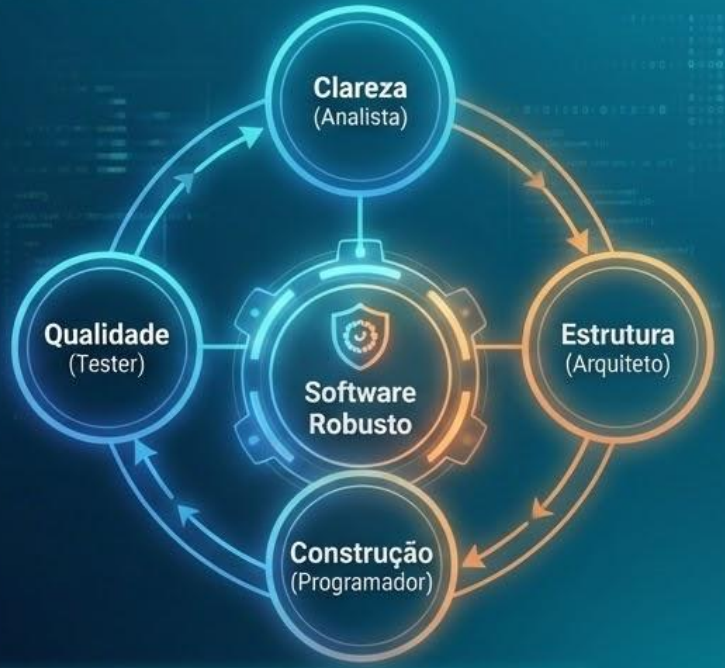
Evidências (prints/logs) + Status de Homologação

"Testar não é 'caçar bug': é garantir conformidade e confiança."

Conclusão: 4 Papéis, 1 Software Robusto

Clareza + Arquitetura + Execução + Validação = Qualidade real.

- **Analista** garante **clareza**: requisitos completos e rastreáveis
- **Arquiteto** garante **estrutura**: decisões técnicas e padrões que sustentam o sistema
- **Programador** garante **execução**: implementação integrada, coesa e sustentável
- **Tester** garante **confiança**: validação e evidências antes da entrega
- A ausência de qualquer um aumenta **risco**: retrabalho, falhas e custo elevado
- Quando os 4 atuam juntos, a entrega fica previsível, escalável e segura



"Time alinhado + arquitetura sólida = entrega certa, com qualidade e menos custo."