

# Teste estáticos

## Noções Básicas [↗](#)

### Exame Manual [↗](#)

- Consiste na revisão manual dos produtos de trabalho, sem a execução do código.

### Análise Estática [↗](#)

- Utiliza ferramentas como SonarQube, verificação ortográfica/gramatical (ex: Microsoft Word), entre outras.
- O código não é executado, ou seja, nada é “rodado”.
- Objetivos:
  - Melhorar a qualidade.
  - Detectar defeitos precocemente.
  - Avaliar legibilidade, integridade, correção, testabilidade e consistência.
- Pode ser usada tanto para **verificação** quanto para **validação**.
- Técnicas de revisão ajudam a garantir que as histórias de usuário sejam completas, compreensíveis e com critérios de aceite testáveis.
- Testadores devem fazer perguntas para esclarecer e melhorar as histórias — **não tenha medo de perguntar**.
- Pode ser mais eficiente que testes dinâmicos, exigindo menos esforço (sem necessidade de casos de teste).
- Ferramentas de análise estática costumam ser integradas ao CI (Integração Contínua).

### Exemplos de Ferramentas: [↗](#)

- Verificadores ortográficos.
- Ferramentas de legibilidade.
- Analisadores de código como o SonarQube.

---

## Produtos de Trabalho Avaliáveis por Testes Estáticos [↗](#)

Qualquer item produzido durante o projeto pode ser revisado ou analisado estaticamente, como:

- Especificações
- Histórias de usuário
- Arquitetura
- Código
- Páginas web
- Documentação
- Contratos
- Planos de projeto, teste e casos de teste
- Cartas de teste
- Etc.

---

## Valor do Teste Estático [↗](#)

- Detecta defeitos nas fases iniciais do ciclo de vida do software (SDLC), alinhado ao princípio do **teste antecipado**.
- Encontra problemas que testes dinâmicos não detectam.

- Aumenta a confiança e qualidade dos produtos.
  - Ajuda stakeholders a validarem se os requisitos refletem suas necessidades reais.
  - Melhora a comunicação entre os envolvidos.
  - Reduz custos gerais do projeto ao evitar retrabalho em fases mais avançadas.
  - Detecta defeitos de forma mais eficiente do que os testes dinâmicos.
- 

## Diferenças entre Teste Estático e Dinâmico [↗](#)

- **Complementares**, pois identificam tipos diferentes de defeitos.
- Estático foca no conteúdo dos produtos de trabalho; dinâmico, no comportamento do software em execução.
- Defeitos encontrados estaticamente são mais baratos e rápidos de corrigir.

## Defeitos Comuns Detectados Estaticamente: [↗](#)

- Requisitos: inconsistências, ambiguidades, contradições, omissões, imprecisões, redundâncias.
  - Projeto: algoritmos ineficientes, acoplamento alto, baixa coesão.
  - Código: variáveis não usadas, código inacessível ou duplicado.
  - Conformidade: desvios de padrões de codificação.
  - Interfaces: especificações incorretas (ex: unidade de medida inconsistente).
  - Segurança: vulnerabilidades detectáveis sem execução.
- 

## Processo de Feedback e Revisão [↗](#)

### Benefícios do Feedback Antecipado e Frequente: [↗](#)

- Comunica possíveis problemas de forma precoce.
  - Evita mal-entendidos e facilita a implementação correta de alterações.
  - Aumenta a compreensão do que está sendo desenvolvido.
  - Direciona o foco da equipe para funcionalidades de maior valor e impacto.
- 

## Atividades do Processo de Revisão [↗](#)

### Planejamento [↗](#)

Definição de:

- Escopo e objetivo da revisão.
- Produto a ser revisado.
- Características de qualidade a avaliar.
- Critérios de saída, padrões, esforço e prazos.

### Início da Revisão [↗](#)

- Garantir que todos estejam preparados e tenham acesso ao produto.
- Definir funções, responsabilidades e materiais necessários.

### Revisão Individual [↗](#)

- Cada participante avalia o produto de forma independente, registrando defeitos, recomendações e dúvidas.

### Comunicação e Análise [↗](#)

- Compartilhar anomalias encontradas.

- Analisar e decidir ações para cada item (nem toda anomalia é um defeito).

## Correção e Relatório [↗](#)

- Registrar defeitos e acompanhar correções.
  - Produto pode ser aceito quando os critérios de saída forem atendidos.
  - Resultados da revisão devem ser documentados.
- 

## Funções e Responsabilidades em Revisões Formais [↗](#)

*Uma mesma pessoa pode exercer mais de uma função.*

- **Gerente:** decide o que será revisado e fornece os recursos (tempo, equipe).
  - **Autor:** cria e corrige o produto.
  - **Moderador (Facilitador):** conduz as reuniões, gerencia o tempo e promove um ambiente seguro.
  - **Relator (Registrador):** registra anomalias e informações da revisão.
  - **Revisor:** avalia o produto. Pode ser alguém da equipe, um especialista ou stakeholder.
  - **Líder da Revisão:** organiza o processo e define os participantes.
- 

## Tipos de Revisão [↗](#)

- **Informal:**
    - Sem processo estruturado.
    - Foco em detectar e resolver rapidamente.
    - Pode ser feita por colegas.
    - Documentação é opcional.
    - Muito comum em metodologias ágeis.
  - **Acompanhamento (Walkthrough):**
    - Conduzido pelo autor.
    - Serve para instrução, alinhamento, validação, geração de ideias, etc.
    - Revisores podem (ou não) revisar previamente.
  - **Revisão Técnica:**
    - Conduzida por especialistas, com liderança de um moderador.
    - Foco em consenso técnico, qualidade e confiança.
  - **Inspeção:**
    - Mais formal.
    - Prioriza a detecção de anomalias.
    - Coleta métricas para melhorar o processo.
    - O autor não pode atuar como líder ou relator.
- 

## Fatores de Sucesso para Revisões [↗](#)

- Estabelecer objetivos e critérios de saída claros e mensuráveis.
- Selecionar o tipo de revisão adequado ao contexto.
- Revisar por partes pequenas para manter a atenção dos participantes.
- Fornecer feedback aos stakeholders e autores.
- Garantir tempo para preparação dos revisores.

- Obter apoio da gestão.
- Integrar revisões à cultura organizacional.
- Capacitar os participantes quanto às suas funções.
- Facilitar reuniões eficazes.