

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**

Testes funcionais e não funcionais

Introdução aos testes de software e tipos de testes

Aula 1

Código da aula: [SIS]ANO1C3B1S2A1

Exposição



Objetivo da aula

Conhecer os conceitos acerca de testes de software, princípios e práticas essenciais.



Competências da Unidade (técnicas e socioemocionais)

- Desenvolver soluções de software (*back-end*, *front-end* e *full-stack*) utilizando técnicas, métodos, ferramentas e linguagens de programação diversas;
- Executar manutenção de programas;
- Resolução de problemas.



Recursos didáticos

- Caderno para anotações;
- Recurso audiovisual para exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou internet.



Duração da aula

50 minutos

Compreender os testes de software: princípios e práticas essenciais

Os testes de software são essenciais para garantir a qualidade e o funcionamento correto das aplicações. Eles detectam defeitos, validam funcionalidades e garantem que requisitos do usuário e expectativas de negócio sejam atendidos.

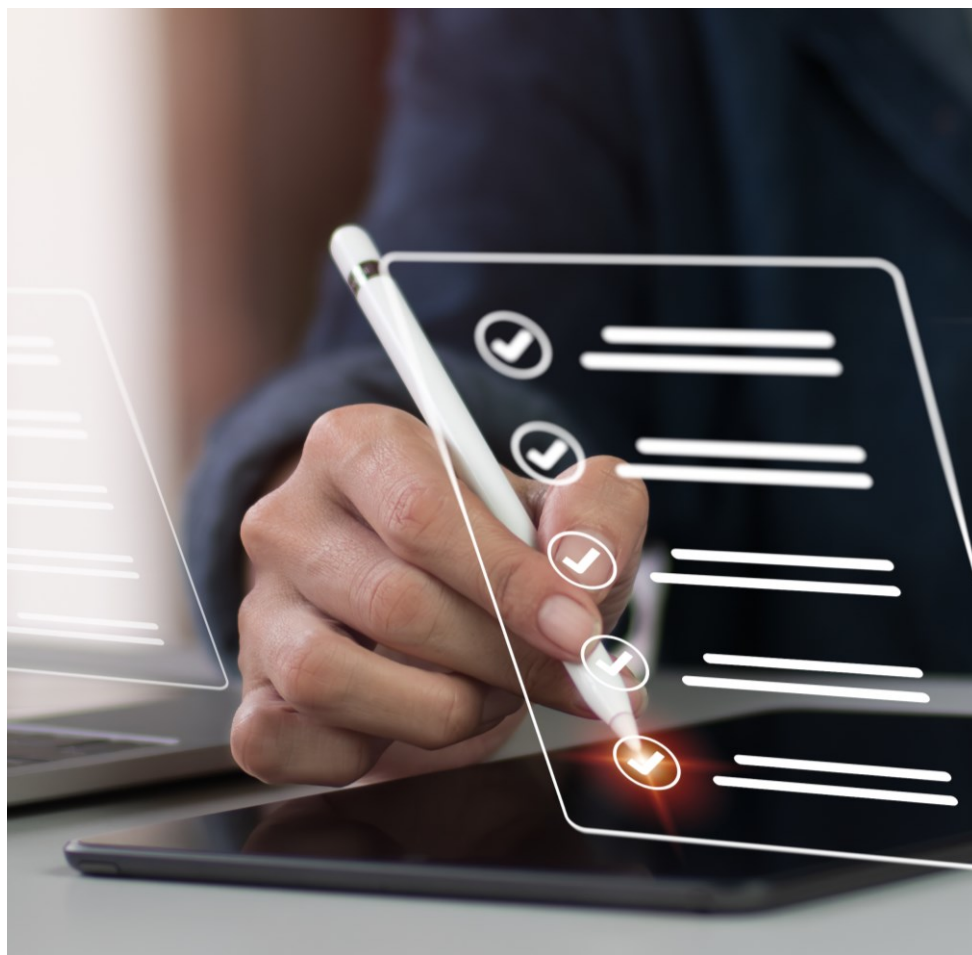
- ✓ Conhecer objetivos e princípios dos testes de software;
- ✓ Aplicar processos de teste de software: visão geral das fases do teste;
- ✓ Integrar níveis de teste de software: diferenciação entre testes unitários, de integração, de sistema e de aceitação.

Exposição

O que são testes de software?

Os **testes de software** são processos utilizados para avaliar a qualidade do software e determinar se ele **atende aos critérios de exigência** estabelecidos.

Eles envolvem a execução de um programa ou aplicação com a intenção de encontrar erros de software e garantir que o comportamento do sistema esteja conforme o esperado.



© Getty Images



Importante

A etapa de teste em software é muito importante para garantir a qualidade da entrega e evitar problemas na operação da empresa por possíveis problemas.

Exposição

Objetivos dos testes de software

Os principais objetivos dos testes de software incluem:



Verificação

Confirmar que o software atende aos requisitos técnicos e funcionais especificados.



Validação

Assegurar que o produto de software atende às expectativas e necessidades do usuário final.



Descoberta de defeitos

Identificar e documentar qualquer falha que o software apresente em relação às expectativas e especificações.



Garantia de qualidade

Proporcionar aos *stakeholders* a confiança de que o software possui um nível aceitável de qualidade.



Prevenção de defeitos

Utilizar as informações obtidas por meio dos testes para prevenir a ocorrência de erros futuros.

Fonte: GRAHAM, 2012. Elaborado especialmente para curso.

Vamos
fazer uma
atividade

Análise de situação-problema

Motivação: Vocês são uma equipe de testadores de QA (*Quality Assurance*) recém-contratados por uma *startup* que está desenvolvendo uma aplicação de *to-do list* chamada TaskMaster.

A TaskMaster permite que os usuários criem, editem e excluam tarefas, marquem tarefas como completas e filtrem tarefas por status. A aplicação está em sua fase inicial de desenvolvimento e vocês foram encarregados de assegurar que todas as funcionalidades operem conforme esperado, antes do lançamento.

Objetivo: Criar um plano de teste de software simples para uma aplicação de *to-do list*.

Exemplo hipotético elaborado especialmente para o curso.



Vamos
fazer uma
atividade

Análise de situação-problema

Requisitos da aplicação:

- Os usuários devem ser capazes de criar uma nova tarefa com um título e descrição;
- Os usuários devem ser capazes de editar o título e descrição de uma tarefa existente;
- Os usuários devem ser capazes de excluir uma tarefa;
- Os usuários devem ser capazes de marcar uma tarefa como completa;
- Os usuários devem ser capazes de filtrar tarefas por todas, completas e incompletas.

Exemplo hipotético elaborado especialmente para o curso.

Vamos
fazer uma
atividade

Criar um plano de teste de software simples para uma aplicação de *to-do list*.



30 minutos



Em grupos de até 6 pessoas

Análise de situação-problema

- 1** Entender os requisitos: discutir e fazer perguntas para entender todos os requisitos da aplicação de *to-do list*;

Criar casos de teste: com base nos requisitos acima, criar casos de teste que cubram todas as funcionalidades.
- 2** Certifiquem-se de incluir casos para testar a funcionalidade normal, bem como casos de borda e condições de erro;
- 3** Priorizar testes: decidir a ordem em que os testes devem ser executados. Considerar a importância da funcionalidade, a facilidade de teste e o risco associado.

Registro



Envio





O que nós
aprendemos
hoje?

Hoje desenvolvemos:

- 1** O conhecimento sobre o papel e a importância da etapa de testes em software e o motivo pelo qual os softwares devem ser entregues com testes realizados;
- 2** A análise dos diferentes objetivos que devem ser aplicados na etapa de verificação de software;
- 3** A criação de um breve relatório de teste de software com base em um cenário proposto de aplicação.

© Getty Images



Saiba mais

Para conhecer mais detalhes sobre onde podemos começar a executar testes de software, assista ao vídeo: **PESSONIZANDO. TESTE DE SOFTWARE: por onde começar**. Disponível em: <https://www.youtube.com/watch?v=NnamjfPYuiY>. Acesso em: 4 jan. 2024.

```
void _decode_(char cbuff **buff)
{
    if (step == AES_LOC_PASS) {
        src = cbuff->load();
        dest = getattr(&ptr, &mod,
            if (mod != NULL) as dest)
        dest += buffer->TABLE(mod)
        mask += (previous_aes12
            if (mask & SIG_KERNEL) !=
        return _ERROR_
        return 0;
    }
}
```


Referências da aula

- GRAHAM, D.; BLACK, R.; VEENDAAL, E. **Foundations of Software Testing ISTQB Certification**. São Paulo: Cengage Learning EMEA, 2012.
- PESSONIZANDO. TESTE DE SOFTWARE: por onde começar. Disponível em: <https://www.youtube.com/watch?v=NnamjfPYuiY>. Acesso em: 4 jan. 2024.
- Identidade visual: imagens © Getty Images

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**