



## Atividade de Verificação e Validação

---

Alunos:

Anderson Viana Dias

José Augusto Oliveira Sant'Ana

**A equipe deve apresentar os seguintes elementos ao realizar a pesquisa sobre duas ferramentas de testes de software para cada uma das seis categorias abordadas no artigo "Software Verification and Validation Technologies and Tools":**

1. Nome da ferramenta;
2. Tipo de ferramenta;
3. Intenção de cada ferramenta;
4. Identificação das funcionalidades de cada ferramenta;
5. Comparação entre as duas ferramentas de uma mesma categoria, buscando levantar suas diferenças.
6. Quais são as lacunas nas ferramentas de uma categoria que podem permitir a construção de uma "nova"?

---

O conteúdo apresentado faz a demonstração da importância e o impacto dos testes de softwares, cada falha de segurança ou dentro a bruta estrutura do projeto pode acarretar o fim de uma empresa. Cada projeto tem um nível de requisitos a serem levantados, sendo um deles a linguagem escolhida para atender o objetivo do projeto, vemos que cada linguagem de programação tem suas próprias ferramentas de testes, sejam estes de ambiente ou testes unitários.

---

### Ferramentas de teste funcional.

#### **pytest**

**Tipo de ferramenta:** Framework de teste unitário e funcional em Python.

**Intenção da ferramenta:** O pytest tem como intenção oferecer uma estrutura simplificada e flexível para criar e executar testes automatizados em projetos Python, abrangendo tanto testes de unidade quanto testes funcionais.

**Identificação das funcionalidades da ferramenta:** O pytest permite a criação de testes de forma concisa usando funções e marcações, suporta assertivas detalhadas e personalizáveis, realiza a descoberta automática de testes em um diretório e gera relatórios abrangentes de resultados. Além disso, oferece integração com outras ferramentas de teste e CI/CD, facilitando a incorporação dos testes no fluxo de desenvolvimento.

## PyUnit

**Tipo de ferramenta:** Framework de teste unitário em Python.

**Intenção da ferramenta:** O PyUnit é projetado para possibilitar a criação e execução de testes de unidade automatizados em projetos Python, visando garantir a qualidade e confiabilidade do código.

**Identificação das funcionalidades da ferramenta:** PyUnit oferece recursos para escrever testes de unidade, criar suites de testes organizadas, executar testes individualmente ou em grupo, gerar relatórios de resultados e integrar-se com outras ferramentas de teste. Ele segue uma abordagem mais clássica, onde os testes são definidos por meio de classes e métodos.

---

### Comparação entre as duas ferramentas de uma mesma categoria, buscando levantar suas diferenças.

O pytest se destaca por sua sintaxe concisa e expressiva, que facilita a escrita de testes e a compreensão dos resultados. Ele também oferece a funcionalidade de descoberta automática de testes, o que economiza tempo na configuração. Além disso, a ampla gama de plugins disponíveis e a integração com outras ferramentas o tornam uma opção flexível.

Em contrapartida, o PyUnit segue uma abordagem mais tradicional de orientação a objetos para a criação de testes. Isso pode ser preferível em cenários onde a estrutura de classes e métodos é mais natural. No entanto, a sintaxe tende a ser mais verbosa em comparação com o pytest.

### Quais são as lacunas nas ferramentas de uma categoria que podem permitir a construção de uma "nova"?

As lacunas nas ferramentas de teste funcional podem incluir a falta de recursos de teste mais avançados, como a capacidade de executar testes de carga ou testes de integração complexos. Além disso, pode haver espaço para uma ferramenta que ofereça uma experiência de usuário ainda mais simplificada e intuitiva na criação e execução de testes funcionais, especialmente para desenvolvedores iniciantes. Uma nova ferramenta também poderia buscar uma integração mais aprofundada com outras ferramentas de desenvolvimento e CI/CD, proporcionando um fluxo de trabalho mais coeso.

## Ferramentas de teste de manutenibilidade.

### Complexity Report

**Tipo de ferramenta:** Ferramenta de análise de código e métricas de complexidade.

**Intenção da ferramenta:** A intenção do Complexity Report é avaliar a manutenibilidade do código-fonte de um software, identificando áreas que podem ser problemáticas devido a complexidade excessiva. A ferramenta visa ajudar os desenvolvedores a melhorar a qualidade e legibilidade do código, facilitando a manutenção futura.

**Identificação das funcionalidades da ferramenta:** O Complexity Report realiza análise estática do código para calcular métricas de complexidade, como complexidade ciclomática e aninhamento de estruturas condicionais. Ele pode gerar relatórios que destacam trechos de código complexos, fornecendo insights sobre quais partes do código podem precisar de refatoração para melhorar a manutenibilidade.

### CPPCheck

**Tipo de ferramenta:** Ferramenta de análise estática de código.

**Intenção da ferramenta:** A intenção do CPPCheck é identificar potenciais erros e problemas no código-fonte escrito em linguagens C e C++. Ele foca em identificar vulnerabilidades, vazamentos de memória, uso inadequado de ponteiros, entre outros problemas que podem afetar a segurança e manutenibilidade do código.

**Identificação das funcionalidades da ferramenta:** O CPPCheck analisa o código-fonte por meio de análise estática, identificando padrões que indicam erros ou más práticas. Ele pode detectar problemas como vazamentos de memória, uso de ponteiros nulos, acesso a memória desalocada e uso incorreto de funções. A ferramenta gera relatórios detalhados que auxiliam os desenvolvedores a corrigir esses problemas antes que eles causem problemas de segurança ou dificuldades de manutenção.

---

### **Comparação entre as duas ferramentas de uma mesma categoria, buscando levantar suas diferenças.**

A diferença entre Complexity Report e CPPCheck reside na abordagem para avaliar a manutenibilidade do código. O Complexity Report pode ser uma ferramenta que calcula métricas de complexidade do código-fonte, como a complexidade ciclomática, facilitando a identificação de áreas de código que podem ser difíceis de manter ou entender. Por outro lado, o CPPCheck é uma ferramenta de análise estática focada em identificar erros e problemas específicos de C/C++ que podem impactar a manutenibilidade, como vazamentos de memória e uso inadequado de ponteiros.

### **Quais são as lacunas nas ferramentas de uma categoria que podem permitir a construção de uma "nova"?**

Uma possível lacuna nas ferramentas de teste de manutenibilidade é a falta de uma abordagem holística que combine métricas de complexidade do código com análise estática detalhada para identificar problemas específicos que afetam a manutenibilidade. Uma "nova" ferramenta poderia visar essa integração, oferecendo uma visão abrangente da qualidade e manutenibilidade do código, ao mesmo tempo em que fornece insights sobre problemas específicos que podem surgir durante a manutenção. Além disso, uma nova ferramenta poderia explorar a automação de refatorações recomendadas para melhorar a manutenibilidade do código.

## **Ferramentas de teste de usabilidade.**

### **UserTesting**

**Tipo de ferramenta:** Plataforma de teste de usabilidade.

**Intenção da ferramenta:** UserTesting visa facilitar a realização de testes de usabilidade e coletar feedback dos usuários sobre produtos e interfaces, ajudando as equipes de design e desenvolvimento a melhorar a experiência do usuário.

### **Identificação das funcionalidades da ferramenta:**

Recrutar participantes, definir cenários e tarefas, gravar sessões com áudio e vídeo, coletar métricas de desempenho e satisfação e gerar análises e relatórios abrangentes.

### **Website Grader**

**Tipo de ferramenta:** Ferramenta de avaliação de websites.

**Intenção da ferramenta:** O Website Grader é projetado para avaliar a eficácia geral de um site, levando em consideração vários fatores que afetam a usabilidade e a performance.

**Identificação das funcionalidades da ferramenta:**

Avaliar o desempenho abrange velocidade de carregamento e adaptação a dispositivos móveis. Otimizar para SEO é essencial na busca online. Analisar redes sociais revela a percepção do público. Identificar oportunidades de melhoria fecha o ciclo de aprimoramento constante.

---

**Comparação entre as duas ferramentas de uma mesma categoria, buscando levantar suas diferenças.**

O UserTesting é mais focado em testes de usabilidade e proporciona uma abordagem mais interativa, permitindo a realização de testes com usuários reais, gravações de sessões e feedback direto. O Website Grader, por outro lado, é voltado para a avaliação automática de websites, fornecendo uma visão geral das métricas de desempenho e SEO, mas não oferece a interação com usuários reais como o UserTesting.

**Quais são as lacunas nas ferramentas de uma categoria que podem permitir a construção de uma "nova"?**

Poderia ser desenvolvida uma nova ferramenta que combine aspectos do UserTesting e do Website Grader, proporcionando testes de usabilidade mais automatizados e detalhados. Essa nova ferramenta poderia realizar avaliações de usabilidade, desempenho e SEO de forma integrada, permitindo interações com usuários reais, bem como análises automáticas, oferecendo um pacote completo para a melhoria contínua das experiências online.

## Ferramentas de teste de segurança.

### Bandit

**Tipo de ferramenta:** Ferramenta de análise de segurança de código-fonte.

**Intenção da ferramenta:** A ferramenta Bandit é usada para identificar vulnerabilidades de segurança no código-fonte de aplicações Python, ajudando desenvolvedores a garantir que seus códigos não contenham brechas de segurança conhecidas.

**Identificação das funcionalidades da ferramenta:**

Realizar uma análise estática do código-fonte em Python é essencial para identificar possíveis vulnerabilidades. Isso envolve a detecção de problemas de segurança, como injeções SQL e crosssite scripting (XSS), que podem comprometer a integridade e a segurança de um sistema. Ao fazer essa análise, busca-se identificar fragilidades no código que possam ser exploradas por invasores mal-intencionados.

### Vega

**Tipo de ferramenta:** Plataforma de teste de segurança de aplicações web.

**Intenção da ferramenta:** Vega é projetada para auxiliar na avaliação de segurança de aplicações web, identificando vulnerabilidades e fornecendo informações sobre como corrigi-las.

**Identificação das funcionalidades da ferramenta:**

---

**Comparação entre as duas ferramentas de uma mesma categoria, buscando levantar suas diferenças.**

O Bandit foca especificamente na análise de código-fonte Python em busca de vulnerabilidades, enquanto o Vega é mais abrangente e se concentra na avaliação automatizada de segurança de aplicações web. O Bandit é voltado para desenvolvedores que desejam verificar seus códigos por vulnerabilidades específicas, enquanto o Vega é mais adequado para testadores de segurança que buscam identificar vulnerabilidades em aplicações web de forma geral.

**Quais são as lacunas nas ferramentas de uma categoria que podem permitir a construção de uma "nova"?**

Poderia ser desenvolvida uma nova ferramenta que combine as funcionalidades do Bandit e do Vega, proporcionando uma análise de segurança mais abrangente que inclua tanto a análise de código-fonte quanto a avaliação de aplicações web. Essa nova ferramenta poderia oferecer uma abordagem mais integrada para identificar e corrigir vulnerabilidades em todas as camadas de uma aplicação, resultando em uma avaliação mais completa da segurança.

**Ferramentas de teste de performance.****Google PageSpeed Insights**

**Tipo de ferramenta:** Online, baseada em navegador.

**Intenção da ferramenta:** A ferramenta Google PageSpeed Insights tem como objetivo avaliar a velocidade e o desempenho de uma página da web, oferecendo insights sobre como melhorar a velocidade de carregamento e a experiência do usuário.

**Identificação das funcionalidades da ferramenta:** A ferramenta analisa uma página da web e fornece uma pontuação de desempenho, indicando o quão bem otimizada está a página. Ela oferece sugestões específicas para melhorias, como compressão de imagens, minificação de recursos, redução do tempo de resposta do servidor e outros elementos que podem impactar o desempenho da página.

**OpenSTA**

**Tipo de ferramenta:** Software de teste de carga de código aberto.

**Intenção da ferramenta:** O OpenSTA é uma ferramenta de teste de carga projetada para simular o tráfego realista em um sistema, avaliando como ele lida com uma grande quantidade de solicitações. Isso ajuda a identificar gargalos e pontos fracos no desempenho do sistema.

**Identificação das funcionalidades da ferramenta:** O OpenSTA permite criar cenários de teste complexos em que várias solicitações podem ser enviadas simultaneamente para avaliar o desempenho sob carga. Ele coleta métricas de desempenho, como tempos de resposta e taxas de erro, e fornece informações sobre como o sistema se comporta sob diferentes cargas.

---

**Comparação entre as duas ferramentas da mesma categoria, buscando levantar suas diferenças.**

O Google PageSpeed Insights e o OpenSTA são ferramentas de teste de desempenho, mas com focos diferentes. O PageSpeed Insights é uma ferramenta online que se concentra em avaliar o desempenho de

páginas da web, oferecendo insights e sugestões para otimização. Ele fornece uma visão mais voltada para a experiência do usuário e a otimização de recursos web.

Por outro lado, o OpenSTA é uma ferramenta de código aberto que se concentra em testes de carga, simulando cargas realistas em sistemas para avaliar seu desempenho sob pressão. Ele é mais orientado para identificar como um sistema responde a um grande número de solicitações e qual é o seu limite de capacidade.

**Quais são as lacunas nas ferramentas de uma categoria que podem permitir a construção de uma "nova"?**

Uma lacuna que pode permitir a construção de uma nova ferramenta de teste de desempenho é uma abordagem mais holística que combine os aspectos de avaliação de desempenho de páginas web com testes de carga em sistemas. Isso permitiria aos desenvolvedores obter insights não apenas sobre a otimização da experiência do usuário, mas também sobre como o sistema como um todo lida com diferentes cargas de tráfego. Além disso, uma nova ferramenta poderia oferecer uma análise mais detalhada das métricas de desempenho, incluindo aspectos como latência, consumo de recursos e estabilidade sob carga extrema.