

TESTES DA APLICAÇÃO (INTEGRADOS, FUNCIONAIS, NÃO FUNCIONAIS) E MÉTRICAS DE TESTES (USABILIDADE X FUNCIONALIDADE)

A ARQUITETURA DA INFORMAÇÃO PRECISA PENSAR NA PARTE PRÁTICA DAS APLICAÇÕES. PENSAR SE SEUS OBJETIVOS FINAIS FORAM ALCANÇADOS. ISSO PODE SER GARANTIDO ATRAVÉS DOS MAIS DIFERENTES TIPOS DE TESTES, INCLUSIVE COM A INTERAÇÃO DO PRINCIPAL ENVOLVIDO E INTERESSADO NO PROCESSO: O USUÁRIO.

AUTOR(A): PROF. ANTONIO ANDRADE DOS SANTOS

Testes

Nosso principal foco nesta aula é:



Legenda: EXECUÇÃO DOS TESTES.

Em aulas anteriores discutimos sobre as fases de um projeto. Similar ao que já foi exposto, Leme filho (2003) em seu livro sobre “metodologia de desenvolvimento de sistemas”, apresenta as fases de um projeto como sendo: Análise; Projeto; Construção; Homologação; Implantação e Manutenção.

Cada uma das fases de um projeto tem sua importância. No entanto, a fase chamada de homologação (onde se dedica aos testes), precisa de especial atenção. Pois a geração de erros ou falhas durante a execução de um software pode colocá-lo em descrédito e até levar um projeto ao fracasso.

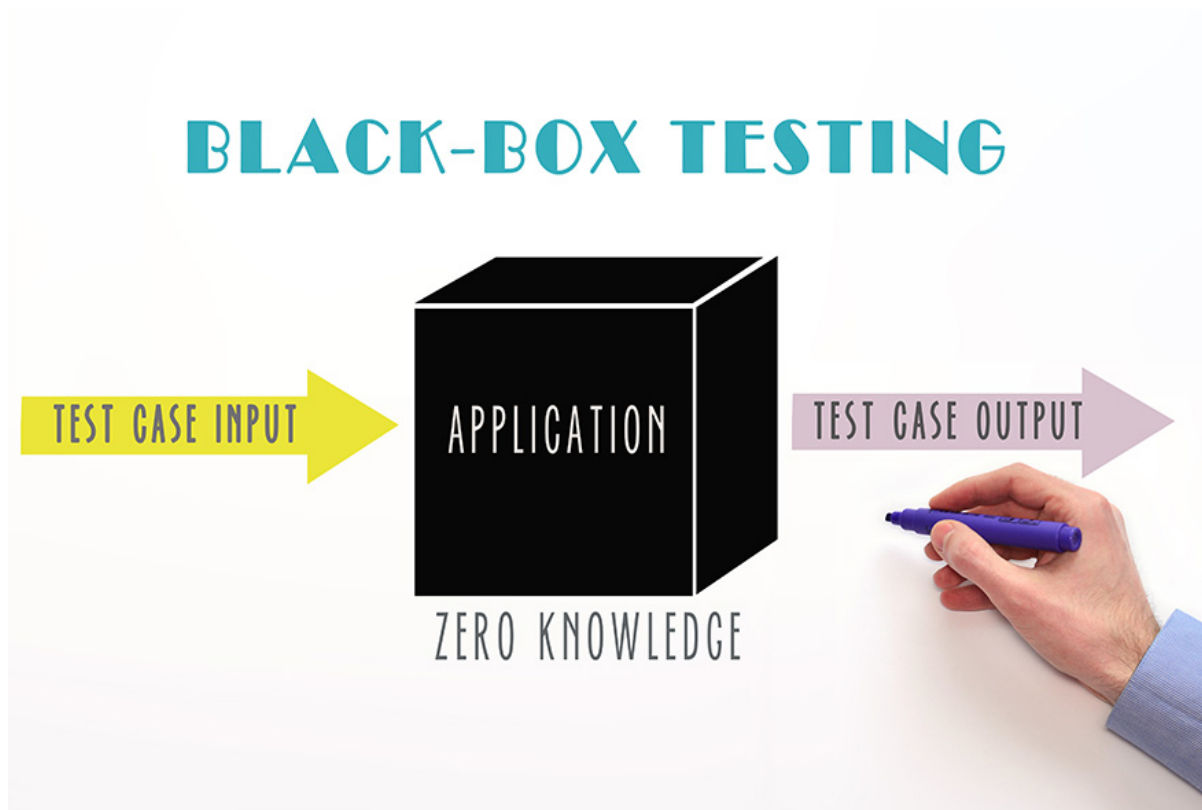
Dentro do conceito de engenharia de software, existem duas técnicas para se efetuar testes em sistemas denominados de: teste de caixa preta e teste de caixa branca.

Vejamos um pouco sobre os conceitos desses tipos de testes.

Teste caixa preta

O teste caixa preta deve ser realizado a partir do conhecimento das funções específicas que o produto projetado deve produzir. (LEME FILHO, 2003, p.84).

Veja a imagem ilustrativa, que demonstra o teste das entradas, valida as saídas, mas sem a necessidade de saber como o software em si funciona:



Legenda: TESTE CAIXA-PRETA

Segundo Leme Filho (2003, p.85) neste teste todas as entradas devem ser testadas e todas as saídas devem acontecer de acordo com o resultado previamente projetado e esperado. Além de que, deve ser validada a integridade dos dados.

Sendo mais específico, neste tipo de teste deve-se atentar à:

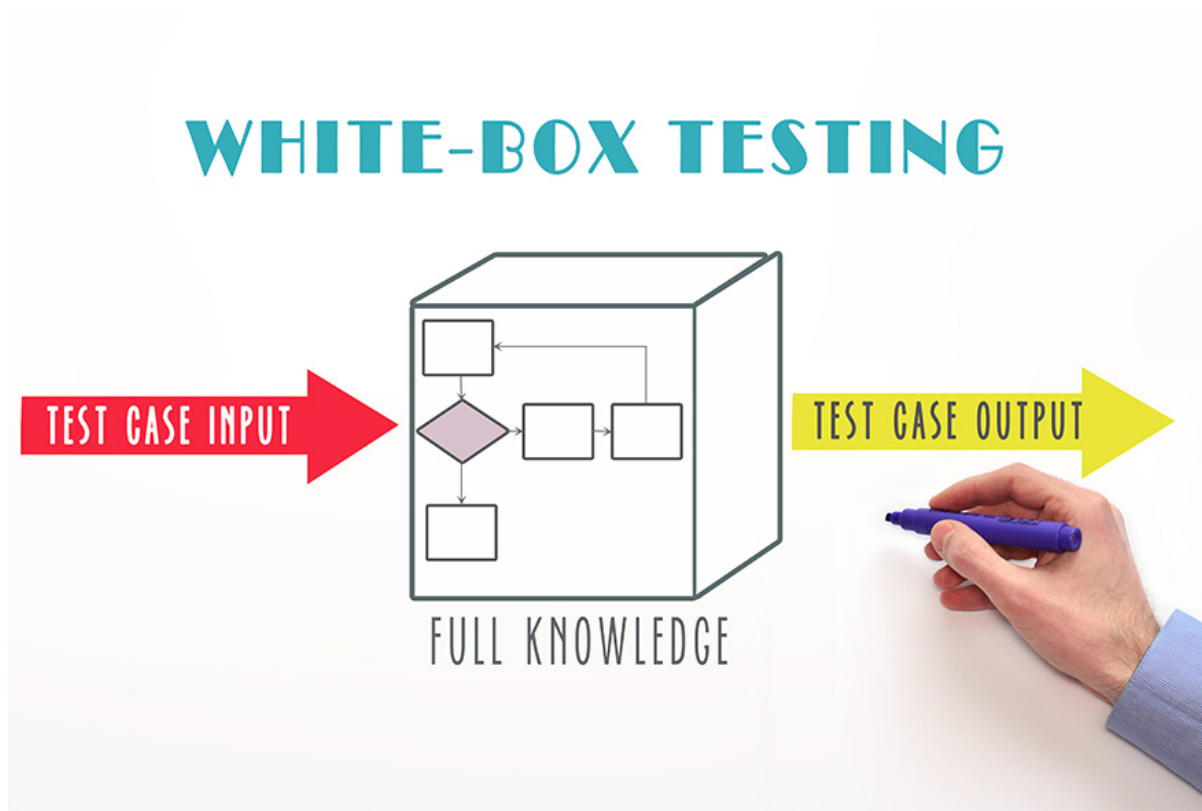
- Quantidade de caracteres dos campos, que podem causar estouro de campo;
- Caracteres incompatíveis com o tipo de campo;
- Caracteres especiais que podem causar problemas no banco de dados;
- Datas inválidas;
- Solicitação do usuário deve vir com o resultado esperado;
- Padrão de nomenclatura de comandos, botões, ícones, etc.
- Acesso rápido ou teclas de atalho às funcionalidades;
- Consultas ou comandos que não geram nenhum resultado.

Estes itens devem fazer parte de um modelo padrão de testes para que nunca se deixe de fazê-los, ocasionando erro para o usuário que dê a impressão de que não houve testes.

Mas, também há o teste realizado com o conhecimento das funcionalidades internas do produto. O chamado teste de caixa branca.

Teste caixa branca

O que muda neste tipo de teste é que o profissional de testes conhece as funcionalidades desenvolvidas no produto. Conforme ilustra a figura abaixo:



Legenda: TESTE CAIXA-BRANCA

Um exemplo citado por Leme Filho (2003, p87) é de que o cálculo de comissão de um vendedor leve em consideração os descontos concedidos, uma vez que isso foi definido pelo usuário.

Além dos tradicionais testes de caixa preta e caixa branca, o documento abaixo enumera mais dezesseis tipos diferentes de testes que podem ser realizados em uma fábrica de software.

Veja a lista dos tipos de testes:

Todos os testes são importantes e devem ser analisados para serem realizados com mais ou menos intensidade em cada projeto.

É importante destacar também a questão de que a usabilidade não seja prejudicada pela funcionalidade ou vice-versa.

Usabilidade x Funcionalidade

O documento acima mostra que o teste de usabilidade avalia as interações do homem com a máquina, a facilidade de navegação nas telas, a consistência da interface com o usuário, a estética, a facilidade do uso tornando o sistema intuitivo, enquanto que o teste de funcionalidade procura avaliar se todos os requisitos solicitados pelo cliente foram realizados satisfatoriamente.

Fique atento a cada um dos testes que podem ser realizados. Além do teste de caixa preta e caixa branca, os testes de funcionalidade, usabilidade, segurança, desempenho, stress, estrutura, concorrência, carga, configuração, instalação, aceitação, unitário, integrado, de sistema, componente e regressão.

Sim, são muitos os tipos de testes que podem e devem ser realizados em um produto. Por isso que há profissionais especializados e específicos para esta função no escopo de um projeto.


Vamos abordar logo abaixo, sobre Plano de Testes para os projetos. No entanto, vamos dar uma ligeira passada sobre casos de uso e como elaborá-lo, uma vez que estes artefatos estão diretamente ligados à compreensão da análise.

Embora uma disciplina específica para discutir o assunto, vamos apenas comentar sobre, para que você já vá se familiarizando com esse tipo de documentação.

Casos de uso (*User Case*)

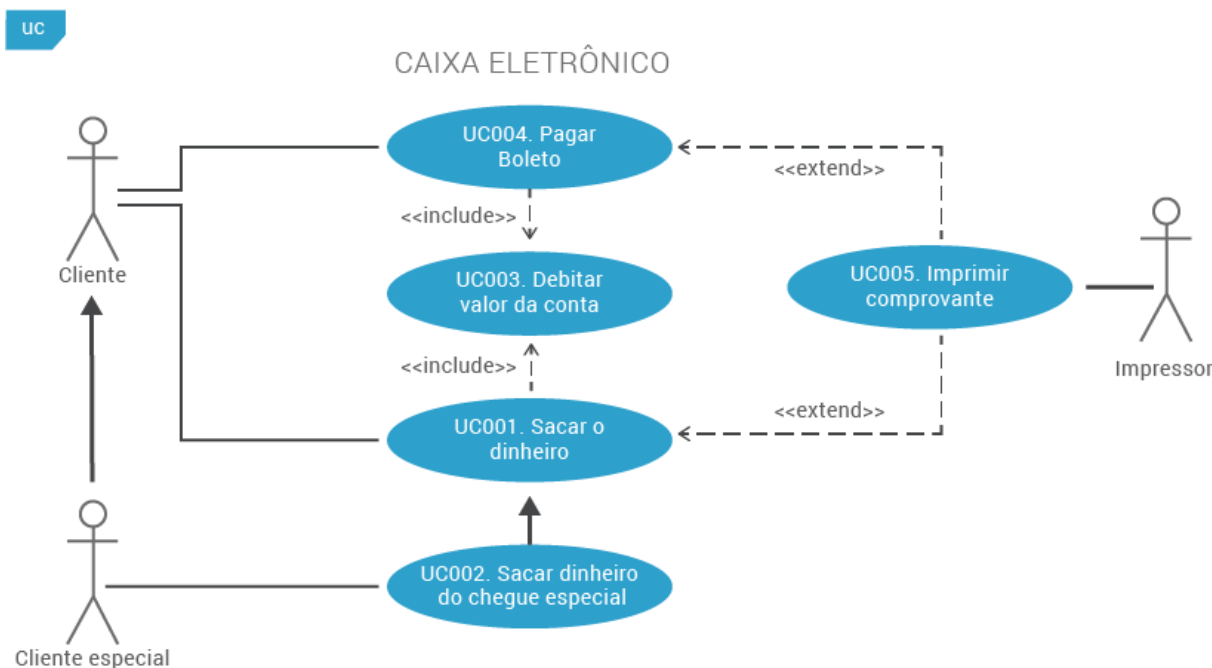
Um caso de uso deve descrever as funcionalidades e o comportamento de um sistema ou parte dele. (NOGUEIRA, 2006).

Veja a figura abaixo que descreve os símbolos ao se desenhar um caso de uso.

ELEMENTO	REPRESENTAÇÃO GRÁFICA	OBJETIVO
Ator		Define um conjunto coerente de papéis que os usuários do sistema podem desempenhar ao interagir com ele. Pode ser considerado tanto um indivíduo como um sistema externo.
Caso de Uso		Define uma sequência de ações realizadas por um sistema que produz um resultado de valor observável para determinado ator.
Relacionamento Simples		Linha que interliga os atores aos casos de uso que eles têm permissão de executar
Relacionamento Herança		Ideia de generalização. Pode ser aplicado tanto entre casos de uso como entre atores. Dessa maneira, é possível aproveitar características do caso de uso ou ator mais genérico.
Relacionamento Extensão		Representa que a ação do sistema que está sendo estendida pode ser executada pelo caso de uso que a estendeu. A seta é direcionada para o caso de uso que poderá executar o caso de uso opcional.
Relacionamento Inclusão		Representa que a ação do sistema será executada pelo caso de uso que a inclui. A seta é direcionada para o caso de uso que será incluído

Legenda: ELEMENTOS DE UM DIAGRAMA DE CASOS DE USO

Na próxima imagem, um exemplo de diagrama de caso de uso onde são expostas as funcionalidades de parte de um sistema: “caixa eletrônico”, onde se pode observar os atores, as ações e a relação entre eles.



Legenda: EXEMPLO DE UM DIAGRAMA DE CASOS DE USO

O trabalho com diagramas nos remete ao UML (*Unified Modeling Language*) ou Linguagem de Modelagem Unificada.

UML é uma linguagem que define uma série de artefatos que nos ajuda na tarefa de modelar e documentar os sistemas orientados a objetos.

(RIBEIRO, 2018)

VEJA MAIS:

UML: <http://www.uml.org/> (<http://www.uml.org/>)

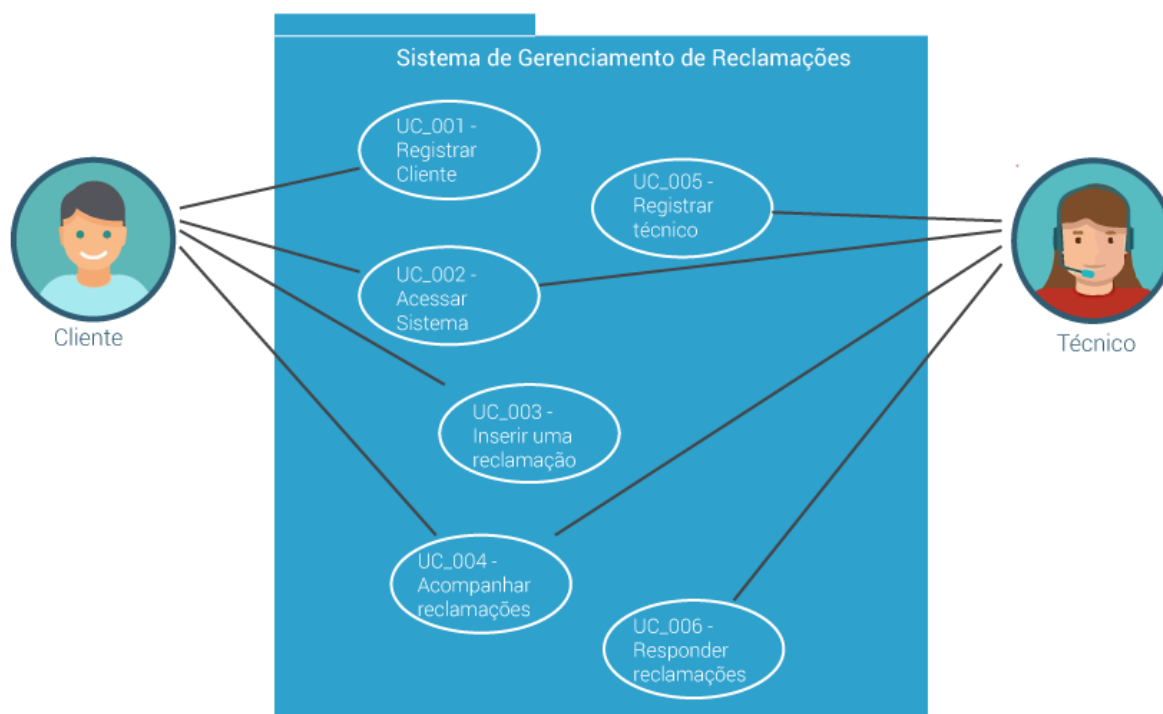
Diagramas de caso de uso UML: diretrizes: <https://msdn.microsoft.com/pt-br/library/dd409432.aspx?f=255&MSPPErrror=-2147217396> (<https://msdn.microsoft.com/pt-br/library/dd409432.aspx?f=255&MSPPErrror=-2147217396>)

Ferramenta para desenhar diagramas - The UML Modeller: <https://umbrello.kde.org> (<https://umbrello.kde.org>)

Manual do Umbrello UML Modeller: https://docs.kde.org/trunk4/pt_BR/kdesdk/umbrello/ (https://docs.kde.org/trunk4/pt_BR/kdesdk/umbrello/)

Mesmo sem saber construir um diagrama de caso de uso, observando o diagrama abaixo, se torna fácil compreender as operações de um sistema ou trecho dele.

Concorda? Veja:



Legenda: DIAGRAMA DE CASOS DE USO DO SISTEMA DE GERENCIAMENTO DE RECLAMAÇÕES

Desenhar um diagrama pode ser feito até se rabiscando um pedaço de papel. No entanto há ferramentas que fazem isso para os analistas documentarem com o projeto todas as especificações de um sistema. Também é importante, descrever de forma narrativa e organizada um caso de uso. Ou seja, é preciso que seja feito um levantamento de requisitos e documentado em um artefato, o que chamamos de especificação de casos de uso.

Clique em cada item do infográfico abaixo para ver como se elabora uma especificação de caso de uso:

Mas enfim, vamos ao tão esperado Plano de Testes que é o foco principal desta aula.

Plano de Testes

Um Plano de Testes é um documento elaborado para registrar as especificações dos testes a serem realizados em um projeto.

Neste documento são registradas as seguintes informações:

- Identificação da Empresa / Órgão / Setor/ Programa;
- Versões revisadas anteriormente (se não for a primeira);
- Descrição do nome do teste;
- Quais Aspectos a serem testados;
- Aspectos a serem testados;

- Casos de testes;
- Critérios de completeza e sucesso;
- Procedimentos de testes;
- Entradas e saídas esperadas;

Faça o download do arquivo abaixo para utilizar como modelo de Plano de Testes:

Porem, o Plano de Testes será composto de um ou mais casos de testes. Vamos falar um pouco sobre este outro modelo.

Casos de Testes



Caso de Teste: descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado.

(DIAS NETO, 2018).

No modelo de caso de testes deve conter os seguintes itens:

- Identificação da empresa;
- Identificação do projeto;
- Versão e data;
- Histórico de revisões anteriores (se não for a primeira;
- Índice;
 - Propósito;
 - Público alvo;
 - Escopo;
 - Definições, Acrônimos e Abreviações.
 - Referências
- Atualização dos testes;
- Casos de teste;
 - Título do caso de teste;
 - Dados;
 - Procedimentos;

Faça o download do modelo de casos de testes abaixo:

Exemplo de Plano de Testes:

Baixe o exemplo de Plano de Testes abaixo para ter uma ideia de como elaborar um para seu projeto.

VEJA MAIS:

A CELEPAR Informática do Paraná elaborou em 2009 um Guia Técnicas de Teste - Metodologia Celepar. Veja: <https://goo.gl/ZQrCdG> (<https://goo.gl/ZQrCdG>)

Tipos de teste de software: <http://testesdesoftware.com/tipos-de-teste-de-software/> (<http://testesdesoftware.com/tipos-de-teste-de-software/>)

Métricas de teste de software

Esta é uma questão delicada, uma vez que segundo a Quality (2018), uma métrica é um indicador de qualidade que servem para responder questões relacionadas ao software em desenvolvimento.

Quem efetua testes pode ser confundido por alguns desenvolvedores como àquele que quer atrapalhar ou colocar em cheque seu trabalho. Pode gerar até intrigas. Mas no objetivo não é esse. Muito pelo contrário, os desenvolvedores devem ver os profissionais de testes como parceiros, que vão ajudar o produto ser levado ao cliente final sem erros as vezes grosseiros.

Muito mais complicado se falar em métricas, pois parece colocar o trabalho de toda equipe dúvida. É preciso maturidade de cada integrante da equipe.

Segundo a Quality (2018), alguns questionamentos podem ser realizado e analisado para se obter uma visão da qualidade do software, tais como:

- Que componentes possuem defeitos e bloqueiam os testes?
- De que forma estão sendo priorizadas as correções de defeitos detectados?
- Em que ponto do software se encontra mais inconformidades?
- Qual a abrangência dos testes realizados?

Subramanyam (2015) fala sobre cinco pontos para avaliar a qualidade de softwares.

Segundo Subramanyam (2015) deve ser avaliado o alcance do software, em relação à complexidade de lidar com vários sistemas e idiomas interligados das aplicações modernas; a profundidade que determina a arquitetura e contextualização da aplicação; avaliação da explicitação do conhecimento através das melhores práticas da engenharia da produção; deve orientar sobre as possíveis melhorias com métricas adicionais; e que todas as métricas possam ser realizadas de forma automatizada.

No entanto, Subramanyam (2015) recomenda que a avaliação de métricas de software só devem ser realizadas, se a equipe estiver certa de que a avaliação está sendo realizada da forma correta. Pois uma avaliação errada é pior do que não ter.

Proposta de atividade em sala

- Conhecimento

- Abordar sobre os diversos tipos de testes e suas importâncias.
 - Abordar sobre casos de uso, Plano de Testes, casos de testes e métricas de teste de software.
- Habilidade
 - Buscar os modelos de casos de testes e Plano de Testes.
- Atitude
 - Criar um documento de caso de testes e um documento de Plano de Testes.

O que foi visto neste tópico

Nesta aula foi abordado sobre diversos tipos de testes que podem e devem ser realizados nos softwares para evitar inconformidades e garantir o sucesso do projeto.

Foi abordado brevemente sobre Casos de uso (*User Case*) e UML.

Foi explanado sobre Plano de Testes e Casos de Testes, também sobre métricas de teste de software.

Teste seu aprendizado

ATIVIDADE FINAL

Qual das alternativas abaixo melhor representa a diferença entre teste de caixa preta e teste de caixa branca?

- A. O teste de caixa preta não requer conhecimento da aplicação e o teste de caixa branca envolve conhecimento das funcionalidades do produto.
- B. O teste de caixa branca não requer conhecimento da aplicação e o teste de caixa preta envolve conhecimento das funcionalidades do produto.
- C. O teste de caixa preta requer conhecimento da aplicação e o teste de caixa branca não envolve conhecimento das funcionalidades do produto.
- D. Os testes de caixa branca e caixa preta não tem diferença entre elas, tanto faz utilizar uma ou outra.

Qual das alternativas abaixo melhor identifica o que é um Caso de Testes?

- A. Descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado.
- B. É um documento elaborado para registrar as especificações dos testes a serem realizados em um projeto.

C. Caso de Testes é uma metodologia de desenvolvimento de sistemas.

D. Caso de Testes é uma ferramenta de análise.

Qual das alternativas abaixo melhor define um Plano de Testes?

A. É um documento elaborado para registrar as especificações dos testes a serem realizados em um projeto.

B. Descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado.

C. Plano de Testes é uma metodologia de desenvolvimento de sistemas.

D. Plano de Testes é uma ferramenta de análise.

REFERÊNCIA

DIAS NETO, Arilo Claudio. Casos de Teste: Aprimore seus casos e procedimentos de teste. Disponível em: <<https://www.devmedia.com.br/casos-de-teste-aprimore-seus-casos-e-procedimentos-de-teste/30526>> (<https://www.devmedia.com.br/casos-de-teste-aprimore-seus-casos-e-procedimentos-de-teste/30526>)>. Acesso em 05 jan 2018.

LEME FILHO, Trajano. Metodologia de Desenvolvimento de Sistemas. Axcel Books do Brasil Editora. São Paulo. 2003.

NOGUEIRA, Admilson. Casos de Uso (Cenários). Postado em 16/01/2006. Disponível em: <<https://imasters.com.br/artigo/3811/uml/casos-de-uso-cenarios?trace=1519021197&source=single>> (<https://imasters.com.br/artigo/3811/uml/casos-de-uso-cenarios?trace=1519021197&source=single>)>. Acesso em 05 jan 2018.

QUALITY. As Métricas e o Teste. Disponível em: <<http://qualidadedesoftware.com.br/2011/08/17/as-metricas-e-o-teste/2011/>> (<http://qualidadedesoftware.com.br/2011/08/17/as-metricas-e-o-teste/2011/>)>. Acesso em 05 jan 2018.

RIBEIRO, Leandro. O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML. Disponível em <<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>> (<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>)>. Acesso em 05 jan 2018.

SUBRAMANYAM, Jitendra. Cinco métricas para avaliar a qualidade de seus softwares. Publicado em 09 de Setembro de 2015. Disponível em: <<http://computerworld.com.br/cinco-metricas-para-avaliar-qualidade-de-seus-softwares>> (<http://computerworld.com.br/cinco-metricas-para-avaliar-qualidade-de-seus-softwares>)>. Acesso em 05 jan 2018.

