

Engenharia de  
Software



Anhanguera

**AVALIE**  
SUA PROFISSÃO

QUANDO APARECER EM SEU  
PORTAL UMA AVALIAÇÃO SOBRE  
SEU CURSO, RESPONDA:



NOTAS

**9 ou 10**

SIGNIFICA QUE VOCÊ INDICA

NOTAS

**7 ou 8**

SIGNIFICA QUE VOCÊ NÃO INDICA



# Anhanguera



Anhanguera



## **IMPORTÂNCIA DA QUALIDADE DE SOFTWARE**

Os conceitos de qualidade de software estão ligados aos métodos, às ferramentas, aos processos e às medições, os quais guiam a equipe de desenvolvimento de forma mais assertiva e garantem que a entrega seja feita dentro dos padrões de qualidade estabelecidos entre as partes.

Essa área do conhecimento da engenharia de software se preocupa em utilizar métodos, processos e normas nas atividades de desenvolvimento de software, a fim de se agregar qualidade nos processos e produtos finais. Para isso, é necessário que se tenha conhecimentos acerca de qualidade de software, qualidade do produto e qualidade dos processos. Como você pode ter percebido, existem diversas competências necessárias para se garantir a qualidade de um desenvolvimento.



# Anhanguera

Os aspectos ligados à qualidade estão na grande maioria de produtos e serviços que você utiliza em sua casa. Mas como isso se relaciona à qualidade de software? Analise a seguinte situação: uma pessoa fez uma filmagem no smartphone dela, mas deseja editar o vídeo, de forma que possa cortar algumas partes e remover o áudio. Após baixar diversos aplicativos e testar muitos deles, o vídeo foi danificado e não é mais possível fazer a sua restauração. Você não acha isso muito grave?

Pois bem, talvez esse vídeo tenha sido algo casual, mas e se fosse uma filmagem de festa, batizado, casamento, formatura, etc.? A perda para o usuário talvez fosse irreversível. Esse é apenas um exemplo dos inúmeros erros que podem ser encontrados em softwares para diversos dispositivos. O que um profissional de tecnologia da área de desenvolvimento pode fazer para que tais situações sejam evitadas? A resposta a essa pergunta será vista nesta seção de estudos.



# Anhanguera

## CONCEITO-CHAVE

Caro aluno, certamente alguma vez na vida você já deve ter adquirido um produto ou serviço, o qual, no momento da compra, parecia perfeito, mas, ao utilizá-lo, as coisas acabaram saindo bem diferente do prometido. Como consumidor, isso traz muita frustração. E isso não é uma exclusividade da Tecnologia da Informação. Corriqueiramente, ocorre também na telefonia, na alimentação, no e-commerce, na entrega de encomendas, de roupas, etc.



# Anhanguera

## QUALIDADE DE SOFTWARE

No que se diz respeito à desenvolvimento de softwares, a qualidade visabasicamente atender as necessidades e expectativas do cliente, a fim de cumprir os requisitos acordados no início do projeto de desenvolvimento de software. O tema é bem mais abrangente do que essa simples definição e é isso que torna os assuntos relacionados à qualidade de software de extrema importância para os profissionais de T.I.



Segundo Zanin et al. (2018), a qualidade de software está ligada aos aspectos de conformidade com os requisitos funcionais e não funcionais encontrados nos desenvolvimentos de softwares. Para compreender melhor, observe a definição dos requisitos:

**Requisitos funcionais:** especificam uma função que o sistema ou que determinado componente do sistema deve realizar. Essa descrição é feita do ponto de vista do usuário e pode ser determinante para o comportamento do sistema, de modo que precisa ficar bem claro o que deve ocorrer com uma entrada no software e qual a saída gerada. Para melhor exemplificar esse conceito, imagine que o seu cliente descreva que um software deve, após o usuário incluir os produtos no carrinho de compras, apresentar a opção de pagamento em crédito, débito ou boleto bancário.

A nível de desenvolvimento, esse funcionamento é de extrema importância, pois determina quais são as funcionalidades que o software deve apresentar. Por exemplo: inserir, alterar, excluir, listar produtos, etc.



# Anhanguera

**Requisitos não funcionais:** esses requisitos podem estar relacionados a necessidades que devem ser atendidas, porém não utilizam funcionalidades. Esse conceito está diretamente ligado à qualidade de software, ou seja, tratadas premissas técnicas que o software deve desempenhar. Um exemplo é a solicitação de um cliente para que determinada aplicação web seja responsiva, isto é, que permita ao usuário utilizá-la em um desktop e em um smartphone.

Esses requisitos não funcionais podem ter um detalhamento técnico mais específico, como: o sistema deve se comunicar obrigatoriamente com o SGBD MySQL; a solução desenvolvida tem que ser voltada para o sistema operacional Linux; a consulta de novos clientes deve permitir que seja efetuada off-line, entre diversos outros requisitos não funcionais, os quais podem ocorrer nas atividades de desenvolvimento de software.





# Anhanguera





Compreender os requisitos permite entender de que forma os aspectos relacionados à qualidade de software estão intimamente ligados com os processos de desenvolvimento. Para Pressman (2006), o ciclo de desenvolvimento de um software é dividido em seis partes, conforme pode ser observado na Figura 2.1





# Anhanguera

Ainda segundo o autor, o termo qualidade de software pode ser definido como o atendimento das conformidades funcionais com um desempenho esperado pelo patrocinador. Porém, para Sommerville (2011), a qualidade nas atividades de desenvolvimento de software deve ser mais abrangente, devendo ser gerenciada e compreendida em três níveis:

**1.Organizacional:** a preocupação é de um nível mais amplo, no qual o objetivo é o estabelecimento de padrões de trabalho de desenvolvimento de software. Esses frameworks devem agrupar as melhores práticas para que os erros e falhas sejam minimizados.



# Anhanguera

**2. Projeto:** envolve o desenvolvimento com base em padrões determinados por gestores de projetos. Isso pode variar conforme o projeto, a política da empresa, a utilização de frameworks, entre outras peculiaridades.

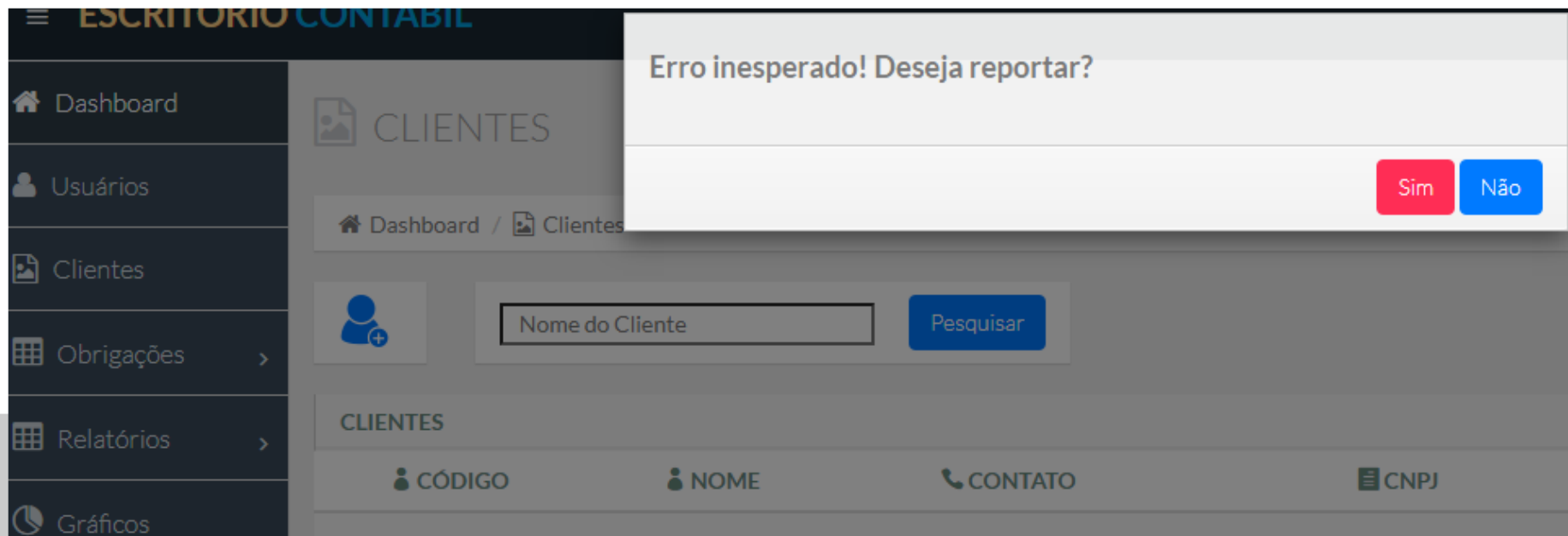
**3. Planejamento:** deve haver um plano de qualidade, ou seja, parte da equipe deve ficar responsável pela verificação dos requisitos de qualidade acordados entre as partes. Tanto os processos quanto os produtos desenvolvidos devem ser revisitados a fim de se evitar que algo passe despercebido.



# Anhanguera

## FALHAS DE SOFTWARE

Segundo Zanin et al. (2018), a falha de software pode ser um comportamento inesperado do sistema e pode ser ocasionado por um ou mais erros. Para melhor compreensão de como a falha de software pode afetar um sistema, observe a Figura 2.2.





# Anhanguera

Nesse exemplo, acima do aceite, o sistema retorna uma mensagem de erro, impedindo, assim, que um novo cliente seja cadastrado, pois a falta da confirmação do aceite que impede a finalização do processo. Isso, a nível de usuário, é bem impactante. Imagine que esse sistema seja de uma loja de departamentos, a qual depende desse sistema para gerar as vendas e as notas fiscais. Certamente o prejuízo seria muito grande e preocupante.

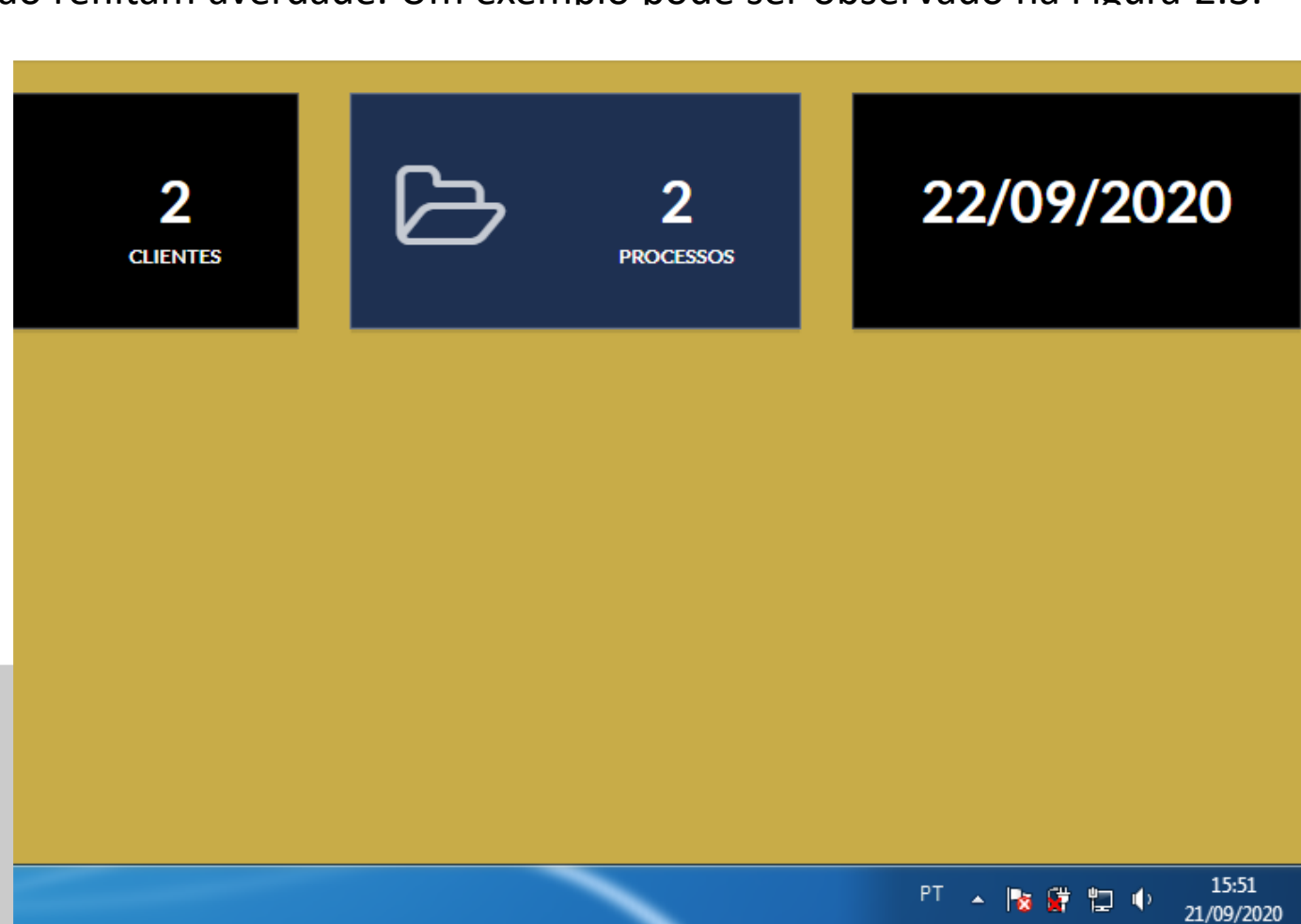
Vale aqui ressaltar que o desenvolvedor é o responsável pela implementação das rotinas que cuidam das tratativas para falhas e erros. Elas podem ser exibidas por meio de uma mensagem ao usuário, podem levar a uma página nova com alguma imagem que remeta a falhas e erros ou podem, ainda, utilizar um report para enviar os problemas a algum repositório, com o intuito de que os desenvolvedores ajustem o sistema.



# Anhanguera

## ERROS DE SOFTWARE

Segundo Zanin et al. (2018), grande parte dos erros de software estão relacionados a execuções incorretas, o que faz com que os resultados gerados não reflitam a verdade. Um exemplo pode ser observado na Figura 2.3.





# Anhanguera

Observe, nesse exemplo, a data do sistema e a data marcada no relógio. O erro de data no sistema pode ter um impacto muito grande na integridade dos seus dados, fazendo com que os lançamentos não sejam confiáveis quanto a sua data. Isso para setores de venda, logísticos, contábeis e fiscais pode ser desastroso.





# Anhanguera

## **DEFEITO DE SOFTWARE**

Segundo Zanin et al. (2018), o defeito de software se refere a uma implementação incorreta, que ocasiona um erro, uma interrupção de serviço ou, ainda, um mal funcionamento. Um exemplo pode ser um sistema de cadastro que, após o preenchimento de todos os campos obrigatórios, executa o processo de inserção de dados, não retorna erros, porém não faz a inserção das informações no banco de dados.

Quanto às atividades de desenvolvimento, os defeitos são mais difíceis de serem encontrados, pois, em alguns casos, erros de lógica, de falha de escrita do código ou de qualquer outra referência não retornam para auxiliar o desenvolvedor a encontrar o problema.



# Anhanguera

## **BUGS**

Esse termo se popularizou entre os jovens para fazer referência a comportamentos inesperados de softwares. Segundo Zanin et al.(2018), um bug de sistema diz respeito a erros e falhas inesperados, que normalmente são de maiores complexidades e demandam mais tempo e conhecimento técnico para que sejam encontrados e solucionados.



# Anhanguera

## **GARANTIA DA QUALIDADE**

Sommerville (2011) mostra que a garantia da qualidade é conhecida como Software Quality Assurance (SQA). A sua abrangência se estende por todo o ciclo de vida do projeto de desenvolvimento de software e deve:

Possuir ferramentas e/ou métodos que permitam a análise dos desenvolvimentos e dos testes.

Efetuar revisões técnicas nos componentes e na funcionalidade, devendo ser feitas em cada uma das fases.

Controlar a documentação por meio de versionamento.

Atribuir métodos para se garantir padrões de desenvolvimento e das boas práticas, as quais atendem as necessidades das equipes de desenvolvimento.

Obter mecanismos de aferição.



# Anhanguera

Para Pressman (2006), a garantia da qualidade diz respeito aos procedimentos, métodos e ferramentas utilizados por profissionais de Tecnologia da Informação para se garantir padrões acordados entre as partes durante todo o ciclo de vida do desenvolvimento de um software. É importante lembrar que os padrões de qualidade podem variar conforme o projeto e, por isso, a garantia da qualidade deve ser guiada pelo que foi acordado entre as partes.

Com o intuito de que você possa compreender melhor como a garantia da qualidade é, de fato, operacionalizada em atividades de desenvolvimento em um meio profissional, vamos analisar um case. Imagine que foi solicitado por um escritório de advocacia um programa para gerenciar os processos advocatícios. Em primeiro lugar, foi dito que tanto os advogados quanto os clientes deverão fazer login no sistema por um único local e de forma transparente. Tempos depois foi apresentada a tela de login ao cliente, conforme pode ser observado na Figura 2.4.



# Anhanguera

Repare que o acordo entre as partes era o login transparente. Dessa forma, o menu dropdown para escolher o tipo do usuário não deveria ter sido utilizado natela. A garantia da qualidade deve utilizar as ferramentas para detectar o problema, pois esse ajuste tem um impacto tanto no desenvolvimento do front-end quanto no do back-end, porque a lógica para se efetuar um login no sistema muda completamente.

The screenshot shows a login interface with a dark blue background. At the top is a white padlock icon. Below it, the word "LOGIN" is written in white. There is a dropdown menu labeled "Selecione o Acesso" with a downward arrow. The dropdown is open, showing three options: "Selecione o Acesso", "Advogado" (highlighted in blue), and "Cliente". Below the dropdown is a password field with a key icon and the label "Senha". At the bottom is a yellow button labeled "ENTRAR". In the bottom left corner is a link "Cadastro" and in the bottom right corner is a link "Esqueceu a sua senha?" in red text.



Segundo Zanin et al. (2018), as diversas engenharias utilizam o processo de medição, para verificar se determinado desenvolvimento está em conformidade e segurança para ser utilizado. Porém, diferentemente das demais áreas de conhecimento, a engenharia de software não utiliza leis quantitativas ou medidas absolutas, mas um conjunto de medidas que dão um feedback quanto aos aspectos qualitativos do software.

Boehn, Brown e Lipow (1977) determinam que, para efetuar a medição da qualidade de um software, deve-se determinar quais são as funcionalidades que devem/podem ser medidas e de que forma precisam ser medidas. A fim de orientar os desenvolvedores no alinhamento das métricas, os autores sugerem o esquema demonstrado na Figura 2.5.



# Anhanguera

Em sua estrutura mais alta, o software deve possuir elementos como usabilidade e manutenibilidade. E, em sua estrutura média, a portabilidade, a confiabilidade, a eficiência, a engenharia humana, a facilidade de teste, a facilidade de entendimento e a facilidade de modificação. Dessa estrutura média, derivam as primitivas (que dão as características dos elementos de nível médio). As primitivas podem ser verificadas por meio de checklists, ou seja, representam a atividade de desenvolvimento em si, que deve ser verificada quanto ao atendimento da garantia da qualidade.



Anhanguera





# Anhanguera

## SISTEMA NO NAVEGADOR

## SISTEMA NO MOBILE

Usuários Clientes Produtos

### USUÁRIOS

[Novo Usuário](#)

Nome	E-mail	Senha
Serginho	teste@teste.com	81dc9bdb52d04dc20036
Testando	teste@teste.com	81dc9bdb52d04dc20036

Usuários

Clientes

Produtos

### USUÁRIOS

[Novo Usuário](#)

Nome	E-mail	Senha
Serginho	teste@teste.com	81d
Testando	teste@teste.com	81d



# Anhanguera

Nesse exemplo, o sistema de cadastro está com a funcionalidade de adicionar um novo usuário e exibir os cadastrados no sistema. Do lado esquerdo é apresentada a exibição desse sistema em navegador de desktop, já do lado direito está a apresentação em um smartphone. Com isso, vamos utilizar a árvore de qualidade de software apresentada na Figura 2.5 para analisar esse cenário.

No primeiro caminho: Utilidade geral → Portabilidade → Independente do dispositivo, já se observa que o software apresenta um problema de responsividade, o que demonstra que o sistema não é portátil.

Outro exemplo em que a qualidade é atendida pode ser verificado no caminho: Utilidade Geral → Usabilidade → Confiabilidade → Precisão. Isso ocorre, pois, quando uma transação de inserção de dados é feita, a operação se completa corretamente.



# Anhanguera

Segundo Sommerville (2011), as vantagens encontradas ao se operacionalizarmos processos de garantia da qualidade dependem de um esforço coletivo, que proporciona a economia de recursos durante todo o ciclo de vida do projeto de desenvolvimento de software. Ainda que de maneira bem específica, os processos de qualidade podem proporcionar algumas vantagens, listadas a seguir:

**Padronização:** boas práticas passam a ser adotadas pelas equipes de desenvolvimento.

**Aumento de produtividade:** ao se minimizar os retrabalhos por meio da verificação da qualidade, o tempo de produtividade é otimizado.

**Satisfação do cliente:** com as ferramentas de verificação da qualidade, a equipe de desenvolvimento realiza as atividades de forma mais assertiva.

**Economia de recursos:** redução nos custos operacionais como um todo, sendo observados pontos como: tempo, custo com colaborador, custos gerais de operação.

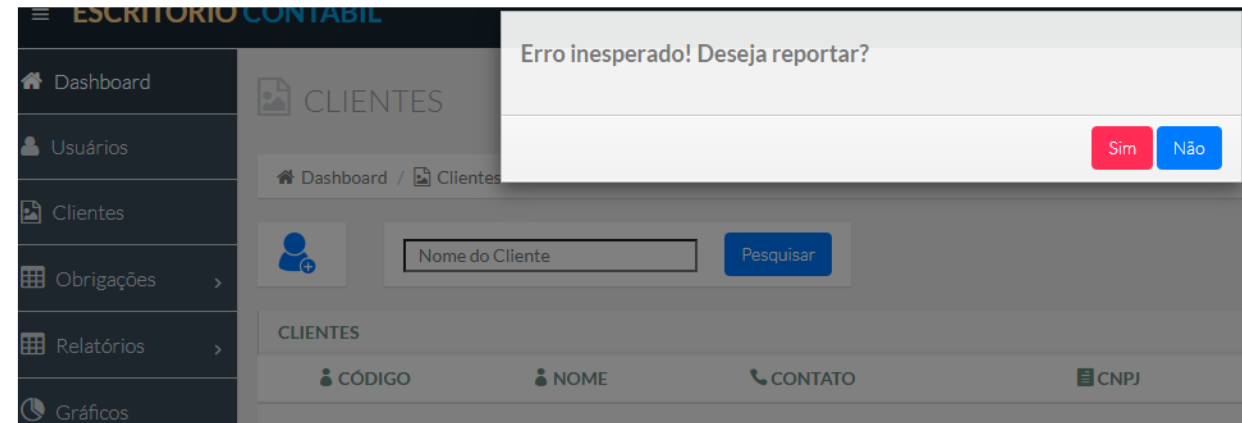
**Retrabalho:** evita que grandes correções e ajustes sejam necessários, fazendo com que muitas vezes as atividades com dependências funcionais necessitem ser paralisadas.

O desenvolvimento de softwares é uma atividade extremamente complexa. Poresse motivo, ter ferramentas de qualidade é um grande diferencial para se garantir que o produto final atenda às necessidades e às expectativas do cliente. Porém, em determinadas situações, algumas funcionalidades não executam as operações como o desejado. Observe a figura a seguir:

Nesse exemplo, o cliente seleciona a opção para excluir determinado registro, confirma a exclusão no modal, porém o registro não é excluído do banco de dados e, conseqüentemente, continua sendo apresentado na lista.

Com base na situação apresentada, assinale a alternativa com o tipo do problema.

- a. Falha de software.
- b. Erro de software.
- c. Defeito de software.
- d. Bug.
- e. Anomalia.





# Anhanguera

Toda e qualquer técnica, método, ferramenta ou tecnologia que se utilize em determinada atividade de desenvolvimento de software tem a intenção de fazer com que algumas vantagens sejam alcançadas a fim de se agregar qualidade.

Ao utilizar as ferramentas de garantia da qualidade, a equipe de desenvolvimento visa alcançar algumas vantagens. A partir do apresentado, analise as asserções a seguir e a relação proposta entre elas.

I. Padronização das tarefas, que passam a se tornar boas práticas para a equipe de desenvolvimento.

**POIS**

II. As atividades tendem a ser repetitivas e repeti-las será uma grande vantagem aos desenvolvedores, à equipe, à empresa e ao cliente.

A seguir, assinale a alternativa correta:

- a. As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- b. A asserção I é uma proposição verdadeira e a asserção II é uma proposição falsa.
- c. As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- d. A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e. As asserções I e II são proposições falsas.