

# Ambientes de testes

**Disciplina:** Qualidade de *Software*



## Conteúdos:

Ambientes de testes.

## Habilidade(s):

- Utilizar *softwares* de apoio no teste de sistemas;
- Compreender o relatório e a documentação de testes unitários de *software*;
- Construir e padronizar o relatório de teste conforme as exigências do mercado.

# Bloco 1

---

Entendendo os principais requisitos da documentação  
de teste.

# Quem é você?

Com base no seu humor atual, se você fosse um **tipo de documentação**, qual seria?

CPF



Simples e básico

Curriculum



Focado na profissão

Passaporte



Em busca de explorar

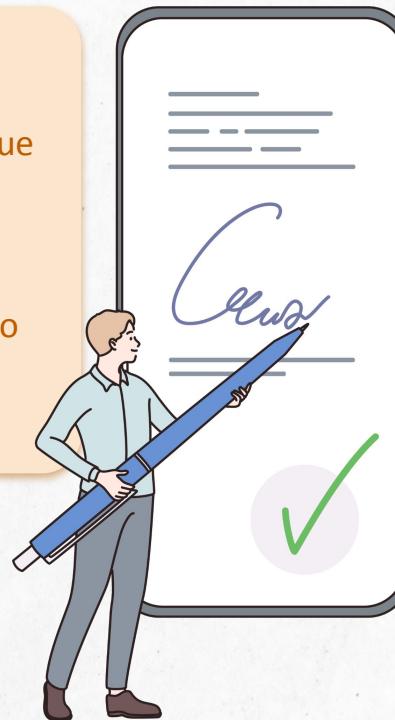
Testamento



Queria ser herdeiro

Diariamente, o uso de documentos é essencial por uma variedade de razões que envolvem organização, identificação, comunicação e legalidade, por exemplo.

No desenvolvimento de um *software*, isso não é diferente.



# Por que a documentação é necessária?

Documentar ajuda a registrar os casos de teste, os cenários e os resultados de forma clara e organizada.

Isso permite que a equipe compreenda facilmente o que foi **testado** e quais foram os **resultados obtidos**.



Fique ligado na norma IEEE 829!

Elá define uma série de documentos recomendados para um projeto de teste de *software*.

A seguir, conheceremos um pouco mais desse assunto.



# Informações sobre a documentação de teste

Do mesmo modo que uma série de documentos é necessária para que tenhamos cidadania, durante a realização dos testes de um *software* é preciso que o processo de testes garanta a sua documentação necessária.

Os documentos podem variar a depender do projeto, mas podem abranger:

Plano de teste

Plano de defeitos

Especificação de casos de teste

Plano de teste de carga

Especificação de procedimentos de teste

Plano de teste de segurança

Plano de configuração e entrega

Plano de teste de instalação

# Requisitos para a realização dos testes

Ao longo do processo de teste, também é necessário seguir alguns protocolos.

Por isso, é importante se atentar a algumas etapas exigidas, como a de **requisitos de administração** e a de **relatório de teste**. Observe o exemplo.

Você está trabalhando em um aplicativo de comércio eletrônico, mas se depara com um problema: as promoções não estão aparecendo como o esperado.

Sendo assim, você decide apostar na realização de um teste.

## Requisitos de administração

► Você deve relatar o problema, incluindo os detalhes relevantes, como as etapas que levaram ao problema, a data e a hora.

## Requisitos de relatório

► Após realizar uma bateria de testes, você precisa criar um relatório de teste que documente os resultados.

# Vamos fazer uma dinâmica?

Para cada gíria que você conseguir identificar corretamente, marque um ponto!

*Stalker*

*Cancelar*

*Pisa menos*

*Spoiler*

*Contatinho*

*10/10*

*Shippar*

*Low profile*

*Auge*

*Crush*

*Cringe*

*Trollar*



# Manual da internet

Hoje em dia, graças à linguagem da internet, é cada vez mais dinâmico se comunicar pelas redes sociais.

Existem tantas gírias e expressões que é necessário construir um **glossário** para entender todas elas!

Um glossário é incluído em livros, artigos, documentos técnicos, *sites* ou outros materiais para ajudar os leitores a entender o significado de termos que podem ser desconhecidos ou especializados.



# Glossário

Durante a realização dos testes, também é necessário construir um glossário para identificar os termos desconhecidos, porque ele ajuda a garantir uma comunicação clara e consistente entre todos os membros da equipe de teste, além de desenvolvedores e gerentes de projeto.

Alguns exemplos podem ser:

- usuário final;
- interface de usuário (UI);
- API (Interface de Programação de Aplicativos);
- *bug*;
- plataforma cruzada;
- *sprint*.



# Bloco 2

---

Plano de teste e as suas principais características.



## Com base no que você sabe

Responda: durante o processo de testes, quantos **planos de testes** podem ser realizados?

- a Apenas um.
- b Vários.



## Resposta correta

A quantidade de planos de teste varia de acordo com o tamanho do projeto.

a Apenas um.

b Vários.

# Plano de teste

Em geral, é um documento que detalha como um *software* será testado, garantindo que ele atenda aos requisitos e às funcionalidades especificadas.

Segundo a IEEE-829, o plano de teste deve conter:

## Introdução

Engloba o identificador do plano de teste, bem como o escopo, as referências, o nível na sequência, a classe de teste e a visão das condições.

## Detalhes para este nível do plano de teste

Concentra os tópicos como itens de teste e os seus identificadores, a matriz de rastreabilidade do teste e as funcionalidades a serem testadas.

## Gerência do testes

Contém aspectos como as tarefas do teste, as necessidades de ambientes, as responsabilidades e a integração entre as partes envolvidas.

## Simples e reproduzível

Apresenta os procedimentos de garantia de qualidade, as métricas, a cobertura do teste, o glossário e os procedimentos de alteração do documento e histórico.

# Identificador do teste

É utilizado para vincular o plano de teste a um projeto específico, permitindo que os membros da equipe saibam a qual projeto e a qual contexto o plano pertence.

É como nomear as pastas do seu computador!

Em uma empresa que esteja trabalhando no projeto de desenvolvimento de um aplicativo chamado “Software XYZ”, o identificador do plano de teste para esse projeto pode ser algo como “ST-XYZ-001”. Aqui, “ST” representa o plano de teste, “XYZ” identifica o projeto “Software XYZ” e “001” é um número sequencial para os planos de teste desse mesmo projeto.



## Nível de sequência do teste

Refere-se ao nível de teste em que o plano de teste está situado.

Isso ajuda a categorizar o plano de acordo com a finalidade dos testes a serem realizados.

Os diferentes níveis de teste podem incluir teste de unidade, teste de integração, teste de sistema, teste de aceitação, por exemplo.

Se um plano de teste está designado para o nível de teste de sistema, isso significa que, após a integração de diferentes componentes, ele abrange a verificação do funcionamento do sistema como um todo.



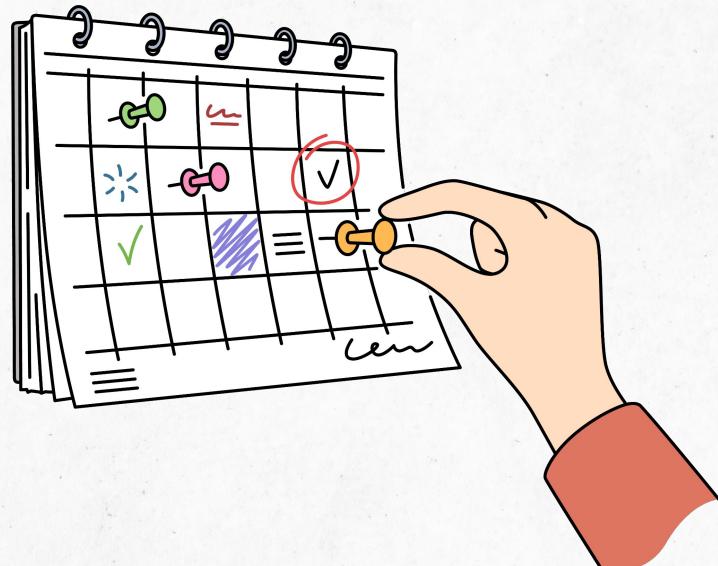
# Classe de teste e visão das condições de teste

Relaciona-se com a categorização das diferentes técnicas de teste que serão aplicadas a cada nível de teste.

Definem abordagens específicas para testar diferentes aspectos do *software*, como testes de limite, testes positivos e negativos e testes exploratórios.

Para a classe de testes de requisitos, a equipe de teste verifica se todas as especificações do sistema foram atendidas.

No contexto de um plano de teste de aceitação, a classe de teste pode incluir testes de requisitos e testes de usabilidade.



# Detalhes para este nível do plano de teste

O plano de teste especifica os itens que serão testados ou usados nos testes, podendo conter:

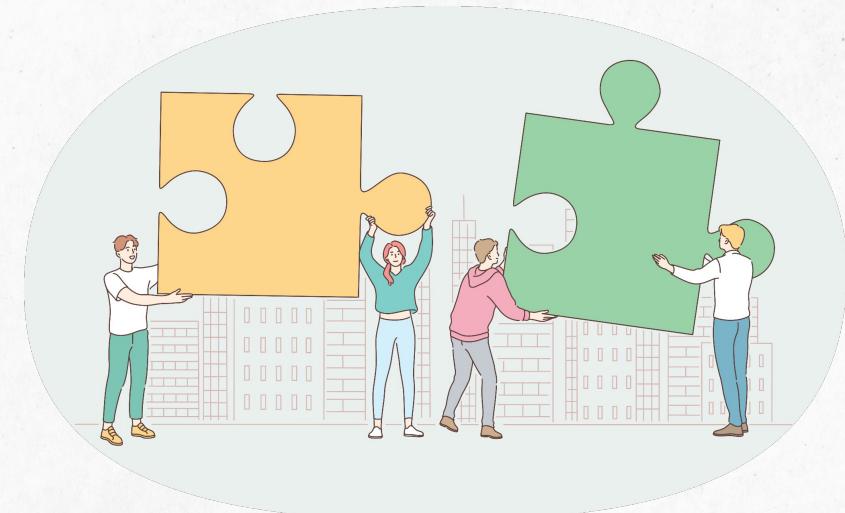
procedimentos

classes

módulos

requisitos

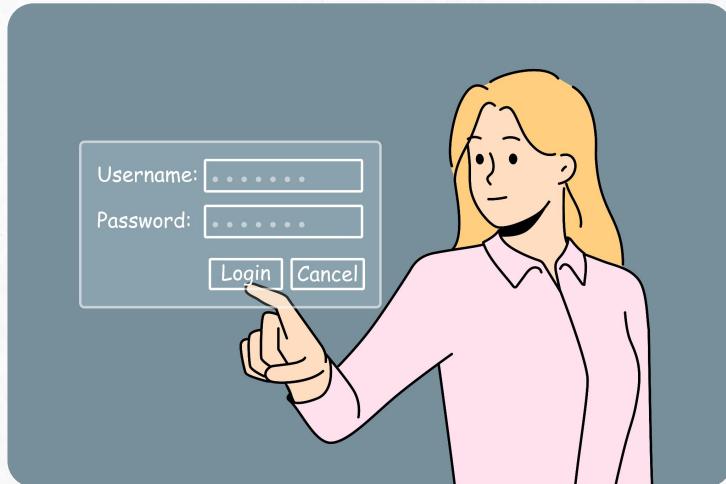
componentes



## Matriz de rastreabilidade

É uma ferramenta que estabelece a conexão entre os artefatos de teste, como os procedimentos de teste, casos de teste e *scripts* de teste, e os requisitos ou outros artefatos do projeto.

Suponha que um requisito específico do sistema seja o seguinte: “o *software* deve permitir que o usuário faça *login* usando um nome de usuário e senha”. A matriz de rastreabilidade deve mostrar quais casos de teste foram criados para verificar este requisito, como o teste de *login* com credenciais válidas e o teste de *login* com credenciais inválidas.



# Bloco 3

---

Entendendo um pouco mais sobre os requisitos do  
plano de teste.

# Vamos recapitular?

Analise os casos a seguir e os classifique como **identificador**, **nível de sequência**, **classe** e **visões das condições, detalhes para este nível ou matriz de rastreabilidade**.

A) O projeto AppX está avançando para a fase de teste de sistema. Nela, o foco é garantir que todas as funcionalidades trabalhem corretamente e em conjunto.



B) O plano de teste para o projeto AppY recebeu o identificador “PT-APPY-003”.



C) O aplicativo AppZ passou por revisões e definições de requisitos.



D) O plano de teste para o projeto AppW estipulou que os módulos A, B e C serão testados no nível de teste de aceitação.



E) Uma etapa está sendo criada para o projeto AppV, que conecta os casos de teste aos requisitos específicos do aplicativo.



# Vamos recapitular?

Analise os casos a seguir e os classifique como **identificador**, **nível de sequência**, **classe** e **visões das condições, detalhes para este nível ou matriz de rastreabilidade**.

A) O projeto AppX está avançando para a fase de teste de sistema. Nela, o foco é garantir que todas as funcionalidades trabalhem corretamente e em conjunto.



Nível de sequência

B) O plano de teste para o projeto AppY recebeu o identificador “PT-APPY-003”.



Identificador do plano de teste

C) O aplicativo AppZ passou por revisões e definições de requisitos.



Detalhes para este nível

D) O plano de teste para o projeto AppW estipulou que os módulos A, B e C serão testados no nível de teste de aceitação.



Classe e visões das condições

E) Uma etapa está sendo criada para o projeto AppV, que conecta os casos de teste aos requisitos específicos do aplicativo.



Matriz de rastreabilidade

## Funcionalidades a serem testadas

Aqui, você pode identificar as funcionalidades específicas do *software* que precisarão ser testadas.

As funcionalidades podem incluir a conversão de banco de dados, a geração de relatórios, a segurança, a recuperação e o desempenho.

Se tratando de um aplicativo de comércio eletrônico, algumas funcionalidades testadas podem ser a realização de pedidos, a segurança das informações do cliente, a capacidade de pesquisa de produtos e a exibição adequada dos detalhes dos produtos.



## Abordagem do teste

É quando é possível definir a abordagem que será adotada para a realização dos testes. Isso inclui determinar o nível de teste (unitário, integração, sistema ou aceitação) e a estratégia para abordá-lo.

Para um sistema de gerenciamento de inventário, a abordagem do teste pode incluir testes unitários para componentes individuais, testes de integração para verificar a comunicação entre diferentes módulos e testes de sistema para garantir que todas as funcionalidades funcionem corretamente em conjunto.



## Critérios de liberação e falha dos itens

Aqui, são estabelecidos os critérios que determinam se um item, como uma funcionalidade ou um módulo, está pronto para ser lançado ou se falhou nos testes realizados.

Um critério de liberação pode ser que todas as funcionalidades do *software* tenham passado em, pelo menos, 95% dos casos de teste, não havendo defeitos de alta severidade pendentes.



## Entregas do teste

Você identifica os documentos e artefatos que serão entregues como parte do projeto de teste.

Isso pode incluir planos de teste, casos de teste, procedimentos de teste, relatórios de resultados, dados de teste, programas e componentes relacionados.

As entregas do teste podem incluir o plano *máster* de teste, planos de teste específicos para cada nível, casos de teste elaborados e relatórios detalhando os resultados dos testes realizados, além das ferramentas de teste utilizadas.



## Requisitos de suspensão e retomada

É quando você define os critérios que determinam quando os testes podem ser suspensos temporariamente. Aqui, também se define se os testes podem ser retomados.

Os testes podem ser suspensos temporariamente se uma funcionalidade crítica apresentar um número significativo de defeitos.



# Gerência de testes

Quando se lista as atividades e as tarefas que serão realizadas para gerenciar o projeto de teste.

As atividades de gerência de teste podem incluir revisar a documentação de desenvolvimento, preparar o ambiente de teste com os recursos necessários, elaborar casos de teste de acordo com os requisitos, bem como registrar e documentar os resultados dos testes.



# Responsabilidades

Aqui, identifica-se os grupos e os indivíduos responsáveis por diferentes aspectos do projeto de teste, o que inclui as equipes de desenvolvimento, os testadores, os analistas de suporte, os usuários e os gerentes de qualidade.

As responsabilidades podem incluir que o gerente de projeto de teste seja responsável por supervisionar todo o projeto de teste. Os desenvolvedores também podem ser responsáveis por fornecer os itens de teste e, por fim, os testadores podem ser responsáveis por preparar e executar os testes.



# Interface entre as partes envolvidas

Descrição de como será realizada a comunicação entre as diferentes partes envolvidas no projeto, incluindo equipes internas e recursos externos.

A interface entre a equipe de desenvolvimento e a equipe de teste pode envolver reuniões regulares de *status*, compartilhamento de documentação e relatórios de progresso.



## Necessidades de equipe e de treinamento

Identifica-se as habilidades e as competências necessárias para a equipe de teste. Além disso, aqui se definem os requisitos de treinamento, garantindo que a equipe esteja preparada para realizar os testes de maneira eficaz.

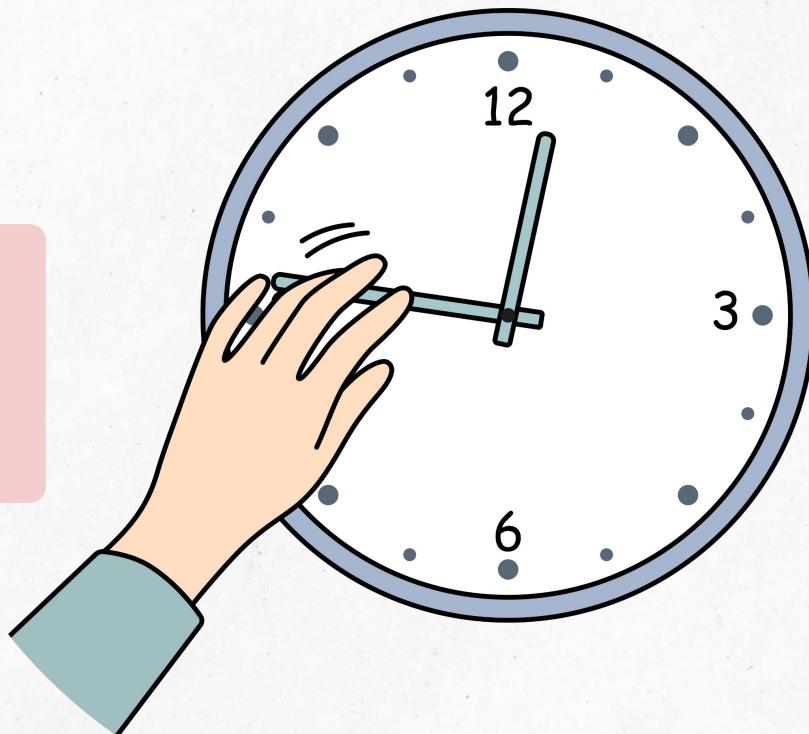
As necessidades de equipe podem incluir habilidades em testes automatizados, o conhecimento de linguagens de programação específicas e a capacidade de lidar com ferramentas de teste. Os requisitos de treinamento podem incluir *workshops* sobre novas ferramentas de teste ou metodologias.



# Cronograma, estimativa e custos

Detalhamento do cronograma do projeto de teste, incluindo estimativas de tempo e custos para as diferentes atividades.

Um cronograma pode listar as datas de início e de conclusão de cada atividade, considerando da revisão da documentação até a execução dos testes. As estimativas de custo podem incluir despesas com recursos.



# Bloco 4

---

Os principais meios de garantir a qualidade.

## Antes de tudo...

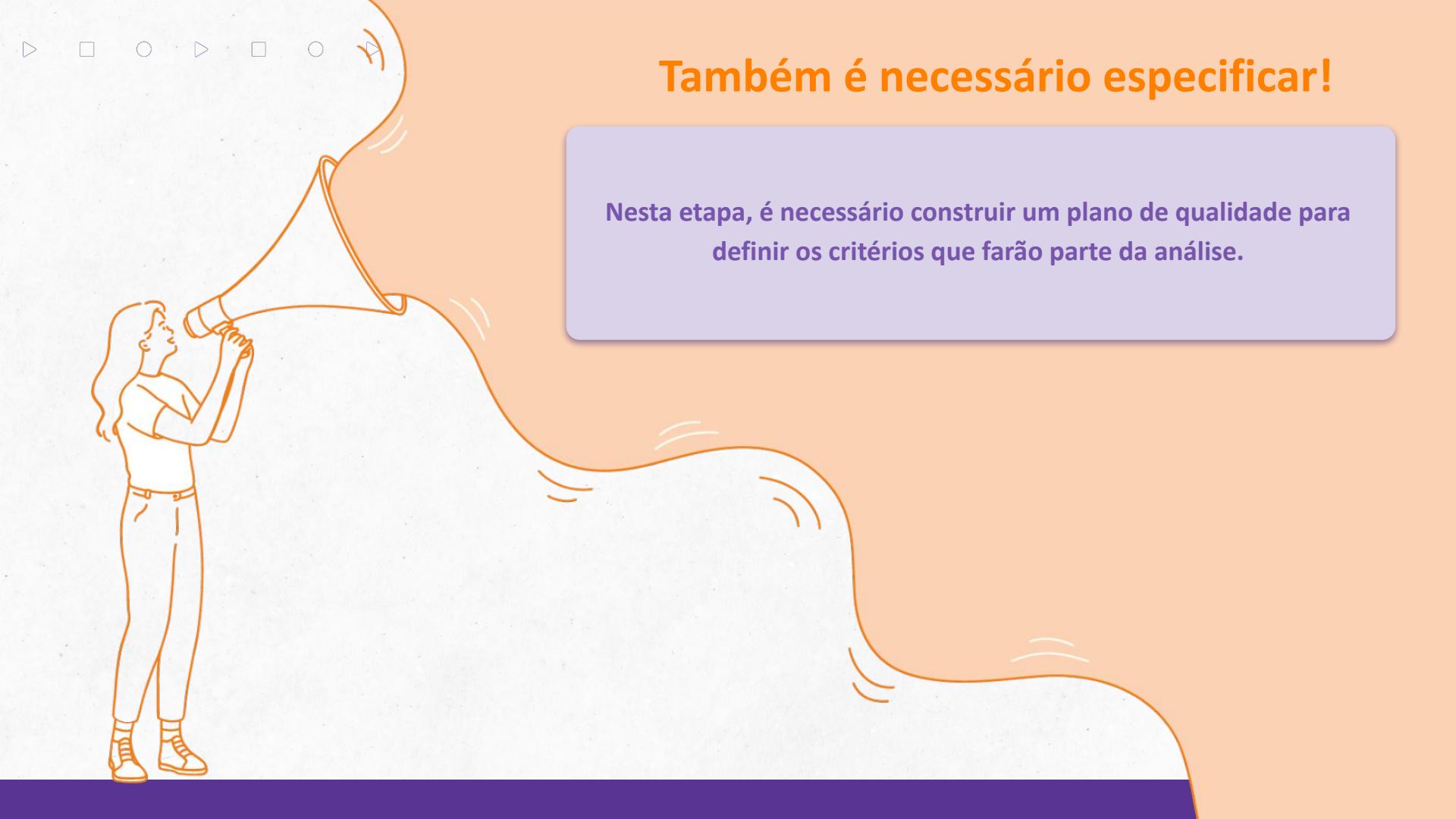
Você sabe qual é a diferença entre garantir e controlar a quantidade?



## A resposta correta é

O controle de qualidade envolve atividades e processos realizados durante a produção. A garantia, por sua vez, engloba a implementação de processos e de sistemas que visam prevenir problemas, assegurando que os produtos ou os serviços sejam produzidos de acordo com os padrões definidos desde o início.



A stylized illustration on the left side of the slide. It features a woman with long hair, wearing a white t-shirt and orange pants, standing and holding a megaphone. A large, orange, speech-bubble-like shape surrounds her head and extends down her body. Inside this shape, there is a white area where the text is placed.

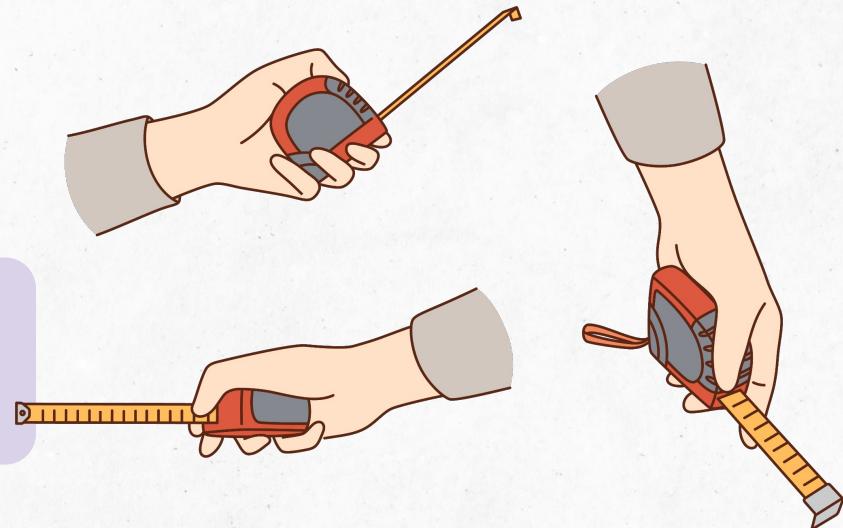
# Também é necessário especificar!

Nesta etapa, é necessário construir um plano de qualidade para definir os critérios que farão parte da análise.

# Métricas

Aqui, são definidos os indicadores e as métricas que serão coletados durante o processo de teste. Essas métricas podem ser usadas para avaliar o progresso, a qualidade e a eficácia dos testes realizados.

Uma métrica importante pode ser a taxa de defeitos encontrados por caso de teste. Isso ajudará a equipe a avaliar a eficácia dos testes na identificação de problemas no *software*.



# Cobertura dos testes

Determina até que ponto o *software* será testado em termos de funcionalidades, requisitos ou linhas de código.

No teste de sistema, a cobertura pode ser definida em termos de porcentagem de requisitos testados. Por exemplo, o objetivo pode ser testar, pelo menos, 90% dos requisitos do sistema.



## Alterações do documento e histórico

Trata-se das alterações que podem ocorrer no plano de teste e como elas serão gerenciadas. As mudanças são registradas junto com os responsáveis pela alteração e aprovação.

Qualquer alteração no plano de teste deve ser registrada em uma planilha de controle de alterações, indicando quem fez a alteração, quando foi feita e qual foi o motivo.



## Desenho ou *design* do teste

Detalha-se os aspectos específicos do projeto ou desenho de teste, como funcionalidades a serem testadas, abordagem refinada, casos de teste, critérios de aprovação e falha e entregáveis.

Se o *software* é um aplicativo de comércio eletrônico, o desenho de teste pode incluir uma lista de funcionalidades a serem testadas, como “registro de usuário”, “adicionar ao carrinho” e “realizar pagamento”.



# Desenho ou *design* do teste

Deve conter os campos a seguir.

## Introdução

Identificador, escopo e referências.

## Detalhes deste nível

*Features* a serem testadas, abordagem refinada, casos de teste com e as suas respectivas identificações, critérios de passagem e falha por característica ou funcionalidade e entregáveis.

## Geral

Glossário, procedimentos de alterações do documento e histórico de alterações.

# Introdução

Deve conter as informações mostradas a seguir.

**Identificador:** um código único que identifica o projeto de teste e se relaciona com o plano de teste;

**Escopo:** especifica uma parte do escopo do plano de teste ao qual o documento se refere. Pode abranger um grupo de funcionalidades a serem testadas;

**Referências:** lista as referências internas e externas relevantes para o plano de teste, incluindo a autorização do projeto, os planos relacionados, além dos requisitos e dos regulamentos.

# Detalhes do nível

1

**Features ou funcionalidades a serem testadas:** devem ser listadas todas as funcionalidades cobertas por este projeto.

2

**Abordagem refinada:** devem ser detalhadas as técnicas de teste que serão adotadas.

3

**Casos de teste com a sua respectiva identificação:** uma breve descrição dos casos de teste.

4

**Critérios de passagem e falha por característica ou funcionalidade.**

5

**Entregáveis:** uma lista de entregas do projeto de teste.

# Geral

Engloba os itens apresentados a seguir.

1

Para não haver dúvidas, um glossário contendo os termos importantes que são utilizados no documento.

2

Registro das alterações feitas no documento, incluindo os responsáveis pelas alterações e as aprovações.

# Bloco 5

---

Aprendendo a utilizar as plataformas Kubernetes e  
Github Actions.



Dê *play* no conhecimento!



Vamos assistir a um vídeo?

## Discussões e reflexões



# Bloco 6

---

Hora de praticar!

## Vamos fazer um desafio?

Separem-se em grupos para realizar a simulação de um **ambiente de teste**.

Aqui, vocês terão a oportunidade de experimentar o uso do Kubernetes e do GitHub para criar um ambiente de teste para um aplicativo simples.

Com o código-fonte do professor, criem um repositório no GitHub e façam o *upload*;

Criem um arquivo Dockerfile para o aplicativo, bem como uma imagem Docker a partir do Dockerfile. Depois, implante essa mesma imagem em um *cluster* Kubernetes;

Crie um arquivo de manifesto Kubernetes (yaml) para implantar o aplicativo como um *pod*;

Peça aos grupos para acessarem o aplicativo implantado a partir do endereço IP ou domínio configurado;

Os grupos devem interagir com o aplicativo e verificar se está funcionando conforme o esperado.

► □ ○ ▶ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷

E aí, foi difícil? Compartilhe com a turma!



# Fechamento

Que bom!

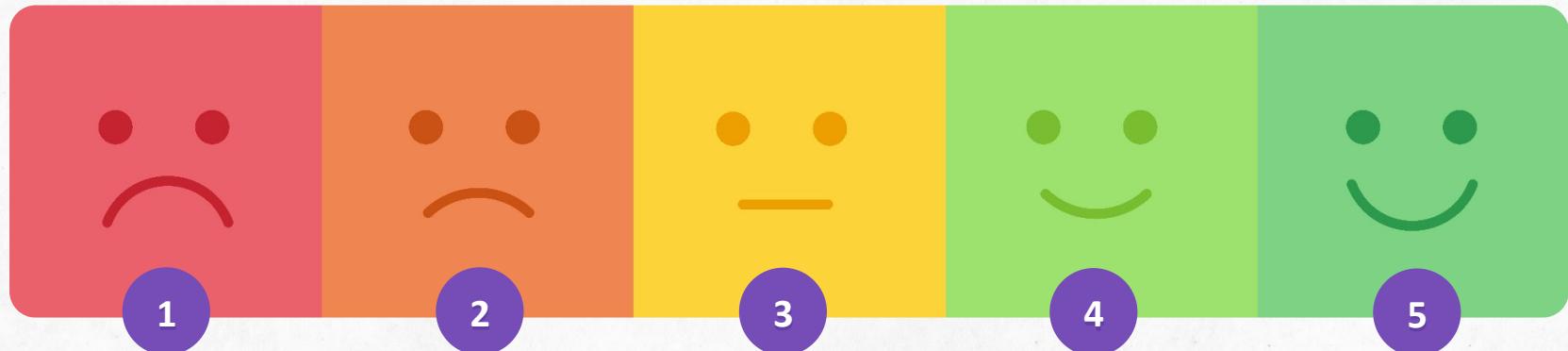
Que pena...

Que tal?

## Fechamento: avaliação de aprendizagem

Chegou o momento de avaliarmos o nosso nível de compreensão em relação ao conteúdo da aula!

Classifique o seu **nível de compreensão** em uma escala de 1 a 5, sendo 1 para “ainda estou perdido” e 5 para “entendi tudo e gostei do que aprendi!”. Em seguida, justifique a sua resposta.



# Referências Bibliográficas

PROZ EDUCAÇÃO. Apostila de *Qualidade de software*. 2023.