



# Jovem Programador

Qualidade e Testes de  
Software

Prof. Vanessa Roberg

Aula 02

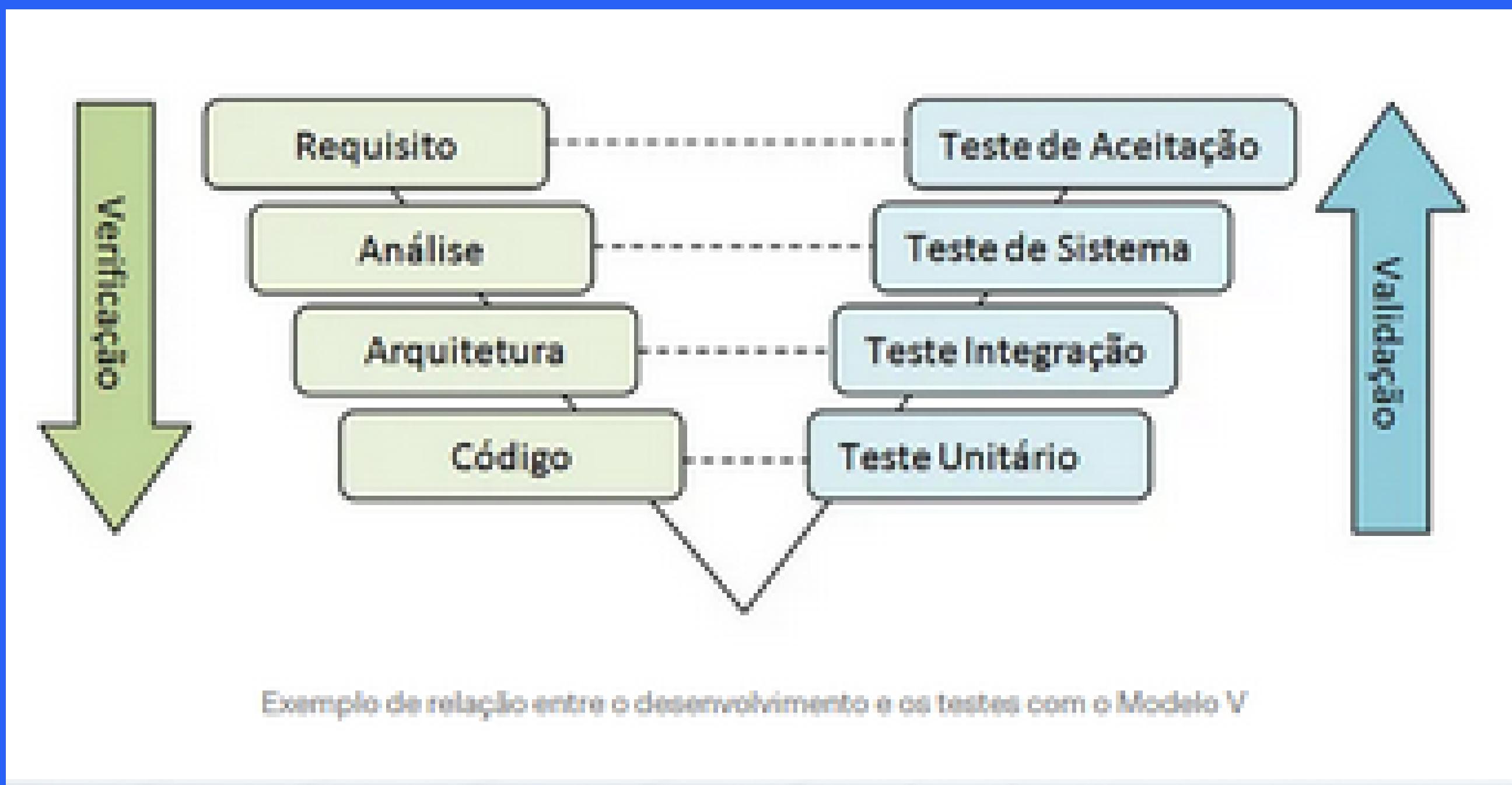
# Agenda

- Modelo em V;
- Metodologia FURPS;
- Tipos de Testes de Software;
- Ambiente de Testes



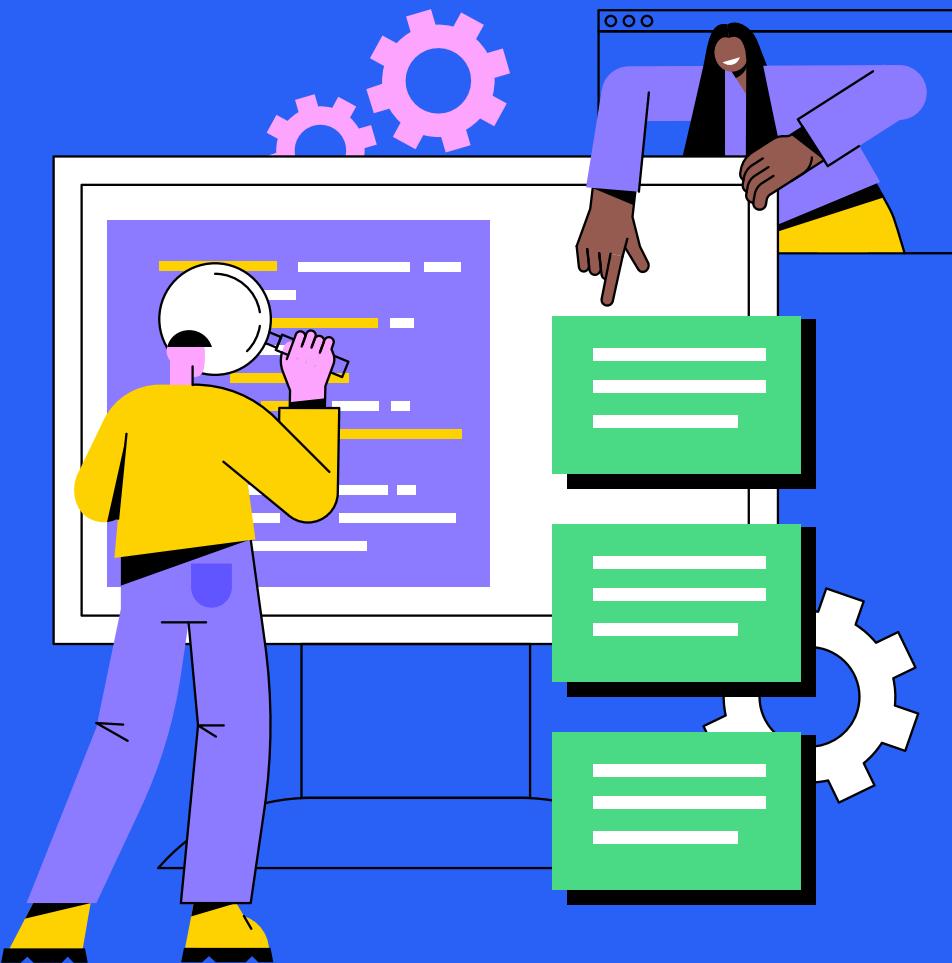
# Modelo em V

## Verificação e Validação



# Modelo em V

## Verificação e Validação



O modelo em V é um modelo de processo de desenvolvimento de software que enfatiza a realização de atividades de teste em paralelo com as atividades de desenvolvimento.

# Principais etapas do modelo em V



- Requisitos
- Especificação
- Arquitetura/Design de Alto Nível
- Design Detalhado
- Codificação
- Teste de Unidade
- Integração
- Teste de Aceitação
- Manutenção



# Análise de requisitos



Nesta etapa, os requisitos do software são levantados e documentados. Isso envolve entender as necessidades dos usuários e definir as funcionalidades e características do sistema.



# Especificação



Com base nos requisitos, uma especificação detalhada do software é elaborada. Isso inclui a definição de como o sistema deve se comportar, suas interfaces, restrições e outros detalhes relevantes.



# Arquitetura/Design



Nesta fase, a arquitetura do sistema é projetada, incluindo a definição dos componentes principais, suas interações e interfaces. O objetivo é criar uma visão geral do sistema e estabelecer uma base sólida para o desenvolvimento.





# Design Detalhado



Com base na arquitetura, o design detalhado é realizado. Isso envolve a definição das estruturas de dados, algoritmos, interfaces de componentes individuais e outras especificações técnicas necessárias para a implementação.



# Codificação



A etapa de codificação envolve a implementação real do software com base no design detalhado. Os programadores escrevem o código fonte do sistema, seguindo as práticas de programação adequadas e as diretrizes estabelecidas.



# Teste de Unidade



Nesta fase, os componentes individuais do sistema, como funções ou módulos, são testados de forma isolada para verificar se funcionam corretamente de acordo com as especificações. O objetivo é identificar e corrigir erros ou defeitos no nível mais baixo do software.





# Integração



Após os testes de unidade, os componentes são integrados para formar o sistema completo. Os testes de integração são realizados para verificar se os componentes funcionam corretamente juntos, se as interfaces são compatíveis e se a comunicação entre eles está adequada.





# Teste de Sistema



Nesta etapa, o sistema completo é testado como um todo. O objetivo é verificar se o software atende aos requisitos definidos anteriormente e se funciona corretamente em seu ambiente de implantação.

Diferentes técnicas de teste, como testes de funcionalidade, desempenho e segurança, podem ser aplicadas.





# Aceitação



O teste de aceitação é realizado pelos usuários finais ou pelos representantes do cliente para validar se o software está de acordo com as expectativas e requisitos do usuário. É uma etapa importante para garantir a satisfação do cliente e a conformidade do sistema.



# Implantação



Após a conclusão dos testes e a aprovação do software, ele é implantado em seu ambiente de produção. Isso envolve a instalação, configuração e preparação do sistema para uso pelos usuários finais.

# Manutenção



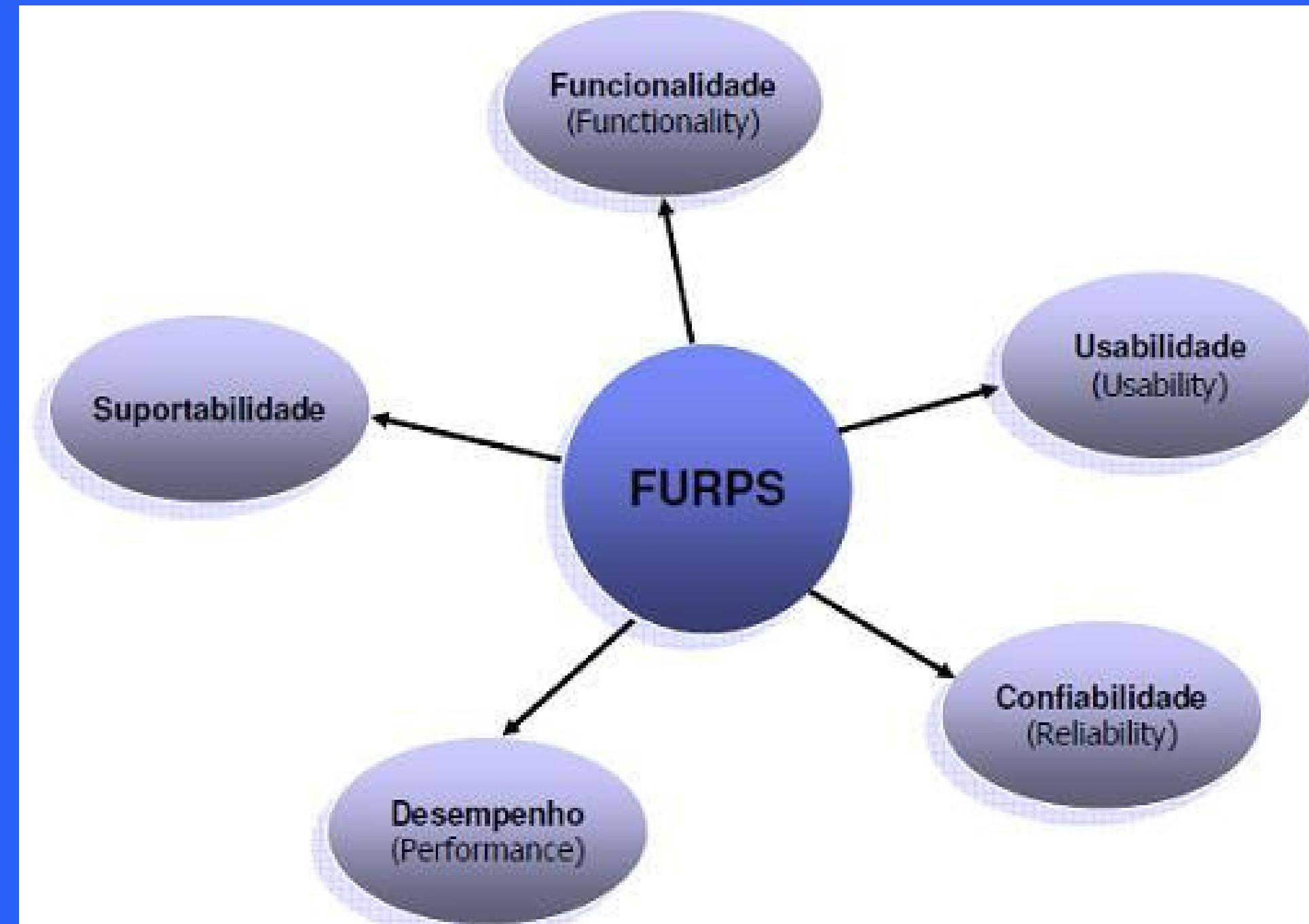
A fase de manutenção ocorre após a implantação e envolve a correção de erros, aprimoramentos e atualizações do software ao longo do tempo, à medida que novas necessidades surgem ou problemas são identificados.



# Metodologia FURPS



# FURPS representa seis categorias principais de requisitos:

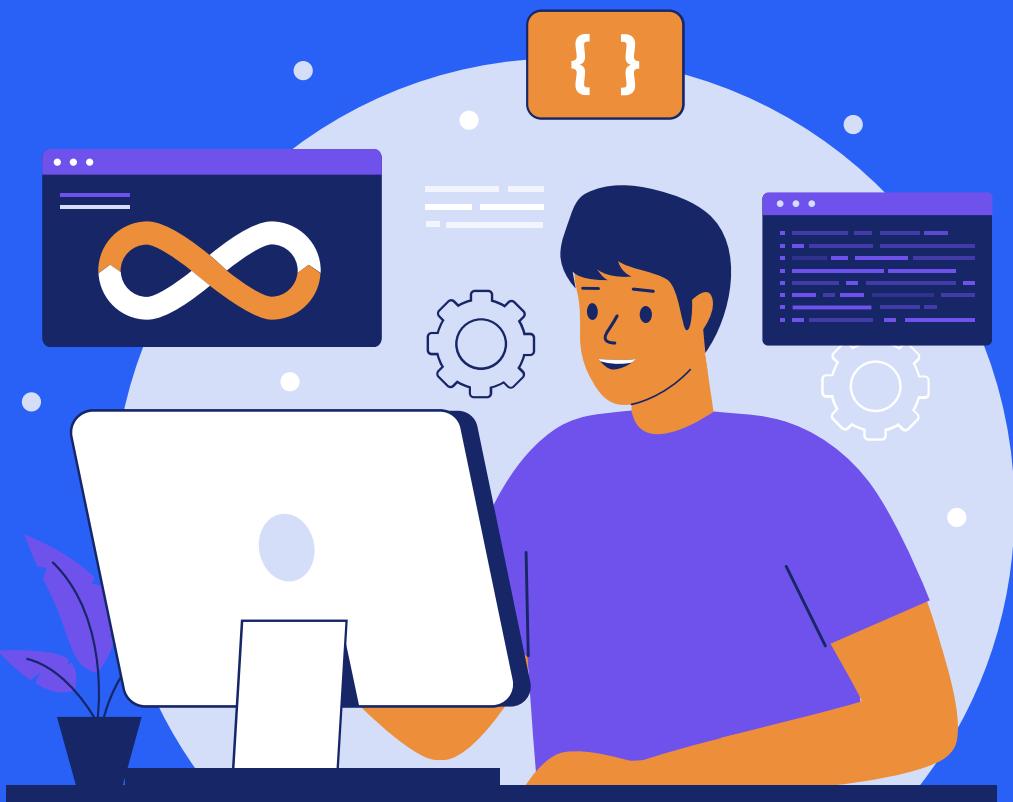


# Metodologia FURPS



Essa metodologia faz parte do Rational Unified Process (RUP).

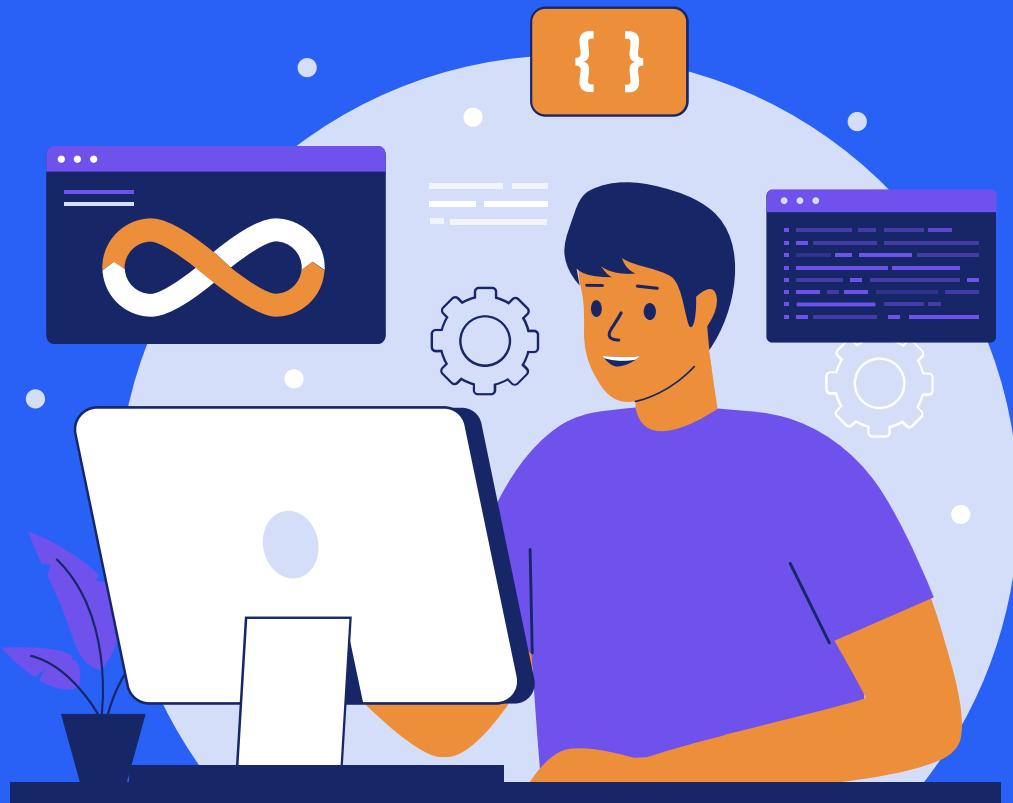
# Metodologia FURPS



\* RUP – É um processo de engenharia de software criado para apoiar o desenvolvedor. Foi criado pela empresa de engenharia de software, a Rational Software Corporation e em 2003 foi comprado pela IBM.



# FURPS - Funcionalidade



Funcionalidade: A capacidade do software de fornecer as funcionalidades necessárias para atender às necessidades do usuário.

Isto é, estar de acordo com as especificações definidas.

Tipos de testes: Testes funcionais e Testes de regressão.





# FURPS - Funcionalidade



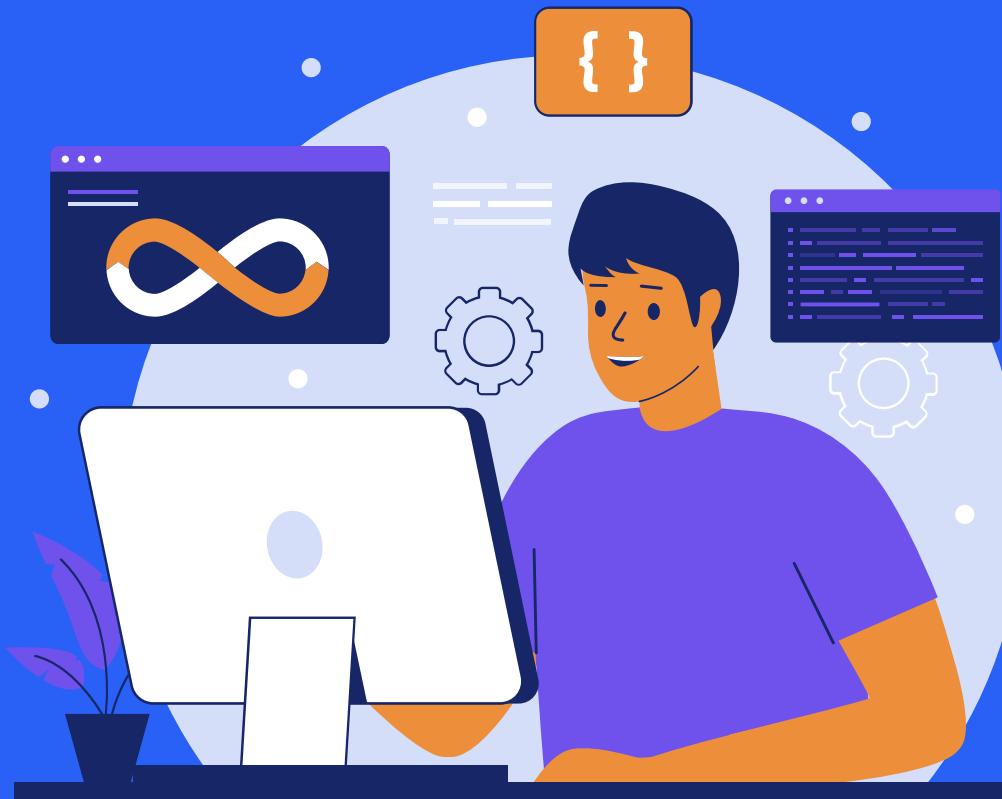
Testes funcionais: Se concentra na validação do comportamento do software em relação aos requisitos funcionais especificados.

Testes de regressão: Tem como objetivo garantir que as alterações feitas em um software não afetem negativamente as funcionalidades existentes.





# FURPS - Usabilidade



Usabilidade: A facilidade com que o software pode ser utilizado pelo usuário final.

Tipos de Testes: Testes de Interface e Teste de usabilidade.





# FURPS - Usabilidade

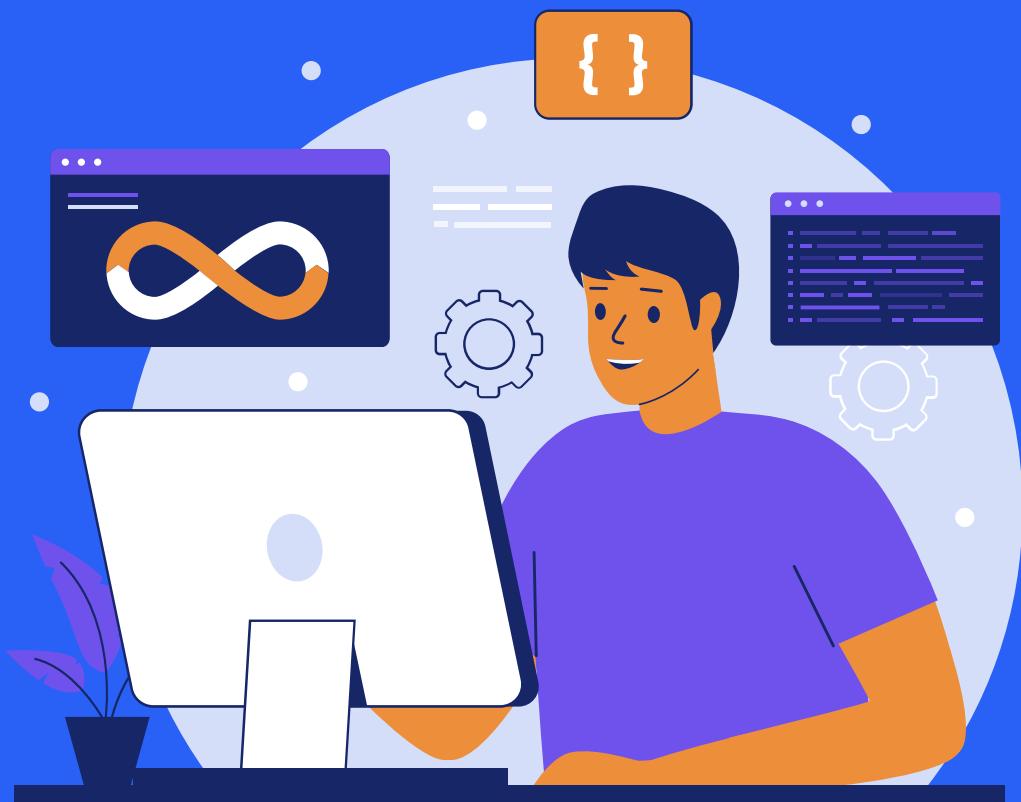


Testes de Interface: Tem como objetivo identificar problemas de interface, como botões que não funcionam, menus confusos, problemas de layout e outros problemas que afetam a usabilidade e a segurança do usuário.

Teste de usabilidade: Tem como objetivo avaliar a facilidade de uso e da experiência do usuário de um software. Esses testes são realizados para garantir que o software seja fácil de usar, intuitivo e forneça uma boa experiência ao usuário.



# FURPS - Confiabilidade



Confiabilidade: A capacidade do software de ser executado sem falhas e de fornecer os resultados esperados.

Tipos de Testes: Teste de estrutura e Teste de Estresse.



# FURPS - Confiabilidade

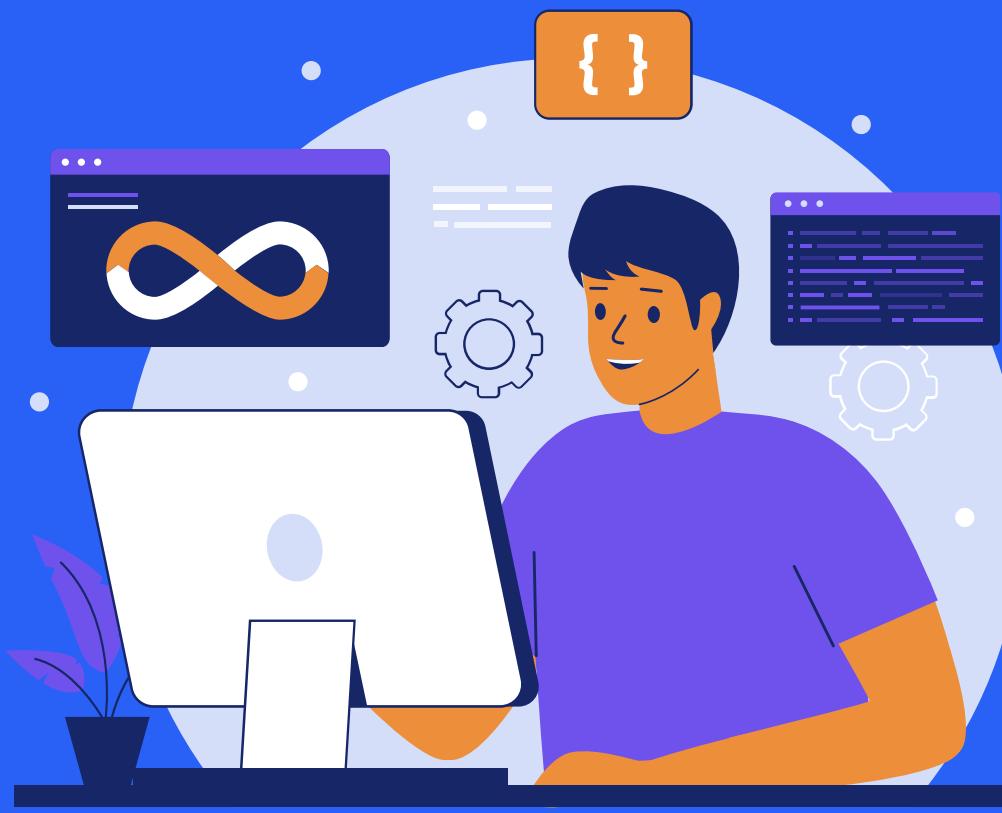


Teste de estrutura: Tem como objetivo avaliar a qualidade do código do software. A ideia é testar se o código fonte está livre de erros de programação, como: loops infinitos e outras vulnerabilidades.

Teste de Estresse: Tem como objetivo avaliar a capacidade de um software em lidar com condições extremas, tais como altas cargas de trabalho, grandes volumes de dados ou picos de tráfego.



# FURPS - Desempenho



Desempenho: A capacidade do software de atender às necessidades de desempenho, como tempo de resposta e velocidade de processamento.

Tipos de Testes:  
Teste de Carga



# FURPS - Desempenho



Tipos de Testes:

Teste de Carga: Tem como objetivo avaliar o desempenho do software sob uma determinada carga de trabalho, validando se o software pode lidar com a carga esperada sem diminuir seu desempenho ou falhar.



# FURPS - Suportabilidade



Suportabilidade: A capacidade do software de ser mantido e evoluído, além de ser compatível com outros sistemas.

Tipos de Testes:  
Testes de Instalação



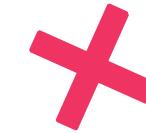
# FURPS - Suportabilidade



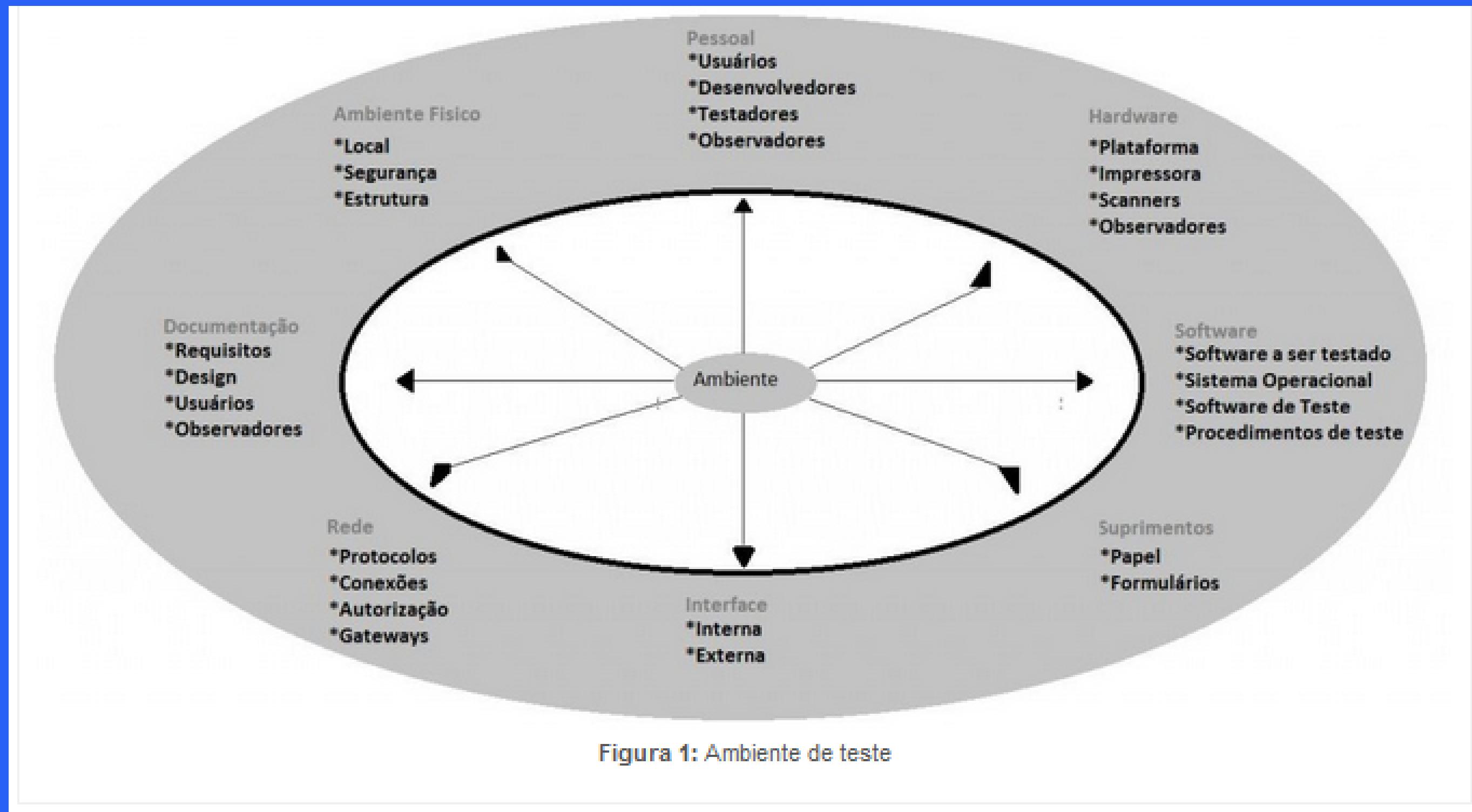
Testes de Instalação: Os testes de instalação de software são realizados para verificar se o software pode ser instalado corretamente em diferentes ambientes de hardware e software. Esses testes ajudam a garantir que o software seja instalado de forma segura e eficiente, sem afetar o desempenho de outros aplicativos ou componentes do sistema.



# Ambiente de Testes



**Na fase que antecede a elaboração dos cenários de teste, deve ser planejado o ambiente em que o produto será testado. Essa etapa tem relação direta com a escolha da estratégia de teste que será utilizada no projeto.**





# Ambiente de Testes



Planejamento: Antes de elaborar o ambiente de teste, é necessário realizar um planejamento detalhado do que deve ser testado e quais são os requisitos de hardware e software necessários para executar os testes.





# Ambiente de Testes



Seleção de ferramentas: Selecionar as ferramentas de teste adequadas é importante para automatizar o processo de teste e reduzir o tempo de execução. É importante escolher as ferramentas que melhor se adéquem aos requisitos específicos do projeto.





# Ambiente de Testes



Ambiente de teste separado: O ambiente de teste deve ser separado do ambiente de produção, para evitar conflitos de software ou hardware e garantir que os testes não afetem nenhuma operação do meu cliente.





# Ambiente de Testes



Configuração do ambiente: Uma vez selecionadas as ferramentas de teste, é necessário configurar o ambiente de teste, que inclui a instalação do software, configuração dos bancos de dados, configuração da rede, configuração do sistema operacional, etc.





# Ambiente de Testes



Criação de dados de teste: Para realizar os testes, é necessário criar dados de teste relevantes e realistas. É importante simular diferentes cenários e condições de uso para garantir que o software esteja funcionando corretamente em todas as situações possíveis.

É possível também, solicitar um backup do banco de dados do cliente para restaurar e realizar os testes com dados reais.



# Ambiente de Testes



Manutenção do ambiente: O ambiente de teste deve ser atualizado regularmente para garantir que esteja funcionando corretamente e que os testes estejam sendo executados adequadamente.





# Atividade!





Obrigada!

