

# QUALIDADE E TESTE DE SOFTWARE

---

Ricardo Beck e Daniel Callegari

“

A tecnologia não faz nada. Ela faz aquilo que  
nós fizemos com ela.

”

*Gabriela Cardozo Ferreira*

# Conheça o livro da disciplina

## CONHEÇA SEUS PROFESSORES

3

*Conheça os professores da disciplina.*

## EMENTA DA DISCIPLINA

4

*Veja a descrição da ementa da disciplina.*

## BIBLIOGRAFIA DA DISCIPLINA

5

*Veja as referências principais de leitura da disciplina.*

## O QUE COMPÕE O MAPA DA AULA?

6

*Confira como funciona o mapa da aula.*

## MAPA DA AULA

7

*Links de artigos científicos, informativos e vídeos sugeridos.*

## RESUMO DA DISCIPLINA

27

*Relembre os principais conceitos da disciplina.*

## AVALIAÇÃO

28

*Veja as informações sobre o teste da disciplina.*

# Conheça seus professores



## **RICARDO BECK**

Professor Convidado

Ricardo Beck tem um histórico profissional de mais de 20 anos de experiência em Qualidade de Software, com certificação internacional em Teste de Software (CSTE) tendo trabalhado com um extenso portfólio de produtos multinacionais, aplicando sempre estratégias e processos no estado da arte. Há 10 anos vem se dedicando também à gerência de projetos atuando como Scrum Master / Product Owner e mais recentemente sendo gerente sênior de pesquisa e desenvolvimento na HP Brasil. Ricardo tem certificações em gestão de pessoas pela FGV-Rio, SCM, PO e Agile Coaching pela Scrum Alliance além de fotografia pela ESPM.

## **DANIEL CALLEGARI**

Professor PUCRS

Doutor em Ciência da Computação (2010), Mestre em Ciência da Computação (1999) e Bacharel em Informática (1997) pela Pontifícia Universidade Católica do Rio Grande do Sul. Possui Especialização em Gestão Empresarial (Sebrae / ANFE / Itália) e Certificações Microsoft e IBM. Associado da Sociedade Brasileira de Computação (SBC).



# *Ementa da Disciplina*

Introdução aos conceitos de teste unitário, teste de integração, teste de UI. Introdução aos conceitos de garantia de qualidade de software. Estudo de métricas voltadas ao controle de qualidade no desenvolvimento de software.

# Bibliografia da Disciplina

As publicações destacadas têm acesso gratuito.

## Bibliografia básica

LAPORTE, Claude Y. , APRIL, Alain. Software quality assurance. Hoboken: John Wiley & Sons, 2018.

DELAMARO, Márcio Eduardo (Org.); MALDONADO, José Carlos (Org.); JINO, Mário (Org.). Introdução ao teste de software. 2a. Ed. Rio de Janeiro: Elsevier, 2016.

MALDONADO, José; et al. . Automatização de Testes de Software com Ferramentas de Software Livre. Rio de Janeiro: Elsevier, 2018.

## Bibliografia complementar

BURNSTEIN, Ilene. Practical software testing: a process-oriented approach. 5a. Ed. New York, NY: Springer, 2022.

GOERICKE, S. The Future of Software Quality Assurance. Springer Open, 2020.

VICENZI, A. et all. Automatização de Teste de Software com Ferramentas de Software Livre. 1a. Ed. Rio de Janeiro: Elsevier, 2018.

GALIN, D. Software quality: concepts and practice. Hoboke: John Wiley & Sons, 2018.

BRAGA, Pedro Henrique Cacique(org.). Testes de Software. São Paulo: Pearson Education do Brasil, 2016.

# O que compõe o Mapa da Aula?

## MAPA DA AULA

São os capítulos da aula, demarcam momentos importantes da disciplina, servindo como o norte para o seu aprendizado.

## EXERCÍCIOS DE FIXAÇÃO

Questões objetivas que buscam reforçar pontos centrais da disciplina, aproximando você do conteúdo de forma prática e exercitando a reflexão sobre os temas discutidos. Na versão online, você pode clicar nas alternativas.

## PALAVRAS-CHAVE

Conceituação de termos técnicos, expressões, siglas e palavras específicas do campo da disciplina citados durante a videoaula.

## VÍDEOS

Assista novamente aos conteúdos expostos pelos professores em vídeo. Aqui você também poderá encontrar vídeos mencionados em sala de aula.

## PERSONALIDADES

Apresentação de figuras públicas e profissionais de referência mencionados pelo(a) professor(a).

## LEITURAS INDICADAS

A jornada de aprendizagem não termina ao fim de uma disciplina. Ela segue até onde a sua curiosidade alcança. Aqui você encontra uma lista de indicações de leitura. São artigos e livros sobre temas abordados em aula.



## FUNDAMENTOS

Conteúdos essenciais sem os quais você pode ter dificuldade em compreender a matéria. Especialmente importante para alunos de outras áreas, ou que precisam relembrar assuntos e conceitos. Se você estiver por dentro dos conceitos básicos dessa disciplina, pode tranquilamente pular os fundamentos.

## CURIOSIDADES

Fatos e informações que dizem respeito a conteúdos da disciplina.

## DESTAQUES

Frases dos professores que resumem sua visão sobre um assunto ou situação.

## ENTRETENIMENTO

Inserções de conteúdos para tornar a sua experiência mais agradável e significar o conhecimento da aula.

## CASE

Neste item, você relembra o case analisado em aula pelo professor.

## MOMENTO DINÂMICA

Aqui você encontra a descrição detalhada da dinâmica realizada pelo professor.

# Mapa da Aula

Os tempos marcam os principais momentos das videoaulas.

## AULA 1 • PARTE 1

“Toda matéria voltada a qualidade é uma sensação, uma percepção.”

“ 01:12

02:31



### O que é qualidade?

Qualidade é uma sensação de que o produto atende as necessidades do usuário. Se essa sensação é tangível, imediatamente gera-se uma confiança automática, o grau de conformidade de um sistema, componente ou processo com seus respectivos requisitos.

### PALAVRAS-CHAVE

**QAI:** O Quality Assurance Institute concentra-se na criação de produtos e serviços de educação e treinamento para abordar o desenvolvimento de competências, avaliações e certificações profissionais de TI.

Q 02:40

Um projeto complexo precisa de uma multidisciplinaridade, pois envolve muitas facetas em um mesmo ecossistema. O ciclo de vida ágil de projetos em software reflete isso, trazendo agilidade, flexibilidade e adaptabilidade, entregando pequenos valores constantemente.

### PALAVRAS-CHAVE

**IEEE:** O Instituto de Engenheiros Eletricistas e Eletrônicos ou Instituto de Engenheiros Eletrotécnicos e Eletrônicos é uma organização profissional sem fins lucrativos, norte-americana, e é a maior organização profissional do mundo dedicada ao avanço da tecnologia em benefício da humanidade.

Q 03:07

03:21



### PALAVRA-CHAVE

**ABNT:** A Associação Brasileira de Normas Técnicas é o órgão responsável pela normalização técnica no Brasil. É uma entidade privada, sem fins lucrativos e de utilidade pública, fundada em 1940.

## PALAVRAS-CHAVE

**ISO:** A International Organization for Standardization refere-se a uma série de regras com o objetivo de realizar a normatização de condutas e processos em organizações e entidades públicas, nos mais diferentes segmentos no mercado.

## EXERCÍCIO DE FIXAÇÃO

Qual é a finalidade dos Stubs e Esqueletos?



03:21



08:36



*O teste é um exercício de disciplina.*



14:13

### Unidade para teste

O conceito de unidade às vezes é vago ou muito simples e refere-se à menor parte indivisível do código. Essas unidades devem ser isoladas entre si, para garantir o mínimo de interferência externa quando os estímulos forem recebidos.

Para a criação de níveis de isolamento são feitos os Stubs e esqueletos, que irão sempre responder a um valor conhecido quando a classe ou método depender de um agente externo.



20:21

## PALAVRA-CHAVE

**Banco de dados:** São conjuntos de arquivos relacionados entre si, com informações gerais, armazenando informações importantes para uma empresa ou informações com finalidade de pesquisa ou estudo científico.



25:23

### Estratégias de teste unitário

As estratégias unitárias passam por diversos pontos, um dos mais clássicos é o TDD (Test Driven Development) em que o desenvolvedor cria um caso de teste, falha esse teste e depois escreve o código que vai fazer esse teste passar. Outra forma de teste unitário é o BDD (Behavior Driven Development) em que se entende como unidade, um componente isolado e cria-se promessas para esse componente. A integração de testes pode ser feita localmente, pipeline ou rodar a cada commit. As diferentes estratégias irão causar impactos tanto na qualidade quanto no tempo de desenvolvimento.



29:10



*Quando um desenvolvedor começa a trabalhar, ele já começa a pensar em como fazer.*





## PALAVRAS-CHAVE

### Mission-critical application:

Programa ou conjunto de programas relacionados que deve funcionar continuamente para que um negócio ou segmento de negócios seja bem-sucedido.



30:23



33:50



*O código precisa ser coberto em termos de requisitos funcionais e não funcionais.*



## Processos de Build

O regime de processos de build requisitado, de acordo com a especificação ou característica do código, pode ser de commit a cada validação ou teste em cima do código completo do dia.

O processo de review e merge deve acontecer com uma forma ou camada de qualidade.

Os testes de integração avaliam o ecossistema no qual os diferentes métodos se integram compondo a solução.



35:16

## AULA 1 • PARTE 2

00:00



### Front-end e Back-end

Nos testes de interface com usuário temos alguns pontos críticos: alinhamento, tipografia, iconografia, ancoragem e vocabulário. Os times de qualidade, desenvolvimento e design devem trabalhar em conjunto para que esses pontos sejam claros e simples.

Quando falamos entre a interface gráfica e os testes de front-end, em geral, a linha é exatamente com relação a entradas e saídas, e o quão consistente os elementos visuais são durante a minha experiência.

“Consistência traz  
qualidade.”

02:25



“A qualidade cai porque  
a frustração sobe.”



06:02

17:27



## PALAVRA-CHAVE

**Tabela ASCII:** O código padrão americano para o Intercâmbio de Informação é um sistema de representação de letras, algarismos e sinais de pontuação e de controle, através de um sinal codificado em forma de código binário.

## Árvore do sistema: parte I

O docente demonstra como podemos construir de forma visual as entradas de testes de um sistema. Destaca algumas possíveis preocupações com o preenchimento dos dados que devemos ter no mapa mental e as formas de interações.



10:04

22:50



## PALAVRA-CHAVE

**Overflow:** É quando o computador executa uma conta e o resultado não pode ser representado, por ser maior do que o valor máximo permitido.

## Árvore do sistema: parte II

O professor Ricardo segue a demonstração do mapa mental, apresentando as saídas de tela e arquivos. Destaca que deve ser pensado, nas interfaces gráficas, sobre o tempo de concepção da solução, pois existe uma ampla variação de população, gerando o problema sobre a localização dos textos e strings de acordo com a linguagem utilizada.



25:42

36:48



## Acessibilidade

A acessibilidade é um conjunto de características que permitem que pessoas consigam ver, experimentar e diferenciar a solução de uma maneira ampla. O processo de testes deve avaliar diferentes perspectivas, como: uso do teclado para completar ações, modos de tela, resolução de tela e alto contraste.

## AULA 1 • PARTE 3

### EXERCÍCIO DE FIXAÇÃO

O custo do defeito é dado pelo:

00:52



#### Quais estratégias aplicar?

O custo do defeito é dado pelo tempo em que ele está presente. Se é pego pelo cliente, o custo é na reputação da empresa ou do desenvolvedor, impactando na retomada dos processos.

Na estratégia é importante priorizar itens, visto que o tempo é escasso na qualidade. A estratégia de qualidade deve ser norteadas pelas diferentes prioridades do software.

08:33



#### Teste de performance: parte I

O teste de performance deve ser feito sempre que possível e o mais cedo possível, pois seus critérios em geral não estão bem definidos. É importante ter o entendimento do que é uma baseline, quantos usuários em média vão usar, quais são os limites de escalonamento e quais são os SLA's definidos. A definição do teste de performance é feita pelo entendimento do ambiente.

### PALAVRAS-CHAVE

**Baseline:** É o plano dividido em fases aprovadas como o escopo do projeto, o custo, o cronograma e as mudanças técnicas aprovadas.



09:11

09:41



#### PALAVRA-CHAVE

**SLA:** Service Level Agreement consiste num contrato entre duas partes: entre a entidade que pretende fornecer o serviço e o cliente que deseja se beneficiar deste.

## Teste de performance: parte II

A liberação do software deve estar sempre atrelada àquilo que foi definido. Deve-se entender o cenário apresentado e, a partir disso, propor melhorias; a definição de baseline estabelece o uso ótimo da solução, expandindo até o ponto de quebra do sistema como um todo.

Uma das unidades mais básicas em testes de performance são o tempo de fila e tempo de resposta. O tempo de resposta é o tempo total somando fila e processamento.

O SLA define o tempo de resposta ou quantidade de tempo que o produto ou serviço deve estar disponível para o usuário em termos de tempo. Através das avaliações de performance podemos identificar possíveis falhas ou problemas com SLA's, assim como promover redundância para mitigar possíveis riscos.

## EXERCÍCIO DE FIXAÇÃO

Refere-se a um conjunto de passos e resultados esperados buscando uma única habilidade:



21:09



25:54



*O importante, muito além da ferramenta, é como usar, como extrair.*



31:08

## Conceitos de garantia de qualidade

Quando falamos em garantia de qualidade nos referimos a repetibilidade de processos, que força o time de qualidade a ter uma documentação robusta com relação ao seu dia a dia. Essa documentação pode variar entre diversas possibilidades como: estratégia de testes, plano de testes, casos de testes, pré-requisitos, passos de testes, resultados esperados e relatórios de testes.



## AULA 1 • PARTE 4

00:00



### Repetibilidade do processo

O processo de qualidade tem que ser repetível, ou seja, sempre ter o mesmo processo, os mesmos passos, na mesma máquina, no mesmo ambiente. Além disso, deve ser auditável a qualquer momento, para isso os processos devem ser bem documentados e atualizados. A padronagem do ambiente de teste é extremamente importante; para que o processo seja repetível são necessárias condições controladas.

“Quando acontecem grandes mudanças de código é necessário que se julgue toda a questão do teste como um todo.”



04:20



A questão do escopo de teste é relativa diretamente ao ambiente que tem que ser rodado e a prioridade que esse ambiente tem, com as características que ele tem.”



11:14

12:55



### Relatórios de teste

Os relatórios de teste devem conter: percentual de testes que passaram e falharam e o número de defeito por cada teste. Isso é um mínimo que se precisa para entender o estado atual. Então, os relatórios têm que trazer em si as características de um percentual e prioridade contra um número de defeitos achados. A priorização ocorre de diversos aspectos, os mais comuns são a probabilidade de ocorrência e o impacto para o usuário.

### Métricas de controle de qualidade

Existem diversas métricas para controle de qualidade, as mais comuns são: número de defeitos por prioridade (versão) e número de defeitos por sistema/browser/requisitos.



22:30

### PALAVRA-CHAVE

25:35



### PALAVRAS-CHAVE

**Teste A/B:** Método de teste através do qual comparam-se elementos aleatórios com duas variantes, A e B, em que estes são o controle e o tratamento de uma experiência controlada, com o objetivo de melhorar a percentagem de aprovação.



25:48

**Streaming:** Tecnologia de transmissão de dados pela internet, principalmente áudio e vídeo, sem a necessidade de baixar o conteúdo.

## EXERCÍCIO DE FIXAÇÃO

Como é denominado o cálculo que auxilia o entendimento dos gastos comprometidos com a qualidade:

28:26



### ROI e estratégias de qualidade

O retorno sobre o investimento (ROI) é o cálculo que auxilia o entendimento dos gastos comprometidos com a qualidade e qual o prejuízo caso não tivesse sido aplicado.

As estratégias de qualidade mudam a cada sprint, então devem ser feitas investigações e adaptações ao teste.

32:19



“Se eu vejo que aquele produto me atende e funciona para aquilo que eu gostaria que ele funcionasse, eu sinto a qualidade.”

## PALAVRAS-CHAVE

**API:** Application Programming Interface são um conjunto de padrões que fazem parte de uma interface, elas permitem a criação de plataformas de maneira mais simples e prática para desenvolvedores.



37:57

## AULA 2 • PARTE 1

02:14



“As métricas têm que estar dentro de um ambiente, dentro de um sentido, dentro de um conceito.”

### Projetos ágeis

Em projetos ágeis, a qualidade tem que estar presente em todos os passos. Os responsáveis pelo teste devem usar a comunicação e a diplomacia para garantir a qualidade.

O desenvolvimento ágil tem diversas etapas: desenvolvimento, review, feedback, aprovação, documentação e incorporação das mudanças e ajustes.



04:12



## EXERCÍCIO DE FIXAÇÃO

Em projetos ágeis, a qualidade tem que estar presente em:



*A gerência desses conflitos e diplomacia são fundamentais para o bom andamento do projeto.*



08:20

14:24



### Como atuar?

A maneira que o profissional de teste deve atuar é direta, conversando com cada uma das pessoas, trazendo os desenvolvedores e designers para reuniões que são necessárias.

O ciclo de vida do processo depende muito de como a qualificação trabalha, devendo ser contínuo e apresentando melhorias.



*O comum do dia a dia é que o ambiente do cliente seja mais básico, mais simples ou menos atualizado que aquele do desenvolvimento.*



25:56

26:57



## PALAVRA-CHAVE

**Firmware:** Conjunto de instruções operacionais que são programadas diretamente no hardware de equipamentos eletrônicos.

### Ferramentas

Cada uma das fases do ciclo de vida de um processo ágil tem uma ferramenta para auxiliar e que traz muito valor quando bem usado. O professor apresenta as ferramentas para cada fase do projeto e destaca que todas as facilidades dependem muito do conhecimento das pessoas que estão agindo.



29:40

31:14



## PALAVRA-CHAVE

**Load balancing:** O balanceamento de carga é uma técnica para distribuir a carga de trabalho uniformemente entre dois ou mais computadores, enlaces de rede, UCPs, discos rígidos ou outros recursos, a fim de otimizar a utilização de recursos.



*O que eu automatizo em testes?  
Eu automatizo aquilo que é  
frequente.*



33:09

## AULA 2 • PARTE 2

00:00



### Perfis

Um SDET (Software Development Engineer in Test) tem como característica o domínio da linguagem de programação, é visto como um profissional mais completo, podendo auxiliar nos processos dependendo da demanda e não é, necessariamente, especializado em teste.

Ricardo apresenta as certificações de QA test existentes. Os certificadores são unidades ou órgãos que se mantêm vigentes por anos e trazem consigo uma valorização.



*A diplomacia faz com que  
exista uma certa troca, um  
certo respeito mútuo entre as  
diferentes partes.*



02:15

## PALAVRAS-CHAVE

**KPI:** Indicador-chave de desempenho são ferramentas de gestão para se realizar a medição e o consequente nível de desempenho e sucesso de uma organização ou de um determinado processo.



06:26

10:51



### O que são KPI's?

Normalmente, os KPI's trazem um peso maior ou menor de acordo com a especificação de software. A escolha dos indicadores deve ser feita de acordo com a natureza da solução, produto ou serviço. Quanto mais métricas são adicionadas, mais complicado se torna a verificação do projeto e análises providas pela mesma.



## PALAVRAS-CHAVE

**Capability building model:** Um modelo de capacidade é uma maneira de entender seu negócio de uma perspectiva holística, para transformá-lo com intenção.



13:59

18:41



“Times de qualificação normalmente trabalham com grande número de métricas.”

### Correlacionando qualidade e métricas

Para prestar atenção naquilo que importa, se utiliza a correlação específica:

- **Nível de teste unitário:** em geral, são vistas correlações entre quantidade de linhas de código adicionadas/modificadas por período.
- **Nível de teste de integração:** defeitos por componentes, componentes mais alterados x defeitos encontrados;
- **Nível de teste de sistema:** quantidade de defeitos achados por tempo, resolvidos por tempo e idade dos defeitos por prioridade.



20:11

31:33



### Métricas e gerência de projetos

Ao analisar um gráfico com as métricas estabelecidas, define-se uma linha de tendência para a ocorrência da release, comparando semana a semana. Gráficos mais simples e diretos podem trazer maiores impactos. Iniciamos pelo mais simples e, na reunião de projeto, perguntas são feitas e novas métricas são geradas, aumentando a complexidade.

31:51



“Quanto mais coisas eu vou medir, mais difícil é.”

“100% significa que não tenha defeitos? Não, significa que a criatividade do processo de qualidade parou por ali.”



35:13

## AULA 2 • PARTE 3

### Gerência de projetos

Em projetos com projeção, entendimento ou vontade de uma release maior, é comum se trabalhar em três eixos: pessoas, escopo e tempo. Quanto maior o escopo precisamos de mais tempo ou mais pessoas, quanto menor o tempo se faz necessário diminuir o escopo ou aumentar as pessoas; alguns gráficos conseguem auxiliar no gerenciamento desses eixos. Através da gerência, a tomada de decisão se torna mais simples, baseando-se em fatos.



00:14



05:33



*O processo de qualificação vai trazer esse valor, esse apoio às decisões de gerência.*



07:31



*A capacidade do desenvolvimento de produzir código é diretamente proporcional à capacidade de desenvolvimento de produzir novos defeitos.*

### Perspectivas gráficas

As perspectivas gráficas “mission criticals” são as mais complexas e que dão maior trabalho para manter. Nos demais casos, trabalha-se com diferentes formatos, padrões e métricas assemelhadas. O professor Ricardo Beck apresenta algumas métricas e como elas são construídas para montar os gráficos.



10:50



26:07

### JMeter

O docente apresenta a ferramenta JMeter, uma ferramenta Java que realiza testes de carga e de estresse em recursos estáticos ou dinâmicos, oferecidos por sistemas computacionais, através de usuários virtuais.

### PALAVRAS-CHAVE



30:13

**FTP:** Protocolo de Transferência de Arquivos é um protocolo padrão/genérico independente de hardware sobre um modo de transferir arquivos e também é um programa de transferência.

## AULA 2 • PARTE 4

### LoadRunner

O LoadRunner é uma ferramenta utilizada para testar aplicativos, medindo o comportamento do sistema e o desempenho sob carga. Ricardo demonstra algumas características da ferramenta e sua utilização.



00:00



### EXERCÍCIO DE FIXAÇÃO

Porque os testes no LoadRunner devem ser parados gradualmente?

### PALAVRAS-CHAVE

**Memory leak:** Vazamento de memória é um fenômeno que ocorre quando um programa de computador gerencia incorretamente alocações de memória de maneira que certa memória não é liberada quando não é mais necessária.



12:32



15:21

### PerfMon

O Windows tem a possibilidade de entender como é que se dá a alocação local para um teste de performance da máquina. PerfMon é uma ferramenta de monitoramento do sistema introduzido no Windows NT 3.1, que monitora várias atividades em um computador, como CPU ou uso de memória.



20:57

### Teste A/B

O teste A/B compara duas versões de conteúdo para identificar qual versão tem maior atratividade para os usuários finais. Esse teste proporciona maiores benefícios no uso contínuo; a cadência correta pode trazer uma melhoria em performance atreladas ao serviço.



33:17

“Através das verificações de hipóteses, eu aumento ou melhoro o meu conhecimento nos processos e minha experiência.”



33:26

### Analytics

Analytics são ferramentas pela quais se monitoram as interações a fim de entender o engajamento geral. Elas nos trazem informações como: tráfego, engajamento, conversões e tendências de performance ao longo do tempo.

## AULA 3 • PARTE 1

### Introdução e revisão

A grande área de qualidade tem um grande objetivo que é assegurar que o software cumpra com as suas especificações e atenda às necessidades dos clientes. O profissional da área constrói uma série de especificações através de entrevistas com o cliente, entende as reais necessidades e redige uma série de documentos com as especificações de como o sistema deve se comportar.

- **Revendo os conceitos:**

Verificação da correção daquilo que se construiu com base nos requisitos que originalmente foram levantados, ou seja, uma busca pelos defeitos.

A validação com o cliente deve ser feita de preferência no ambiente final ou infraestrutura muito similar aonde o software vai efetivamente ser utilizado.



02:40



### EXERCÍCIO DE FIXAÇÃO

O processo de busca pelos defeitos, refere-se a:

06:28



“Os requisitos podem mudar ao longo do tempo.”

### Técnicas e níveis de teste

Temos dois tipos de técnicas de verificação:

- **Técnicas estáticas:** não requerem que o sistema seja executado;
- **Técnicas dinâmicas:** requerem trabalhar com uma representação executável do sistema.

Podemos dividir os testes de software em níveis:

- **Teste de Unidade:** cada componente vai ser testado sem comunicação com nenhuma outra parte;
- **Teste de Integração:** testar as interfaces entre os componentes;
- **Teste de sistema:** analisa as funcionalidades do sistema como um todo;
- **Teste de regressão:** re-testa após reparo de falhas ou inclusão de funcionalidades.



10:37



21:28

### Terminologia

São alguns termos importantes:

- **Caso de teste:** é um subconjunto de entradas e saídas que são planejadas dentro de um ambiente controlado de execução;
- **Roteiro de teste:** é um conjunto de casos de teste;
- **Salas de Guerra:** sala com vários monitores que apresentam a capacidade do software ao longo do tempo;
- **Dimensões de teste:** tipos de testes (o que vou testar), técnicas de teste (como testar) e níveis de teste (quando testar).

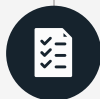
“Para o usuário as coisas se apresentam como uma falha.”

“Teste de software é um tema imenso.”

### Técnicas fundamentais

As técnicas fundamentais de teste podem ser classificadas em:

- **Caixa preta:** teste funcional, baseado na especificação formal e usado mais comumente no nível de sistema. Vantagens: independe da linguagem, manual ou automatizado, e pode ser utilizado em diversos níveis;
- **Caixa branca:** teste estrutural, baseado na estrutura do programa e usado mais comumente nos níveis de sistema de unidade e integração. Formas de trabalho: baseado em diversos fluxos de controle da aplicação, complexidade ou fluxo de dados.



### EXERCÍCIO DE FIXAÇÃO

Envolve a definição de um subconjunto de entradas e saídas que são planejadas dentro de um ambiente controlado de execução:



24:24



27:47



31:38



### EXERCÍCIO DE FIXAÇÃO

Qual o elemento chave da técnica funcional?

41:25



### PALAVRA-CHAVE

**Complexidade ciclomática:** É uma métrica de software usada para indicar a complexidade de um programa de computador, mede a quantidade de caminhos de execução independentes a partir de um código fonte.

## Casos reais

Daniel discorre sobre alguns casos reais ilustrando a importância dos testes, como o primeiro bug de computador, em 1947, onde um inseto havia entrado no hardware causando um problema, o que denominou os problemas que ocorrem em nossos sistemas como bug (inseto). Ele explica ainda outros casos que estão cada vez mais presentes em nosso cotidiano.



44:01

51:31



## PALAVRA-CHAVE

**Bug do milênio:** Foi o termo usado para se referir ao problema previsto para ocorrer em todos os sistemas informatizados na passagem do ano de 1999 para 2000, devido ao formato de data utilizado nos sistemas mais antigos.

## AULA 3 • PARTE 2

## Teste no nível de unidades

Uma unidade é a menor parte testável em um sistema. As técnicas mais comuns utilizadas nesse nível são drivers e stubs, junto com uma série de ferramentas.

O professor demonstra alguns exemplos de unidades em Javascript com Jest, Java com JUnit.



00:34

04:02



“É importante lembrar que o código de teste também é código.”

06:44



## Teste no nível de integração

A técnica de desenvolvimento de software como a TDD (test-driven development) se relaciona com o conceito de verificação e validação, e pode ser utilizada no nível de integração.

Os testes do nível de integração são testes que vão seguir naturalmente a partir de testes das diversas unidades e das inspeções. Embora funcionem em separado, as unidades devem ser testadas em conjunto.

## Drivers e stubs

São códigos de apoio de testes no nível de integração:

- **Drivers** são responsáveis pelo controle dos testes, exercitam as unidades sob teste;
- **Stubs** simulam o comportamento da unidade utilizando o mínimo de esforço.

É importante destacar que falhas detectadas durante o teste de integração podem ser um sinal de um teste insatisfatório no nível das unidades.

No nível de integração podem ser utilizadas técnicas top-down ou bottom-up.



14:50

## CURIOSIDADE

### Caso do foguete Ariane

O foguete Ariane, da comunidade Europeia, teve várias operações. Quando foi lançado o foguete Ariane 4, existia um módulo de software que calcula a inclinação horizontal, que possuía como uma das suas entradas, a velocidade do foguete. Alguns anos depois, foi construído um foguete maior, Ariane 5, reutilizando esse módulo o que causou um acidente. Nas investigações, a velocidade do novo foguete era muito maior e acabou causando um overflow que desequilibrou a trajetória do foguete.

### Integração e orientação a objetos

Os softwares orientados a objetos possuem algumas características que devem ser levadas em consideração na integração:

- Comportamento dependente do estado: testes devem considerar o estado em que o objeto se encontra e em cada momento que for feito a chamada dos métodos;
- Encapsulamento: às vezes é necessário ter acesso à informação privada, pode ser necessário ferir o encapsulamento para testes;
- Herança: qual é o efeito dos métodos novos e dos métodos sobrescritos, sobre a definição anterior;
- Polimorfismo e ligação dinâmica: um cuidado especial dado que uma única chamada pode estar ligada a diferentes métodos;
- Classes abstratas: deve-se ter pelo menos uma classe derivada a partir da qual consiga exercitar os métodos que herdou.

17:33



## PALAVRA-CHAVE

**Buffer overflow:** Situação em que um programa em execução tenta gravar dados fora do buffer de memória que não se destina a armazenar esses dados.

18:30



22:04



“Nenhum sistema é construído da noite para o dia.”

## EXERCÍCIO DE FIXAÇÃO

Como se denomina a técnica de construção de sistema de cima para baixo?

23:11



## EXERCÍCIO DE FIXAÇÃO

Qual o teste que avalia o comportamento funcional e requisitos de qualidade?

37:29



### Aceitação e regressão

Os testes de sistema, regressão e aceitação vão lidar o comportamento do sistema como um todo, mas cada um com um propósito diferente. O teste de sistema verifica contra uma especificação de requisitos, o teste de aceitação verifica de acordo com as necessidades do usuário, e o teste de regressão é o caso em que os testes são realizados novamente para verificar se algo não quebrou o que estava sendo feito anteriormente.



42:09



*O resultado de um teste de sistema é o comportamento externamente observado desse sistema.*



## PALAVRAS-CHAVE

**FURPS:** Acrônimo em inglês que representa um modelo para classificação de atributos de qualidade de software: Funcionalidade, Usabilidade, Confiabilidade, Desempenho e Suportabilidade.



43:14

## AULA 3 • PARTE 3

00:00



### Exemplos práticos: parte I

O docente demonstra exemplos práticos de testes automatizados nos níveis de unidades e integração em Javascript com JUnit e Javascript com Jest.

Demonstra como os testes realizados passam e cria funções realizando novos testes para identificar se o código é válido.

### Exemplos práticos: parte II

Daniel segue os exemplos práticos demonstrando a utilização do Jest, neste momento iniciando com o teste antes do método. Criando uma coleção de testes que vai passar por todos os casos definidos no método, o professor busca a comparação entre testes para identificar se estão funcionais.



21:27



37:25

**Exemplos práticos: parte III**

O professor destaca também que é muito comum fazer chamadas de API's, e apresenta o site [jasonplaceholder.typicode.com](https://jasonplaceholder.typicode.com), onde é possível testar API's. Demonstra como realizar um teste em uma base de usuários diretamente do browser.

**AULA 3 • PARTE 4****Selenium**

Daniel demonstra um exemplo prático de uma base de testes de IMC utilizando a ferramenta de teste Selenium. Realiza o teste automatizado em nível de sistema.



00:00

10:42

**Lighthouse**

O docente demonstra a utilização da ferramenta Lighthouse da Google, um plugin com o qual é possível gerar relatórios dentre cinco categorias de teste, apresentando diversas informações sobre as páginas web e sua performance.

“Quanto mais cedo a gente conseguir identificar um problema e levar isso para a equipe consertá-lo, mais barato é esse processo.”



20:33

21:47

**Sugestões: parte I**

O professor traz algumas sugestões:

- Montar e disponibilizar uma série de dashboards para o time de desenvolvimento e operações;
- Leitura: Os 25 erros de software mais perigosos.
- <https://www.sans.org/top25-software-errors/>

**Sugestões: parte II**

O professor sugere ainda como fonte adicional o OWASP, uma comunidade online que cria e disponibiliza de forma gratuita artigos, metodologias, documentação, ferramentas e tecnologias no campo da segurança de aplicações web.

Traz ainda a sugestão de leitura do código-fonte do JUnit disponível no GitHub, destacando que nenhum software está livre de falhas.



36:11

43:25



“Nenhum software é 100% seguro.”

# Resumo da disciplina

Veja, nesta página, um resumo dos principais conceitos vistos ao longo da disciplina.

## AULA 1

Toda matéria voltada a qualidade é uma sensação, uma percepção.



O importante é como usar, como extrair.



Consistência traz qualidade.



## AULA 2

Times de qualificação trabalham com grande número de métricas.



As métricas têm que estar dentro de um ambiente.



O processo de qualificação vai trazer o apoio às decisões de gerência.



## AULA 3

Os requisitos podem mudar ao longo do tempo.



Se ela não existir antes de você começar a testar, ela não vai existir quando o teste tiver terminado.



O código de teste também é código.



# Avaliação

Veja as instruções para realizar a avaliação da disciplina.

Já está disponível o teste online da disciplina. O prazo para realização é de **dois meses a partir da data de lançamento das aulas**.

Lembre-se que cada disciplina possui uma avaliação online.

A nota mínima para aprovação é 6.

Fique tranquilo! Caso você perca o prazo do teste online, ficará aberto o teste de recuperação, que pode ser realizado até o final do seu curso.

A única diferença é que a nota máxima atribuída na recuperação é 8.

**PUCRS** online

 **uol** edtech.