

# Testing I

## Fundamentação

Falamos de qualidade o tempo todo, mas sabemos realmente o que é qualidade? Para entender este termo, podemos nos basear na definição formal: "Qualidade de software é o grau em que um sistema, componente ou processo atende aos requisitos especificados e às necessidades ou expectativas do cliente ou usuário" (IEEE Std 610, 1990).

Na indústria de tecnologia, a qualidade passa a ser importante no objetivo principal de toda equipe de trabalho, ou seja, a entrega eficiente e a curto prazo de um produto que tenha a maior qualidade possível. Embora toda a equipe seja responsável pela qualidade do produto de software, o testador do sistema será a pessoa responsável por orientar e liderar as atividades relacionadas à qualidade para garantir que as necessidades do cliente sejam atendidas.

Essas atividades estão incluídas no **ciclo de vida de teste de software**, o que nos ajuda a organizar nosso trabalho. Logo, conhecer e aprofundar essas atividades nos permitirá entregar um produto de qualidade.

## Objetivos de aprendizagem

* Conhecer e compreender a importância dos testes e da qualidade do software.
* Compreender e aprofundar-se sobre o ciclo de vida do processo de teste.
* Compreender e exercitar diferentes técnicas de teste.
* Criar e executar casos de teste apropriadamente.
* Compreender como relatar bugs e a importância deles.
* Conhecer os diferentes níveis e tipos de testes para determinar a conveniência de seu uso.
* Compreender e aprofundar-se nos testes de componentes (TDD e BDD).
* Conhecer e realizar testes de API.
* Aprender sobre diferentes frameworks de automação de teste e modelos de casos de teste automáticos.

## Metodologia de ensino-aprendizagem

A partir da Digital House, propomos um modelo educacional que inclui ambientes de aprendizagem síncrona e assíncrona com uma abordagem que une teoria e prática, por meio de aprendizagem ativa e colaborativa.

Nossa proposta inclui aulas ao vivo com seu grupo de estudantes e docentes, que você pode participar de onde estiver. Além disso, temos um campus virtual customizado, no qual você encontra aulas virtuais com atividades, vídeos, apresentações e recursos interativos, para fazer no seu próprio ritmo antes de cada aula ao vivo. Ao longo da sua experiência de aprendizagem na Digital House, você poderá desenvolver competências técnicas e sociais, como trabalho em equipe, criatividade, responsabilidade, compromisso, comunicação eficaz e autonomia.

Na Digital House, utilizamos a metodologia "Sala de Aula Invertida". O que isso quer dizer? Toda semana, vamos pedir para você se preparar para a próxima, lendo textos, assistindo vídeos, fazendo atividades, entre outros recursos. Dessa forma, ao chegar ao encontro ao vivo, você terá mais preparo para abordar o tema de maneira mais rica.

Utilizamos atividades e estratégias baseadas em métodos participativos e ativos para colocar você em movimento, porque só você sabe o que está fazendo por si. Por isso, organizamos as aulas para que você realmente trabalhe nela e possa colocar em prática as diferentes ferramentas, linguagens e habilidades que compõem a formação de um programador. Concebemos a aula como um espaço de trabalho.

Uma das questões centrais da nossa metodologia de ensino é aprender fazendo. Por esta razão, os exercícios estarão muito presentes ao longo do curso, ou seja, a prática de atividades de vários tipos e níveis de complexidade que lhe permitirão consolidar a aprendizagem e verificar se a assimilou corretamente. Desta forma, é possível alcançar uma aprendizagem mais significativa e profunda, assimilar o conhecimento de maneira mais eficaz e duradoura, relacionar o que foi aprendido com a realidade dos programadores web, promover autonomia e autoconhecimento, melhorar a análise, relação e compreensão de conceitos, que ajuda a exercitar uma infinidade de habilidades.

A aprendizagem entre pares é um dos elementos centrais da nossa metodologia, por esta razão, em cada aula propomos que trabalhe em mesas de trabalho com seus colegas, ao longo do curso iremos variar a composição dos grupos para potencializar a cooperação. O que se propõe é uma mudança de perspectiva sobre o curso em questão, onde o aluno não é mais visto percorrendo seu percurso acadêmico individualmente, mas como parte de uma equipe que resulta da soma das potencialidades de cada um. A distribuição em grupos de trabalho estimula a diversidade e o aproveitamento do potencial de cada integrante para melhorar o desempenho da equipe.

A explicação recíproca como eixo do trabalho diário não só facilita o aprendizado dos colegas, mas sobretudo potencializa a consolidação do conhecimento por parte de quem explica. Responsabilidade, autonomia e proatividade são promovidas, tudo no âmbito da cooperação, levando a ressignificar a experiência de aprendizagem relacionada a emoções positivas.

O trabalho cooperativo permite estabelecer relações responsáveis ​​e duradouras, aumenta a motivação e o comprometimento, além de promover um bom desenvolvimento cognitivo e social. A cooperação surge diante da dúvida. Se alguém tiver uma pergunta, perguntará a algum integrante do grupo designado que esteja disponível. Se a dúvida persistir, o facilitador é acionado. Se não resolverem, o facilitador pedirá a todos que parem para cooperar em equipe na resolução do conflito que levantou a dúvida. É assim que todos os integrantes da mesa vão debater em busca da solução. Caso ainda não consigam resolver, anotarão a questão que será abordada de forma assíncrona pelo suporte técnico ou de forma síncrona, na próxima aula pelo professor.



Todos os dias, ao final do dia, estudantes reconhecerão um dos integrantes do grupo com quem compartilharam aquele dia. O critério para esse reconhecimento é a cooperação.

Cada grupo terá um facilitador que será escolhido a partir dos reconhecimentos e gerando um sistema de rotação, onde qualquer pessoa poderá desempenhar esse papel. O facilitador não é uma figura estática, mas cumpre um papel dinâmico e versátil. O facilitador é um estudante que mobiliza a realização dos objetivos comuns da equipe colocando em jogo a cooperação. É quem compartilha seu potencial com a mesa em favor do restante da equipe e que, portanto, promove a cooperação.

## Informações sobre a matéria

##### Modalidade 100% remota

##### Número total de semanas: 9

##### Número de aulas ao vivo semanais: 3

* Número de aulas virtuais no Playground: 23
* Número total de aulas ao vivo: 23

## Requisitos e correlações

Quais disciplinas o aluno deve ter sido aprovado previamente?

* Introdução à Informática
* Programação Imperativa

Quais disciplinas você estuda a seguir?

* Projeto Integrador
* Testing II

## Modalidade de trabalho

Nossa proposta educacional é especialmente pensada para esta modalidade 100% a distância, por meio da aprendizagem ativa e colaborativa seguindo nosso pilar de "aprender fazendo". Os ambientes de aprendizagem são síncronos e assíncronos, com uma abordagem que une teoria e prática, para que ambas estejam presentes em todos os momentos.

Temos um Campus Virtual próprio, onde encontraremos atividades, vídeos, apresentações e recursos interativos com instâncias de trabalho individual e em equipe para aprofundar cada um dos conceitos. Além disso, realizaremos encontros online e ao vivo com o grupo de estudantes e docentes, aos quais poderemos participar de onde estivermos por meio de uma plataforma de videoconferência com nossa câmera e microfone para gerar uma experiência próxima.

## Metodologia de avaliação

A avaliação formativa é um processo contínuo que gera informações sobre a formação de nossos alunos e de nós como educadores.

Ao mesmo tempo, gera-se um conhecimento de caráter feedback, ou seja, tem uma função de conhecimento, pois nos permite conhecer os processos de ensino e aprendizagem. Também tem uma função de melhoria contínua porque nos permite saber onde estamos no processo, para validar se continuamos no caminho planejado ou se precisamos tomar novas decisões para cumprir os objetivos propostos.

Finalmente, a avaliação desempenha um papel importante em termos de promoção do desenvolvimento de competências muito valiosas.

Nosso objetivo é fugir da avaliação tradicional, onde muitas vezes é um momento difícil, entediante e tenso. Para isso, vamos utilizar a gamificação, que é uma técnica em que elementos do jogo são aplicados para tornar o conteúdo mais atraente, os participantes se sentem motivados e imersos no processo, utilizam os conteúdos de aprendizagem como desafios que realmente desejam superar e aprendem com o erro.

Por sua vez, para o registro da referida formação, utiliza-se um conjunto de instrumentos, para os quais é imprescindível utilizar a mais ampla variedade e técnicas de análise possíveis.

## Critérios de aprovação

* Realização de atividades de Playground (80% concluídas).
* Presença nas reuniões síncronas (90% de presença).
* Obter nota igual ou superior a 7 na avaliação parcial.
* Obter pontuação igual ou superior a 7 na nota final da disciplina.

## Conteúdos

Módulo 1: Fundamentos do Teste de Software e Gestão de Defeitos

Introdução ao processo de teste tradicional. Começando com a história do teste, o que um testador deve ter em mente em suas atividades diárias e terminando com o conhecimento em um dos principais objetivos de todo testador: encontrar defeitos. Neste módulo, você aprenderá o que é um defeito, como relatá-lo e gerenciá-lo.

Aula 1: Primeiros passos

* Um pouco de história
* Os 7 princípios do teste
* Aspecto psicológico do teste
* Os 3 pilares do desenvolvimento de software
* Qualidade
* Validação versus Verificação

Aula 2: Gestão de defeitos

* O que é um defeito?
* Erros, defeitos e falhas
* Ciclo de vida de um defeito
* Gestão de defeitos

Aula 3: Fundamentos e Gestão de Defeitos - Revisão

* Fechamento da semana (Aulas 1 e 2)

Módulo 2: Design de testes

Durante o design de teste, as condições de teste são transformadas em casos de teste. Neste módulo, serão apresentadas as diferentes técnicas de teste para gerar um caso de teste eficiente.

Aula 4: Design de testes

* Casos de teste
* Teste negativo e Teste positivo
* "Happy Path"
* Casos de uso
* Requisitos

Aula 5: Níveis e tipos de testes

* Modelos de desenvolvimento de software
* Níveis de teste
* Teste de componentes
* Teste de integração
* Teste de sistema
* Teste de aceitação
* Tipos de teste
* Teste funcional
* Teste não funcional

Aula 6: Design de testes, Níveis e tipos de testes - Revisão

* Fechamento da semana (Aulas 4 e 5)

Aula 7: Técnicas de testes

* Caixa preta e Caixa branca
* Análise de valor limite
* Partição de equivalência
* Teste de tabela de decisão
* Testes baseados na experiência

Módulo 3: Análise de teste

Durante a análise do teste, a base de teste é analisada para identificar "o que testar". Essas atividades de análise de teste não apenas verificam se os requisitos são consistentes, elas são devidamente expressas e completas, mas também validam se os requisitos atendem adequadamente às necessidades dos clientes, usuários e outros envolvidos.

Neste módulo, aprenderemos como se envolver nos testes desde os estágios iniciais e como diferentes pontos de vista nos processos de revisão ajudam a melhorar a qualidade do produto.

Aula 8: Testes estáticos e dinâmicos

* Testes estáticos
* Testes estáticos e Testes dinâmicos
* Processo de revisão

Aula 9: Técnicas de teste e Testes estáticos e dinâmicos - Revisão

* Fechamento da semana (Aulas 7 e 8)

Módulo 4: Implementação de testes

Durante a fase de implementação do teste, devemos responder à seguinte pergunta: está tudo pronto para realizar o teste? Após fazer as configurações necessárias, procedemos à execução dos testes. Neste módulo, aprenderemos como organizar, controlar e relatar a execução dos testes.

Aula 10: Implementação e execução do teste

* Execução de casos de teste
* Criação de suíte
* Smoke suites
* Suítes de regressão

Módulo 5: Planejamento de testes

O planejamento de teste envolve atividades que definem os objetivos do teste e a abordagem para atender aos objetivos do teste. Neste módulo, aprenderemos como os testes são organizados em diferentes ambientes de trabalho até chegarmos à entrega final.

Aula 11: Organização do teste

* Ambientes de teste (DEV, QA, UAT, STG, PROD)
* Métricas e relatórios

Aula 12: Implementação e execução do teste - Organização do teste - Revisão

* Fechamento da semana (Revisão da primeira parte da matéria)

Aula 13: Avaliação I

Módulo 6: Teste Unitário

Aula 14: Introdução ao teste de componente

* Debugging
* Introdução ao teste de unidade

Aula 15: Teste Unitário

* Teste de unidade
* Primeiro teste unitário com JavaScript
* Teste de cobertura de sentença
* Teste de cobertura de decisão
* Ferramentas de cobertura de teste

Aula 16: Teste Unitário - Revisão

* Fechamento da semana (Aulas 14 e 15)

Módulo 7: BE testing

Nesse módulo, será apresentado o conceito de API - sigla para “*application programming interface*”. Como funciona e quais as melhores ferramentas e formas de verificar o seu funcionamento.

Aula 17: Introdução - Teste de API

* Teste de back end
* APIs
* Protocolo HTTP
* Postman - Parte 1

Aula 18: Teste de API

* Postman - Parte 2
* GET e POST
* Testes (js)
* Coleções em postman / variáveis de ambiente

Aula 19: Teste de API - Revisão

* Fechamento da semana (Aulas 17 e 18)

Módulo 8: Introdução à Automação de testes

Neste módulo, será apresentado o mundo da automação de teste, com ênfase em demonstrar como isso melhora a qualidade do software e reduz os tempos de teste.

Aula 20: Fundamentos da automação de teste

* Introdução à automação
* Padrões de design
* Page Object Model
* Introdução ao Selenium
* Nosso primeiro teste com o Selenium
* Selenium WebDriver

Aula 21: Automação de teste

* Introdução ao Robot Framework
* Nosso primeiro teste com o Robot Framework

Aula 22: Automação de teste - Revisão

* Fechamento da semana (Revisão da segunda parte da matéria)

Aula 23: Avaliação II