

**Matéria: 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**

Na Digital House, propomos um modelo educacional que inclui ambientes de aprendizagem síncronos e assíncronos com uma abordagem que vincula teoria e prática, por meio da aprendizagem ativa e colaborativa.

Nossa proposta inclui aulas ao vivo com seu grupo de alunos e professores, nas quais você pode participar de onde estiver. Além disso, contamos com um campus virtual customizado, onde você encontrará aulas virtuais, com atividades, vídeos, apresentações e recursos interativos, para fazer no seu ritmo antes de cada aula presencial.

Ao longo da sua experiência de aprendizagem na Digital House você será capaz de desenvolver habilidades técnicas e soft skills, tais como: trabalho em equipe, criatividade, responsabilidade, compromisso, comunicação eficaz e autonomia.

Na Digital House usamos a metodologia “sala de aula invertida”. O que isso significa? Toda semana vamos pedir que você se prepare para a próxima, lendo textos, assistindo vídeos, fazendo atividades, entre outros recursos. Desta forma, quando você chegar na reunião ao vivo, você estará preparado para abordar o assunto de uma forma mais rica.

Utilizamos atividades e estratégias baseadas em métodos participativos e ativos para você se movimentar, já que só se sabe o que faz por si mesmo. Por isso, organizamos as aulas para que você possa realmente trabalhar e colocar em prática as diferentes ferramentas, linguagens e habilidades que constituem 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 isso, ao longo do curso os exercícios estarão muito presentes, ou seja, a prática de atividades de vários tipos e níveis de complexidade que lhe permitirão consolidar a aprendizagem e verificar se assimilou corretamente. Desta forma, consegue-se uma aprendizagem mais significativa e profunda, a assimilação do conhecimento de forma mais eficaz e duradoura, relacionar o que foi aprendido com a realidade dos desenvolvedores, promover autonomia e autoconhecimento, melhorar a análise, relacionamento e compreensão de conceitos, além de ajudar a exercitar uma infinidade de habilidades.

A aprendizagem entre pares é um dos elementos centrais da nossa metodologia, portanto, a cada aula iremos propor que trabalhe em mesas de trabalho com os seus colegas, ao longo do curso iremos variar a composição dos grupos para reforçar a cooperação. O que se propõe é uma mudança de visão sobre o curso em questão, o aluno não é mais considerado 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 das potencialidades de cada integrante para a melhoria do desempenho da equipe.

A explicação recíproca como eixo do cotidiano do trabalho não só facilita o aprendizado dos colegas, mas, sobretudo, potencializa a consolidação do conhecimento por parte de quem explica. Promovem-se a responsabilidade, a autonomia, a pró-atividade, tudo no quadro da cooperação. O que leva a ressignificar a experiência de aprendizagem que está ligada a emoções positivas.

O trabalho cooperativo permite relacionamentos responsáveis ​​e duradouros, aumenta a motivação e o comprometimento, além de promover o bom desenvolvimento cognitivo e social. A cooperação surge diante da dúvida. Se um aluno tem uma pergunta, ele pergunta a um membro disponível do grupo designado. Se a dúvida persistir, o facilitador é convocado. Se eles não resolverem, o facilitador pedirá a todos para cooperarem como uma equipe na resolução do conflito que levantou a dúvida. É assim que todos os membros da mesa vão debater, procurando a solução. Se ainda não conseguirem resolver, irão anotar 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, os alunos reconhecem 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 pode passar por essa função. O facilitador não é uma figura estática, mas cumpre um papel dinâmico e versátil. O facilitador é um aluno que mobiliza a realização dos objetivos comuns da equipe colocando em ação a cooperação. É quem partilha com a mesa o seu potencial a favor do resto da equipa e, por isso, promove a cooperação.

**Informação da matéria**

* Modalidade 100% remota.
* Número total de semanas: 9
* Número de encontros síncronos semanais: 3
* Aulas virtuais em nosso campus Playground: 18
* Número de aulas ao vivo: 18

**Requisitos de correlatividades**

Quais disciplinas o aluno deve ter sido aprovado previamente?

* Introdução à Informática (Bimestre 1, Ano 1)
* Programação Imperativa (Bimestre 1, Ano 1)

Quais disciplinas você estuda a seguir?

* Projeto Integrativo (Bimestre 4, Ano 1)
* Teste II (Bimestre 2, Ano 2)

**Modalidade de trabalho**

Nossa proposta educacional é pensada especialmente 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 vincula teoria e prática, de modo que ambas estão presentes o tempo todo.

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 reuniões online e ao vivo com o grupo de alunos e professores, às quais podemos nos associar 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 registo 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**

* Realizar as atividades do playground (80% concluído)
* Presença em encontros síncronos (90% de presença) \*
* Obter uma pontuação de 7 ou mais na avaliação final.
* Obter uma pontuação de 7 ou mais na nota final da disciplina.

**Conteúdos**

**Módulo 1: Fundamentos do Teste de Software**

Introdução ao processo de teste tradicional. Começando com a história do teste, o que um testador deve ter em mente em suas atividades diárias, o processo de teste formal e terminando com o conhecimento dos tipos e níveis de teste para saber quais testes realizar de acordo com o contexto do produto a ser testado.

**Aula 1: Primeiros passos**

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

**Aula 2: Fundamentos do teste**

* 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 3: Fundamentos do teste - Revisão**

* Fechamento da semana (Aulas 1 e 2)

**Módulo 2: Gestão de defeitos**

Um dos principais objetivos de todo testador é encontrar defeitos. Neste módulo, você aprenderá o que é um defeito, como relatá-lo e gerenciá-lo.

**Aula 4: Gestão de defeitos**

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

**Módulo 3: Design de testes**

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

**Aula 5: Design de testes**

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

**Aula 6: Defeitos e 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 4: Implementação de testes**

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

**Aula 8: Implementação e execução do teste**

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

**Aula 9: Testes - Revisão**

* Fechamento da semana (Aulas 7 e 8)

**Módulo 5: 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, eles 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 10: Testes estáticos e dinâmicos**

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

**Módulo 6: 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: Testes estáticos e dinâmicos - Organização do teste - Revisão**

* Fechamento da semana (Aulas 10 e 11)

**Aula 13: Avaliação I**

**Módulo 7: Introdução ao TDD**

Recentemente, a metodologia de *test driven development* (desenvolvimento orientado a testes) foi imposta como uma forma de trabalho porque os testes são necessários nas fases iniciais do projeto. Neste módulo, iniciaremos o caminho desse processo de desenvolvimento que consiste na codificação de testes, desenvolvendo e refatorando continuamente o código construído.

**Aula 14: Introdução ao teste de componente**

* Debugging
* Introdução ao teste de unidade

**Aula 15: Fechamento da semana**

* Fechamento da semana (Aulas 13 e 14)

**Aula 16: Teste de Componente**

* Teste de unidade
* Teste de cobertura de sentença
* Teste de cobertura de decisão
* Ferramentas de cobertura de teste
* Test driven development
* Mock e stub

**Módulo 8: 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 I

**Aula 18: Fechamento da semana II**

* Fechamento da semana (Aulas 16 e 17)

**Aula 19: Teste de API**

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

**Módulo 9: Introdução à Automação**

Neste módulo será apresentado ao aluno 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

**Aula 21: Fechamento da semana III**

* Fechamento da semana (Aulas 19 e 20)

**Aula 22: Automação de teste**

* Critérios para automatização e seleção de ferramentas
* Selenium WebDriver
* Criando nosso primeiro projeto
* Nosso primeiro teste

**Aula 23: Avaliação II**

**Aula 24: Fechamento da semana IV**

* Fechamento da semana (Aulas 22 e 23)

**Módulo 10: Full stack e DevOps tester**

Neste módulo continuaremos a aprender mais sobre o mundo da automação e apresentaremos as atividades relacionadas às posições de liderança, como líder de QA, release owner, gerente de QA. Iremos mencionar e mostrar as ferramentas DevOps atuais para a área de teste, enfatizando como elas melhoram e refinam todos os processos de QA.

**Aula 25: Full stack tester - Frameworks**

* Frameworks para automações mais comuns
* Documento de plano de teste
* Liberar documento de plano de teste

**Aula 26: DevOps tester - Teste contínuo**

* Entrega contínua e desenvolvimento contínuo de teste

**Aula 27: Full stack e DevOps tester - Revisão**

* Fechamento da semana (Aulas 23 e 24)