Advanced Master

Engenharia e Qualidade de Software

tech universidade tecnológica

tech universidade tecnológica

Advanced Master
Engenharia e Qualidade de Software

- Modalidade: **online**
- Duração: **2 anos**
- Certificado: **TECH Universidade Tecnológica**
- Horário: **no seu próprio ritmo**
- Provas: **online**

Acesso ao site: www.techtitute.com/br/informatica/advanced-master/advanced-master-engenharia-qualidade-software

Índice

01 Apresentação do programa — pág. 4

02 Por que estudar na TECH? — pág. 8

03 Plano de estudos — pág. 12

04 Objetivos de ensino — pág. 34

05 Oportunidades profissionais — pág. 40

06 Metodologia de estudo — pág. 44

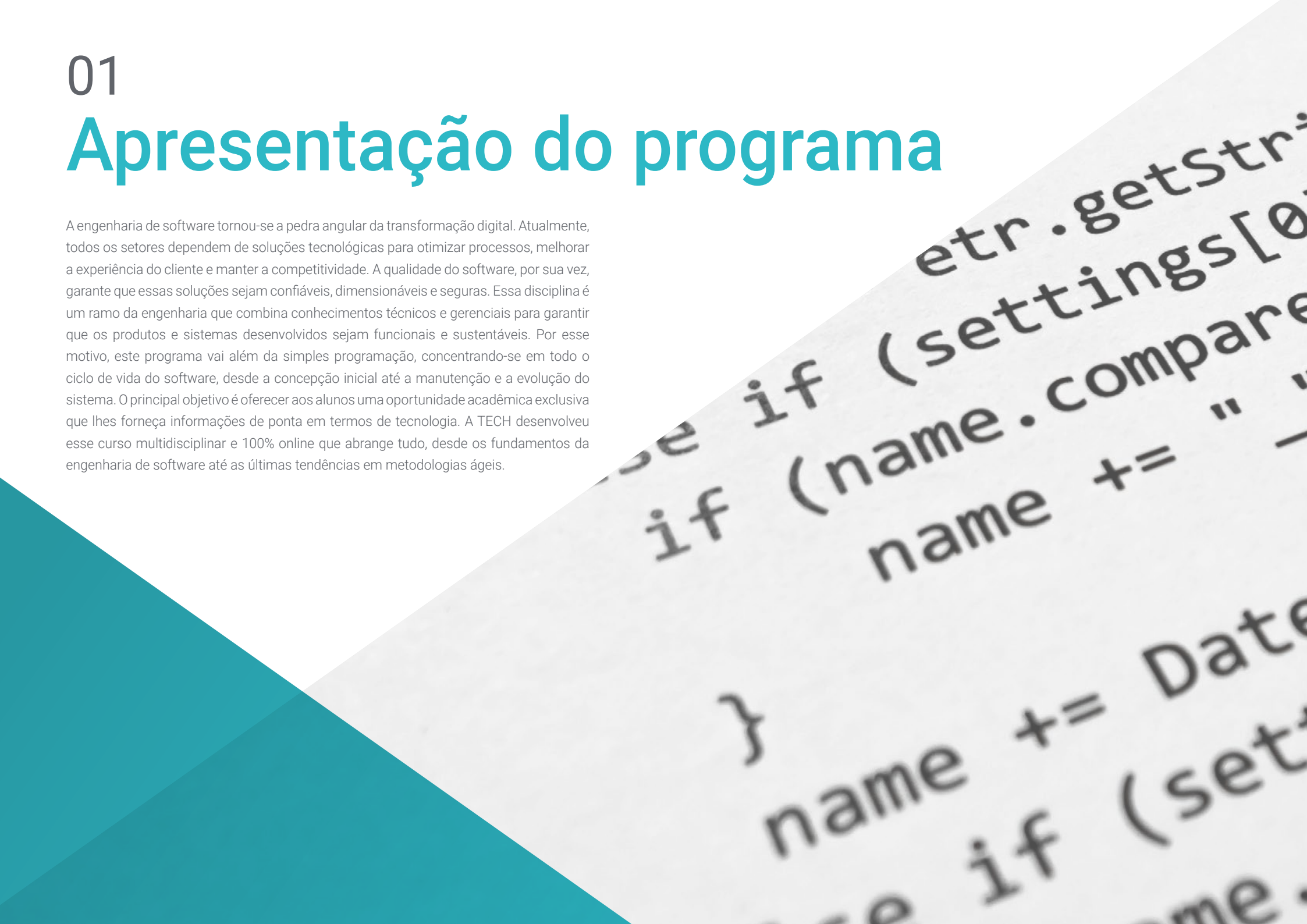
07 Equipe de professores — pág. 54

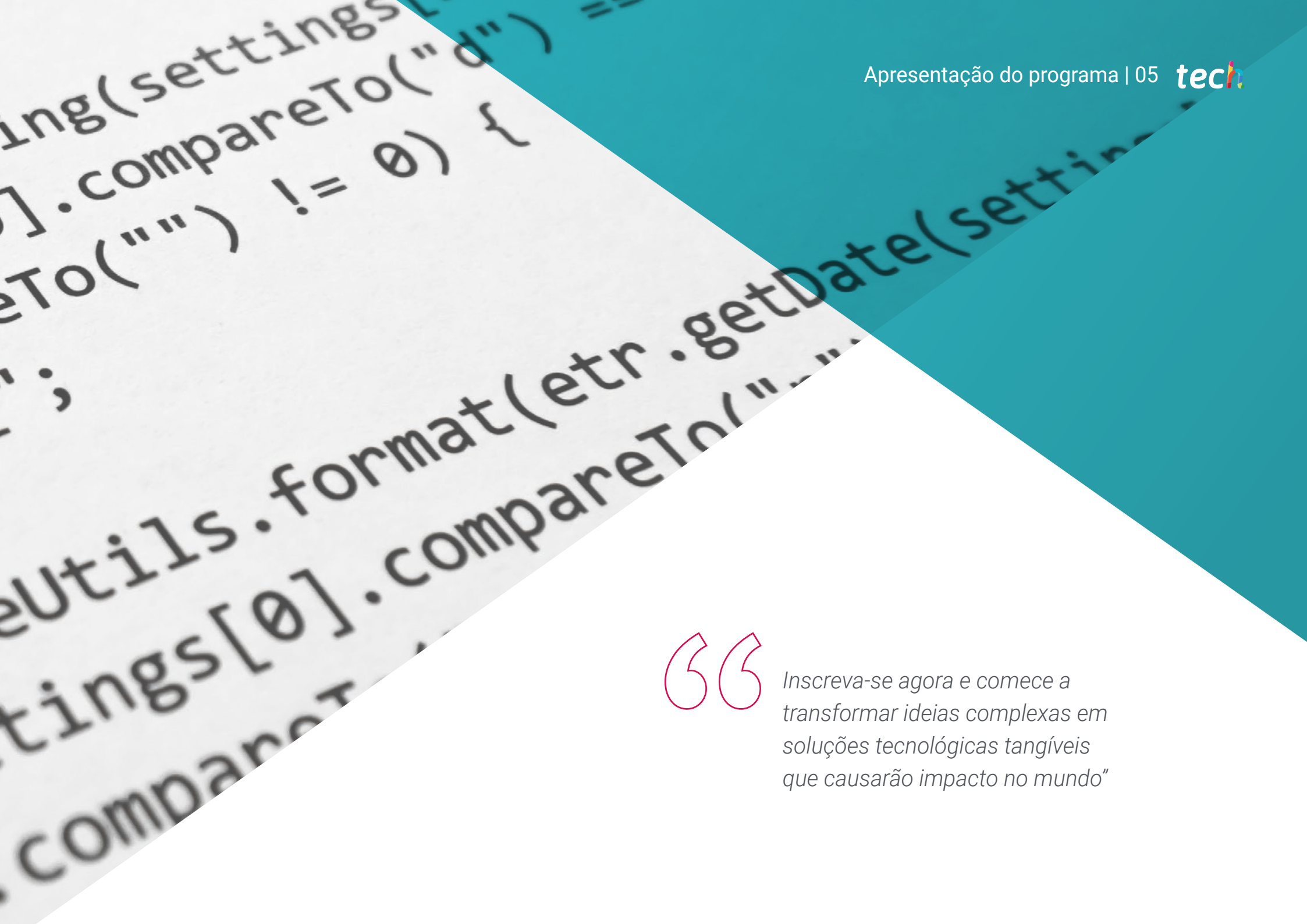
08 Certificação — pág. 60

01

Apresentação do programa

A engenharia de software tornou-se a pedra angular da transformação digital. Atualmente, todos os setores dependem de soluções tecnológicas para otimizar processos, melhorar a experiência do cliente e manter a competitividade. A qualidade do software, por sua vez, garante que essas soluções sejam confiáveis, dimensionáveis e seguras. Essa disciplina é um ramo da engenharia que combina conhecimentos técnicos e gerenciais para garantir que os produtos e sistemas desenvolvidos sejam funcionais e sustentáveis. Por esse motivo, este programa vai além da simples programação, concentrando-se em todo o ciclo de vida do software, desde a concepção inicial até a manutenção e a evolução do sistema. O principal objetivo é oferecer aos alunos uma oportunidade acadêmica exclusiva que lhes forneça informações de ponta em termos de tecnologia. A TECH desenvolveu esse curso multidisciplinar e 100% online que abrange tudo, desde os fundamentos da engenharia de software até as últimas tendências em metodologias ágeis.





Inscreva-se agora e comece a transformar ideias complexas em soluções tecnológicas tangíveis que causarão impacto no mundo”

A qualidade do software garante que os sistemas não apenas atendam aos requisitos funcionais, mas também sejam intuitivos, seguros e sustentáveis a longo prazo. Isso é particularmente relevante em setores críticos, como finanças, saúde ou transporte, onde as falhas podem ter consequências graves. Além disso, ao priorizar a qualidade, ele garante que as empresas possam se adaptar com agilidade aos constantes avanços tecnológicos e responder com eficácia às crescentes demandas do mercado.

Com o uso de metodologias como desenvolvimento ágil, *DevOps* e a implementação de padrões internacionais de qualidade, a engenharia de software garante a entrega de produtos em prazos mais curtos. Além disso, os custos controlados e um nível de qualidade que minimiza os erros críticos foram ampliados pela integração de tecnologias emergentes, como inteligência artificial, o *cloud computing* e a cibersegurança. Nesse contexto, o programa elaborado pela TECH tem como objetivo formar profissionais altamente qualificados em design, desenvolvimento, gerenciamento e garantia de qualidade de software. Para adquirir as habilidades necessárias, o programa de estudos do Advanced Master inclui os conceitos mais atualizados de gerenciamento de projetos de tecnologia e gerenciamento estratégico. Essa abordagem representa um valor agregado tanto para engenheiros que já ocupam cargos de responsabilidade e desejam atualizar seus conhecimentos quanto para aqueles que desejam liderar equipes e projetos pela primeira vez nesse campo.

Uma das principais vantagens desse programa é que ele será 100% online, o que elimina a necessidade de deslocamento e de se ajustar a horários específicos. Além disso, os alunos aprenderão através do método *Relearning*, que se adapta ao seu ritmo de estudo. Essa abordagem flexível é muito útil, pois permite que os alunos organizem suas obrigações diárias, sejam elas profissionais ou familiares, de forma eficiente, alcançando assim o desenvolvimento pleno.

Este **Advanced Master em Engenharia e Qualidade de Software** conta com o conteúdo mais completo e atualizado do mercado. Suas principais características são:

- Desenvolvimento de casos práticos apresentados por especialistas na área de informática
- O conteúdo gráfico, esquemático e extremamente útil, fornece informações científicas e práticas sobre as disciplinas essenciais para o exercício da profissão
- Exercícios práticos em que o processo de autoavaliação é realizado para melhorar a aprendizagem
- Sua ênfase especial em metodologias inovadoras em Engenharia de Software e Qualidade de Software
- Aulas teóricas, perguntas a especialistas, fóruns de discussão sobre temas controversos e trabalhos de reflexão individual
- Disponibilidade de acesso a todo o conteúdo a partir de qualquer dispositivo, fixo ou portátil, com conexão à Internet

Com a TECH, você não aprenderá apenas a desenvolver software, mas a criar sistemas que fazem a diferença na vida das pessoas e das empresas”

Domine as habilidades e ferramentas de engenharia mais avançadas com a metodologia de ensino mais inovadora do cenário acadêmico atual"

Aumente suas expectativas profissionais aprendendo 100% online, sem interferir em suas responsabilidades pessoais e familiares.

Torne-se um profissional de engenharia líder, pronto para aprender em qualquer lugar do mundo.

O corpo docente deste programa é formado por profissionais da área de informática, cuja experiência é somada a este curso, além de reconhecidos especialistas de sociedades conceituadas e universidades de prestígio

O conteúdo multimídia desenvolvido com a mais recente tecnologia educacional, oferece ao profissional uma aprendizagem situada e contextual, ou seja, um ambiente simulado que proporcionará um estudo imersivo e programado para capacitar em situações reais.

Este programa se fundamenta na Aprendizagem Baseada em Problemas, através da qual o aluno deverá resolver as diferentes situações de prática profissional que surgirem ao longo do programa. Para isso, o profissional contará com a ajuda de um inovador sistema de vídeo interativo, realizado por especialistas reconhecidos nesta área.

02

Por que estudar na TECH?

A TECH é a maior universidade digital do mundo. Com um impressionante catálogo de mais de 14.000 programas universitários, disponíveis em 11 idiomas, a TECH se posiciona como líder em empregabilidade, com uma taxa de inserção profissional de 99%. Além disso, conta com um vasto corpo docente formado por mais de 6.000 professores de prestígio internacional.

Estude na maior universidade digital do mundo e garanta seu sucesso profissional. O futuro começa na TECH”

A melhor universidade online do mundo de acordo com a FORBES

A conceituada revista Forbes, especializada em negócios e finanças, destacou a TECH como «a melhor universidade online do mundo». Foi o que afirmou recentemente em um artigo de sua edição digital, no qual faz referência à história de sucesso dessa instituição, «graças à oferta acadêmica que oferece, à seleção de seu corpo docente e a um método de aprendizagem inovador destinado a formar os profissionais do futuro».

A melhor equipe de professores top internacional

A equipe de professores da TECH é composta por mais de 6.000 profissionais de renome internacional. Professores, pesquisadores e executivos seniores de multinacionais, incluindo Isaiah Covington, técnico de desempenho do Boston Celtics; Magda Romanska, pesquisadora principal do Harvard MetaLAB; Ignacio Wistumba, presidente do departamento de patologia molecular translacional do MD Anderson Cancer Center; e D.W. Pine, diretor de criação da revista TIME, entre outros.

A maior universidade digital do mundo

A TECH é a maior universidade digital do mundo. A TECH é a maior universidade digital do mundo. Somos a maior instituição educacional, com o melhor e mais amplo catálogo educacional digital, 100% online, abrangendo a grande maioria das áreas do conhecimento. Oferecemos o maior número de cursos próprios, pós-graduações e graduações oficiais do mundo. No total, são mais de 14.000 programas universitários em onze idiomas diferentes, o que nos torna a maior instituição de ensino do mundo.

Os planos de estudos mais completos do panorama universitário

A TECH oferece os planos de estudos mais completos do cenário universitário, com programas que abrangem conceitos fundamentais e, ao mesmo tempo, os principais avanços científicos em suas áreas específicas. Além disso, esses programas são continuamente atualizados para garantir aos alunos a vanguarda acadêmica e as habilidades profissionais mais procuradas. Dessa forma, os programas da universidade proporcionam aos seus alunos uma vantagem significativa para impulsionar suas carreiras rumo ao sucesso.

Um método de aprendizado único

A TECH é a primeira universidade a utilizar o *Relearning* em todos os seus cursos. É a melhor metodologia de aprendizagem online, credenciada com certificações internacionais de qualidade de ensino, fornecidas por agências educacionais de prestígio. Além disso, esse modelo acadêmico disruptivo é complementado pelo "Método do Caso", configurando assim uma estratégia única de ensino online. Também são implementados recursos didáticos inovadores, incluindo vídeos detalhados, infográficos e resumos interativos.

A universidade online oficial da NBA

A TECH é a Universidade Online Oficial da NBA. Por meio de um acordo com a maior liga de basquete do mundo, oferece aos seus alunos programas universitários exclusivos, além de uma grande variedade de recursos educacionais voltados para o negócio da liga e outras áreas da indústria esportiva. Cada programa tem um plano de estudos único e conta com palestrantes convidados excepcionais: profissionais com trajetórias esportivas destacadas que compartilham suas experiências sobre os temas mais relevantes.

Líderes em empregabilidade

A TECH se consolidou como a universidade líder em empregabilidade. 99% dos seus alunos conseguem um emprego na área que estudaram em até um ano após a conclusão de qualquer programa da universidade. Um número semelhante obtém uma melhoria imediata em sua carreira. Isso é possível graças a uma metodologia de ensino baseada na aquisição de competências práticas, essenciais para o desenvolvimento profissional.

Google Partner Premier

A gigante da tecnologia Google concedeu à TECH o selo Google Partner Premier. Esse reconhecimento, disponível apenas para 3% das empresas no mundo, destaca a experiência eficaz, flexível e adaptada que a universidade oferece aos seus alunos. O reconhecimento não apenas credencia o máximo rigor, desempenho e investimento nas infraestruturas digitais da TECH, mas também coloca essa universidade como uma das empresas de tecnologia mais avançadas do mundo.

A Universidade mais bem avaliada por seus alunos

O site de avaliação Global score posicionou a TECH como a universidade mais bem avaliada do mundo por seus alunos. Esse portal de avaliações, o mais confiável e prestigiado, pois verifica e valida a autenticidade de cada opinião publicada, concedeu à TECH a sua classificação mais alta, 4,9 de 5, com base em mais de 1000 avaliações recebidas. Esses números colocam a TECH como referência absoluta de universidade internacional.

03

Plano de estudos

O plano de estudos do Advanced Master em Engenharia e Qualidade de Software foi projetado para oferecer uma especialização abrangente e avançada em todas as principais áreas da engenharia de software. Os primeiros módulos se concentram nos fundamentos, que vão desde o design de software e o gerenciamento de requisitos até as arquiteturas de tecnologia e as metodologias ágeis. À medida que o programa avança, os alunos se aprofundam em áreas mais especializadas, como automação de testes, integração contínua e garantia de qualidade. Além disso, são incorporadas disciplinas sobre gerenciamento de projetos de tecnologia, nas quais os participantes aprendem a gerenciar equipes multidisciplinares.

Este Advanced Master o prepara para ser o especialista que faz a diferença no setor de Engenharia e Qualidade de Software"

Módulo 1. Qualidade do Software Níveis de Desenvolvimento TRL

1.1. Elementos que influenciam na Qualidade do Software (I) A dívida técnica
- 1.1.1. A dívida técnica Causas e consequências
- 1.1.2. Qualidade do Software Princípios generais
- 1.1.3. Software sem princípios e com princípios de qualidade
  - 1.1.3.1. Consequências
  - 1.1.3.2. Necessidade de aplicar princípios de qualidade de software
- 1.1.4. Qualidade do Software Tipologia
- 1.1.5. Software de qualidade Características específicas

1.2. Elementos que influenciam na Qualidade do Software (II) Custos associados
- 1.2.1. Qualidade do Software Elementos influentes
- 1.2.2. Qualidade do Software Ideias equivocadas
- 1.2.3. Qualidade do Software Custos associados

1.3. Modelos de Qualidade de Software (I) Gestão do conhecimento
- 1.3.1. Modelos de Qualidade Geral
  - 1.3.1.1. Gestão da Qualidade Total
  - 1.3.1.2. Modelo Europeu de Excelência Empresarial (EFQM)
  - 1.3.1.3. Modelo Seis-sigma
- 1.3.2. Modelos da Gestão do conhecimento
  - 1.3.2.1. Modelo Dyba
  - 1.3.2.2. Modelo Seks
- 1.3.3. Fábrica de experiência e paradigma QIP
- 1.3.4. Modelos de Qualidade de Uso (25010)

1.4. Modelos de Qualidade de Software (III) Qualidade em dados, processos e modelos SEI
- 1.4.1. Modelo de Qualidade de Dados
- 1.4.2. Modelo de processo de software
- 1.4.3. Software & *Systems Process Engineering Metamodel Specification* (SPEM)
- 1.4.4. Modelos SEI
  - 1.4.4.1. CMMI
  - 1.4.4.2. SCAMPI
  - 1.4.4.3. IDEAL

1.5. Normas ISO de Qualidade de Software (I) Análise de padrões
- 1.5.1. Norma ISO 9000
  - 1.5.1.1. Norma ISO 9000
  - 1.5.1.2. Família ISO de normas de Qualidade (9000)
- 1.5.2. Outras normas ISO relacionadas à Qualidade
- 1.5.3. Normas de Modelagem de qualidade (ISO 2501)
- 1.5.4. Normas de Medida de qualidade (ISO 2502n)

1.6. Normas ISO de Qualidade de Software (II) Requisitos e avaliação
- 1.6.1. Normas sobre requisitos de Qualidade (2503n)
- 1.6.2. Normas sobre avaliação de Qualidade (2504n)
- 1.6.3. ISO/IEC 24744:2007

1.7. Níveis de desenvolvimento TRL I: Níveis 1 a 4
- 1.7.1. Nivel TRL
- 1.7.2. Nível 1: princípios básicos
- 1.7.3. Nível 2: conceito e/ou aplicação
- 1.7.4. Nível 3: função crítica analítica
- 1.7.5. Nível 4: validação de componentes em um ambiente de laboratório

1.8. Níveis de desenvolvimento do TRL (II) Níveis de 5 a 9
- 1.8.1. Nível 5: validação de componentes em um ambiente relevante
- 1.8.2. Nível 6: modelo de sistema/subsistema
- 1.8.3. Nível 7: demonstração em ambiente real
- 1.8.4. Nível 8: sistema completo e certificado
- 1.8.5. Nível 9: sucesso em ambiente real

1.9. Níveis de desenvolvimento TRL Usos
- 1.9.1. Exemplo de uma empresa com ambiente de laboratório
- 1.9.2. Exemplo de uma empresa I+D+i
- 1.9.3. Exemplo de uma empresa de I+D+i industrial
- 1.9.4. Exemplo de uma empresa mista laboratório-engenharia

1.10. Qualidade do Software Detalhes importantes
- 1.10.1. Detalhes metodológicos
- 1.10.2. Detalhes técnicos
- 1.10.3. Detalhes em gestão de projetos Software
  - 1.10.3.1. Qualidade dos sistemas informáticos
  - 1.10.3.2. Qualidade do Produto Software
  - 1.10.3.3. Qualidade do processo de Software

Módulo 2. Desenvolvimento de Projetos Software Documentação Funcional e Técnica

2.1. Gestão de projetos
- 2.1.1. Gestão de projetos na Qualidade do Software
- 2.1.2. Gerenciamento de projetos. Vantagens
- 2.1.3. Gerenciamento de projetos. Tipologia

2.2. Metodologia em gestão de projetos
- 2.2.1. Metodologia em gestão de projetos
- 2.2.2. Metodologia de projetos Tipologia
- 2.2.3. Metodologia em gestão de projetos Aplicação

2.3. Fase de identificação de requisitos
- 2.3.1. Identificação dos requisitos de um projeto
- 2.3.2. Gestão das reuniões de projetos
- 2.3.3. Documentação a fornecer

2.4. Modelo
- 2.4.1. Fase inicial
- 2.4.2. Fase de análise
- 2.4.3. Fase de construção
- 2.4.4. Fase de testes
- 2.4.5. Entrega

2.5. Modelo de dados a ser utilizado
- 2.5.1. Determinação do novo modelo de dados
- 2.5.2. Identificação do plano de migração de dados
- 2.5.3. Conjunto de dados

2.6. Impacto em outros projetos
- 2.6.1. Impacto de um projeto Exemplos

2.7. *MUST* do Projeto
- 2.7.1. *MUST* do Projeto
- 2.7.2. Identificação dos *MUST* do projeto
- 2.7.3. Identificação dos pontos de execução para a entrega de um projeto

2.8. A equipe responsável pela construção do projeto
- 2.8.1. Funções de atuar de acordo com o projeto
- 2.8.2. Contato com RH para contratação
- 2.8.3. Entrega e cronograma do projeto

2.9. Aspectos técnicos de um projeto de Software
- 2.9.1. Arquiteto do projeto Aspectos técnicos
- 2.9.2. Líderes técnicos
- 2.9.3. Construção do projeto Software
- 2.9.4. Avaliação da Qualidade do Código Sonar

2.10. Entrega do projeto
- 2.10.1. Análise funcional
- 2.10.2. Modelos de dados
- 2.10.3. Diagramas de estado
- 2.10.4. Documentação técnica

Módulo 3. *Testing* de Software Automação de Testes

3.1. Modelos de Qualidade do Software
- 3.1.1. Qualidade do produto
- 3.1.2. Qualidade do processo
- 3.1.3. Qualidade de uso

3.2. Qualidade do processo
- 3.2.1. Qualidade do processo
- 3.2.2. Modelos de maturidade
- 3.2.3. Norma ISO 15504
  - 3.2.3.1. Propósitos
  - 3.2.3.2. Contexto
  - 3.2.3.3. Fases

3.3. Normas ISO/IEC 15504
- 3.3.1. Categorias de processo
- 3.3.2. Processo de desenvolvimento. Exemplo
- 3.3.3. Fragmento de perfil
- 3.3.4. Fases

3.4. CMMI (*Capability Maturity Model Integration*)
- 3.4.1. CMMI Integração de Modelos de Maturidade de capacidades
- 3.4.2. Modelos e áreas Tipologia
- 3.4.3. Áreas de processo
- 3.4.4. Níveis de capacidade
- 3.4.5. Gestão de processos
- 3.4.6. Gestão de projetos

3.5. Gestão de mudanças e repositórios
- 3.5.1. Gestão de troca de software
  - 3.5.1.1. Item de configuração Integração contínua
  - 3.5.1.2. Linhas
  - 3.5.1.3. Fluxogramas
  - 3.5.1.4. *Branches (Branch)*
- 3.5.2. Repositório
  - 3.5.2.1. Controle de versão
  - 3.5.2.2. Equipamento de trabalho e uso do repositório
  - 3.5.2.3. Integração contínua no repositório

3.6. *Team Foundation Server* (TFS)
- 3.6.1. Instalação e configuração
- 3.6.2. Criação de um projeto de equipamento
- 3.6.3. Incorporação de conteúdo no controle do código fonte
- 3.6.4. *TFS on Cloud*

3.7. *Testing*
- 3.7.1. Motivação para a realização de testes
- 3.7.2. Testes de verificação
- 3.7.3. Teste Beta
- 3.7.4. Implementação e manutenção

3.8. Teste de carga
- 3.8.1. *Load testing*
- 3.8.2. Teste com *LoadView*
- 3.8.3. Teste com K6 *Cloud*
- 3.8.4. Teste com *Loader*

3.9. Testes unitários de estresse e resistência
- 3.9.1. Motivação para testes unitários
- 3.9.2. Ferramentas para *Unit Testing*
- 3.9.3. Motivação para testes de estresse
- 3.9.4. Teste usando o *StressTesting*
- 3.9.5. Motivação para testes de Resistência
- 3.9.6. Teste usando o *LoadRunner*

3.10. A escalabilidade. Projeto de Software Escalável
- 3.10.1. Escalabilidade e Arquitetura de Software
- 3.10.2. Independência entre camadas
- 3.10.3. O acoplamento entre camadas Padrões de arquitetura

Módulo 4. Metodologias de Gestão de Projetos Software Metodologias Waterfall comparadas com Metodologias Ágeis

4.1. Metodologia Waterfall
- 4.1.1. Metodologia Waterfall
- 4.1.2. Metodologia Waterfall Influência na Qualidade do Software
- 4.1.3. Metodologia Waterfall Exemplos

4.2. Metodologia Agile
- 4.2.1. Metodologia Agile
- 4.2.2. Metodologia Agile Influência na Qualidade do Software
- 4.2.3. Metodologia Agile Exemplos

4.3. Metodologia SCRUM
- 4.3.1. Metodologia SCRUM
- 4.3.2. Manifesto SCRUM
- 4.3.3. Aplicação do SCRUM

4.4. Painel Kanban
- 4.4.1. Método Kanban
- 4.4.2. Painel Kanban
- 4.4.3. Painel Kanban Exemplos de aplicação

4.5. Gestão de projetos em Waterfall
- 4.5.1. Fases em um projeto
- 4.5.2. Visão em um projeto Waterfall
- 4.5.3. Entregas a considerar

4.6. Gestão de projetos em SCRUM
- 4.6.1. Fases em um projeto SCRUM
- 4.6.2. Visão em um projeto SCRUM
- 4.6.3. Entregas a considerar

4.7. Waterfall vs SCRUM. Comparativa
- 4.7.1. Abordagem de um projeto piloto
- 4.7.2. Projeto aplicando Waterfall. Exemplo
- 4.7.3. Projeto aplicando SCRUM Exemplo

4.8. Visão do cliente
- 4.8.1. Documentos em um Waterfall
- 4.8.2. Documentos em um SCRUM
- 4.8.3. Comparativa

4.9. Estrutura do Kanban
- 4.9.1. Histórias de usuário
- 4.9.2. Backlog
- 4.9.3. Análise do Kanban

4.10. Projetos híbridos
- 4.10.1. Construção do projeto
- 4.10.2. Gestão de projetos
- 4.10.3. Entregas a considerar

Módulo 5.TDD (*Test Driven Developement*). Projeto de Software guiado por Testes

5.1. TDD *Test Driven Development*
- 5.1.1. TDD *Test Driven Development*
- 5.1.2. TDD Influência do TDD sobre a Qualidade
- 5.1.3. Projeto e desenvolvimento baseado em testes Exemplos

5.2. Ciclo de TDD
- 5.2.1. Escolha de um requisito
- 5.2.2. Realização de testes Tipologia
  - 5.2.2.1. Testes unitários
  - 5.2.2.2. Teste de integração
  - 5.2.2.3. Testes *End To End*
- 5.2.3. Verificação do teste Falhas
- 5.2.4. Criação de Implementação
- 5.2.5. Execução de testes automatizados
- 5.2.6. Eliminação de duplicação
- 5.2.7. Atualização de listas de requisitos
- 5.2.8. Repetição do ciclo TDD
- 5.2.9. Ciclo TDD Exemplo teórico e prático

5.3. Estratégias de implementação de TDD
- 5.3.1. Falsa implementação
- 5.3.2. Implementação triangular
- 5.3.3. Implementação óbvia

5.4. TDD Uso Vantagens e Desvantagens
- 5.4.1. Vantagens do uso
- 5.4.2. Limitações de uso
- 5.4.3. Equilíbrio de qualidade na implementação

5.5. TDD Boas práticas
- 5.5.1. Regras TDD
- 5.5.2. Regra 1: Realizar um teste prévio que falhe antes de codificar na produção
- 5.5.3. Regra 2: não escrever mais de um teste unitário
- 5.5.4. Regra 3: não escrever mais código do que o necessário
- 5.5.5. Erros e anti-padrões a serem evitados em TDD

5.6. Simulação de projeto real para uso de TDD (I)
- 5.6.1. Descrição geral do projeto (Empresa A)
- 5.6.2. Implementação do TDD
- 5.6.3. Exercícios propostos
- 5.6.4. Exercícios *Feedback*

5.7. Simulação de projeto real para uso de TDD (II)
- 5.7.1. Descrição geral do projeto (Empresa B)
- 5.7.2. Implementação do TDD
- 5.7.3. Exercícios propostos
- 5.7.4. Exercícios *Feedback*

5.8. Simulação de projeto real para uso de TDD (III)
- 5.8.1. Descrição geral do projeto (Empresa C)
- 5.8.2. Implementação do TDD
- 5.8.3. Exercícios propostos
- 5.8.4. Exercícios *Feedback*

5.9. Alternativas a TDD *Test Driven Development*
- 5.9.1. TCR (*Test Commit Revert*)
- 5.9.2. BDD (*Behavior Driven Development*)
- 5.9.3. ATDD (*Acceptance Test Driven Development*)
- 5.9.4. TDD Comparativa teórica

5.10. TDD TCR, BDD y ATDD. Comparação prática
- 5.10.1. Definição do problema
- 5.10.2. Resolução com TCR
- 5.10.3. Resolução com BDD
- 5.10.4. Resolução com ATDD

Módulo 6. DevOps Gestão de Qualidade do Software

6.1. DevOps Gestão de Qualidade do Software
- 6.1.1. DevOps
- 6.1.2. DevOps e Qualidade do Software
- 6.1.3. DevOps Benefícios da Cultura DevOps

6.2. DevOps Relação com Agile
- 6.2.1. Entrega acelerada
- 6.2.2. Qualidade
- 6.2.3. Redução de custos

6.3. Implementação de DevOps
- 6.3.1. Identificação de problemas
- 6.3.2. Implantação em uma companhia
- 6.3.3. Métricas de implantação

6.4. Ciclo de entrega do Software
- 6.4.1. Métodos de projeto
- 6.4.2. Convênios
- 6.4.3. Folha de rota

6.5. Desenvolvimento do código livre de erros
- 6.5.1. Código de manutenção
- 6.5.2. Modelos de desenvolvimento
- 6.5.3. *Testing* de código
- 6.5.4. Desenvolvimento do Software à nível de código. Boas práticas

6.6. Automatização
- 6.6.1. Automatização Tipos de provas
- 6.6.2. Custo de automatização e manutenção
- 6.6.3. Automatização Mitigação de erros

6.7. Implantações
- 6.7.1. Avaliação de objetivos
- 6.7.2. Projeto de um processo automático e adaptado
- 6.7.3. Retroalimentação e capacidade de resposta

6.8. Gestão de Incidentes
- 6.8.1. Preparação para incidentes
- 6.8.2. Análise e resolução de incidentes
- 6.8.3. Como evitar erros futuros

6.9. Automatização da implantação
- 6.9.1. Preparação para implantações automáticas
- 6.9.2. Avaliação da saúde do processo automático
- 6.9.3. Métricas e capacidade de retorno

6.10. Boas práticas Evolução do DevOps
- 6.10.1. Guia de boas práticas aplicando o DevOps
- 6.10.2. DevOps Metodologia para a equipe
- 6.10.3. Evitando nichos

Módulo 7. DevOps e Integração Contínua Soluções Práticas Avançadas no Desenvolvimento de Software

7.1. Fluxo de entrega de Software
- 7.1.1. Identificação de atores e artefatos
- 7.1.2. Projeto do fluxo de entrega de Software
- 7.1.3. Fluxo de entrega do Software Requisitos entre etapas

7.2. Automatização de processos
- 7.2.1. Integração contínua
- 7.2.2. Implantação contínua
- 7.2.3. Configuração de ambientes e gestão de segredos

7.3. Pipelines declarativos
- 7.3.1. Diferenças entre pipelines tradicionais, como código e declarativos
- 7.3.2. Pipelines declarativos
- 7.3.3. Pipelines declarativos em Jenkins
- 7.3.4. Comparação de fornecedores de integração contínua

7.4. Portas de qualidade e retroalimentação enriquecida
- 7.4.1. Portas de qualidade
- 7.4.2. Padrões de qualidade com portas de qualidade Manutenção
- 7.4.3. Requisitos de negócio em pedidos de integração

7.5. Gestão de Artefatos
- 7.5.1. Artefatos e ciclo de vida
- 7.5.2. Sistemas de armazenamento e gestão de artefatos
- 7.5.3. Segurança na gestão de artefatos

7.6. Implantação contínua
- 7.6.1. Desdobramento contínuo como containers
- 7.6.2. Implantação contínua com o PaaS

7.7. Melhoria do tempo de execução do Pipeline: análise estática e *Git Hooks*
- 7.7.1. Análise estática
- 7.7.2. Regras de estilo do código
- 7.7.3. *Git Hookse Tests unitários*
- 7.7.4. O impacto da infraestrutura

7.8. Vulnerabilidades em containers
- 7.8.1. Vulnerabilidades em containers
- 7.8.2. Digitalização de imagens
- 7.8.3. Relatórios periódicos e alertas

Módulo 8. Design de Bases de Dados (BD) Normalização e Rendimento Qualidade de software

8.1. Projeto da base de dados
- 8.1.1. Bases de Dados Tipologia
- 8.1.2. Bases de dados utilizadas atualmente
  - 8.1.2.1. Relacional
  - 8.1.2.2. Chave-Valor
  - 8.1.2.3. Baseado em grafos
- 8.1.3. Qualidade dos dados

8.2. Projeto do Modelo de Entidade-Relacionamento (I)
- 8.2.1. Modelo de Entidade-Relacionamento Qualidade e documentação
- 8.2.2. Entidades
  - 8.2.2.1. Entidade forte
  - 8.2.2.2. Entidade fraca
- 8.2.3. Atributos
- 8.2.4. Conjunto de relacionamentos
  - 8.2.4.1. 1 para 1
  - 8.2.4.2. 1 para muitos
  - 8.2.4.3. Muitos para 1
  - 8.2.4.4. Muitos para muitos
- 8.2.5. Pinos
  - 8.2.5.1. Chave primária
  - 8.2.5.2. Chave estrangeira
  - 8.2.5.3. Chave primária entidade fraca
- 8.2.6. Restrições
- 8.2.7. Cardinalidade
- 8.2.8. Herança.
- 8.2.9. Agregação

8.3. Modelo Entidade-Relacionamento (II) Ferramentas
- 8.3.1. Modelo Entidade-Relacionamento Ferramentas
- 8.3.2. Modelo Entidade-Relacionamento Exemplos práticos
- 8.3.3. Modelo Entidade-Relacionamento viáveis
  - 8.3.3.1. Amostra Visual
  - 8.3.3.2. Amostra em representação de tabela

8.4. Normalização da base de dados (DB) (I) Considerações na Qualidade do Software
- 8.4.1. Normalização da BD e Qualidade
- 8.4.2. Dependência
  - 8.4.2.1. Dependência funcional
  - 8.4.2.2. Propriedades da dependência funcional
  - 8.4.2.3. Propriedades deduzidas
- 8.4.3. Pinos

8.5. Normalização da base de dados (DB) (II) Formas Normais e Regras do Codd
- 8.5.1. Formulários normais
  - 8.5.1.1. Primeira Forma Normal (1FN)
  - 8.5.1.2. Segunda Forma Normal (2FN)
  - 8.5.1.3. Terceira Forma Normal (3FN)
  - 8.5.1.4. Forma normal Boyce-Codd (FNBC)
  - 8.5.1.5. Quarta Forma Normal (4FN)
  - 8.5.1.6. Quint Forma Normal (5FN)
- 8.5.2. Regras Codd
  - 8.5.2.1. Regra 1: Informação
  - 8.5.2.2. Regra 2: acesso garantido
  - 8.5.2.3. Regra 3: tratamento sistemático de valores nulos
  - 8.5.2.4. Regra 4: descrição da base de dados
  - 8.5.2.5. Regra 5: sub-linguagem integral
  - 8.5.2.6. Regra 6: atualização de visualizações
  - 8.5.2.7. Regra 7: Inserir e atualizar
  - 8.5.2.8. Regra 8: independência física
  - 8.5.2.9. Regra 9: independência lógica
  - 8.5.2.10. Regra 10: independência da integridade
    - 8.5.2.10.1. Regras de integridade
  - 8.5.2.11. Regra 11: Distribuição
  - 8.5.2.12. Regra 12: não subversão
- 8.5.3. Exemplos práticos

8.6. Armazém de dados/Sistema OLAP
- 8.6.1. Armazém de dados
- 8.6.2. Tabela Fato
- 8.6.3. Tabela de dimensões
- 8.6.4. Criação do sistema OLAP Ferramentas

8.7. Rendimento da base de dados (DB)
- 8.7.1. Otimização de índices
- 8.7.2. Otimização de consultas
- 8.7.3. Particionamento de tabelas

8.8. Simulação de projeto real para Design BD (I)
- 8.8.1. Descrição geral do projeto (Empresa A)
- 8.8.2. Aplicação do projeto de Base de Dados
- 8.8.3. Exercícios propostos
- 8.8.4. Exercícios propostos *Feedback*

8.9. Simulação de projeto real para Design BD (II)
- 8.9.1. Descrição geral do projeto (Empresa B)
- 8.9.2. Aplicação do projeto de Base de Dados
- 8.9.3. Exercícios propostos
- 8.9.4. Exercícios propostos. *Feedback*

8.10. Relevância da otimização da BBDD na Qualidade do Software
- 8.10.1. Otimização do projeto
- 8.10.2. Otimização do código de consulta
- 8.10.3. Otimização do código de procedimentos armazenados
- 8.10.4. Influência dos *Triggers* na Qualidade do Software Recomendações de uso.

Módulo 9. Projeto de Arquitetura Escalável A Arquitetura no Ciclo de Vida do Software

9.1. Projeto de Arquitetura Escalável (I)
- 9.1.1. Arquiteturas Escaláveis
- 9.1.2. Princípios de uma arquitetura escalável
  - 9.1.2.1. Confiável
  - 9.1.2.2. Escalabilidade
  - 9.1.2.3. Manutenção
- 9.1.3. Tipos de escalabilidade
  - 9.1.3.1. Vertical
  - 9.1.3.2. Horizontal (Scalling-out)
  - 9.1.3.3. Combinada

9.2. Arquiteturas DDD (*Domain-Driven Design*)
- 9.2.1. O modelo DDD Orientação ao domínio
- 9.2.2. Camadas, compartilhamento de responsabilidades e padrões do projeto
- 9.2.3. Desacoplamento como base para a qualidade

9.3. Projeto de Arquitetura Escalável (II) Benefícios, limitações e estratégias do projeto
- 9.3.1. Arquitetura Escalável Benefícios
- 9.3.2. Arquitetura Escalável Limites
- 9.3.3. Estratégias para o desenvolvimento de arquiteturas escaláveis (Tabelas descritiva)

9.4. Ciclo de Qualidade de Software (I) Fases
- 9.4.1. Ciclo de vida do Software
  - 9.4.1.1. Etapa de planejamento
  - 9.4.1.2. Fase de análise
  - 9.4.1.3. Etapa do projeto
  - 9.4.1.4. Etapa da implementação
  - 9.4.1.5. Etapa de testes
  - 9.4.1.6. Etapa de instalação/ implantação
  - 9.4.1.7. Etapa de uso e manutenção

9.5. Modelos de Ciclo de vida do Software
- 9.5.1. Modelo em cascata
- 9.5.2. Modelo repetitivo
- 9.5.3. Modelo em espiral
- 9.5.4. Modelo Big Bang

9.6. Ciclo de Qualidade do Software (II) Automatização
- 9.6.1. Ciclo de vida do desenvolvimento do software. Soluções
  - 9.6.1.1. Integração e desenvolvimento contínuo (CI/CD)
  - 9.6.1.2. Metodologia Agile
  - 9.6.1.3. DevOps / Operações de Produção
- 9.6.2. Tendências futuras
- 9.6.3. Exemplos práticos

9.7. Arquitetura no ciclo de vida do Software
- 9.7.1. Benefícios
- 9.7.2. Limites
- 9.7.3. Ferramentas

9.8. Simulação de projeto real para projeto de arquitetura Software (I)
- 9.8.1. Descrição geral do projeto (Empresa A)
- 9.8.2. Aplicação do projeto de arquitetura do Software
- 9.8.3. Exercícios propostos
- 9.8.4. Exercícios propostos *Feedback*

9.9. Simulação do projeto real para projeto de arquitetura Software (II)
- 9.9.1. Descrição geral do projeto (Empresa B)
- 9.9.2. Aplicação do projeto de arquitetura do Software
- 9.9.3. Exercícios propostos
- 9.9.4. Exercícios propostos *Feedback*

9.10. Simulação do projeto real para projeto de arquitetura Software (III)
- 9.10.1. Descrição geral do projeto (Empresa C)
- 9.10.2. Aplicação do projeto de arquitetura do Software
- 9.10.3. Exercícios propostos
- 9.10.4. Exercícios propostos. *Feedback*

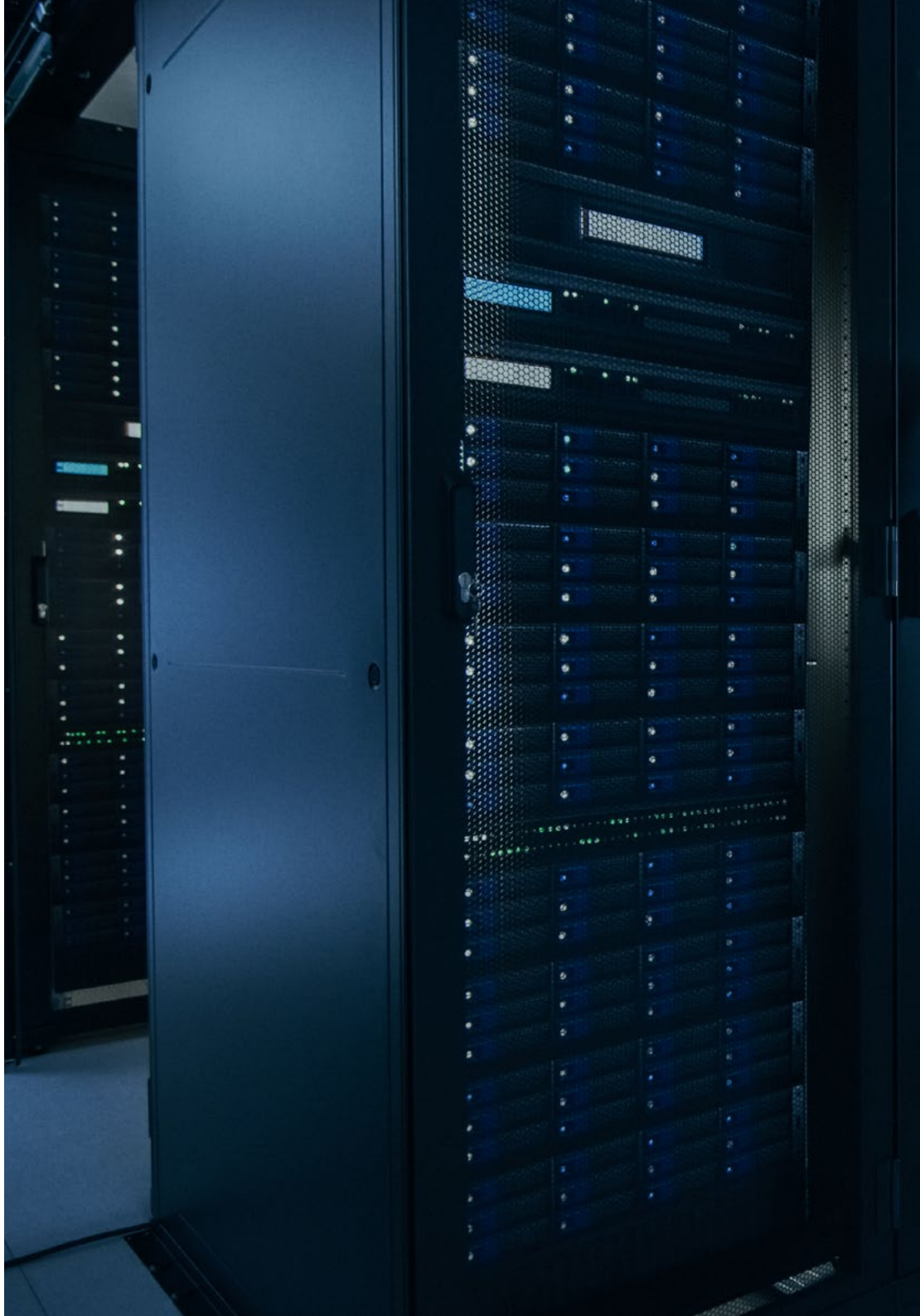
Módulo 10. Critérios de Qualidade ISO, IEC 9126 Métricas de Qualidade do Software

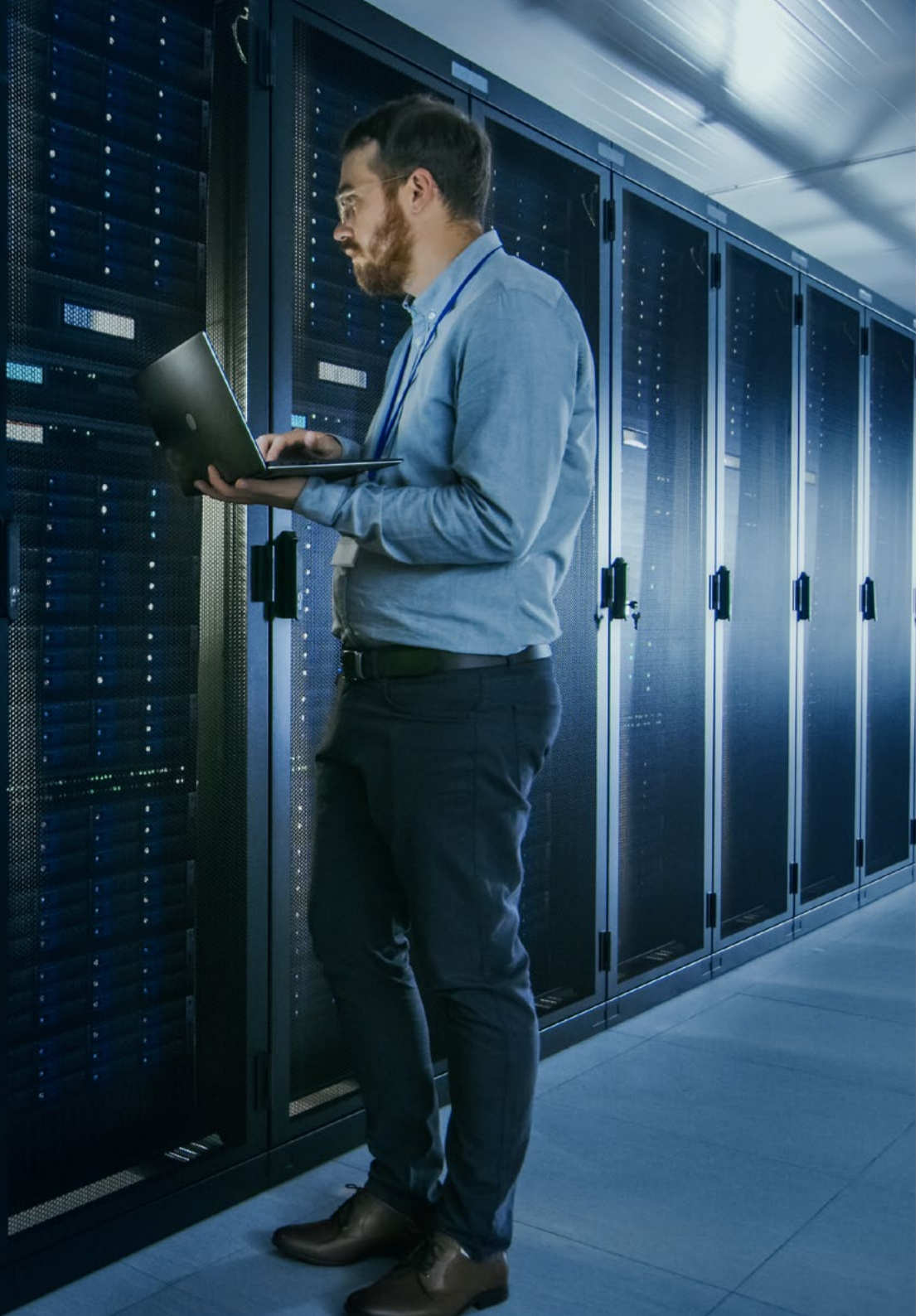
10.1. Critérios de qualidade Norma ISO, IEC 9126
- 10.1.1. Critérios de Qualidade
- 10.1.2. Qualidade do Software Justificativa Norma ISO, IEC 9126
- 10.1.3. Medição da qualidade do Software como um indicador chave

10.2. Critérios de qualidade do software Características
- 10.2.1. Confiabilidade
- 10.2.2. Funcionalidade
- 10.2.3. Eficiência
- 10.2.4. Usabilidade
- 10.2.5. Capacidade de manutenção
- 10.2.6. Portabilidade

10.3. Norma ISO, IEC 9126 (I) Apresentação
- 10.3.1. Descrição da Norma ISO, IEC 9126
- 10.3.2. Funcionalidade
- 10.3.3. Confiabilidade

10.3.4. Usabilidade
10.3.5. Capacidade de manutenção
10.3.6. Portabilidade
10.3.7. Qualidade em uso
10.3.8. Métricas de Qualidade de Software
10.3.9. Métricas de Qualidade em ISO 9126
10.4. Norma ISO, IEC 9126 (II) Modelos McCall e Boehm
10.4.1. Modelo McCall: Fatores de qualidade
10.4.2. Modelo Boehm
10.4.3. Nível Intermediário Características
10.5. Métricas de Qualidade de Software (I) Elementos
10.5.1. Medida
10.5.2. Métricas
10.5.3. Índices
10.5.3.1. Tipos de indicadores
10.5.4. Medidas e modelos
10.5.5. Alcance das métricas do Software
10.5.6. Classificação das Métricas do Software
10.6. Medidas de Qualidade do Software (II) Prática de medição
10.6.1. Coleta de dados métricos
10.6.2. Medição dos atributos internos do produto
10.6.3. Medição dos atributos externos do produto
10.6.4. Medição de recursos
10.6.5. Métricas para sistemas orientados a objetos
10.7. Projeto de um indicador único de qualidade do Software
10.7.1. Indicador único como um qualificador global
10.7.2. Desenvolvimento do indicador, justificação e aplicação
10.7.3. Exemplos de aplicação Necessidade de conhecer os detalhes
10.8. Simulação de projeto real para Medidas de Qualidade (I)
10.8.1. Descrição geral do projeto (Empresa A)
10.8.2. Aplicação da medição de qualidade
10.8.3. Exercícios propostos
10.8.4. Exercícios propostos *Feedback*





10.9. Simulação do projeto real para Medidas de Qualidade (II)
- 10.9.1. Descrição geral do projeto (Empresa B)
- 10.9.2. Aplicação da medição de qualidade
- 10.9.3. Exercícios propostos
- 10.9.4. Exercícios propostos *Feedback*

10.10. Simulação do projeto real para Medidas de Qualidade (III)
- 10.10.1. Descrição geral do projeto (Empresa C)
- 10.10.2. Aplicação da medição de qualidade
- 10.10.3. Exercícios propostos
- 10.10.4. Exercícios propostos *Feedback*

Módulo 11. Metodologias, Desenvolvimento e Qualidade em Engenharia de Software

11.1. Desenvolvimento de software baseado em modelos
- 11.1.1. A necessidade de
- 11.1.3. Modelagem de objetos
- 11.1.4. UML
- 11.1.5. Ferramentas CASE

11.2. Modelagem de aplicações e padrões de design com UML
- 11.2.1. Modelagem avançada de requisitos
- 11.2.2. Modelagem estática avançada
- 11.2.3. Modelagem dinâmica avançada
- 11.2.4. Modelagem de componentes
- 11.2.5. Introdução aos padrões de design com UML
- 11.2.6. Adapter
- 11.2.7. Factory
- 11.2.8. Singleton
- 11.2.9. Strategy
- 11.2.10. Composite
- 11.2.11. Facade
- 11.2.12. Observer

11.3. Engenharia orientada por modelos
- 11.3.1. Introdução
- 11.3.2. Metamodelagem de sistemas
- 11.3.3. MDA
- 11.3.4. DSL
- 11.3.5. Refinamento de modelos com OCL
- 11.3.6. Transformações de modelos

11.4. Ontologias em Engenharia de Software
- 11.4.1. Introdução
- 11.4.2. Engenharia ontológica
- 11.4.3. Aplicação de Ontologias em Engenharia de Software

Módulo 12. Gerenciamento de projetos de Software

12.1. Gestão dos *stakeholders* e de alcance
- 12.1.1. Identificação das partes interessadas
- 12.1.2. Desenvolver um plano para a gestão das partes interessadas
- 12.1.3. Gerenciamento do engajamento das partes interessadas
- 12.1.4. Monitoramento do engajamento das partes interessadas
- 12.1.5. O objetivo do projeto
- 12.1.6. A gestão de alcance e seu plano
- 12.1.7. Requisitos para a coleta
- 12.1.8. Definir a declaração do escopo
- 12.1.9. Criar o WBS (EDT)
- 12.1.10. Verificar e controlar o escopo

12.2. Desenvolvimento do cronograma
- 12.2.1. A gestão de Tempo e seu plano
- 12.2.2. Definir as atividades
- 12.2.3. Sequenciamento de atividades
- 12.2.4. Recursos estimados de atividades
- 12.2.5. Estimativa da duração as atividades
- 12.2.6. Desenvolvimento da cronologia e cálculo do caminho crítico
- 12.2.7. Controle do horário

12.3. Desenvolvimento do orçamento e resposta aos riscos
- 12.3.1. Estimativa de custos
- 12.3.2. Desenvolver o orçamento e a curva S
- 12.3.3. Controle de custos e método do valor agregado
- 12.3.4. Os conceitos de risco
- 12.3.5. Como fazer uma análise de risco
- 12.3.6. Desenvolvimento do plano de resposta

12.4. Comunicação e recursos humanos
- 12.4.1. Planejamento da gestão das comunicações
- 12.4.2. Análise das exigências de comunicação
- 12.4.3. Tecnologia das comunicações
- 12.4.4. Modelos de comunicação
- 12.4.5. Métodos de comunicação
- 12.4.6. Plano de gestão de comunicações
- 12.4.7. Gerenciando as comunicações
- 12.4.8. Gestão de recursos humanos
- 12.4.9. Principais atores e seus papéis nos projetos
- 12.4.10. Tipos de organizações
- 12.4.11. Organização do projeto
- 12.4.12. A equipe de trabalho

12.5. Aquisições
- 12.5.1. O processo de Compras
- 12.5.2. Planejamento
- 12.5.3. Busca de fornecedores e solicitação de propostas
- 12.5.4. Adjudicação do contrato
- 12.5.5. Administração de contratos
- 12.5.6. Os contratos
- 12.5.7. Tipos de contrato
- 12.5.8. Negociação de contratos

12.6. Implementação, monitoramento, controle e fechamento
- 12.6.1. Grupos de Processo:
- 12.6.2. Implementação do projeto
- 12.6.3. Monitoramento e controle de projetos
- 12.6.4. Encerramento do projeto

12.7. Responsabilidade profissional
- 12.7.1. Responsabilidade profissional
- 12.7.2. Características da responsabilidade social e profissional
- 12.7.3. Código de ética do líder do projeto
- 12.7.4. Responsabilidades PMP®
- 12.7.5. Exemplos de responsabilidade
- 12.7.6. Benefícios da profissionalização

Módulo 13. Plataformas de desenvolvimento de software

13.1. Introdução ao desenvolvimento de aplicações
- 13.1.1. Aplicações desktop
- 13.1.2. Linguagem de programação
- 13.1.3. Ambientes de desenvolvimento integrado
- 13.1.4. Aplicações Web
- 13.1.5. Aplicações móveis
- 13.1.6. Aplicações de nuvens

13.2. Desenvolvimento de aplicações e interface gráfica em Java
- 13.2.1. Ambientes de desenvolvimento integrado para Java
- 13.2.2. Principais IDEs para Java
- 13.2.3. Introdução à plataforma de desenvolvimento Eclipse
- 13.2.4. Introdução à plataforma de desenvolvimento NetBeans
- 13.2.5. Modelo de visualização do controlador para interfaces gráficas de usuário
- 13.2.6. Projetar uma interface gráfica no Eclipse
- 13.2.7. Projetando uma interface gráfica no NetBeans

13.3. Depuração e testes em Java
- 13.3.1. Teste e depuração de programas Java
- 13.3.2. Depuração no Eclipse
- 13.3.3. Depuração no NetBeans

13.4. Desenvolvimento de aplicações e interface gráfica em. NET
- 13.4.1. Net Framework
- 13.4.2. Componentes de plataforma de desenvolvimento. NET
- 13.4.3. Visual Studio .NET
- 13.4.4. Ferramentas GUI .NET
- 13.4.5. A GUI com Windows Presentation Foundation
- 13.4.6. Depuração e compilação de uma aplicação WPF

13.5. Programação para redes .NET
- 13.5.1. Introdução à programação da rede .NET
- 13.5.2. Pedidos e respostas em .NET
- 13.5.3. Uso de protocolos de aplicação em .NET
- 13.5.4. Segurança de programação da rede .NET

13.6. Ambientes de desenvolvimento de aplicativos móveis
- 13.6.1. Aplicações móveis
- 13.6.2. Aplicações móveis Android
- 13.6.3. Passos para o desenvolvimento do Android
- 13.6.4. O IDE Android Studio

13.7. Desenvolvimento de aplicações no ambiente do Android Studio
- 13.7.1. Instalar e lançar o Android Studio
- 13.7.2. Executando um aplicativo Android
- 13.7.3. Desenvolvimento de GUI no Android Studio
- 13.7.4. Início das atividades no Android Studio

13.8. Depuração e publicação de aplicativos Android
- 13.8.1. Depuração de um aplicativo no Android Studio
- 13.8.2. Memorização de aplicações no Android Studio
- 13.8.3. Publicação de um aplicativo no Google Play

13.9. Desenvolvimento de aplicações para a nuvem
- 13.9.1. *Cloud Computing*
- 13.9.2. Níveis de *Cloud*: SaaS, PaaS, IaaS
- 13.9.3. Principais plataformas de desenvolvimento de nuvens
- 13.9.4. Referências Bibliográficas

13.10. Introdução à Google Cloud Platform
- 13.10.1. Noções básicas de Google Cloud Platform
- 13.10.2. Serviços de Google Cloud Platform
- 13.10.3. Ferramentas de Google Cloud Platform

Módulo 14. Computação de clientes Web

14.1. Introdução ao HTML
- 14.1.1. Estrutura de um documento
- 14.1.2. Cor
- 14.1.3. Texto
- 14.1.4. Links de hipertexto
- 14.1.5. Imagens
- 14.1.6. Listas
- 14.1.7. Tabelas
- 14.1.8. Estruturas(*frames*)
- 14.1.9. Formulários
- 14.1.10. Elementos específicos para tecnologias móveis
- 14.1.11. Itens fora de uso

14.2. Folhas de Estilo Web (CSS)
- 14.2.1. Elementos e estrutura de uma folha de estilo
  - 14.2.1.1. Criação de folhas de estilo
  - 14.2.1.2. Aplicação de estilos Seletores
  - 14.2.1.3. Herança de estilo e cascata
  - 14.2.1.4. Formatação de página usando estilos
  - 14.2.1.5. Estrutura da página usando estilos O modelo de caixa
- 14.2.2. Estilos de design para diferentes dispositivos
- 14.2.3. Tipos de folhas de estilo: estáticas e dinâmicas As Pseudoclasses
- 14.2.4. Boas práticas no uso de folhas de estilo

14.3. Introdução e história do JavaScript
- 14.3.1. Introdução
- 14.3.2. História do JavaScript
- 14.3.3. Ambiente de desenvolvimento a ser utilizado

14.4. Noções básicas de programação web
- 14.4.1. Sintaxe básica do JavaScript
- 14.4.2. Tipos de dados e operadores primitivos
- 14.4.3. Variáveis e domínios
- 14.4.4. Cadeias de texto e *template literals*
- 14.4.5. Números e Booleans
- 14.4.6. Comparar

14.5. Estruturas complexas JavaScript
- 14.5.1. Vetores ou *arrays* e objetos
- 14.5.2. Sets
- 14.5.3. Mapas
- 14.5.4. Disjunções
- 14.5.5. Circuitos

14.6. Funções e objetivos
- 14.6.1. Definição e invocação de funções
- 14.6.2. Argumentação
- 14.6.3. Funções das setas
- 14.6.4. Funções de retorno de chamada ou *callback*
- 14.6.5. Funções de ordem superior
- 14.6.6. Objetos literais
- 14.6.7. O objeto *this*
- 14.6.8. Objetos como namespaces: o objeto Math e o objeto Date

14.7. O Modelo de Objeto de Documento (DOM)
- 14.7.1. O que é DOM?
- 14.7.2. Um pouco de história
- 14.7.3. Navegação e obtenção de elementos
- 14.7.4. Um DOM virtual com JSDOM
- 14.7.5. Selecionadores de consultas ou *query selectors*
- 14.7.6. Navegação através de propriedades
- 14.7.7. Atribuição de atributos aos elementos
- 14.7.8. Criação e modificação de nós
- 14.7.9. Atualizando o estilo dos elementos DOM

14.8. Desenvolvimento web moderno
- 14.8.1. Fluxo de eventos e *listeners*
- 14.8.2. *Toolkits* sistemas modernos de web e alinhamento
- 14.8.3. Modo estritamente JavaScript
- 14.8.4. Mais sobre funções
- 14.8.5. Promessas e funções assíncronas
- 14.8.6. *Closures*
- 14.8.7. Programação funcional
- 14.8.8. POO em JavaScript

14.9. Usabilidade da web
- 14.9.1. Introdução à Usabilidade
- 14.9.2. Definição de usabilidade
- 14.9.3. Importância do web design centrado no usuário
- 14.9.4. Diferenças entre acessibilidade e usabilidade
- 14.9.5. Vantagens e problemas em combinar acessibilidade e usabilidade
- 14.9.6. Vantagens e dificuldades na implementação de websites utilizáveis
- 14.9.7. Métodos de usabilidade
- 14.9.8. Análise das necessidades do usuário
- 14.9.9. Princípios do desenho conceituais Prototipagem orientada ao usuário
- 14.9.10. Diretrizes para a criação de websites utilizáveis
  - 14.9.10.1. Diretrizes de usabilidade de Jakob Nielsen
  - 14.9.10.2. Diretrizes de usabilidade de Jakob Nielsen
- 14.9.11. Avaliação da usabilidade

14.10. Acessibilidade Web
- 14.10.1. Introdução
- 14.10.2. Definição de acessibilidade web
- 14.10.3. Tipos de deficiências
  - 14.10.3.1. Deficiências temporárias ou permanentes
  - 14.10.3.2. Deficiência visual
  - 14.10.3.3. Deficiência auditiva
  - 14.10.3.4. Deficiências motoras
  - 14.10.3.5. Deficiências neurológicas ou cognitivas
  - 14.10.3.6. Dificuldades relacionadas ao envelhecimento
  - 14.10.3.7. Restrições decorrentes do meio ambiente
  - 14.10.3.8. Obstáculos ao acesso ao site
- 14.10.4. Ajudas técnicas e produtos de assistência para superar barreiras
  - 14.10.4.1. Ajudas para os cegos
  - 14.10.4.2. Ajudas para pessoas com baixa visão
  - 14.10.4.3. Ajudas para pessoas daltônicas
  - 14.10.4.4. Ajudas para deficientes auditivos
  - 14.10.4.5. Ajudas para pessoas com deficiência motora
  - 14.10.4.6. Ajudas para pessoas com deficiências cognitivas e neurológicas
- 14.10.5. Vantagens e dificuldades na implementação da acessibilidade web
- 14.10.6. Regulamentos e normas de acessibilidade web
- 14.10.7. Órgãos reguladores de acessibilidade web
- 14.10.8. Comparação de normas e padrões
- 14.10.9. Diretrizes para o cumprimento de regulamentos e normas
  - 14.10.9.1. Descrição das principais diretrizes (imagens, links, vídeos, etc.)
  - 14.10.9.2. Diretrizes para navegação acessível
    - 14.10.9.2.1. Perceptibilidade
    - 14.10.9.2.2. Operabilidade
    - 14.10.9.2.3. Compreensibilidade
    - 14.10.9.2.4. Robustez
- 14.10.10. Descrição do Processo de Conformidade de Acessibilidade da Web
- 14.10.11. Níveis de conformidade
- 14.10.12. Critérios de conformidade
- 14.10.13. Requisitos de conformidad
- 14.10.14. Metodologia para avaliar a acessibilidade dos websites

Módulo 15. Computação de Servidor Web

15.1. Introdução à programação de servidor: PHP
- 15.1.1. Noções básicas de programação do lado do servidor
- 15.1.2. Sintaxe básica do PHP
- 15.1.3. Geração de conteúdo HTML com PHP
- 15.1.4. Ambientes de desenvolvimento e testes: XAMPP

15.2. PHP avançado
- 15.2.1. Estruturas de controle com PHP
- 15.2.2. Funções em PHP
- 15.2.3. Manuseio de *arrays* em PHP
- 15.2.4. Manuseio de cordas com PHP
- 15.2.5. PHP orientado a objetos

15.3. Modelos de dados
- 15.3.1. Conceito de dados Ciclo de vida dos dados
- 15.3.2. Tipos de dados
  - 15.3.2.1. Noções básicas
  - 15.3.2.2. Registros
  - 15.3.2.3. Dinâmica

15.4. O modelo relacional
- 15.4.1. Descrição
- 15.4.2. Entidades e tipos de entidades
- 15.4.3. Elementos de dados Atributos
- 15.4.4. Relacionamentos: tipos, subtipos, cardinalidade
- 15.4.5. Senha: Tipos de senhas
- 15.4.6. Padronização Formulários normais

15.5. Construção do modelo de dados lógicos
- 15.5.1. Especificação das tabelas
- 15.5.2. Definição de colunas
- 15.5.3. Especificação de senhas
- 15.5.4. Conversão para formas normais Dependência

15.6. O modelo de dados físicos Arquivos de dados
- 15.6.1. Descrição dos arquivos de dados
- 15.6.2. Tipos de arquivo
- 15.6.3. Modos de acesso
- 15.6.4. Organização dos arquivos

15.7. Acesso ao banco de dados a partir do PHP
- 15.7.1. Introdução ao MariaDB
- 15.7.2. Trabalhando com um banco de dados MariaDB: a linguagem SQL
- 15.7.3. Acesso ao banco de dados MariaDB a partir do PHP
- 15.7.4. Introdução ao MySql
- 15.7.5. Trabalhar com um banco de dados MySql: linguagem SQL
- 15.7.6. Acessar o banco de dados MySql a partir do PHP

15.8. Interação do cliente a partir do PHP
- 15.8.1. Formulários PHP
- 15.8.2. Cookies
- 15.8.3. Gestão da sessão

15.9. Arquitetura da aplicação Web
- 15.9.1. O modelo View Controller Modelo do modelo
- 15.9.2. Controles
- 15.9.3. Modelo
- 15.9.4. Vistas

15.10. Introdução aos serviços web
- 15.10.1. Introdução ao XML
- 15.10.2. Arquiteturas orientadas a serviços (SOA): serviços web
- 15.10.3. Criação de serviços web SOAP e REST
- 15.10.4. O protocolo SOAP
- 15.10.5. O protocolo REST

Módulo 16. Gestão da Segurança

16.1. A Segurança da informação
- 16.1.1. Introdução
- 16.1.2. A segurança da informação implica confidencialidade, integridade e disponibilidade
- 16.1.3. A segurança é uma questão econômica
- 16.1.4. A segurança é um processo
- 16.1.5. A Classificação da informação
- 16.1.6. A segurança da informação envolve a gestão de riscos
- 16.1.7. A segurança é articulada com controles de segurança
- 16.1.8. A segurança é tanto física quanto lógica
- 16.1.9. A segurança envolve as pessoas

16.2. O profissional de segurança da informação
- 16.2.1. Introdução
- 16.2.2. A Segurança da informação como profissão
- 16.2.3. Certificações (ISC)2
- 16.2.4. O padrão ISO 27001
- 16.2.5. Boas práticas de segurança na gestão de serviços de TI
- 16.2.6. Modelos de maturação para segurança da informação
- 16.2.7. Outras certificações, normas e recursos profissionais

16.3. Controle de acesso
- 16.3.1. Introdução
- 16.3.2. Requisitos de controle de acesso
- 16.3.3. Mecanismos de autenticação
- 16.3.4. Métodos de autorização
- 16.3.5. Acesso à contabilidade e auditoria
- 16.3.6. Tecnologias Triple A

16.4. Programas, processos e políticas de segurança da informação
- 16.4.1. Introdução
- 16.4.2. Programas de gestão de segurança
- 16.4.3. Gestão de riscos
- 16.4.4. Concepção da política de segurança

16.5. Planos de continuidade de negócios
- 16.5.1. Introdução aos BCPs
- 16.5.2. Fase I e II
- 16.5.3. Fase III e IV
- 16.5.4. Manutenção do BCP

16.6. Procedimentos para a proteção adequada da empresa
- 16.6.1. Redes DMZ
- 16.6.2. Sistemas de detecção de intrusão
- 16.6.3. Requisitos de controle de acesso
- 16.6.4. Aprendendo com o atacante: Honeypot

16.7. Arquitetura de segurança Prevenção
- 16.7.1. Visão geral. Atividades e modelo estratificado
- 16.7.2. Defesa perimetral (Firewalls, WAFs, IPS, etc.)
- 16.7.3. Defesa do ponto final (equipamentos, servidores e serviços)

16.8. Arquitetura de segurança Detecção
- 16.8.1. Detecção e monitoramento geral
- 16.8.2. Logs, quebra de tráfego criptografado, gravação e Siems
- 16.8.3. Alertas e inteligência

16.9. Arquitetura de segurança Reação
- 16.9.1. Reação Produtos, serviços e recursos
- 16.9.2. Gestão de Incidentes
- 16.9.3. CERTS e CSIRTs

16.10. Arquitetura de segurança Recuperação
- 16.10.1. Resiliência, conceitos, requisitos e padrões comerciais
- 16.10.2. Resiliência Soluções de TI
- 16.10.3. Gestão de Crise e Governança

Módulo 17. Segurança de software

17.1. Problemas de segurança de software
- 17.1.1. Introdução ao Problemas de segurança de software
- 17.1.2. Vulnerabilidades e sua classificação
- 17.1.3. Propriedades Software seguro
- 17.1.4. Referências

17.2. Princípios do projeto de segurança de Software
- 17.2.1. Introdução
- 17.2.2. Princípios do projeto de segurança de Software
- 17.2.3. Tipos de S-SDLC
- 17.2.4. Segurança do Software nas fases S-SDLC
- 17.2.5. Metodologias e normas
- 17.2.6. Referências

17.3. Segurança do ciclo de vida do Software nas fases de requisitos e projeto
- 17.3.1. Introdução
- 17.3.2. Modelagem de ataque
- 17.3.3. Casos de abuso
- 17.3.4. Engenharia de requisitos de segurança
- 17.3.5. Análise de risco Arquitetônico
- 17.3.6. Padrões de design
- 17.3.7. Referências

17.4. Segurança do ciclo de vida do Software nas fases de codificação, testes e operação
- 17.4.1. Introdução
- 17.4.2. Teste de segurança baseado em risco
- 17.4.3. Revisão do código
- 17.4.4. Teste de penetração
- 17.4.5. Operações de segurança
- 17.4.6. Revisão externa
- 17.4.7. Referências

17.5. Aplicações de codificação segura I
- 17.5.1. Introdução
- 17.5.2. Práticas de codificação seguras
- 17.5.3. Manuseio e validação das entradas
- 17.5.4. Excesso de memória
- 17.5.5. Referências

17.6. Aplicações de codificação segura II
- 17.6.1. Introdução
- 17.6.2. Integers Overflows, erros de truncagem e problemas com conversões de tipo entre números inteiros
- 17.6.3. Erros e exceções
- 17.6.4. Privacidade e confidencialidade
- 17.6.5. Programas privilegiados
- 17.6.6. Referências

17.7. Segurança no desenvolvimento e na nuvem
- 17.7.1. Segurança no desenvolvimento; metodologia e prática
- 17.7.2. Modelos PaaS, IaaS,CaaS e SaaS
- 17.7.3. Segurança na nuvem e para serviços na nuvem

17.8. Cifrado
- 17.8.1. Fundamentos da criptologia
- 17.8.2. Criptografia simétrica e assimétrica
- 17.8.3. Criptografia em repouso e em trânsito

17.9. Orquestração e Automação de Segurança (SOAR)
- 17.9.1. Complexidade do processamento manual; necessidade de automatizar tarefas
- 17.9.2. Produtos e serviços
- 17.9.3. Arquitetura SOAR

17.10. Segurança no teletrabalho
- 17.10.1. Necessidades e cenários
- 17.10.2. Produtos e serviços
- 17.10.3. Segurança no teletrabalho

Módulo 18. Administração do Servidor Web

18.1. Introdução aos servidores web
- 18.1.1. O que é um servidor web?
- 18.1.2. Arquitetura e operação de um servidor web
- 18.1.3. Recursos e conteúdo em um servidor web
- 18.1.4. Servidores de aplicação
- 18.1.5. Servidores Proxy
- 18.1.6. Principais servidores web no mercado
- 18.1.7. Estatísticas de uso do servidor Web
- 18.1.8. Segurança do servidor Web
- 18.1.9. Balanceamento de carga em servidores web
- 18.1.10. Referências

18.2. Manejo do protocolo HTTP
- 18.2.1. Funcionamento e estrutura.
- 18.2.2. Descrição dos pedidos ou request methods
- 18.2.3. Códigos de status
- 18.2.4. Cabeçalhos
- 18.2.5. Codificação do conteúdo Páginas de código
- 18.2.6. Realizar solicitações HTTP na Internet utilizando um proxy, livehttpheaders analisando o protocolo utilizado

18.3. Descrição de arquiteturas distribuídas através de múltiplos servidores
- 18.3.1. Modelo em 3 camadas
- 18.3.2. Tolerância a falhas
- 18.3.3. Compartilhamento de carga
- 18.3.4. Armazenamento do estado da sessão
- 18.3.5. Armazenamento em cache

18.4. Internet Information Services (IIS)
- 18.4.1. O que é IIS?
- 18.4.2. História e evolução do IIS
- 18.4.3. Principais benefícios e características do IIS7 e mais
- 18.4.4. IIS7 e arquitetura posterior

18.5. Instalação, administração e configuração do IIS
- 18.5.1. Preâmbulo
- 18.5.2. Instalação de Internet Information Services (IIS)
- 18.5.3. Ferramentas de administração do IIS
- 18.5.4. Criação, configuração e administração do site
- 18.5.5. Instalando e gerenciando extensões no IIS

18.6. Segurança avançada no IIS
- 18.6.1. Preâmbulo
- 18.6.2. Autenticação, autorização e controle de acesso no IIS
- 18.6.3. Criação de um site seguro no IIS com SSL
- 18.6.4. Políticas de segurança implementadas no IIS 8.x

18.7. Introdução ao Apache
- 18.7.1. O que é Apache?
- 18.7.2. Principais vantagens do Apache
- 18.7.3. Principais características do Apache
- 18.7.4. Arquitetura

18.8. Instalação e configuração da ferramenta
- 18.8.1. Instalação Inicial da ferramenta
- 18.8.2. Configuração do Apache

18.9. Instalação e configuração dos diferentes módulos no Apache
- 18.9.1. Instalação dos módulos Apache
- 18.9.2. Tipos de módulos
- 18.9.3. Configuração segura do Apache

18.10. Segurança avançada
- 18.10.1. Autenticação, autorização e controle de acesso
- 18.10.2. Métodos de autenticação
- 18.10.3. Configuração segura do Apache com SSL

Módulo 19. Auditoria de Segurança

19.1. Introdução aos sistemas de informação e sua auditoria
- 19.1.1. Introdução aos sistemas de informação e o papel da auditoria informática
- 19.1.2. Definições de Auditoria de TI e Controle de TI interno
- 19.1.3. Funções e objetivos da auditoria de TI
- 19.1.4. Diferenças entre controle interno e auditoria de TI

19.2. Controles Internos de Sistemas de Informação
- 19.2.1. Fluxograma funcional de um centro de processamento de dados
- 19.2.2. Classificação dos controles dos sistemas de informação
- 19.2.3. A Regra de Ouro

19.3. O processo e as fases da auditoria dos Sistemas de Informação
- 19.3.1. Avaliação de risco (RRA) e outras metodologias de auditoria de TI
- 19.3.2. Execução de uma auditoria de Sistemas de Informação Fases de auditoria
- 19.3.3. Principais habilidades do auditor de Sistemas de Informação

19.4. Auditoria técnica de segurança de sistemas e redes
- 19.4.1. Auditorias técnicas de segurança Teste de intrusão Conceitos preliminares
- 19.4.2. Auditorias de segurança do sistema Ferramentas de apoio
- 19.4.3. Auditorias de segurança de rede Ferramentas de apoio

19.5. Auditoria técnica da segurança da internet e dos dispositivos móveis
- 19.5.1. Auditorias de segurança de Internet Ferramentas de apoio
- 19.5.2. Auditoria da segurança da dispositivos móveis Ferramentas de apoio
- 19.5.3. Anexo 1. Estrutura do relatório executivo e relatório técnico
- 19.5.4. Anexo 2. Inventário de ferramentas
- 19.5.5. Anexo 3. Metodologias

19.6. Sistemas de gestão de segurança da informação
- 19.6.1. Segurança IS: propriedades e fatores de influência
- 19.6.2. Risco empresarial e gerenciamento de risco: implementação de controles
- 19.6.3. Sistema de Gestão de Segurança da Informação (SGSI): conceito e fatores críticos de sucesso
- 19.6.4. SGSI-Modelo PDCA
- 19.6.5. ISMS ISO-IEC 27001: contexto organizacional
- 19.6.6. Contexto organizacional
- 19.6.7. Liderança
- 19.6.8. Planejamento
- 19.6.9. Suporte
- 19.6.10. Operação
- 19.6.11. Avaliação de desempenho
- 19.6.12. Melhorias
- 19.6.13. Anexo à ISO 27001/ISO-IEC 27002: objetivos e controles
- 19.6.14. Auditoria do SGSI

19.7. Realização da auditoria
- 19.7.1. Procedimento
- 19.7.2. Técnicas

19.8. Rastreabilidade
- 19.8.1. Metodologias
- 19.8.2. Análise

19.9. Custódia
- 19.9.1. Técnicas
- 19.9.2. Resultados

19.10. Relatório e apresentação de provas
- 19.10.1. Tipos de relatórios
- 19.10.2. Análise dos dados
- 19.10.3. Apresentação de provas

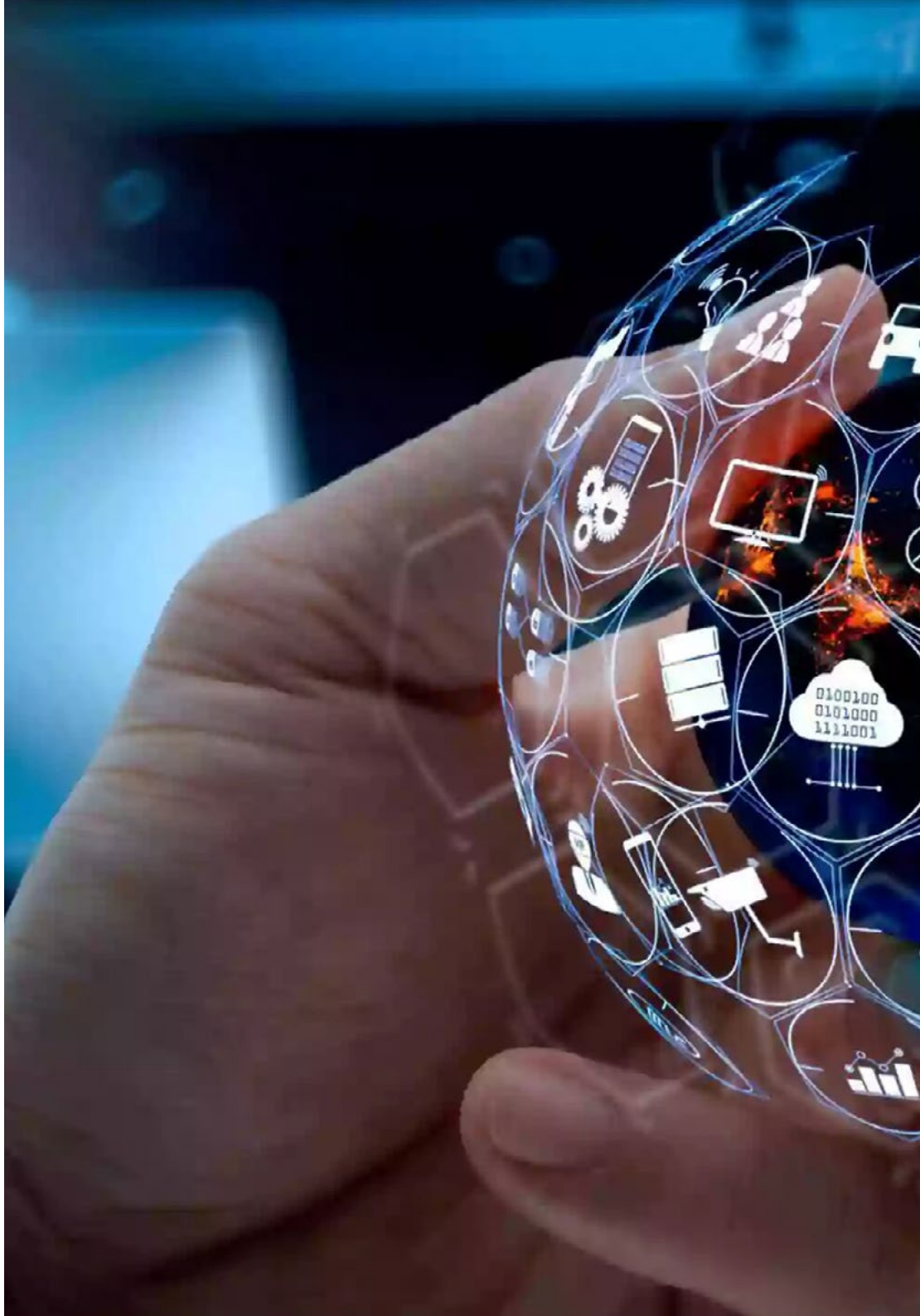
Módulo 20. Segurança de Aplicações Online

20.1. Vulnerabilidades e problemas de segurança em aplicações online
- 20.1.1. Introdução à segurança em aplicações online
- 20.1.2. Vulnerabilidades de segurança no projeto de aplicações web
- 20.1.3. Vulnerabilidades de segurança na implementação de aplicações web
- 20.1.4. Vulnerabilidades de segurança na implantação de aplicações web
- 20.1.5. Listas oficiais de vulnerabilidades de segurança

20.2. Políticas e normas para a segurança de aplicações online
- 20.2.1. Pilares segurança de aplicações online
- 20.2.2. Política de segurança
- 20.2.3. Sistemas de gestão de segurança da informação
- 20.2.4. Ciclo de vida de desenvolvimento segurança de software
- 20.2.5. Normas segurança de aplicações

20.3. Segurança no projeto de aplicações web
- 20.3.1. Introdução à segurança de aplicações Web
- 20.3.2. Segurança no projeto de aplicações web

20.4. Testando a segurança online e a segurança das aplicações web
- 20.4.1. Análises e testes de segurança de aplicações Web
- 20.4.2. Segurança no implantação e produção de aplicações web



20.5. Segurança dos serviços web
- 20.5.1. Introdução à segurança dos serviços web
- 20.5.2. Funções e tecnologias de segurança dos serviços web

20.6. Testando a segurança online e a segurança dos serviços web
- 20.6.1. Avaliação de segurança dos serviços web
- 20.6.2. Proteção online Firewalls e gateways XML

20.7. Hacking ético, malware e forensic
- 20.7.1. Hacking ético
- 20.7.2. Análise de malware
- 20.7.3. Análise Forense

20.8. Boas práticas para garantir a segurança na aplicação
- 20.8.1. Manual de boas práticas no desenvolvimento de aplicações online
- 20.8.2. Manuais de Boas práticas na implementação de aplicações Online

20.9. Erros comuns que minam a segurança da aplicação
- 20.9.1. Erros comuns no desenvolvimento
- 20.9.2. Erros comuns na hospedagem
- 20.9.3. Erros comuns na produção

Um plano de estudos abrangente que o levará a dominar a área de Big Data e a se tornar um arquiteto de estratégia de negócios bem-sucedido"

04

Objetivos de ensino

Esse Advanced Master em Engenharia e Qualidade de Software tem como objetivo treinar profissionais altamente qualificados no projeto, desenvolvimento e gerenciamento de sistemas de software de alta qualidade. Com um foco especial na qualidade e no gerenciamento estratégico de projetos de tecnologia, o programa também promove a capacidade de adaptação aos rápidos desenvolvimentos do setor. Isso garante que os alunos não apenas estejam preparados para enfrentar os desafios do futuro, mas também sejam capazes de liderar a inovação no campo do software.

Transforme ideias em soluções tecnológicas eficazes com a arte da engenharia de software"

Objetivos gerais

- Desenvolver habilidades avançadas em projeto, desenvolvimento e manutenção de sistemas de software complexos e dimensionáveis, aplicando as melhores práticas e metodologias de engenharia de software
- Capacitar os alunos em garantia de qualidade de software, fornecendo-lhes ferramentas e técnicas para garantir a confiabilidade, a segurança e o desempenho das soluções tecnológicas
- Promover a liderança em gestão de projetos de tecnologia, desenvolvendo competências em gestão de equipes multidisciplinares, planejamento estratégico e tomada de decisões em ambientes dinâmicos
- Promover a capacidade de adaptação aos rápidos avanços tecnológicos, por meio da especialização em novas ferramentas, técnicas e tendências que lhes permitam permanecer na vanguarda da engenharia de software
- Desenvolver competências em gerenciamento de qualidade em todo o ciclo de vida do software, desde o planejamento inicial até a manutenção e o aprimoramento contínuo dos sistemas
- Fortalecer as habilidades de comunicação e trabalho em equipe, essenciais para colaborar efetivamente com diferentes partes interessadas, gerenciar expectativas e garantir o sucesso de projetos tecnológicos

Aprimore suas habilidades e torne-se um líder na criação de soluções de tecnologia de ponta"

Objetivos específicos

Módulo 1. Qualidade do Software Níveis de Desenvolvimento TRL

- Compreender os diferentes níveis de maturidade tecnológica e sua relação com a qualidade do software
- Avaliar o desenvolvimento do software em cada estágio do TRL e como isso afeta a qualidade do produto final

Módulo 2. Desenvolvimento de Projetos Software Documentação Funcional e Técnica

- Desenvolver habilidades para criar documentação funcional e técnica clara e detalhada em projetos de software
- Analisar a importância da documentação precisa para o gerenciamento de projetos e a qualidade do software

Módulo 3. *Testing* de Software Automação de Testes

- Desenvolver habilidades para projetar e executar testes automatizados em aplicativos de software
- Implementar soluções de *testing* eficientes usando ferramentas de automação de teste

Módulo 4. Metodologias de Gestão de Projetos Software Metodologias Waterfall comparadas com Metodologias Ágeis

- Analisar as diferenças entre as metodologias Waterfall e Agile no gerenciamento de projetos de software
- Avaliar os benefícios e as limitações de cada metodologia de acordo com o tipo de projeto

Módulo 5.TDD (*Test Driven Developement*). Projeto de Software guiado por Testes

- Desenvolver habilidades para escrever testes unitários antes de escrever o código de produção
- Melhorar a qualidade do software implementando o TDD no processo de desenvolvimento

Módulo 6. DevOps Gestão de Qualidade do Software

- Explorar o conceito de DevOps e seu impacto no aprimoramento contínuo da qualidade do software
- Saber como integrar práticas de desenvolvimento e operações para obter um ciclo de vida de software mais ágil e eficiente

Módulo 7. DevOps e Integração Contínua Soluções Práticas Avançadas no Desenvolvimento de Software

- Aprofundar as técnicas avançadas de integração contínua dentro da estrutura de DevOps
- Implementar soluções práticas de integração contínua para automatizar o processo de desenvolvimento e implantação de software

Módulo 8. Design de Bases de Dados (BD) Normalização e Rendimento Qualidade de software

- Analisar os princípios de design de banco de dados, incluindo normalização e otimização de desempenho
- Compreender como o design adequado do banco de dados contribui para a qualidade do software

Módulo 9. Projeto de Arquitetura Escalável A Arquitetura no Ciclo de Vida do Software

- Aprofundar os princípios de design de arquiteturas dimensionáveis e seu impacto na qualidade e no desempenho do software
- Avaliar diferentes padrões de arquitetura para aplicativos de software dimensionáveis

Módulo 10. Critérios de Qualidade ISO, IEC 9126 Métricas de qualidade de software

- Compreender os critérios de qualidade de software de acordo com esses padrões e como aplicá-los
- Implementar métricas de qualidade para avaliar e melhorar continuamente aplicativos de software

Módulo 11. Metodologias, Desenvolvimento e Qualidade em Engenharia de Software

- Estudar em profundidade as metodologias mais utilizadas na Engenharia de Software e sua relação com a Qualidade
- Desenvolver uma abordagem holística que combine desenvolvimento, teste e qualidade em projetos de software

Módulo 12. Gerenciamento de projetos de Software

- Desenvolver habilidades em gerenciamento de projetos de software, do planejamento à execução
- Gerenciar os recursos, o tempo e os riscos associados aos projetos de desenvolvimento de software

Módulo 13. Plataformas de desenvolvimento de software

- Explorar diferentes plataformas de desenvolvimento de software e seus recursos
- Avaliar as plataformas de desenvolvimento em termos de seus recursos, flexibilidade e compatibilidade com diferentes projetos

Módulo 14. Computação de clientes Web

- Analisar como a computação no lado do cliente é realizada no desenvolvimento de aplicativos da Web
- Desenvolver aplicativos que aproveitem a computação no lado do cliente para melhorar a interação e o desempenho

Módulo 15. Computação de Servidor Web

- Explorar as tecnologias e técnicas usadas na computação de servidores da Web
- Compreender o manejo de dados, a lógica comercial e o gerenciamento de usuários no servidor

Módulo 16. Gestão da Segurança

- Avaliar os riscos de segurança nos aplicativos e implementar medidas preventivas
- Implementar controles de segurança em todos os estágios do ciclo de vida do software

Módulo 17. Segurança de software

- Explorar as práticas recomendadas de segurança no desenvolvimento de software
- Analisar as vulnerabilidades mais comuns do software e aprender a atenuá-las.

Módulo 18. Administração do Servidor Web

- Compreender a função dos servidores da Web no desenvolvimento e na implantação de aplicativos
- Desenvolver habilidades na administração e manutenção de servidores da Web

Módulo 19. Auditoria de Segurança

- Avaliar a segurança dos sistemas por meio de auditorias e testes de penetração
- Implementar processos de auditoria contínua para melhorar a segurança do software

Módulo 20. Segurança de Aplicações Online

- Implementar soluções para proteger aplicativos online contra ameaças externas e internas
- Estabelecer políticas de segurança e auditoria para garantir a integridade dos aplicativos online

05

Oportunidades profissionais

Os alunos serão altamente qualificados para assumir funções de liderança no desenvolvimento, implementação e gerenciamento de software de alta qualidade. Graças à sua especialização avançada em áreas importantes, como arquitetura de software, segurança de aplicativos online, gerenciamento de projetos de tecnologia e metodologias ágeis, você poderá liderar equipes de desenvolvimento. Além disso, sua preparação permitirá que você ocupe cargos de gerência em projetos de software, enquanto sua capacidade de inovar e liderar equipes multidisciplinares permitirá que você enfrente os desafios do ambiente digital e contribua significativamente para o sucesso de qualquer organização.

A TECH lhe dá a oportunidade de realizar seus sonhos na disciplina mais empolgante, que transforma ideias em produtos tangíveis capazes de melhorar a vida das pessoas"

Perfil do aluno

O perfil de pós-graduação do Advanced Master em Engenharia e Qualidade de Software é orientado para a formação de profissionais altamente qualificados, capazes de liderar e gerenciar projetos tecnológicos de alto impacto. Para garantir a qualidade, a segurança e a eficiência em todas as fases do desenvolvimento de software, você dominará as metodologias ágeis e tradicionais. Além disso, você será capaz de projetar e desenvolver sistemas de software dimensionáveis, eficientes e seguros, aplicando padrões internacionais de qualidade e metodologias avançadas, como DevOps e integração contínua.

Torne-se um especialista em sucesso empresarial na maior universidade digital do mundo.

- **Segurança de software e sistemas:** Competência na implementação de práticas avançadas de segurança, incluindo proteção de dados e gerenciamento de vulnerabilidades em aplicativos online.
- **Garantia de qualidade de software:** Capacidade de aplicar padrões internacionais (ISO, IEC 9126) e ferramentas de teste automatizadas para garantir a confiabilidade e o desempenho do software.
- **Desenvolvimento de arquiteturas dimensionáveis:** Capacidade de projetar e criar sistemas de software que possam crescer e se adaptar às demandas do mercado sem comprometer a qualidade e a segurança
- **Integração contínua e DevOps:** Capacidade de implementar e gerenciar processos de integração contínua, garantindo a entrega eficiente e contínua de novas funcionalidades de software

Após concluir o Advanced Master, você poderá usar seus conhecimentos e habilidades nos seguintes cargos:

1. **Diretor de Tecnologia (CTO):** Responsável pela direção estratégica da tecnologia em uma empresa, liderando equipes de desenvolvimento e supervisionando a implementação de soluções tecnológicas inovadoras.
2. **Gerente de Qualidade do Software:** Responsável por supervisionar e garantir que os processos e produtos de software atendam aos padrões de qualidade estabelecidos, liderando os testes de software e as iniciativas de melhoria contínua.
3. **Arquiteto de software:** Projetista líder da estrutura e da arquitetura de sistemas de software complexos, garantindo que eles sejam escalonáveis, seguros e eficientes
4. **Líder de projeto de software:** Responsável pelo planejamento, execução e entrega de projetos de software, gerenciando equipes multidisciplinares e garantindo que os projetos sejam concluídos no prazo, dentro do orçamento e com os padrões de qualidade adequados.
5. **Especialista em segurança de computadores:** Responsável pela proteção de aplicativos, infraestrutura e dados contra ameaças cibernéticas, implementando estratégias e políticas de segurança eficazes.
6. **Auditor de segurança de software:** Realiza auditorias abrangentes para identificar vulnerabilidades em aplicativos e sistemas, propondo melhorias e soluções para garantir a segurança do software.

Se você quiser fazer a diferença no mundo digital, escolha esse caminho que o especializa como um especialista na criação de software de qualidade"

06

Metodologia de estudo

A TECH é a primeira universidade do mundo a unir a metodologia dos ***case studies*** com o ***Relearning***, um sistema de aprendizado 100% online baseado na repetição guiada.

Essa estratégia de ensino inovadora foi projetada para oferecer aos profissionais a oportunidade de atualizar conhecimentos e desenvolver habilidades de forma intensiva e rigorosa. Um modelo de aprendizagem que coloca o aluno no centro do processo acadêmico e lhe dá o papel principal, adaptando-se às suas necessidades e deixando de lado as metodologias mais convencionais.

A TECH prepara você para enfrentar novos desafios em ambientes incertos e alcançar o sucesso em sua carreira"

O aluno: a prioridade de todos os programas da TECH

Na metodologia de estudo da TECH, o aluno é o protagonista absoluto.
As ferramentas pedagógicas de cada programa foram selecionadas levando-se em conta as demandas de tempo, disponibilidade e rigor acadêmico que, atualmente, os alunos, bem como os empregos mais competitivos do mercado, exigem.

Com o modelo educacional assíncrono da TECH, é o aluno quem escolhe quanto tempo passa estudando, como decide estabelecer suas rotinas e tudo isso no conforto do dispositivo eletrônico de sua escolha. O aluno não precisa assistir às aulas presenciais, que muitas vezes não poderá comparecer. As atividades de aprendizado serão realizadas de acordo com sua conveniência. O aluno sempre poderá decidir quando e de onde estudar.

> *Na TECH, o aluno NÃO terá aulas ao vivo (das quais poderá nunca participar)".*

Os programas de ensino mais abrangentes do mundo

A TECH se caracteriza por oferecer os programas acadêmicos mais completos no ambiente universitário. Essa abrangência é obtida por meio da criação de programas de estudo que cobrem não apenas o conhecimento essencial, mas também as últimas inovações em cada área.

Por serem constantemente atualizados, esses programas permitem que os alunos acompanhem as mudanças do mercado e adquiram as habilidades mais valorizadas pelos empregadores. Dessa forma, os alunos da TECH recebem uma preparação abrangente que lhes dá uma vantagem competitiva significativa para avançar em suas carreiras.

Além disso, eles podem fazer isso de qualquer dispositivo, PC, tablet ou smartphone.

> *O modelo da TECH é assíncrono, portanto, você poderá estudar com seu PC, tablet ou smartphone onde quiser, quando quiser e pelo tempo que quiser”*

Case studies ou Método de caso

O método de casos tem sido o sistema de aprendizado mais amplamente utilizado pelas melhores escolas de negócios do mundo. Desenvolvido em 1912 para que os estudantes de direito não aprendessem a lei apenas com base no conteúdo teórico, sua função também era apresentar a eles situações complexas da vida real. Assim, eles poderiam tomar decisões informadas e fazer julgamentos de valor sobre como resolvê-los. Em 1924 foi estabelecido como o método de ensino padrão em Harvard.

Com esse modelo de ensino, é o próprio aluno que desenvolve sua competência profissional por meio de estratégias como o *Learning by doing* ou o *Design Thinking*, usados por outras instituições renomadas, como Yale ou Stanford.

Esse método orientado para a ação será aplicado em toda a trajetória acadêmica do aluno com a TECH. Dessa forma, o aluno será confrontado com várias situações da vida real e terá de integrar conhecimentos, pesquisar, argumentar e defender suas ideias e decisões. A premissa era responder à pergunta sobre como eles agiriam diante de eventos específicos de complexidade em seu trabalho diário.

Método *Relearning*

Na TECH os *case studies* são alimentados pelo melhor método de ensino 100% online: o *Relearning*.

Esse método rompe com as técnicas tradicionais de ensino para colocar o aluno no centro da equação, fornecendo o melhor conteúdo em diferentes formatos. Dessa forma, consegue revisar e reiterar os principais conceitos de cada matéria e aprender a aplicá-los em um ambiente real.

Na mesma linha, e de acordo com várias pesquisas científicas, a repetição é a melhor maneira de aprender. Portanto, a TECH oferece entre 8 e 16 repetições de cada conceito-chave dentro da mesma lição, apresentadas de uma forma diferente, a fim de garantir que o conhecimento seja totalmente incorporado durante o processo de estudo.

O Relearning permitirá uma aprendizagem com menos esforço e mais desempenho, fazendo com que você se envolva mais em sua especialização, desenvolvendo seu espírito crítico e sua capacidade de defender argumentos e contrastar opiniões: uma equação de sucesso.

Um Campus Virtual 100% online com os melhores recursos didáticos

Para aplicar sua metodologia de forma eficaz, a TECH se concentra em fornecer aos alunos materiais didáticos em diferentes formatos: textos, vídeos interativos, ilustrações e mapas de conhecimento, entre outros. Todos eles são projetados por professores qualificados que concentram seu trabalho na combinação de casos reais com a resolução de situações complexas por meio de simulação, o estudo de contextos aplicados a cada carreira profissional e o aprendizado baseado na repetição, por meio de áudios, apresentações, animações, imagens etc.

As evidências científicas mais recentes no campo da neurociência apontam para a importância de levar em conta o local e o contexto em que o conteúdo é acessado antes de iniciar um novo processo de aprendizagem. A capacidade de ajustar essas variáveis de forma personalizada ajuda as pessoas a lembrar e armazenar o conhecimento no hipocampo para retenção a longo prazo. Trata-se de um modelo chamado *Neurocognitive context-dependent e-learning* que é aplicado conscientemente nesse curso universitário.

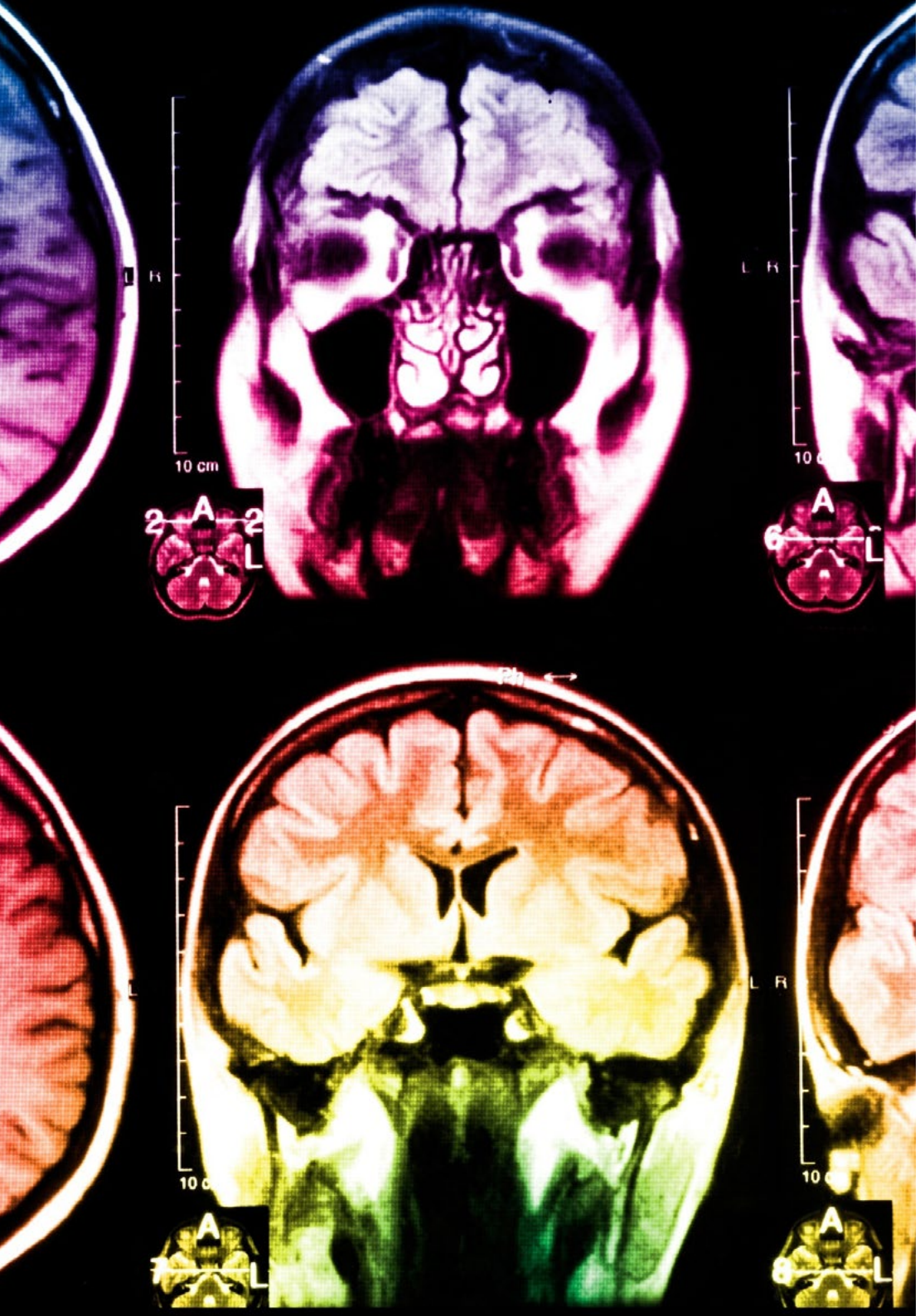
Por outro lado, também para favorecer ao máximo o contato entre mentor e mentorado, é oferecida uma ampla variedade de possibilidades de comunicação, tanto em tempo real quanto em diferido (mensagens internas, fóruns de discussão, serviço telefônico, contato por e-mail com a secretaria técnica, bate-papo, videoconferência etc.).

Da mesma forma, esse Campus Virtual muito completo permitirá que os alunos da TECH organizem seus horários de estudo de acordo com sua disponibilidade pessoal ou obrigações de trabalho. Dessa forma, eles terão um controle global dos conteúdos acadêmicos e de suas ferramentas didáticas, em função de sua atualização profissional acelerada.

> *O modo de estudo online deste programa permitirá que você organize seu tempo e ritmo de aprendizado, adaptando-o à sua agenda”*

A eficácia do método é justificada por quatro conquistas fundamentais:

1. Os alunos que seguem este método não só assimilam os conceitos, mas também desenvolvem a capacidade intelectual através de exercícios de avaliação de situações reais e de aplicação de conhecimentos.
2. A aprendizagem se consolida nas habilidades práticas, permitindo ao aluno integrar melhor o conhecimento à prática clínica.
3. A assimilação de ideias e conceitos se torna mais fácil e eficiente, graças à abordagem de situações decorrentes da realidade.
4. A sensação de eficiência do esforço investido se torna um estímulo muito importante para os alunos, o que se traduz em um maior interesse pela aprendizagem e um aumento no tempo dedicado ao curso.



A metodologia universitária mais bem avaliada por seus alunos

Os resultados desse modelo acadêmico inovador podem ser vistos nos níveis gerais de satisfação dos alunos da TECH.

A avaliação dos alunos sobre a qualidade do ensino, a qualidade dos materiais, a estrutura e os objetivos do curso é excelente. Não é de surpreender que a instituição tenha se tornado a universidade mais bem avaliada por seus alunos de acordo com o índice global score, obtendo 4,9 de 5 pontos.

Acesse o conteúdo do estudo de qualquer dispositivo com conexão à Internet (computador, tablet, smartphone) graças ao fato da TECH estar na vanguarda da tecnologia e do ensino.

Você poderá aprender com as vantagens do acesso a ambientes de aprendizagem simulados e com a abordagem de aprendizagem por observação, ou seja, aprender com um especialista.

Assim, os melhores materiais educacionais, cuidadosamente preparados, estarão disponíveis neste programa:

Material de estudo

O conteúdo didático foi elaborado especialmente para este curso pelos especialistas que irão ministrá-lo, o que permite que o desenvolvimento didático seja realmente específico e concreto.

Posteriormente, esse conteúdo é adaptado ao formato audiovisual, para criar o método de trabalho online, com as técnicas mais recentes que nos permitem lhe oferecer a melhor qualidade em cada uma das peças que colocaremos a seu serviço.

Práticas de aptidões e competências

Serão realizadas atividades para desenvolver as habilidades e competências específicas em cada área temática. Práticas e dinâmicas para adquirir e desenvolver as competências e habilidades que um especialista precisa desenvolver no âmbito da globalização.

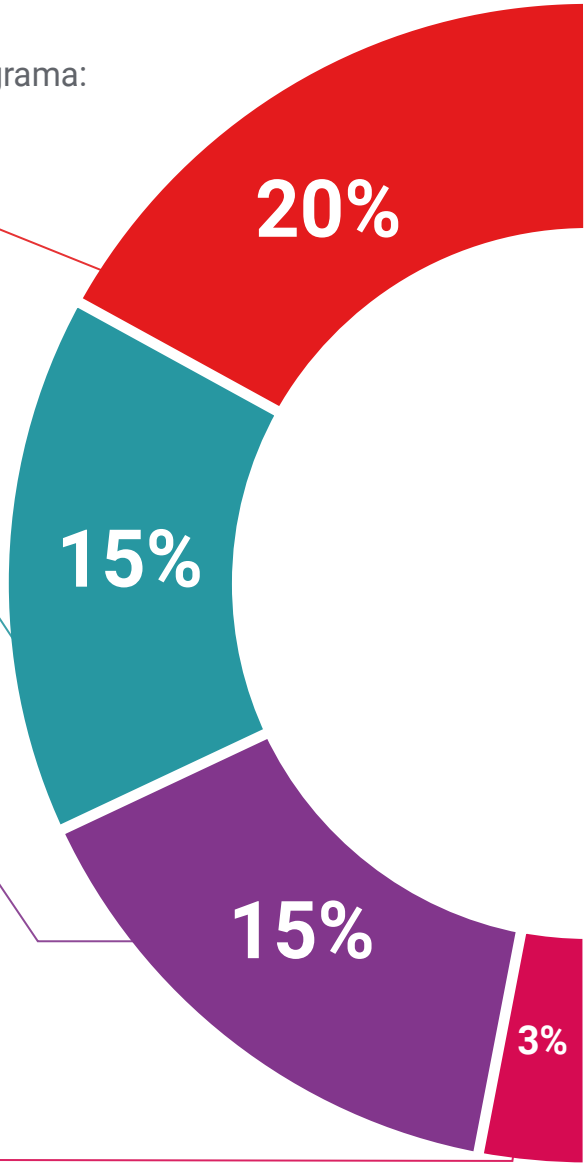
Resumos interativos

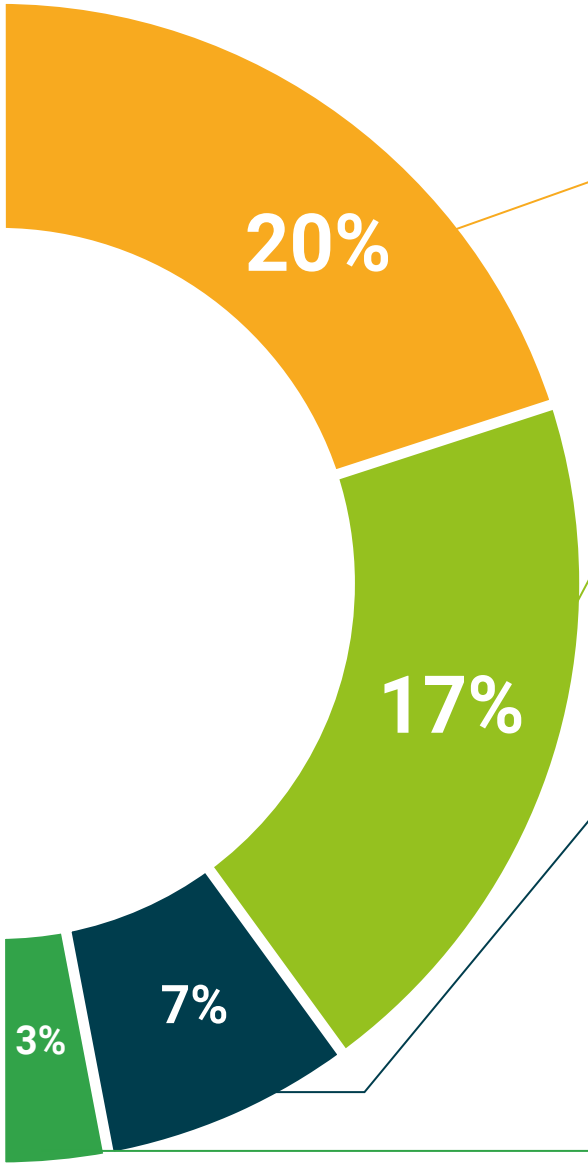
Apresentamos os conteúdos de forma atraente e dinâmica em pílulas multimídia que incluem áudio, vídeos, imagens, diagramas e mapas conceituais com o objetivo de reforçar o conhecimento.

Este sistema exclusivo de capacitação por meio da apresentação de conteúdo multimídia foi premiado pela Microsoft como "Caso de sucesso na Europa"

Leituras complementares

Artigos recentes, documentos científicos, guias internacionais, entre outros. Na biblioteca virtual do estudante você terá acesso a tudo o que for necessário para completar sua capacitação.





Case Studies

Você concluirá uma seleção dos melhores *case studies* da disciplina. Casos apresentados, analisados e orientados pelos melhores especialistas no cenário internacional.

Testing & Retesting

Avaliamos e reavaliamos periodicamente seus conhecimentos ao longo de todo o programa. Fazemos isso em 3 dos 4 níveis da Pirâmide de Miller.

Masterclasses

Há evidências científicas sobre a utilidade da observação de terceiros especialistas.

O *Learning from an expert* fortalece o conhecimento e a memória, e aumenta nossa confiança para tomar decisões difíceis no futuro.

Guias rápidos de ação

A TECH oferece o conteúdo mais relevante do curso em formato de fichas de trabalho ou guias rápidos de ação. Uma forma sintetizada, prática e eficaz de ajudar os alunos a progredirem na aprendizagem.

07

Equipe de professores

Este Advanced Master de Engenharia de Software e Qualidade é ministrado por uma equipe de especialistas em engenharia, além disso, possuem muitos anos de experiência na gestão e desenvolvimento de projetos técnicos e especializados. Com esse diferencial aportado pelos excelentes professores, este programa 100% online oferecerá um completo conteúdo com ênfase em um ambiente simulado de casos práticos e reais, permitindo ao aluno uma aprendizagem situada e contextual.

Participe da profissão mais procurada por especialistas em engenharia de software e qualidade e aproveite esta oportunidade para se tornar um deles”

Diretor Internacional Convidado

Darren Pulsipher é um **arquiteto de software** altamente experiente, um inovador com um histórico internacional distinto em **desenvolvimento de software e firmware**. Na verdade, ele possui habilidades altamente desenvolvidas de **comunicação**, **gestão de projetos** e **negócios**, o que lhe permitiu liderar iniciativas importantes em nível global.

Além disso, ocupou cargos seniores de grande responsabilidade ao longo de sua carreira, incluindo o de **arquiteto-chefe de soluções para o setor público** na **Intel Corporation**, onde promoveu **negócios,, processos e tecnologias modernos** para clientes, parceiros e usuários do **setor público.**. Além disso, ele fundou a **Yoly Inc.**, onde também atuou como **CEO**,trabalhando para desenvolver uma **ferramenta de agregação e diagnóstico** de **redes sociais** baseada em **software como serviço (SaaS)**, usando tecnologias de ***Big Data*** e **Web 2.0**.

Além disso, atuou em outras empresas, como **Diretor Sênior de Engenharia,,** na **Dell Technologies**, onde liderou a **Unidade de Negócios *Big Data* na Nuvem**, iderando equipes nos **Estados Unidos** e na **China** no gerenciamento de grandes projetos e na reestruturação de divisões de negócios para uma integração bem-sucedida. Ele também trabalhou como **Diretor de Tecnologias da Informação (*Chief Information Officer*)** na **XanGo**, onde gerenciou projetos como **soporte de *Help Desk***, e **suporte de produção** e **desenvolvimento de soluções**.

Entre as muitas especialidades em que ele é especialista estão a tecnologia ***Edge to Cloud***, a **cibersegurança**, a **inteligência artificial generativa**, o **desenvolvimento de software**, a **tecnologia de redes**, o **desenvolvimento nativo na nuvem** e o **ecossistema de contêineres**. O conhecimento que ele compartilha por meio do **podcast** e **boletim semanal** ***"Embracing Digital Transformation"***, que ele produziu e apresentou, ajuda as organizações a navegar com sucesso **pela transformação digital** aproveitando as **pessoas**, os **processos** e a **tecnologia**.

Sr. Darren Pulsipher

- Arquiteto-chefe de soluções para o setor público da Intel, Califórnia, EUA
- Apresentador e produtor de *"Embracing Digital Transformation"*, Califórnia
- Fundador e CEO da Yoly Inc., Arkansas
- Diretor Sênior de Engenharia da Dell Technologies, Arkansas
- Diretor de Tecnologia da Informação (*Chief Information Officer*) em XanGo, Utah
- Arquiteto sênior na Cadence Design Systems, Califórnia
- Gerente sênior de processos de projetos na Lucent Technologies, Califórnia
- Engenheiro de software na Cemax-Icon, Califórnia
- Engenheiro de software da ISG Technologies, Canadá
- MBA em Gestão de Tecnologia pela University of Phoenix
- Formado em Ciência da Computação e Engenharia Elétrica pela Universidade Brigham Young

Graças à TECH você será capaz de aprender com os melhores profissionais do mundo"

Direção

Sr. Jerónimo Molina Molina

- Responsável pela Inteligência Artificial no Helphone
- AI Engineer & Software Architect, NASSAT-Internet Satélite em movimento
- Consultor Sênior em Hexa Engenheiros
- Introdutor de Inteligência Artificial (ML e CV)
- Especialista em Soluções Baseadas em Inteligência Artificial nas áreas de Computer Vision, ML/DL y NLP
- Especialista universitário em criação e desenvolvimento de negócios na Bancaixa e na Fundeun
- Engenheiro em Informática, Universidade de Alicante
- Mestrado em Inteligência artificial, Universidade Católica de Ávila
- Executivo de MBA no Fórum Europeu de Campus de Negócios

Professores

Sra. Cándida Rodríguez Míguez

- Desenvolvedor de aplicativos júnior em Getronics
- Cofundadora e líder municipal da rede de IA da Galícia
- Engenheira de software júnior na Indra
- Desenvolvimento web na EDISA
- Formada em Engenharia da Computação pela Universidade de Vigo.
- Mestrado em Engenharia da Computação pela Universidade de Vigo

Sr. Oriol Pi Morell

- Analista Funcional da Fihoca
- Product Owner de Hosting e Mail
- Analista funcional e Software Engineer da Atmira e CapGemini
- Professor na ORACLE Forms CapGemini e na Atmira
- Formado em Engenharia Técnica de Informática de Gestão pela Universidade Autónoma de Barcelona
- Mestrado em Inteligência artificial, Universidade Católica de Ávila
- MBA em Gestão e Administração de Empresas pela IMF Smart Education
- Mestrado em Direção de Sistemas de Informação pela IMF Smart Education
- Pós-graduação em em Padrões de Design pela Universitat Oberta de Catalunya

Sr. Francisco Javier Rodríguez

- Engenheiro Técnico Industrial especialização em Eletricidade Eletrônica
- Técnico de software na HEXA Ingenieros
- Desenvolvedor .Net Sênior / Arquiteto de Soluções Net na Everis
- Analista/Arquiteto *Software* na LaLiga
- Engenheiro On-site da Microsoft no BBVA
- Consultor Técnico-Informático *Freelance*
- Capacitador em Visual Studio, SqlServer, CCNA (roteadores e switch Cisco), Programação web PHP e . puramente em vários centros (Salesianos, Maforem, Dreamsoft).
- Engenheiro Técnico Industrial Cone especialização em Eletricidade, Eletrônica Industrial
- Mestre em Cibernética em .NET, MCAD
- Mestrado da Eidos em Programação Avançada, Nível Expert
- Mestre Web com certificações Dreamweaver, Fireworks, Flash e ActionScript, versões MX

Sr. Marcos Tenrero Morán

- Engenheiro de DevOps na Allot Communications
- Gerente de gerenciamento de ciclo de vida de aplicativos na Cegid Meta4
- Engenheiro de automação de controle de qualidade na Cegid Meta4
- Mestrado em Desenvolvimento de Aplicativos profissionais para Android pela Universidade Galileo, Guatemala.
- Mestrado em Desenvolvimento de Serviços em Nuvem, Node.Js, JavaScript, HTML5 pela Universidad Politécnica de Madrid
- Desenvolvimento Web com Angular-CLI (4), Ionic e Node.Js, Meta4 pela Universidad Rey Juan Carlos
- Formado em Engenharia da Computação pela Universidade Rey Juan Carlos

Sra. Patricia Acebes Tamargo

- Consultora especializada em Big Data
- Departamento de Operações, trabalha com Elasticsearch e Kivana
- Pesquisadora online na *Human Factor* e AI *Aplications* na CTIC Centro de Tecnologia
- Pesquisadora online da Unidade de Negócios do CTIC Centro Tecnológico
- Departamento de Saúde Digital e Envelhecimento Ativo do Centro de Tecnologia CTIC
- Departamento de Ciência de Dados do Centro de Tecnologia CTIC
- PhD em Ciência da Computação em Inteligência Artificial pela Universidade Politécnica de Valência.
- Formada em Economia pela Universidade de Oviedo
- Mestrado em Análise de Dados UCJC
- Mestrado em Inteligência Artificial em Pesquisa na UNED
- Mestrado em *Blockchain, Smart Contracts* e Criptomonedas pela Universidade de Alcalá
- Pós-graduação em Engenharia de *Blockchain* pela EADA
- Mestrado em Economia, Instrumentos, Análise Econômica pela Universidade de Oviedo
- Mestrado em em Fiscalidade Colegio de Economistas

08

Certificação

O Advanced Master em Engenharia e Qualidade de Software garante, além da capacitação mais rigorosa e atualizada, o acesso a um título de Advanced Master emitido pela TECH Universidade Tecnológica

Conclua este programa de estudos com sucesso e receba o seu certificado sem sair de casa e sem burocracias”

Este **Advanced Master em Engenharia e Qualidade de Software** conta com o conteúdo científico mais completo e atualizado do mercado.

Uma vez aprovadas as avaliações, o aluno receberá por correio o certificado* correspondente ao título de **Advanced Master** emitido pela **TECH Universidade Tecnológica.**

O certificado emitido pela **TECH Universidade Tecnológica** expressará a qualificação obtida no Advanced Master, atendendo aos requisitos normalmente exigidos pelas bolsas de empregos, concursos públicos e avaliação de carreira profissional.

Título: **Advanced Master em Engenharia e Qualidade de Software**

Modalidade: **online**

Duração: **2 anos**

tech universidade tecnológica

Outorga o presente

CERTIFICADO

a

Sr./Sra. _____, com documento de identidade nº_____

por ter concluído e aprovado com sucesso o programa de

ADVANCED MASTER

em

Engenharia e Qualidade de Software

Este é um curso próprio desta Universidade, com duração de 3.000 horas, com data de início dd/mm/aaaa e data final dd/mm/aaaa.

A TECH é uma Instituição Particular de Ensino Superior reconhecida pelo Ministério da Educação Pública em 28 de junho de 2018.

Em 17 de junho de 2020

Ma.Tere Guevara Navarro
Reitora

código único TECH: AFWOR23S techtitute.com/titulos

Advanced Master em Engenharia e Qualidade de Software

Conteúdo programático

Curso	Disciplina	Horas	Tipo
1º	Qualidade do Software Níveis de Desenvolvimento TRL	150	OB
1º	Desenvolvimento de Projetos Software Documentação Funcional e Técnica	150	OB
1º	Testing de Software Automação de Testes	150	OB
1º	Metodologias de Gestão de Projetos Software Metodologias Waterfall comparadas com Metodologias Ágeis	150	OB
1º	TDD (Test Driven Developement). Projeto de Software guiado por Testes	150	OB
1º	DevOps Gestão de Qualidade do Software	150	OB
1º	DevOps e Integração Contínua Soluções Práticas Avançadas no Desenvolvimento de Software	150	OB
1º	Design de Bases de Dados (BD) Normalização e Rendimento Qualidade de software	150	OB
1º	Projeto de Arquitetura Escalável A Arquitetura no Ciclo de Vida do Software	150	OB

Curso	Disciplina	Horas	Tipo
2º	Critérios de Qualidade ISO, IEC 9126 Métricas de qualidade de software	150	OB
2º	Metodologias, Desenvolvimento e Qualidade em Engenharia de Software	150	OB
2º	Gerenciamento de projetos de Software	150	OB
2º	Plataformas de desenvolvimento de software	150	OB
2º	Computação de clientes Web	150	OB
2º	Computação de Servidor Web	150	OB
2º	Gestão da Segurança	150	OB
2º	Segurança de software	150	OB
2º	Administração do Servidor Web	150	OB
2º	Auditoria de Segurança	150	OB
2º	Segurança de Aplicações Online	150	OB

Ma.Tere Guevara Navarro
Reitora

tech universidade tecnológica

*Apostila de Haia: Caso o aluno solicite que seu certificado seja apostilado, a TECH EDUCATION providenciará a obtenção do mesmo a um custo adicional.

tech universidade tecnológica

Advanced Master
Engenharia e Qualidade de Software

- Modalidade: **online**
- Duração: **2 anos**
- Certificado: **TECH Universidade Tecnológica**
- Horário: **no seu próprio ritmo**
- Provas: **online**

Advanced Master

Engenharia e Qualidade de Software

tech universidade tecnológica