

# Mestrado Avançado

## Engenharia e Qualidade do Software





## Mestrado Avançado Engenharia e Qualidade do Software

- » Modalidade: online
- » Duração: 2 anos
- » Certificação: TECH Global University
- » Acreditação: 120 ECTS
- » Horário: ao seu próprio ritmo
- » Exames: online

Acesso ao site: [www.techtute.com/pt/informatica/mestrado-avancado/mestrado-avancado-engenharia-qualidade-software](http://www.techtute.com/pt/informatica/mestrado-avancado/mestrado-avancado-engenharia-qualidade-software)

# Índice

01

Apresentação do programa

pág. 4

02

Porquê estudar na TECH?

pág. 8

03

Plano de estudos

pág. 12

04

Objetivos de ensino

pág. 34

05

Oportunidades de carreira

pág. 40

06

Metodología de estudo

pág. 44

07

Corpo docente

pág. 54

08

Certificação

pág. 60

01

# Apresentação do programa

A Engenharia de Software tornou-se no pilar fundamental da transformação digital. Atualmente, todas as indústrias dependem de soluções tecnológicas para otimizar processos, melhorar a experiência do cliente e manter-se competitivas. A qualidade do Software, por sua vez, garante que essas soluções sejam confiáveis, escaláveis e seguras. Esta disciplina é uma ramificação da engenharia que combina conhecimentos técnicos e de gestão para garantir que os produtos e sistemas desenvolvidos sejam funcionais e sustentáveis. Por essa razão, este programa vai além da simples programação, focando-se em todo o ciclo de vida do Software, desde a concepção inicial até à manutenção e evolução do sistema. O objetivo principal é oferecer aos alunos uma oportunidade académica única que os coloque na vanguarda da tecnologia. A TECH desenvolveu esta qualificação multidisciplinar e 100% online, que aborda desde os fundamentos da engenharia de Software até às últimas tendências em metodologias ágeis.

```
...etr.getStrings[0]  
if (settings[0]  
name.compare  
name += "  
}  
name += Date  
if (set  
me.
```

“

*Junte-se agora e comece a transformar  
ideias complexas em soluções tecnológicas  
tangíveis que terão um impacto no mundo”*

A qualidade do Software garante que os sistemas não só cumpram os requisitos funcionais, mas também sejam intuitivos, seguros e sustentáveis a longo prazo. Este aspeto é especialmente relevante em setores críticos, como o financeiro, a saúde ou o transporte, onde as falhas podem resultar em consequências graves. Além disso, ao priorizar a qualidade, assegura-se que as empresas possam adaptar-se de forma ágil aos constantes avanços tecnológicos e responder eficazmente às crescentes exigências do mercado.

Mediante o uso de metodologias como o desenvolvimento ágil, *DevOps* e a implementação de normas internacionais de qualidade, a engenharia de Software garante a entrega de produtos em prazos mais curtos. Adicionalmente, os custos controlados e o nível de qualidade que minimiza erros críticos foram ampliados com a integração de tecnologias emergentes, como a inteligência artificial, *cloud computing* e cibersegurança. Neste contexto, o programa que a TECH desenhou está orientado para formar profissionais altamente capacitados no desenho, desenvolvimento, gestão e garantia da qualidade do Software. Para adquirir as competências necessárias, o plano de estudos do Mestrado Avançado inclui os conceitos mais atualizados sobre gestão de projetos tecnológicos e direção estratégica. Este enfoque representa um valor acrescentado tanto para engenheiros que já ocupam cargos de responsabilidade e desejam atualizar os seus conhecimentos, como para aqueles que aspiram a liderar equipas e projetos pela primeira vez neste âmbito.

Uma das principais vantagens que este programa oferece é que será 100% online, o que elimina a necessidade de deslocações e de ajustar-se a horários específicos. Além disso, os estudantes contarão com o método de aprendizagem *Relearning*, que se adapta ao seu ritmo de estudo. Este enfoque flexível é de grande utilidade, pois permite aos alunos organizar as suas obrigações diárias, sejam profissionais ou familiares, de forma eficiente, alcançando assim um desenvolvimento completo.

Este **Mestrado Avançado em Engenharia e Qualidade do Software** conta com o conteúdo educacional mais completo e atualizado do mercado. As suas principais características são:

- ♦ O desenvolvimento de casos práticos apresentados por especialistas em Informática
- ♦ Os conteúdos gráficos, esquemáticos e eminentemente práticos, concebidos para oferecer uma informação científica e prática sobre as disciplinas indispensáveis para o exercício profissional
- ♦ Os exercícios práticos onde o processo de autoavaliação pode ser efetuado a fim de melhorar a aprendizagem
- ♦ O seu foco especial em metodologias inovadoras na Engenharia e Qualidade do Software
- ♦ As lições teóricas, perguntas aos especialistas, fóruns de discussão sobre temas controversos e trabalhos de reflexão individual
- ♦ A disponibilidade de acesso aos conteúdos a partir de qualquer dispositivo fixo ou portátil com conexão à Internet



*Com a TECH, não aprenderá só a desenvolver Software, mas a criar sistemas que fazem a diferença na vida das pessoas e das empresas”*

“

*Dominar as competências e ferramentas de engenharia mais avançadas com a metodologia de ensino mais inovadora do panorama académico atual”*

O corpo docente do programa inclui profissionais do setor informático que trazem para este programa a experiência do seu trabalho, bem como especialistas reconhecidos de empresas de referência e universidades de prestígio.

O seu conteúdo multimédia, elaborado com a mais recente tecnologia educativa, permitirá ao profissional uma aprendizagem situada e contextual, ou seja, um ambiente simulado que proporcionará um estudo imersivo programado para treinar-se em situações reais.

O design deste plano de estudos está centrado na Aprendizagem Baseada em Problemas, através da qual o aluno terá de tentar resolver as diversas situações de prática profissional que lhe serão apresentadas ao longo do curso académico. Para tal, o profissional contará com a ajuda de um sistema inovador de vídeo interativo desenvolvido por especialistas reconhecidos.

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

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



02

# Porquê 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 línguas, posiciona-se como líder em empregabilidade, com uma taxa de colocação profissional de 99%. Além disso, possui um enorme corpo docente de mais de 6.000 professores de renome internacional.



“

*Estuda na maior universidade digital do mundo e garante o teu sucesso profissional. O futuro começa na TECH”*

**A melhor universidade online do mundo segundo a FORBES**

A prestigiada revista Forbes, especializada em negócios e finanças, destacou a TECH como «a melhor universidade online do mundo». Foi o que afirmaram recentemente num artigo da sua edição digital, no qual fazem eco da história de sucesso desta instituição, «graças à oferta académica que proporciona, à seleção do seu corpo docente e a um método de aprendizagem inovador destinado a formar os profissionais do futuro».

**Forbes**  
Mejor universidad  
online del mundo

**Plan**  
de estudios  
más completo

**Os planos de estudos mais completos do panorama universitário**

A TECH oferece os planos de estudos mais completos do panorama universitário, com programas que abrangem os conceitos fundamentais e, ao mesmo tempo, os principais avanços científicos nas suas áreas científicas específicas. Além disso, estes programas são continuamente atualizados para garantir aos estudantes a vanguarda académica e as competências profissionais mais procuradas. Desta forma, os cursos da universidade proporcionam aos seus alunos uma vantagem significativa para impulsionar as suas carreiras com sucesso.

**O melhor corpo docente top internacional**

O corpo docente da TECH é composto por mais de 6.000 professores de renome internacional. Professores, investigadores e quadros superiores de multinacionais, incluindo Isaiah Covington, treinador de desempenho dos Boston Celtics; Magda Romanska, investigadora principal do Harvard MetaLAB; Ignacio Wistumba, presidente do departamento de patologia molecular translacional do MD Anderson Cancer Center; e D.W. Pine, diretor criativo da revista TIME, entre outros.

Profesorado  
**TOP**  
Internacional



La metodología  
más eficaz

**Um método de aprendizagem único**

A TECH é a primeira universidade a utilizar o *Relearning* em todos os seus cursos. É a melhor metodologia de aprendizagem online, acreditada com certificações internacionais de qualidade de ensino, fornecidas por agências educacionais de prestígio. Além disso, este modelo académico disruptivo é complementado pelo “Método do Caso”, configurando assim uma estratégia única de ensino online. São também implementados recursos didáticos inovadores, incluindo vídeos detalhados, infografias e resumos interativos.

**A maior universidade digital do mundo**

A TECH é a maior universidade digital do mundo. Somos a maior instituição educativa, com o melhor e mais extenso catálogo educativo digital, cem por cento online e abrangendo a grande maioria das áreas do conhecimento. Oferecemos o maior número de títulos próprios, pós-graduações e licenciaturas oficiais do mundo. No total, são mais de 14.000 títulos universitários, em onze línguas diferentes, o que nos torna a maior instituição de ensino do mundo.

**nº1**  
Mundial  
Mayor universidad  
online del mundo

#### A universidade online oficial da NBA

A TECH é a Universidade Online Oficial da NBA. Através de um acordo com a maior liga de basquetebol, oferece aos seus estudantes programas universitários exclusivos, bem como uma grande variedade de recursos educativos centrados no negócio da liga e noutras áreas da indústria desportiva. Cada programa tem um plano de estudos único e conta com oradores convidados excepcionais: profissionais com um passado desportivo distinto que oferecem os seus conhecimentos sobre os temas mais relevantes.

#### Líderes em empregabilidade

A TECH conseguiu tornar-se a universidade líder em empregabilidade. 99% dos seus estudantes conseguem um emprego na área académica que estudaram, no prazo de um ano após a conclusão de qualquer um dos programas da universidade. Um número semelhante consegue uma melhoria imediata da sua carreira. Tudo isto graças a uma metodologia de estudo que baseia a sua eficácia na aquisição de competências práticas, absolutamente necessárias para o desenvolvimento profissional.



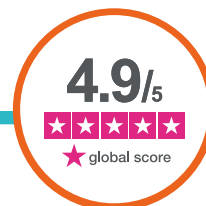
#### Google Partner Premier

O gigante tecnológico americano atribuiu à TECH o distintivo Google Partner Premier. Este prémio, que só está disponível para 3% das empresas no mundo, destaca a experiência eficaz, flexível e adaptada que esta universidade proporciona aos estudantes. O reconhecimento não só acredita o máximo rigor, desempenho e investimento nas infra-estruturas digitais da TECH, mas também coloca esta universidade como uma das empresas de tecnologia mais avançadas do mundo.



#### A universidade mais bem classificada pelos seus alunos

Os alunos posicionaram a TECH como a universidade mais bem avaliada do mundo nos principais portais de opinião, destacando a sua classificação máxima de 4,9 em 5, obtida a partir de mais de 1.000 avaliações. Estes resultados consolidam a TECH como uma instituição universitária de referência internacional, refletindo a excelência e o impacto positivo do seu modelo educativo.



# 03

## Plano de estudos

O plano de estudos do Mestrado Avançado em Engenharia e Qualidade do Software está desenhado para proporcionar uma especialização integral e avançada em todas as áreas chave da engenharia de Software. Os primeiros módulos centram-se nos fundamentos, abrangendo desde o design de Software e a gestão de requisitos até às arquiteturas tecnológicas e metodologias ágeis. À medida que o programa avança, os alunos mergulham em áreas mais especializadas, como a automatização de testes, a integração contínua e a garantia da qualidade. Além disso, são incorporadas disciplinas sobre gestão de projetos tecnológicos, onde os participantes aprendem a liderar equipas multidisciplinares.





“

*Este Mestrado Avançado prepara-o para ser o especialista que faz a diferença no setor da Engenharia e Qualidade do 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 gerais
  - 1.1.3. Softwares sem princípios e com princípios de qualidade
    - 1.1.3.1. Consequências
    - 1.1.3.2. A necessidade de aplicar princípios de qualidade no Software
  - 1.1.4. Qualidade do Software. Tipologia
  - 1.1.5. Software de Qualidade. Traços específicos
- 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 erradas
  - 1.2.3. Qualidade do Software. Custos associados
- 1.3. Modelos de qualidade do Software I. Gestão do conhecimento
  - 1.3.1. Modelos de qualidade gerais
    - 1.3.1.1. Gestão da qualidade total
    - 1.3.1.2. Modelo Europeu de Excelência Empresarial (EFQM)
    - 1.3.1.3. Modelo Six-sigma
  - 1.3.2. Modelos de gestão de 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 na utilização (25010)
- 1.4. Modelos de qualidade do Software (III). Qualidade em dados, processos e modelos SEI
  - 1.4.1. Modelo de qualidade de dados
  - 1.4.2. Modelo do processo Software
  - 1.4.3. Software & Systems Process Engineering Metamodel Specification (SPEM)
  - 1.4.4. Modelos do SEI
    - 1.4.4.1. CMMI
    - 1.4.4.2. SCAMPI
    - 1.4.4.3. IDEAL
- 1.5. Normas ISO de Qualidade do Software (I). Análises das normas
  - 1.5.1. Normas ISO 9000
    - 1.5.1.1. Normas ISO 9000
    - 1.5.1.2. Família ISO de Normas de Qualidade (9000)
  - 1.5.2. Outras normas ISO relacionadas com Qualidade
  - 1.5.3. Normas de Modelação de Qualidade (ISO 2501)
  - 1.5.4. Normas de Medida da Qualidade (ISO 2502n)
- 1.6. Normas ISO de Qualidade do Software (II). Requisitos e avaliação
  - 1.6.1. Normas sobre Requisitos de Qualidade (2503n)
  - 1.6.2. Normas sobre Avaliação da Qualidade (2504n)
  - 1.6.3. ISO/IEC 24744: 2007
- 1.7. Níveis de desenvolvimento TRL (I). Níveis do 1 ao 4
  - 1.7.1. Níveis 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 componente em ambiente de laboratório
- 1.8. Níveis de desenvolvimento TRL (II). Níveis do 5 ao 9
  - 1.8.1. Nível 5: validação de componente em ambiente relevante
  - 1.8.2. Nível 6: modelo 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 empresa I+D+I
  - 1.9.3. Exemplo de empresa I+D+I industrial
  - 1.9.4. Exemplo de empresa mista laboratório-engenharia
- 1.10. Qualidade do Software. Detalhes chave
  - 1.10.1. Detalhes metodológicos
  - 1.10.2. Detalhes técnicos
  - 1.10.3. Detalhes da 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 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. Gestão de projetos. Vantagens
  - 2.1.3. Gestão de projetos. Tipologia
- 2.2. Metodologia na gestão do projeto
  - 2.2.1. Metodologia na gestão do projeto
  - 2.2.2. Metodologias de projetos. Tipologia
  - 2.2.3. Metodologia na gestão de projetos. Aplicação
- 2.3. Fase de Identificação dos Requisitos
  - 2.3.1. Identificação dos requisitos de um projeto
  - 2.3.2. Gestão das reuniões de um projeto
  - 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 utilizar
  - 2.5.1. Determinação do novo Modelo de Dados
  - 2.5.2. Identificação do Plano de Migração de Dados
  - 2.5.3. Jogo de dados
- 2.6. Repercussões noutros projetos
  - 2.6.1. Repercussão 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 equipa de construção do projeto
  - 2.8.1. Papéis a intervir de acordo com o projeto
  - 2.8.2. Contacto com RH para contratação
  - 2.8.3. Entregáveis e calendários 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. Documentos do projeto a entregar
  - 2.10.1. Análise funcional
  - 2.10.2. Modelos de dados
  - 2.10.3. Diagramas de estados
  - 2.10.4. Documentação técnica

## Módulo 3. *Testing* de Software. Automatização 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 maturação
  - 3.2.3. Normativa ISO 15504
    - 3.2.3.1. Propósitos
    - 3.2.3.2. Contexto
    - 3.2.3.3. Etapas
- 3.3. Normativa 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. Etapas
- 3.4. CMMI (*Capability Maturity Model Integration*)
  - 3.4.1. CMMI Integração de Modelos de Maturação de Capacidades
  - 3.4.2. Modelo e áreas. Tipologia
  - 3.4.3. Áreas de processo
  - 3.4.4. Níveis de Capacidade
  - 3.4.5. Administração de processos
  - 3.4.6. Administração de projetos

- 3.5. Gestão de mudança e repositórios
  - 3.5.1. Gestão de mudanças em 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*
  - 3.5.2. Repositório
    - 3.5.2.1. Controlo de versões
    - 3.5.2.2. Equipa de trabalho e utilização 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 equipa
  - 3.6.3. Incorporação de conteúdo no controlo do código fonte
  - 3.6.4. *TFS on Cloud*
- 3.7. *Testing*
  - 3.7.1. Motivação para a realização de provas
  - 3.7.2. Provas de verificação
  - 3.7.3. Provas beta
  - 3.7.4. Implementação e manutenção
- 3.8. Provas de carga
  - 3.8.1. *Load testing*
  - 3.8.2. Provas com *LoadView*
  - 3.8.3. Provas com *K6 Cloud*
  - 3.8.4. Provas com *Loader*
- 3.9. Provas unitárias de stress e de resistência
  - 3.9.1. Motivação das provas unitárias
  - 3.9.2. Ferramentas para *unit testing*
  - 3.9.3. Motivação das provas de stress
  - 3.9.4. Provas usando *StressTesting*
  - 3.9.5. Motivação para as provas de resistência
  - 3.9.6. Provas usando *LoadRunner*
- 3.10. Escalabilidade. Desenho de Software escalável
  - 3.10.1. A escalabilidade e a arquitectura de Software
  - 3.10.2. A independência entre camadas
  - 3.10.3. O Acoplamento entre camadas. Padrões de arquitetura

#### Módulo 4. Metodologias de gestão de projetos Software. Metodologias Waterfall vs. 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 Ágil Influência na Qualidade do Software
  - 4.2.3. Metodologia Ágil Exemplos
- 4.3. Metodologia SCRUM
  - 4.3.1. Metodologia SCRUM
  - 4.3.2. Manifesto SCRUM
  - 4.3.3. Aplicação de SCRUM
- 4.4. Painel Kanban
  - 4.4.1. Método Kanban
  - 4.4.2. Painel Kanban
  - 4.4.3. Painel Kanban Exemplo de aplicação
- 4.5. Gestão do Projeto em Waterfall
  - 4.5.1. Fases num projeto
  - 4.5.2. Visão num projeto Waterfall
  - 4.5.3. Entregáveis a ter em conta
- 4.6. Gestão de projeto em SCRUM
  - 4.6.1. Fases num projeto SCRUM
  - 4.6.2. Visão num projeto SCRUM
  - 4.6.3. Entregáveis a considerar
- 4.7. Waterfall vs SCRUM. Comparação
  - 4.7.1. Abordagem de um projecto piloto
  - 4.7.2. Projeto aplicando Waterfall. Exemplo
  - 4.7.3. Projeto aplicando SCRUM. Exemplo
- 4.8. Visão do cliente
  - 4.8.1. Documentos num Waterfall
  - 4.8.2. Documentos num SCRUM
  - 4.8.3. Comparação

- 4.9. Estrutura de Kanban
  - 4.9.1. Histórias de Utilizador
  - 4.9.2. Backlog
  - 4.9.3. Análise de Kanban
- 4.10. Projetos Híbridos
  - 4.10.1. Construção do projeto
  - 4.10.2. Gestão de projeto
  - 4.10.3. Entregáveis a considerar

## Módulo 5.TDD (*Test Driven Development*). Desenho de software guiado pelas provas

- 5.1. TDD. *Test Driven Development*
  - 5.1.1. TDD. *Test Driven Development*
  - 5.1.2. TDD. Influência do TDD na Qualidade
  - 5.1.3. Design e desenvolvimento baseado em Provas. Exemplos
- 5.2. Ciclo de TDD
  - 5.2.1. Eleição de um requisito
  - 5.2.2. Realização de provas. Tipologias
    - 5.2.2.1. Provas unitárias
    - 5.2.2.2. Testes de Integração
    - 5.2.2.3. Provas *End To End*
  - 5.2.3. Verificação da prova. Falhas
  - 5.2.4. Criação da Implementação
  - 5.2.5. Execução das provas automatizadas
  - 5.2.6. Eliminação da duplicação
  - 5.2.7. Atualização da lista 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. Implementação falsa
  - 5.3.2. Implementação triangular
  - 5.3.3. Implementação óbvia
- 5.4. TDD. Uso. Vantagens e Desvantagens
  - 5.4.1. Vantagens de utilização
  - 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: Fazer um teste prévio que falhe antes de codificar em produção
  - 5.5.3. Regra 2: não escrever mais do que um teste unitário
  - 5.5.4. Regra 3: não escrever mais código do que o necessário
  - 5.5.5. Erros e anti-padrões a evitar numa TDD
- 5.6. Simulação de projeto real para usar TDD (I)
  - 5.6.1. Descrição geral do projeto (Empresa A)
  - 5.6.2. Aplicação da TDD
  - 5.6.3. Exercícios propostos
  - 5.6.4. Exercícios *Feedback*
- 5.7. Simulação de projeto real para usar TDD (II)
  - 5.7.1. Descrição geral do projeto (Empresa B)
  - 5.7.2. Aplicação da TDD
  - 5.7.3. Exercícios propostos
  - 5.7.4. Exercícios *Feedback*
- 5.8. Simulação de projeto real para usar TDD (III)
  - 5.8.1. Descrição geral do projeto (Empresa C)
  - 5.8.2. Aplicação da 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. Implementar DevOps
  - 6.3.1. Identificação de problemas
  - 6.3.2. Implementação numa empresa
  - 6.3.3. Métricas de Implementação
- 6.4. Ciclo de entrega de Software
  - 6.4.1. Métodos de Design
  - 6.4.2. Acordos
  - 6.4.3. Roteiro
- 6.5. Desenvolvimento de Código Livre de Erros
  - 6.5.1. Código de manutenção
  - 6.5.2. Padrões de desenvolvimento
  - 6.5.3. *Testing* de Código
  - 6.5.4. Desenvolvimento de Software a Nível de Código. Boas práticas
- 6.6. Automatização
  - 6.6.1. Automatização Tipos de provas
  - 6.6.2. Custo da automatização e manutenção
  - 6.6.3. Automatização Atenuando erros
- 6.7. Implementações
  - 6.7.1. Avaliação de objetivos
  - 6.7.2. Conceção 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 do incidente
  - 6.8.3. Como evitar erros futuros

- 6.9. Automatização de implantações
  - 6.9.1. Preparação para implantações automáticas
  - 6.9.2. Avaliação da saúde do processo automático
  - 6.9.3. Métricas e capacidade de voltar atrás
- 6.10. Boas práticas Evolução de DevOps
  - 6.10.1. Guia de boas práticas aplicando DevOps
  - 6.10.2. DevOps. Metodologia para a equipa
  - 6.10.3. Evitando nichos

## Módulo 7. DevOps e integração contínua Soluções práticas avançadas no desenvolvimento de Software

- 7.1. Fluxos da entrega de software
  - 7.1.1. Identificação de atores e artefatos
  - 7.1.2. Conceção do fluxo de entrega de Software
  - 7.1.3. Fluxo de entrega de Software. Requisitos entre etapas
- 7.2. Automatização de processos
  - 7.2.1. Integração contínua
  - 7.2.2. Implementação 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 provedores 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 nos pedidos de integração
- 7.5. Gestão de artefactos
  - 7.5.1. Artefatos e ciclo de Vida
  - 7.5.2. Sistemas de armazenamento e gestão de artefactos
  - 7.5.3. Segurança na gestão de Artefactos
- 7.6. Implementação contínua
  - 7.6.1. Implantação contínua como recipientes
  - 7.6.2. Implantação contínua com 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 Hooks* e *Tests Unitarios*
  - 7.7.4. O impacto da infraestrutura
- 7.8. Vulnerabilidade em recipientes
  - 7.8.1. Vulnerabilidade em recipientes
  - 7.8.2. Digitalização de imagens
  - 7.8.3. Relatórios periódicos e alertas

## Módulo 8. Desenho de Bases de Dados (BD). Normalização e rendimento. Qualidade do Software

- 8.1. Desenho de bases de dados
  - 8.1.1. Bases de Dados. Tipologia
  - 8.1.2. Bases de dados usados atualmente
    - 8.1.2.1. Relacionais
    - 8.1.2.2. Chave-Valor
    - 8.1.2.3. Baseadas em gráficos
  - 8.1.3. A Qualidade do dado
- 8.2. Desenho do Modelo Entidade-Relação (I)
  - 8.2.1. Modelo de Entidade-Relação. Qualidade e documentação
  - 8.2.2. Entidades
    - 8.2.2.1. Entidade forte
    - 8.2.2.2. Entidade débil
  - 8.2.3. Atributos
  - 8.2.4. Conjunto de Relações
    - 8.2.4.1. 1 a 1
    - 8.2.4.2. 1 a muitos
    - 8.2.4.3. Muitos a 1
    - 8.2.4.4. Muitos a muitos
  - 8.2.5. Chaves
    - 8.2.5.1. Chave primária
    - 8.2.5.2. Chave estrangeira
    - 8.2.5.3. Chave primária entidade débil
  - 8.2.6. Restrições
  - 8.2.7. Cardinalidade
  - 8.2.8. Herança
  - 8.2.9. Agregação
- 8.3. Modelo Entidade-Relação(II). Ferramentas
  - 8.3.1. Modelo Entidade-Relação. Ferramentas
  - 8.3.2. Modelo Entidade-Relação. Exemplo prático
  - 8.3.3. Modelo de Entidade-Relação viável
    - 8.3.3.1. Mostra visual
    - 8.3.3.2. Mostra em representação de tabelas
- 8.4. Normalização da Bases de Dados (BD) (I). Considerações em Qualidade do Software
  - 8.4.1. Normalização da BD e Qualidade
  - 8.4.2. Dependências
    - 8.4.2.1. Dependência funcional
    - 8.4.2.2. Propriedades da dependência funcional
    - 8.4.2.3. Propriedades deduzidas
  - 8.4.3. Chaves
- 8.5. Normalização da Bases de Dados (BD) (II). Formas Normais e Regras de Codd
  - 8.5.1. Formas 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 de Boyce-Codd (FNBC)
    - 8.5.1.5. Quarta Forma Normal (4FN)
    - 8.5.1.6. Quinta Forma Normal (5FN)
  - 8.5.2. Regras de Codd
    - 8.5.2.1. Regra 1: informação
    - 8.5.2.2. Regra 2: acesso garantido
    - 8.5.2.3. Regra 3: tratamento sistemático dos 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 vistas
    - 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. Exemplo prático

- 8.6. Armazém de dados / Sistema OLAP
  - 8.6.1. Armazém de dados
  - 8.6.2. Tabela de factos
  - 8.6.3. Tabela de dimensões
  - 8.6.4. Criação do sistema OLAP. Ferramentas
- 8.7. Rendimento da Base de Dados (BD)
  - 8.7.1. Optimização de Índices
  - 8.7.2. Optimização de consultas
  - 8.7.3. Partição de tabelas
- 8.8. Simulação de projeto real para Desenho BD (I)
  - 8.8.1. Descrição geral do projeto (Empresa A)
  - 8.8.2. Aplicação do desenho de bases de dados
  - 8.8.3. Exercícios propostos
  - 8.8.4. Exercícios propostos. *Feedback*
- 8.9. Simulação de projeto real para Desenho BD (II)
  - 8.9.1. Descrição geral do projeto (Empresa B)
  - 8.9.2. Aplicação do desenho de Bases de Dados
  - 8.9.3. Exercícios propostos
  - 8.9.4. Exercícios propostos *Feedback*
- 8.10. Relevância da Otimização de BD na Qualidade do Software
  - 8.10.1. Otimização do Design
  - 8.10.2. Otimização do Código de Consultas
  - 8.10.3. Otimização do Código de Procedimentos armazenados
  - 8.10.4. Influência dos *Triggers* na Qualidade do Software. Recomendações de uso

## Módulo 9. Desenho de arquiteturas escaláveis. A Arquitetura no ciclo de vida do Software

- 9.1. Desenho de arquiteturas escaláveis (I)
  - 9.1.1. Arquiteturas escaláveis
  - 9.1.2. Princípios de uma arquitetura escalável
    - 9.1.2.1. Confiável
    - 9.1.2.2. Escalável
    - 9.1.2.3. Sustentável
  - 9.1.3. Tipos de escalabilidade
    - 9.1.3.1. Vertical
    - 9.1.3.2. Horizontal
    - 9.1.3.3. Combinado

- 9.2. Arquiteturas DDD (*Domain-Driven Design*)
  - 9.2.1. O Modelo DDD. Orientação ao Domínio
  - 9.2.2. Camadas, partilha de responsabilidade e padrões de desenho
  - 9.2.3. Desacoplamento como base para a qualidade
- 9.3. Desenho de arquiteturas escaláveis (II). Benefícios, limitações e estratégias de desenho
  - 9.3.1. Arquitetura escalável Benefícios
  - 9.3.2. Arquitetura escalável Limitações
  - 9.3.3. Estratégias para o desenvolvimento de arquiteturas escaláveis (Tabela descritiva)
- 9.4. Ciclo de vida do Software (I). Etapas
  - 9.4.1. Ciclo de vida do Software
    - 9.4.1.1. Etapa de planificação
    - 9.4.1.2. Etapa de análise
    - 9.4.1.3. Etapa de desenho
    - 9.4.1.4. Etapa de implementação
    - 9.4.1.5. Etapa de provas
    - 9.4.1.6. Fase de instalação/implantação
    - 9.4.1.7. Etapa de uso e manutenção
- 9.5. Modelos de ciclos 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 vida do Software (II). Automatização
  - 9.6.1. Ciclos de vida de desenvolvimento de software. Soluções
    - 9.6.1.1. Integração e desenvolvimento contínuos (CI/CD)
    - 9.6.1.2. Metodologias Agile
    - 9.6.1.3. DevOps / Operações de Produção
  - 9.6.2. Tendências futuras
  - 9.6.3. Exemplos práticos
- 9.7. Arquitetura Software no ciclo de vida do Software
  - 9.7.1. Benefícios
  - 9.7.2. Limitações
  - 9.7.3. Ferramentas

- 9.8. Simulação de projeto real para desenho de arquitetura Software (I)
  - 9.8.1. Descrição geral do projeto (empresa A)
  - 9.8.2. Aplicação do Desenho de arquitetura do Software
  - 9.8.3. Exercícios propostos
  - 9.8.4. Exercícios propostos *Feedback*
- 9.9. Simulação de projeto real para desenho de arquitetura Software (II)
  - 9.9.1. Descrição geral do projeto (Empresa B)
  - 9.9.2. Aplicação do Desenho de arquitetura do Software
  - 9.9.3. Exercícios propostos
  - 9.9.4. Exercícios propostos *Feedback*
- 9.10. Simulação de projeto real para desenho de arquitetura Software (III)
  - 9.10.1. Descrição geral do projeto (Empresa C)
  - 9.10.2. Aplicação do Desenho de arquitetura do Software
  - 9.10.3. Exercícios propostos
  - 9.10.4. Exercícios propostos *Feedback*

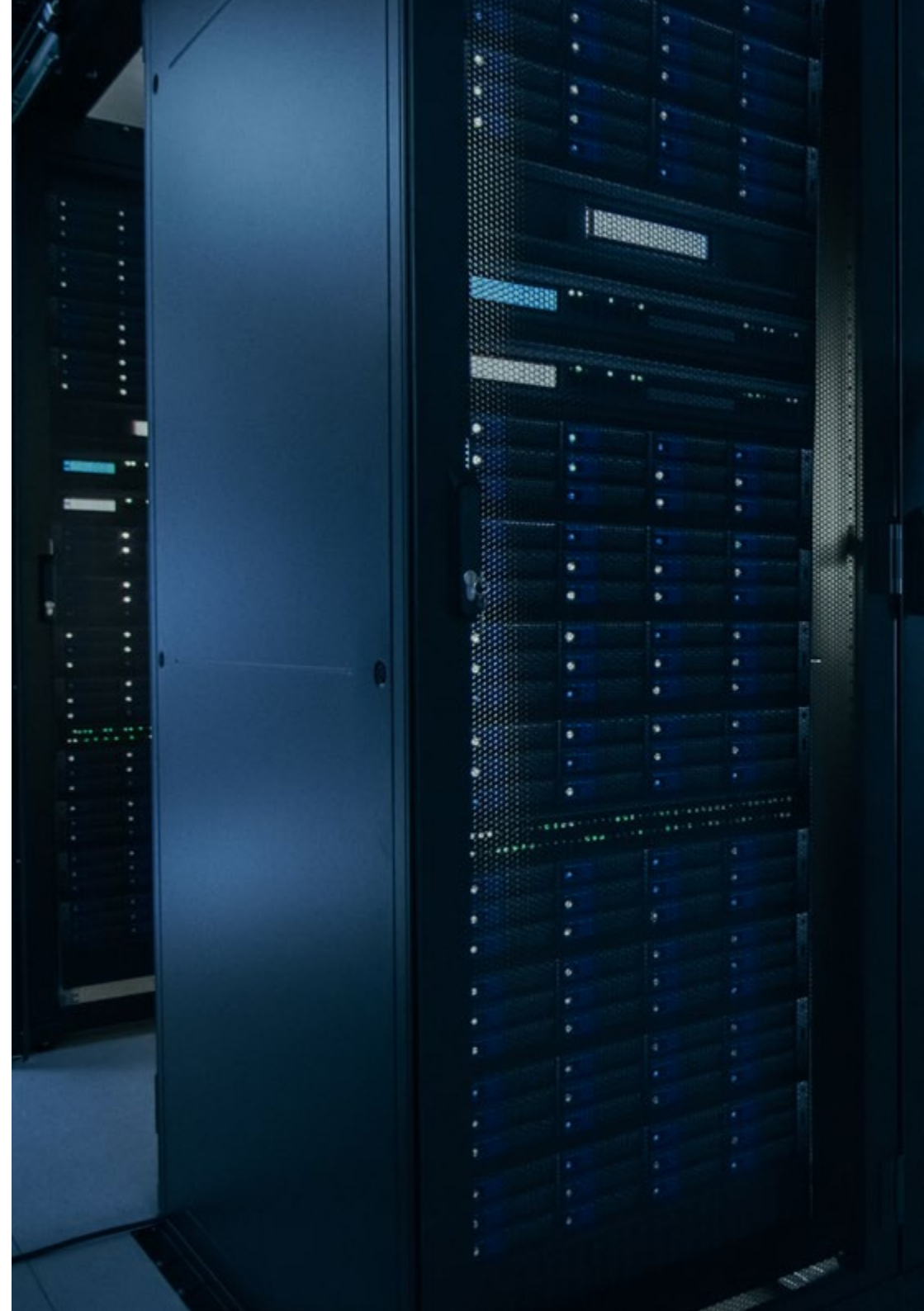
## Módulo 10. Critérios de Qualidade ISO, IEC 9126. Métrica de qualidade do Software

- 10.1. Critérios de qualidade. Norma ISO, IEC 9126
  - 10.1.1. Critérios de qualidade
  - 10.1.2. Qualidade do Software. Justificação Norma ISO, IEC 9126
  - 10.1.3. A Medição da qualidade do Software como indicador chave
- 10.2. Critérios de qualidade do Software. Características
  - 10.2.1. Fiabilidade
  - 10.2.2. Funcionalidade
  - 10.2.3. Eficiência
  - 10.2.4. Usabilidade
  - 10.2.5. Manutenibilidade
  - 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. Fiabilidade
  - 10.3.4. Usabilidade
  - 10.3.5. Manutenibilidade
  - 10.3.6. Portabilidade
  - 10.3.7. Qualidade em uso
  - 10.3.8. Métricas de qualidade do 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 intermédio Características
- 10.5. Métrica de qualidade do Software (I). Elementos
  - 10.5.1. Medida
  - 10.5.2. Métrica
  - 10.5.3. Indicador
    - 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. Medição de qualidade do Software (II). Prática da medição
  - 10.6.1. Recolha de dados métricos
  - 10.6.2. Medição de atributos internos do produto
  - 10.6.3. Medição de atributos externos do produto
  - 10.6.4. Medição de recursos
  - 10.6.5. Métricas para sistemas orientados a objetos
- 10.7. Desenho de um indicador único de qualidade do Software
  - 10.7.1. Indicador único como qualificador global
  - 10.7.2. Desenvolvimento do indicador, justificação e Aplicação
  - 10.7.3. Exemplo de aplicação. Necessidade conhecer o pormenor

- 10.8. Simulação de projeto real para medição 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 de projeto real para medição 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 de projeto real para medição 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. Modelação de objetos
  - 11.1.4. UML
  - 11.1.5. Ferramentas CASE
- 11.2. Modelação de aplicações e padrões de design com UML
  - 11.2.1. Modelação avançada de requisitos
  - 11.2.2. Modelação estática avançada
  - 11.2.3. Modelação dinâmica avançada
  - 11.2.4. Modelação 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. Metamodelação de sistemas
  - 11.3.3. MDA
  - 11.3.4. DSL
  - 11.3.5. Refinamentos 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 da ontologia
  - 11.4.3. Aplicação das ontologias na engenharia de Software

## Módulo 12. Gestão de projetos de software

- 12.1. A gestão dos *stakeholders* e do alcance
  - 12.1.1. Identificar os interessados
  - 12.1.2. Desenvolver o plano para a gestão dos interessados
  - 12.1.3. Gerir o compromisso das partes interessadas
  - 12.1.4. Controlar o compromisso das partes interessadas
  - 12.1.5. O objetivo do projeto
  - 12.1.6. A gestão do alcance e o seu plano
  - 12.1.7. Recolher os requisitos
  - 12.1.8. Definir a declaração do alcance
  - 12.1.9. Criar a WBS (EDT)
  - 12.1.10. Verificar e controlar o alcance
- 12.2. O desenvolvimento do cronograma
  - 12.2.1. A gestão do tempo e o seu plano
  - 12.2.2. Definir as atividades
  - 12.2.3. Estabelecimento da sequência das atividades
  - 12.2.4. Estimativa de recursos das atividades
  - 12.2.5. Estimativa da duração das atividades
  - 12.2.6. Desenvolvimento do cronograma e cálculo do caminho crítico
  - 12.2.7. Controlo do cronograma

- 12.3. O desenvolvimento do orçamento e a resposta ao riscos
  - 12.3.1. Estimar os custos
  - 12.3.2. Desenvolver o orçamento e a curva S
  - 12.3.3. Controlo de custos e método do valor ganho
  - 12.3.4. Os conceitos de risco
  - 12.3.5. Como fazer uma análise de riscos
  - 12.3.6. O desenvolvimento do plano de resposta
- 12.4. A comunicação e os recursos humanos
  - 12.4.1. Planificar a gestão das comunicações
  - 12.4.2. Análise de requisitos de comunicações
  - 12.4.3. Tecnologia das comunicações
  - 12.4.4. Modelos de comunicação
  - 12.4.5. Métodos de comunicação
  - 12.4.6. Plano de gestão das comunicações
  - 12.4.7. Gerir as comunicações
  - 12.4.8. A gestão dos recursos humanos
  - 12.4.9. Principais atores e os seus papéis nos projetos
  - 12.4.10. Tipos de organizações
  - 12.4.11. Organização do projeto
  - 12.4.12. A equipa de trabalho
- 12.5. O aprovisionamento
  - 12.5.1. O processo de aquisições
  - 12.5.2. Planejamento
  - 12.5.3. Procura de fornecedores e solicitação de ofertas
  - 12.5.4. Adjudicação do contrato
  - 12.5.5. Administração do contrato
  - 12.5.6. Os contratos
  - 12.5.7. Tipos de contratos
  - 12.5.8. Negociação do contrato
- 12.6. Execução, monitorização e controlo e encerramento
  - 12.6.1. Os grupos de processos
  - 12.6.2. A execução do projeto
  - 12.6.3. A monitorização e controlo do projeto
  - 12.6.4. O 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 deontológico do líder de projetos
  - 12.7.4. Responsabilidade vs. PMP®
  - 12.7.5. Exemplos de responsabilidade
  - 12.7.6. Benefícios da profissionalização

## Módulo 13. Plataformas de desenvolvimento do Software

- 13.1. Introdução ao desenvolvimento de aplicações
  - 13.1.1. Aplicações de ambiente de trabalho
  - 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 na cloud
- 13.2. Desenvolvimento de aplicações e interface gráfica em Java
  - 13.2.1. Ambientes de desenvolvimento integrados para Java
  - 13.2.2. Principais IDE para Java
  - 13.2.3. Introdução à plataforma de desenvolvimento Eclipse
  - 13.2.4. Introdução à plataforma de desenvolvimento NetBeans
  - 13.2.5. Modelo Vista Controlador para interfaces gráficas do utilizador
  - 13.2.6. Desenhar uma interface gráfica no Eclipse
  - 13.2.7. Desenhar uma interface gráfica no NetBeans
- 13.3. Depuração e provas em Java
  - 13.3.1. Provas e depuração de programas em 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 da plataforma de desenvolvimento .NET
  - 13.4.3. Visual Studio .NET
  - 13.4.4. Ferramentas de .NET para GUI
  - 13.4.5. A GUI com Windows Presentation Foundation
  - 13.4.6. Depurar e compilar uma aplicação WPF

- 13.5. Programação para redes .NET
  - 13.5.1. Introdução à programação para redes .NET
  - 13.5.2. Petições e respostas em .NET
  - 13.5.3. Utilização de protocolos de aplicação em .NET
  - 13.5.4. Segurança na programação para redes .NET
- 13.6. Ambientes de desenvolvimento de aplicações móveis
  - 13.6.1. Aplicações móveis
  - 13.6.2. Aplicações móveis Android
  - 13.6.3. Passos para o desenvolvimento em Android
  - 13.6.4. O IDE Android Studio
- 13.7. Desenvolvimento de aplicações no ambiente Android Studio
  - 13.7.1. Instalar e iniciar o Android Studio
  - 13.7.2. Execução de uma aplicação Android
  - 13.7.3. Desenvolvimento da interface gráfica no Android Studio
  - 13.7.4. Iniciar atividades no Android Studio
- 13.8. Depuração e publicação de aplicações Android
  - 13.8.1. Depuração de uma aplicação no Android Studio
  - 13.8.2. Memorizar aplicações no Android Studio
  - 13.8.3. Publicação de uma aplicação em Google Play
- 13.9. Desenvolvimento de aplicações para a cloud
  - 13.9.1. *Cloud computing*
  - 13.9.2. Níveis de *cloud*: SaaS, PaaS, IaaS
  - 13.9.3. Principais plataformas de desenvolvimento na cloud
  - 13.9.4. Referências bibliográficas
- 13.10. Introdução ao Google Cloud Platform
  - 13.10.1. Noções básicas do Google Cloud Platform
  - 13.10.2. Serviços do Google Cloud Platform
  - 13.10.3. Ferramentas do Google Cloud Platform

## Módulo 14. Computação para o cliente web

- 14.1. Introdução ao HTML
  - 14.1.1. Estrutura de um documento
  - 14.1.2. Cor
  - 14.1.3. Texto
  - 14.1.4. Ligações de hipertexto
  - 14.1.5. Imagens
  - 14.1.6. Listas
  - 14.1.7. Tabelas
  - 14.1.8. Quadros (*Frames*)
  - 14.1.9. Formulários
  - 14.1.10. Elementos específicos para tecnologias móveis
  - 14.1.11. Elementos em desuso
- 14.2. Folhas de estilo web (CSS)
  - 14.2.1. Elementos e estrutura de uma folha de estilos
    - 14.2.1.1. Criação de folhas de estilo
    - 14.2.1.2. Aplicação de estilos. Seletores
    - 14.2.1.3. Herança de estilos e aplicação em cascata
    - 14.2.1.4. Formatação de páginas usando estilos
    - 14.2.1.5. Estrutura de páginas utilizando estilos. O modelo de caixas
  - 14.2.2. Design de estilos para diferentes dispositivos
  - 14.2.3. Tipos de folhas de estilos: 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 que vamos utilizar
- 14.4. Noções básicas de programação web
  - 14.4.1. Sintaxe básica do JavaScript
  - 14.4.2. Tipos de dados primitivos e operadores
  - 14.4.3. Variáveis e âmbitos
  - 14.4.4. Cadeias de texto e *template literals*
  - 14.4.5. Números e booleanos
  - 14.4.6. Comparações

- 14.5. Estruturas complexas em JavaScript
  - 14.5.1. Vetores ou *arrays* e objetos
  - 14.5.2. Conjuntos
  - 14.5.3. Mapas
  - 14.5.4. Disjunções
  - 14.5.5. Loops
- 14.6. Funções e objetos
  - 14.6.1. Definição e invocação de funções
  - 14.6.2. Argumentos
  - 14.6.3. Arrow functions
  - 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 espaços de nomes: o objeto Math e o objeto Date
- 14.7. O modelo de objetos do documento (DOM)
  - 14.7.1. O que é o 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. Seletores de consulta 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. Atualização do estilo dos elementos do DOM
- 14.8. Desenvolvimento web moderno
  - 14.8.1. Fluxo baseado em eventos e *listeners*
  - 14.8.2. *Toolkits* web modernos e sistemas de alinhamento
  - 14.8.3. Modo estrito do JavaScript
  - 14.8.4. Algo 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 web
  - 14.9.1. Introdução à usabilidade
  - 14.9.2. Definição de usabilidade
  - 14.9.3. Importância do web design centrado no utilizador
  - 14.9.4. Diferenças entre acessibilidade e usabilidade
  - 14.9.5. Vantagens e problemas na combinação de acessibilidade e usabilidade
  - 14.9.6. Vantagens e dificuldades na implementação de websites utilizáveis
  - 14.9.7. Métodos de usabilidade
  - 14.9.8. Análise de requerimento de utilizador
  - 14.9.9. Princípios de design conceitual. Criação de protótipos orientados para o utilizador
  - 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 Bruce Tognazzini
  - 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. Deficiências visuais
    - 14.10.3.3. Deficiências auditivas
    - 14.10.3.4. Deficiências motoras
    - 14.10.3.5. Deficiências neurológicas ou cognitivas
    - 14.10.3.6. Dificuldades relacionadas com o envelhecimento
    - 14.10.3.7. Limitações decorrentes do ambiente
    - 14.10.3.8. Obstáculos ao acesso à Web
  - 14.10.4. Ajudas técnicas e produtos de apoio para superar barreiras
    - 14.10.4.1. Ajudas para as pessoas cegas
    - 14.10.4.2. Ajudas para pessoas com baixa visão
    - 14.10.4.3. Ajudas para pessoas daltónicas
    - 14.10.4.4. Ajudas para pessoas com deficiência auditiva
    - 14.10.4.5. Ajudas para pessoas com deficiência motora
    - 14.10.4.6. Ajudas para pessoas com deficiência cognitiva e neurológica

- 14.10.5. Vantagens e dificuldades na implementação da acessibilidade web
- 14.10.6. Regulamentos e normas de acessibilidade da Web
- 14.10.7. Organismos reguladores da acessibilidade da 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 uma navegação acessível
    - 14.10.9.2.1. Percetibilidade
    - 14.10.9.2.2. Operacionalidade
    - 14.10.9.2.3. Compreensibilidade
    - 14.10.9.2.4. Robustez
- 14.10.10. Descrição do processo de conformidade da acessibilidade da Web
- 14.10.11. Níveis de conformidade
- 14.10.12. Critérios de conformidade
- 14.10.13. Requisitos de conformidade
- 14.10.14. Metodologia de avaliação da acessibilidade dos websites

## Módulo 15. Computação para o servidor web

- 15.1. Introdução à programação no servidor: PHP
  - 15.1.1. Conceitos básicos de programação no servidor
  - 15.1.2. Sintaxe básica de PHP
  - 15.1.3. Geração de conteúdo HTML com PHP
  - 15.1.4. Ambientes de desenvolvimento e provas: XAMPP
- 15.2. PHP avançado
  - 15.2.1. Estruturas de controlo com PHP
  - 15.2.2. Funções em PHP
  - 15.2.3. Manipulação de *arrays* em PHP
  - 15.2.4. Manipulação de cadeias com PHP
  - 15.2.5. Orientação a objetos em PHP
- 15.3. Modelos de dados
  - 15.3.1. Conceito de dado. Ciclo de vida dos dados
  - 15.3.2. Tipos de dado
    - 15.3.2.1. Básicos
    - 15.3.2.2. Registos
    - 15.3.2.3. Dinâmicos
- 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. Relações: tipos, subtipos, cardinalidade
  - 15.4.5. Chaves. Tipos de chaves
  - 15.4.6. Normalização. Formas normais
- 15.5. Construção do modelo lógico de dados
  - 15.5.1. Especificação de tabelas
  - 15.5.2. Definição de colunas
  - 15.5.3. Especificação de chaves
  - 15.5.4. Conversão a formas normais. Dependências
- 15.6. O modelo físico de dados. Ficheiros de dados
  - 15.6.1. Descrição dos ficheiros de dados
  - 15.6.2. Tipos de ficheiros
  - 15.6.3. Modos de acesso
  - 15.6.4. Organização de ficheiros
- 15.7. Acesso a bases de dados a partir de PHP
  - 15.7.1. Introdução a MariaDB
  - 15.7.2. Trabalhar com uma base de dados MariaDB: a linguagem SQL
  - 15.7.3. Aceder à base de dados MariaDB a partir de PHP
  - 15.7.4. Introdução a MySQL
  - 15.7.5. Trabalhar com uma base de dados MySQL: a linguagem SQL
  - 15.7.6. Aceder à base de dados MySQL a partir de PHP
- 15.8. Interação com o cliente a partir de PHP
  - 15.8.1. Formulários PHP
  - 15.8.2. Cookies
  - 15.8.3. Gestão de sessões
- 15.9. Arquitetura de aplicações web
  - 15.9.1. O padrão Modelo Vista Controlador
  - 15.9.2. Controlador
  - 15.9.3. Modelo
  - 15.9.4. Vista

- 15.10. Introdução aos serviços web
  - 15.10.1. Introdução ao XML
  - 15.10.2. Arquiteturas orientadas para 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 um assunto económico
  - 16.1.4. A segurança é um processo
  - 16.1.5. A classificação da informação
  - 16.1.6. A segurança da informação implica a gestão de riscos
  - 16.1.7. A segurança é articulada com controlos de segurança
  - 16.1.8. A segurança é tanto física como lógica
  - 16.1.9. A segurança envolve pessoas
- 16.2. O profissional da segurança da informação
  - 16.2.1. Introdução
  - 16.2.2. A segurança da informação como profissão
  - 16.2.3. As certificações (ISC)2
  - 16.2.4. A norma ISO 27001
  - 16.2.5. Boas práticas de segurança na gestão de serviços TI
  - 16.2.6. Modelos de maturidade para a segurança da informação
  - 16.2.7. Outras certificações, normas e recursos profissionais
- 16.3. Controlo de acessos
  - 16.3.1. Introdução
  - 16.3.2. Requisitos de controlo de acessos
  - 16.3.3. Mecanismos de autenticação
  - 16.3.4. Métodos de autorização
  - 16.3.5. Contabilidade e auditoria de acessos
  - 16.3.6. Tecnologias Triple A
- 16.4. Programas, processos e políticas de segurança da informação
  - 16.4.1. Introdução
  - 16.4.2. Programa de gestão da segurança
  - 16.4.3. A gestão de riscos
  - 16.4.4. Conceção de políticas de segurança
- 16.5. Planos de continuidade de negócio
  - 16.5.1. Introdução às PCN
  - 16.5.2. Fase I e II
  - 16.5.3. Fase III e IV
  - 16.5.4. Manutenção do PCN
- 16.6. Procedimentos para a correta proteção da empresa
  - 16.6.1. Redes DMZ
  - 16.6.2. Sistemas de deteção de intrusos
  - 16.6.3. Listas de controlo de acessos
  - 16.6.4. Aprender com o atacante: Honeypot
- 16.7. Arquitetura de segurança. Prevenção
  - 16.7.1. Visão geral. Atividades e modelo de camadas
  - 16.7.2. Defesa perimetral (firewalls, WAFs, IPS, etc.)
  - 16.7.3. Defesa do ponto final (equipamento, servidores e serviços)
- 16.8. Arquitetura de segurança. Deteção
  - 16.8.1. Visão geral deteção e supervisão
  - 16.8.2. Logs, ruptura de tráfico encriptado, 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 Empresariais e Regulamentos
  - 16.10.2. Soluções IT de resiliência
  - 16.10.3. Gestão e governação das crises

## Módulo 17. Segurança no Software

- 17.1. Problemas da Segurança no Software
  - 17.1.1. Introdução ao problema da segurança no Software
  - 17.1.2. Vulnerabilidades e a sua classificação
  - 17.1.3. Propriedades Software seguro
  - 17.1.4. Referências
- 17.2. Princípios de concepção de segurança do software
  - 17.2.1. Introdução
  - 17.2.2. Princípios de concepção de segurança do software
  - 17.2.3. Tipos de S-SDLC
  - 17.2.4. Segurança do software nas fases do S-SDLC
  - 17.2.5. Metodologias e normas
  - 17.2.6. Referências
- 17.3. Segurança no ciclo de vida do Software nas fases de requisitos e desenho
  - 17.3.1. Introdução
  - 17.3.2. Modelação de ataques
  - 17.3.3. Casos de abuso
  - 17.3.4. Engenharia de requisitos de segurança
  - 17.3.5. Análise de risco. Arquitetónico
  - 17.3.6. Padrões de design
  - 17.3.7. Referências
- 17.4. Segurança no ciclo de vida do Software nas fases de codificação, provas e operação
  - 17.4.1. Introdução
  - 17.4.2. Provas de segurança baseadas no risco
  - 17.4.3. Revisão de código
  - 17.4.4. Teste de penetração
  - 17.4.5. Operações de segurança
  - 17.4.6. Revisão externa
  - 17.4.7. Referências
- 17.5. Codificação segura aplicações I
  - 17.5.1. Introdução
  - 17.5.2. Práticas de codificação segura
  - 17.5.3. Manipulação e validação de entradas
  - 17.5.4. Sobrecarga de memória
  - 17.5.5. Referências
- 17.6. Codificação segura aplicações 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 cloud
  - 17.7.1. Segurança no desenvolvimento; metodologia e prática
  - 17.7.2. Modelos IaaS, PaaS e SaaS
  - 17.7.3. Segurança na cloud e para serviços na cloud
- 17.8. Encriptação
  - 17.8.1. Fundamentos de criptologia
  - 17.8.2. Encriptação Simétrica e assimétrica
  - 17.8.3. Encriptação em repouso e em trânsito
- 17.9. Automatização e orquestração de segurança (SOAR)
  - 17.9.1. Complexidade do tratamento manual; necessidade de automatizar tarefas
  - 17.9.2. Produtos e serviços
  - 17.9.3. Arquitetura SOAR
- 17.10. Segurança no Teletrabalho
  - 17.10.1. Necessidade e cenários
  - 17.10.2. Produtos e serviços
  - 17.10.3. Segurança no teletrabalho

## Módulo 18. Administração de Servidores Web

- 18.1. Introdução a servidores web
  - 18.1.1. O que é um servidor web?
  - 18.1.2. Arquitetura e funcionamento de um servidor web
  - 18.1.3. Recursos e conteúdos num servidor web
  - 18.1.4. Servidores de aplicações
  - 18.1.5. Servidores Proxy
  - 18.1.6. Principais servidores web do mercado
  - 18.1.7. Estatística de uso servidores web
  - 18.1.8. Segurança em servidores web
  - 18.1.9. Equilíbrio de carga em servidores web
  - 18.1.10. Referências
- 18.2. Gestão do protocolo HTTP
  - 18.2.1. Funcionamento e estrutura
  - 18.2.2. Descrição de petições ou request methods
  - 18.2.3. Códigos de estado
  - 18.2.4. Cabeçalhos
  - 18.2.5. Codificação do conteúdo. Páginas de códigos
  - 18.2.6. Realização de petições HTTP na Internet utilizando um proxy, livehttpheaders ou método semelhante, analisando o protocolo utilizado
- 18.3. Descrição de arquiteturas distribuídas em múltiplos servidores
  - 18.3.1. Modelo de 3 camadas
  - 18.3.2. Tolerância a falhas
  - 18.3.3. Partilha de carga
  - 18.3.4. Armazéns de estado de sessão
  - 18.3.5. Armazéns de cache
- 18.4. Internet Information Services (IIS)
  - 18.4.1. O que é IIS?
  - 18.4.2. História e evolução de IIS
  - 18.4.3. Principais vantagens e características de IIS7 e posteriores
  - 18.4.4. Arquitetura IIS7 e posteriores
- 18.5. Instalação, administração e configuração de IIS
  - 18.5.1. Preâmbulo
  - 18.5.2. Instalação de Internet Information Services (IIS)
  - 18.5.3. Ferramentas de administração de IIS
  - 18.5.4. Criação, configuração e administração de websites
  - 18.5.5. Instalação e gestão de extensões em IIS
- 18.6. Segurança avançada em IIS
  - 18.6.1. Preâmbulo
  - 18.6.2. Autenticação, Autorização, e Controlo de Acesso em IIS
  - 18.6.3. Configuração de um website seguro em IIS com SSL
  - 18.6.4. Políticas de segurança implementada em IIS 8.x
- 18.7. Introdução a Apache
  - 18.7.1. O que é o Apache?
  - 18.7.2. Principais vantagens do Apache
  - 18.7.3. Características principais do Apache
  - 18.7.4. Arquitetura
- 18.8. Instalação e configuração do Apache
  - 18.8.1. Instalação inicial do apache
  - 18.8.2. Configuração de apache
- 18.9. Instalação e configuração dos diferentes módulos no Apache
  - 18.9.1. Instalação de módulos no Apache
  - 18.9.2. Tipos de módulos
  - 18.9.3. Configuração segura do Apache
- 18.10. Segurança avançada
  - 18.10.1. Autenticação, autorização e controlo 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 a sua auditoria
  - 19.1.1. Introdução aos sistemas de informação e o papel da auditoria informática
  - 19.1.2. Definições de auditoria informática e de controlo interno informático
  - 19.1.3. Funções e objetivos da auditoria informática
  - 19.1.4. Diferenças entre controlo interno e auditoria informática
- 19.2. Controlos internos dos sistemas de informação
  - 19.2.1. Organigrama funcional de um centro de processo de dados
  - 19.2.2. Classificação dos controlos dos sistemas de informação
  - 19.2.3. A regra de ouro
- 19.3. O processo e as fases da auditoria de sistemas de informação
  - 19.3.1. Avaliação de riscos (EDR) e outras metodologias de auditoria informática
  - 19.3.2. Execução de uma auditoria de Sistemas de Informação. Fases de auditoria
  - 19.3.3. Competências fundamentais do auditor de Sistemas de Informação
- 19.4. Auditoria técnica de segurança em sistemas e redes
  - 19.4.1. Auditorias técnicas de segurança. Teste de intrusão. Conceitos prévios
  - 19.4.2. Auditorias de segurança em sistemas. Ferramentas de apoio
  - 19.4.3. Auditorias de segurança em redes. Ferramentas de apoio
- 19.5. Auditoria técnica da segurança da Internet e dos dispositivos móveis
  - 19.5.1. Auditoria de segurança em Internet. Ferramentas de apoio
  - 19.5.2. Auditoria de segurança em dispositivos móveis. Ferramentas de apoio
  - 19.5.3. Anexo 1. Estrutura de relatório executivo e relatório técnico
  - 19.5.4. Anexo 2. Inventário de ferramentas
  - 19.5.5. Anexo 3. Metodologias
- 19.6. Sistema de gestão de segurança da informação
  - 19.6.1. Segurança dos SI: propriedades e fatores de influência
  - 19.6.2. Riscos empresariais e gestão de riscos: implementação de controlos
  - 19.6.3. SG da Segurança da Informação (SGSI): conceito e fatores críticos para o sucesso
  - 19.6.4. SGSI-Modelo PDCA
  - 19.6.5. SGSI ISO-IEC 27001: contexto da organização
  - 19.6.6. Contexto da organização
  - 19.6.7. Liderança
  - 19.6.8. Planejamento

- 19.6.9. Suporte
- 19.6.10. Funcionamento
- 19.6.11. Avaliação do desempenho
- 19.6.12. Melhoria
- 19.6.13. Anexo ao ISO 27001/ISO-IEC 27002: objetivos e controlos
- 19.6.14. Auditoria do SGSI
- 19.7. Realização da auditoria
  - 19.7.1. Procedimentos
  - 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. Relatórios e apresentação de provas
  - 19.10.1. Tipos de relatórios
  - 19.10.2. Análises dos dados
  - 19.10.3. Apresentação das provas

## Módulo 20. Segurança das Aplicações Online

- 20.1. Vulnerabilidades e problemas de segurança nas aplicações online
  - 20.1.1. Introdução à segurança nas aplicações online
  - 20.1.2. Vulnerabilidades de segurança na conceção das aplicações web
  - 20.1.3. Vulnerabilidades de segurança na implementação das aplicações web
  - 20.1.4. Vulnerabilidades de segurança na implementação das aplicações web
  - 20.1.5. Listas oficiais de vulnerabilidades de segurança
- 20.2. Políticas e normas para a segurança das aplicações online
  - 20.2.1. Pilares para a segurança das aplicações online
  - 20.2.2. Políticas de segurança
  - 20.2.3. Sistema de gestão de segurança da informação
  - 20.2.4. Ciclo de vida do desenvolvimento seguro de Software
  - 20.2.5. Padrões para a segurança das aplicações

- 20.3. Segurança na concepção das aplicações web
  - 20.3.1. Introdução à segurança das aplicações web
  - 20.3.2. Segurança na concepção das aplicações web
- 20.4. Teste da segurança e proteção online das aplicações web
  - 20.4.1. Análise e teste da segurança das aplicações web
  - 20.4.2. Segurança na implantação e produção das aplicações web
- 20.5. Segurança dos serviços web
  - 20.5.1. Introdução à segurança dos serviços web
  - 20.5.2. Funções e tecnologias da segurança dos serviços web
- 20.6. Teste da segurança e proteção online dos serviços web
  - 20.6.1. Avaliação da 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 nas aplicações
  - 20.8.1. Manual de boas práticas no desenvolvimento das aplicações online
  - 20.8.2. Manual de boas práticas na implementação das aplicações online
- 20.9. Erros comuns que prejudicam a segurança das aplicações
  - 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 do Big Data e a tornar-se um arquiteto de estratégias empresariais de sucesso”*

# 04

## Objetivos de ensino

O Mestrado Avançado em Engenharia e Qualidade do Software tem como objetivo formar profissionais altamente capacitados no design, desenvolvimento e gestão de sistemas de Software de alta qualidade. Com um enfoque particular na qualidade e na gestão estratégica de projetos tecnológicos, o programa também promove a capacidade de adaptação aos rápidos avanços no setor. Deste modo, garante-se 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 competências avançadas no design, desenvolvimento e manutenção de sistemas de Software complexos e escaláveis, aplicando as melhores práticas e metodologias da engenharia de Software
- ♦ Capacitar os estudantes no asseguramento da qualidade do Software, fornecendo-lhes ferramentas e técnicas para garantir a fiabilidade, segurança e desempenho das soluções tecnológicas
- ♦ Fomentar a liderança na gestão de projetos tecnológicos, desenvolvendo competências na direção de equipas multidisciplinares, planeamento estratégico e tomada de decisões em ambientes dinâmicos
- ♦ Promover a capacidade de adaptação aos rápidos avanços tecnológicos, através da especialização em novas ferramentas, técnicas e tendências que permitem manter-se na vanguarda da engenharia de Software
- ♦ Desenvolver competências na gestão da qualidade em todo o ciclo de vida do Software, desde o planeamento inicial até à manutenção e melhoria contínua dos sistemas
- ♦ Fortalecer as competências de comunicação e trabalho em equipa, essenciais para colaborar eficazmente com diferentes stakeholders, gerir expectativas e garantir o sucesso de projetos tecnológicos



*Melhore as suas competências e torne-se um líder na criação de soluções tecnológicas de vanguarda”*





## Objetivos específicos

### Módulo 1. Qualidade do Software. Níveis de desenvolvimento TRL

- ♦ Compreender os diferentes níveis de maturidade tecnológica e a sua relação com a qualidade do Software
- ♦ Avaliar o desenvolvimento do Software em cada etapa do TRL e como isso impacta na qualidade final do produto

### Módulo 2. Desenvolvimento de projetos Software. Documentação funcional e técnica

- ♦ Desenvolver competências para criar documentação funcional e técnica clara e detalhada em projetos de Software
- ♦ Analisar a importância de uma documentação precisa para a gestão de projetos e a qualidade do Software

### Módulo 3. *Testing* de Software. Automatização de provas

- ♦ Desenvolver competências para desenhar e executar testes automatizados em aplicações de Software
- ♦ Implementar soluções de *testing* eficientes utilizando ferramentas de automatização de testes

### Módulo 4. Metodologias de gestão de projetos Software. Metodologias Waterfall vs. Metodologias Ágeis

- ♦ Analisar as diferenças entre as metodologias Waterfall e Ágeis na gestão de projetos de Software
- ♦ Avaliar os benefícios e limitações de cada metodologia em função do tipo de projeto

### Módulo 5. TDD (*Test Driven Development*). Desenho de software guiado pelas provas

- ♦ Desenvolver competências para escrever testes unitários antes de escrever o código de produção
- ♦ Melhorar a qualidade do Software através da implementação de TDD no processo de desenvolvimento

### Módulo 6. DevOps. Gestão de Qualidade do Software

- ♦ Explorar o conceito de DevOps e o seu impacto na melhoria contínua da qualidade do Software
- ♦ Aprender a integrar as práticas de desenvolvimento e operações para alcançar um ciclo de vida de Software mais ágil e eficiente

### Módulo 7. DevOps e integração contínua Soluções práticas avançadas no desenvolvimento de Software

- ♦ Aprofundar-se nas técnicas avançadas de integração contínua dentro do quadro de DevOps
- ♦ Implementar soluções práticas de integração contínua para automatizar o processo de desenvolvimento e implantação de Software

### Módulo 8. Desenho de Bases de Dados (BD). Normalização e rendimento. Qualidade do Software

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

### Módulo 9. Desenho de arquiteturas escaláveis. A Arquitetura no ciclo de vida do Software

- ♦ Aprofundar-se nos princípios de design de arquiteturas escaláveis e o seu impacto na qualidade e no desempenho do Software
- ♦ Avaliar diferentes padrões de arquitetura para aplicações de Software escaláveis

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

- ♦ Compreender os critérios de qualidade do Software segundo estes padrões e como aplicá-los
- ♦ Implementar métricas de qualidade para avaliar e melhorar continuamente as aplicações de Software

### **Módulo 11. Metodologias, desenvolvimento e qualidade em Engenharia de Software**

- ♦ Aprofundar-se nas metodologias mais utilizadas na Engenharia de Software e a sua relação com a Qualidade
- ♦ Desenvolver uma abordagem integral que combine desenvolvimento, testes e qualidade em projetos de Software

### **Módulo 12. Gestão de projetos de software**

- ♦ Desenvolver competências na gestão de projetos de Software, desde o planeamento até à execução
- ♦ Gerir os recursos, tempos e riscos associados aos projetos de desenvolvimento de Software

### **Módulo 13. Plataformas de desenvolvimento do Software**

- ♦ Explorar as diferentes plataformas de desenvolvimento de Software e as suas características
- ♦ Avaliar plataformas de desenvolvimento em função das suas capacidades, flexibilidade e compatibilidade com diferentes projetos

### **Módulo 14. Computação para o cliente web**

- ♦ Analisar como é realizada a computação no lado do cliente no desenvolvimento de aplicações web
- ♦ Desenvolver aplicações que aproveitem a computação no cliente para melhorar a interação e o desempenho

### **Módulo 15. Computação para o servidor web**

- ♦ Explorar as tecnologias e técnicas utilizadas para a computação no servidor web
- ♦ Compreender o manuseio de dados, a lógica de negócios e a gestão de utilizadores no servidor





### **Módulo 16. Gestão da segurança**

- ♦ Avaliar os riscos de segurança nas aplicações e aplicar medidas preventivas
- ♦ Implementar controlos de segurança em todas as etapas do ciclo de vida do Software

### **Módulo 17. Segurança no Software**

- ♦ Explorar as melhores práticas de segurança no desenvolvimento de Software
- ♦ Analisar as vulnerabilidades mais comuns no Software e aprender a mitigá-las

### **Módulo 18. Administração de Servidores Web**

- ♦ Compreender o papel dos servidores web no desenvolvimento e no lançamento de aplicações
- ♦ Desenvolver competências na administração e manutenção de servidores web

### **Módulo 19. Auditoria de Segurança**

- ♦ Avaliar a segurança dos sistemas através de auditorias e testes de penetração
- ♦ Implementar processos de auditoria contínua para melhorar a segurança do Software

### **Módulo 20. Segurança em Aplicações Online**

- ♦ Implementar soluções para proteger aplicações online contra ameaças externas e internas
- ♦ Estabelecer políticas de segurança e auditoria para garantir a integridade das aplicações online

# 05

## Oportunidades de carreira

O aluno estará altamente capacitado para assumir papéis de liderança no desenvolvimento, implementação e gestão de Software de alta qualidade. Graças à sua especialização avançada em áreas chave como arquitetura de Software, segurança em aplicações online, gestão de projetos tecnológicos e metodologias ágeis, poderá liderar equipas de desenvolvimento. Além disso, a sua preparação permitirá ocupar cargos de direção em projetos de Software, enquanto a sua capacidade de inovar e liderar equipas multidisciplinares torná-lo-á capaz de enfrentar os desafios do ambiente digital e contribuir significativamente para o sucesso de qualquer organização.



“

*A TECH oferece-lhe a oportunidade de concretizar os seus sonhos na disciplina mais apaixonante, que transforma ideias em produtos tangíveis capazes de melhorar a vida das pessoas”*

#### Perfil dos nossos alunos

O perfil do aluno do Mestrado Avançado em Engenharia e Qualidade do Software está orientado a formar profissionais altamente qualificados, capazes de liderar e gerir projetos tecnológicos de grande impacto. Garantindo a qualidade, segurança e eficiência em todas as fases do desenvolvimento de Software, dominará tanto as metodologias ágeis como as tradicionais. Além disso, estará capacitado para projetar e desenvolver sistemas de Software escaláveis, eficientes e seguros, aplicando normas internacionais de qualidade e metodologias avançadas como DevOps e integração contínua.

*Torne-se um especialista que garanta o sucesso das empresas, na maior universidade digital do mundo.*

- ♦ **Segurança em Software e sistemas:** Competência na implementação de práticas avançadas de segurança, incluindo a proteção de dados e a gestão de vulnerabilidades em aplicações online
- ♦ **Garantia da qualidade do Software:** Habilidade para aplicar normas internacionais (ISO, IEC 9126) e ferramentas de teste automatizado para garantir a fiabilidade e o desempenho do Software
- ♦ **Desenvolvimento de arquiteturas escaláveis:** Capacidade para projetar e construir sistemas de Software que possam crescer e adaptar-se às exigências do mercado, sem comprometer a sua qualidade nem segurança
- ♦ **Integração contínua e DevOps:** Habilidade para implementar e gerir processos de integração contínua, garantindo uma entrega eficiente e sem interrupções de novas funcionalidades de Software



Após realizar o Mestrado Avançado, poderá desempenhar os seus conhecimentos e competências nos seguintes cargos:

1. **Diretor de Tecnologia (CTO):** Responsável pela direção estratégica da tecnologia numa empresa, liderando equipas de desenvolvimento e supervisionando a implementação de soluções tecnológicas inovadoras.
2. **Gestor de Qualidade do Software:** Responsável por supervisionar e garantir que os processos e produtos de Software cumpram com os padrões de qualidade estabelecidos, liderando as iniciativas de melhoria contínua e testes de Software.
3. **Arquiteto de Software:** Designer principal da estrutura e da arquitetura de sistemas de Software complexos, garantindo que sejam escaláveis, seguros e eficientes.
4. **Líder de Projetos de Software:** Responsável pelo planeamento, execução e entrega de projetos de Software, gerindo equipas multidisciplinares e garantindo que os projetos sejam concluídos dentro do prazo, orçamento e com os padrões de qualidade adequados.
5. **Especialista em Segurança Informática:** Responsável pela proteção de aplicações, infraestruturas e dados contra ameaças cibernéticas, implementando estratégias e políticas de segurança eficazes.
6. **Auditor de Segurança em Software:** Realiza auditorias exaustivas para identificar vulnerabilidades nas aplicações e sistemas, propondo melhorias e soluções para garantir a segurança do Software.



*Se deseja fazer a diferença no mundo digital, escolha este caminho que o especializa como 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.



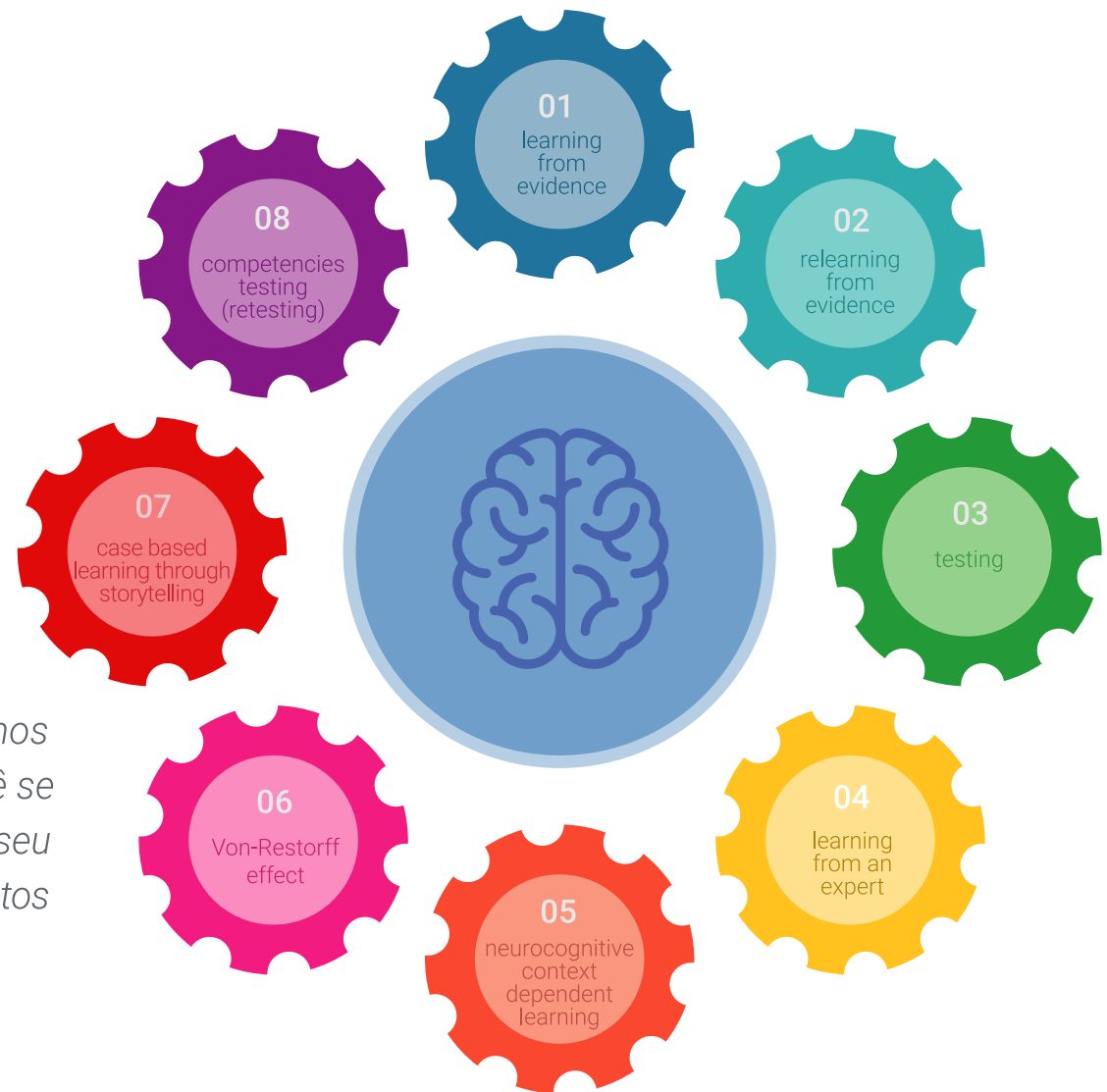
## Método Relearning

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 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 na plataforma de avaliação Trustpilot, com uma pontuação de 4,9 de 5.

*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

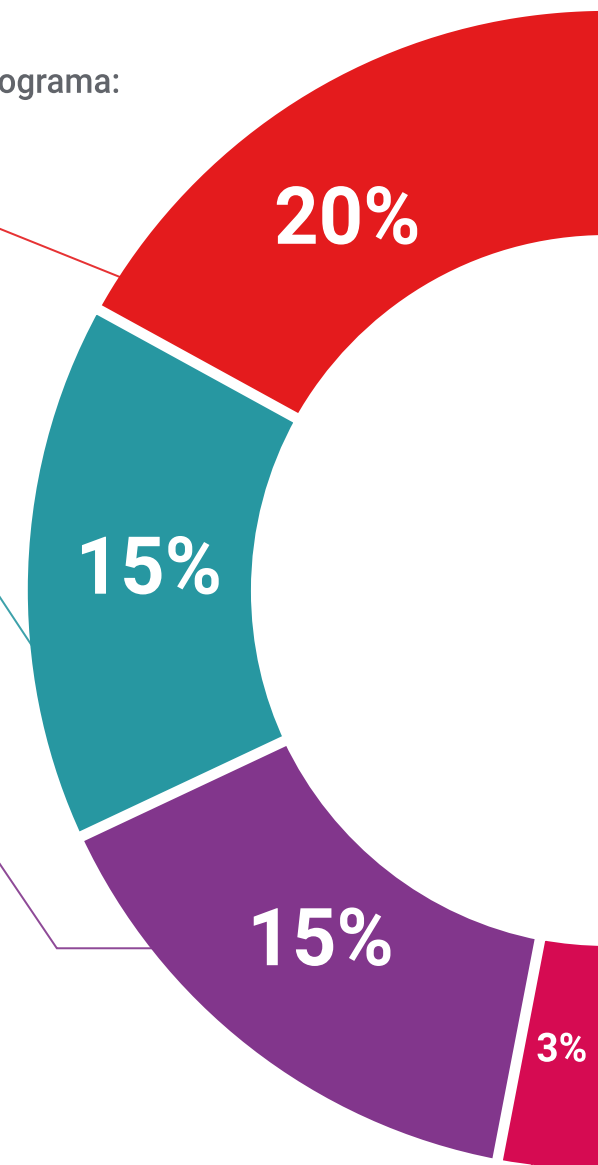
Apresentamos os conteúdos de forma atraente e dinâmica em pílulas multimídia que incluem áudio, vídeos, imagens, diagramas e mapas conceituais com o objetivo de reforçar o conhecimento.

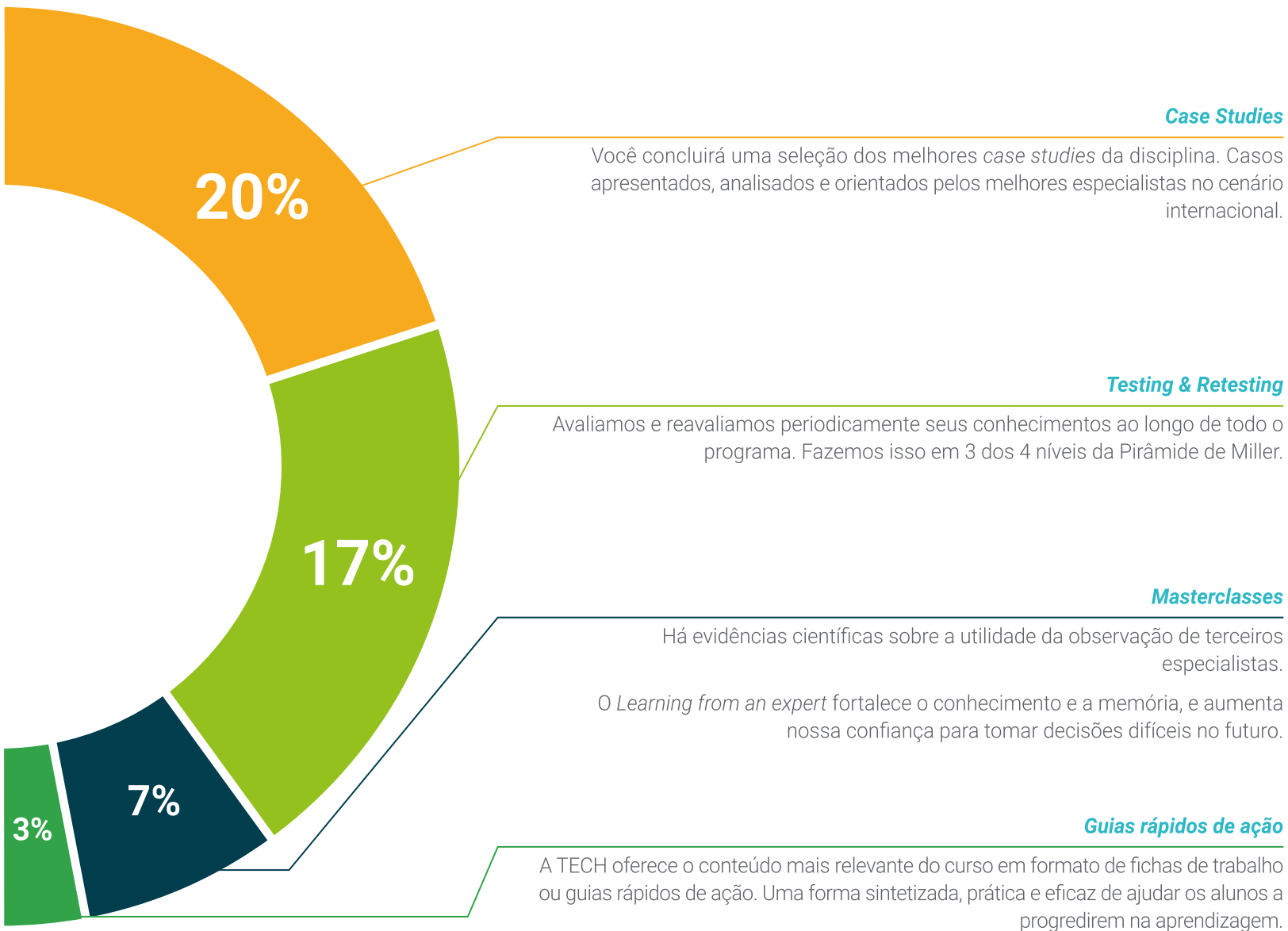
Este sistema exclusivo de capacitação por meio da apresentação de conteúdo multimídia foi premiado pela Microsoft como "Caso de sucesso na Europa"



#### 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



#### Testing & Retesting



#### Masterclasses



#### Guias rápidos de ação



07

# Corpo docente

A gestão e a docência deste Mestrado Avançado de Engenharia e Qualidade do Software é levada a cabo por uma equipa de especialistas em engenharia com muitos anos de experiência na gestão e no desenvolvimento de projetos técnicos e especializados. A sua formação profissional confere a este diploma uma mais valia de qualidade que se refletirá numa melhor contextualização do conteúdo pelo licenciado, bem como na implementação à experiência académica de casos práticos reais, mas sempre com o objetivo de oferecer um programa 100% online, dinâmico, inovador e baseado na realidade imediata do setor.



“

*Junte-se à carreira mais procurada pelos especialistas em engenharia e qualidade de Software e aproveite esta oportunidade para tornar-se um deles”*

## Diretor Internacional Convidado

Darren Pulsipher é um **arquitecto de software** altamente experiente, um inovador com uma carreira internacional de destaque no **desenvolvimento de software e firmware**. De facto, possui competências altamente desenvolvidas em **comunicação, gestão de projetos e negócios**, o que lhe permitiu liderar importantes iniciativas a nível global.

Além disso, ocupou cargos de grande responsabilidade ao longo da sua carreira, como o de **Arquiteto Chefe de Soluções para o Sector Público na Intel Corporation**, onde promoveu **negócios modernos, processos e tecnologias** para clientes, parceiros e utilizadores do **setor público**. Adicionalmente, fundou a **Yoly Inc.**, onde também exerceu funções como **CEO**, trabalhando no desenvolvimento de uma **ferramenta de agregação e diagnóstico de redes sociais** baseada no **Software como Serviço (SaaS)**, utilizando tecnologias de **Big Data** e **Web 2.0**.

Além disso, exerceu funções noutras empresas, como **Director Sénior de Engenharia**, na **Dell Technologies**, onde liderou a **Unidade de Negócios de Big Data na Nuvem**, comandando equipas nos **Estados Unidos** e na **China** para a gestão de projetos de grande envergadura e a reestruturação de divisões empresariais para a sua integração bem-sucedida. Também trabalhou como **Diretor de Tecnologias de Informação (Chief Information Officer)** na **XanGo**, onde geriu projetos como o suporte de **Help Desk**, o suporte à produção e o desenvolvimento de soluções.

Entre as múltiplas especialidades nas quais é perito, destacam-se 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. Conhecimentos que partilhou através do pódcast e boletim semanal **"Embracing Digital Transformation"**, que ele próprio produziu e apresentou, ajudando as organizações a navegar com sucesso na **transformação digital** aproveitando as pessoas, os processos e a tecnologia.



## Sr. Pulsipher, Darren

---

- Arquiteto Chefe de Soluções para o Setor Público na Intel, Califórnia, Estados Unidos
- Apresentador e Produtor de *"Embracing Digital Transformation"*, Califórnia
- Fundador e CEO na Yoly Inc., Arkansas
- Diretor Sênior de Engenharia na Dell Technologies, Arkansas
- Diretor de Tecnologias de Informação (*Chief Information Officer*) na XanGo, Utah
- Arquiteto Sênior na Cadence Design Systems, Califórnia
- Gestor Sênior de Processos de Projetos na Lucent Technologies, Califórnia
- Engenheiro de Software na Cemax-Icon, Califórnia
- Engenheiro de Software na ISG Technologies, Canadá
- MBA em Gestão de Tecnologia pela Universidade de Phoenix
- Licenciatura em Ciências da Computação e Engenharia Elétrica pela Universidade Brigham Young

“

*Graças à TECH, poderá aprender com os melhores profissionais do mundo"*

## Direção



### Sr. Molina Molina, Jerónimo

- ♦ Responsável pela Inteligência Artificial na Helphone
- ♦ AI Engineer & Software Architect na NASSAT, Internet Satélite em Movimento
- ♦ Consultor Sênior na Hexa Ingenieros
- ♦ Introdutor de Inteligência Artificial (ML e CV)
- ♦ Especialista em Soluções Baseadas em Inteligência Artificial nas áreas de Computer Vision, ML/DL y NLP
- ♦ Curso de Especialização em Criação e Desenvolvimento de Empresas no Bancaixa e Fundeun
- ♦ Engenheiro Informático pela Universidade de Alicante
- ♦ Mestrado em Inteligência Artificial pela Universidade Católica de Ávila
- ♦ MBA Executive no Fórum Europeu de Campus Empresarial

## Professores

### Sra. Tenrero Morán, Marcos

- ♦ Junior Application Developer na Getronics
- ♦ Cofundadora e City Leader da rede Galicia AI
- ♦ Engenheira de Software Júnior na Indra
- ♦ Desenvolvedora web na EDISA
- ♦ Licenciatura em Engenharia Informática pela Universidade de Vigo
- ♦ Mestrado Universitário em Engenharia Informática pela Universidade de Vigo

### Sr. Pi Morell, Oriol

- ♦ Analista Funcional na Fihoca
- ♦ Product Owner de Hosting e correio na CDmon
- ♦ Analista Funcional e Software Engineer na Atmira e Capgemini
- ♦ Docente na Capgemini, Forms Capgemini e na Atmira
- ♦ Licenciatura em Engenharia Técnica de Informática de Gestão pela Universidade Autónoma de Barcelona
- ♦ Mestrado em Inteligência Artificial pela Universidade Católica de Ávila
- ♦ MBA em Direção e Administração de Empresas pela IMF Smart Education
- ♦ Mestrado em Direção de Sistemas de Informação pela IMF Smart Education
- ♦ Pós-graduação em Padrões de Design pela Universitat Oberta de Catalunya

**Sr. Martínez Calvo , Francisco Javier**

- ♦ Engenheiro Técnico Industrial Especializado em Eletricidade e Eletrónica
- ♦ Técnico de Software na HEXA Engenheiros
- ♦ Desenvolvedor Sênior .Net/Arquitecto de soluções .Net na Everis
- ♦ Analista/Arquiteto *Software* na LaLiga
- ♦ Engenheiro On-site da Microsoft no BBVA
- ♦ Consultor Técnico-Informático *Freelance*
- ♦ Formador em Visual Studio, SqlServer, CCNA (Routers e Switch Cisco), Programação Web PHP e .Net em vários centros (Salesianos, Maforem, Dreamsoft)
- ♦ Engenheiro Técnico Industrial com Especialização em Eletricidade, Eletrónica Industrial
- ♦ Mestrado Cibernos em .NET, MCAD
- ♦ Mestrado Eidos em Programação Avançada, Nível Especialista
- ♦ Mestrado Web com Certificações Dreamweaver, Fireworks, Flash e ActionScript, Versões MX

**Sr. Tenrero Morán, Marcos**

- ♦ Engenheiro DevOps na Allot Communications
- ♦ Manager de Gestão do Ciclo de Vida das Aplicações na Cegid Meta4
- ♦ Engenheiro de Automação QA na Cegid Meta4
- ♦ Mestrado em Desenvolvimento de Aplicações Profissionais para Android pela Universidade Galileo Guatemala
- ♦ Mestrado em Desenvolvimento de Serviços na Nuvem, Node.Js, JavaScript, HTML5 pela Universidade Politécnica de Madrid
- ♦ Desenvolvimento Web com Angular-CLI (4), Ionic e Node.Js, Meta4 pela Universidade Rey Juan Carlos
- ♦ Licenciatura em Engenharia de Computadores pela Universidade Rey Juan Carlos

**Sra. Acebes Tamargo, Patricia**

- ♦ Consultora Especializada em Big Data
- ♦ Departamento de Operações, trabalha com Elasticsearch e Kibana na Sirt
- ♦ Investigadora na Linha de *Human Factor* e *AI Applications* no CTIC Centro Tecnológico
- ♦ Investigadora na Linha de Unidade de Negócio no CTIC Centro Tecnológico
- ♦ Departamento de Saúde Digital e Envelhecimento Ativo no CTIC Centro Tecnológico
- ♦ Departamento de Ciência de Dados no CTIC Centro Tecnológico
- ♦ Doutoramento em Informática em Inteligência Artificial pela Universidade Politécnica de Valência
- ♦ Licenciatura em Economia pela Universidade de Oviedo
- ♦ Mestrado em Análise de Dados pela UCJC
- ♦ Mestrado em Investigação em Inteligência Artificial pela UNED
- ♦ Mestrado em *Blockchain*, *Smart Contracts* e Criptomoedas pela Universidade de Alcalá
- ♦ Pós-graduação em Engenharia de *Blockchain* pela EADA
- ♦ Mestrado Universitário em Economia, Instrumentos, Análise Económica pela Universidade de Oviedo
- ♦ Mestrado em Fiscalidade pelo Colégio de Economistas

08

# Certificação

O Mestrado Avançado em Engenharia e Qualidade do Software garante, além da formação mais rigorosa e atualizada, o acesso a um certificado de Mestrado Avançado emitido pela TECH Global University.



“

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

Este programa permitirá a obtenção do certificado próprio de **Mestrado Avançado em Engenharia e Qualidade do Software** reconhecido pela TECH Global University, a maior universidade digital do mundo.

Esse título próprio da **TECH Global University**, é um programa europeu de formação contínua e atualização profissional que garante a aquisição de competências na sua área de conhecimento, conferindo um alto valor curricular ao aluno que conclui o programa.

Título: **Mestrado Avançado em Engenharia e Qualidade do Software**

Modalidade: **online**

Duração: **2 anos**

Acreditação: **120 ECTS**



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

futuro  
saúde confiança pessoas  
informação orientadores  
educação certificação ensino  
garantia aprendizagem  
instituições tecnologia  
comunidade compromisso  
atenção personalizada  
conhecimento inovação  
presente qualidade  
desenvolvimento situação



## Mestrado Avançado Engenharia e Qualidade do Software

- » Modalidade: online
- » Duração: 2 anos
- » Certificação: TECH Global University
- » Acreditação: 120 ECTS
- » Horário: ao seu próprio ritmo
- » Exames: online

# Mestrado Avançado

## Engenharia e Qualidade do Software