Mestrado Próprio
Qualidade do Software

tech universidade tecnológica

tech universidade tecnológica

Mestrado Próprio
Qualidade do Software

- Modalidade: **online**
- Duração: **12 meses**
- Certificação: **TECH Universidade Tecnológica**
- Créditos: **60 ECTS**
- Tempo Dedicado: **16 horas/semana**
- Horário: **ao seu próprio ritmo**
- Exames: **online**

Acesso ao site: www.techtitute.com/pt/informatica/mestrado-proprio/mestrado-proprio-qualidade-software

Índice

01 Apresentação — *pág. 4*

02 Objetivos — *pág. 8*

03 Competências — *pág. 16*

04 Direção do curso — *pág. 18*

05 Estrutura e conteúdo — *pág. 28*

06 Metodologia — *pág. 40*

07 Certificação — *pág. 48*

01

Apresentação

O rápido crescimento da indústria e as atuais exigências do mercado conduziram a um elevado nível de dívida técnica em projetos de software. Devido à necessidade imperativa de refletir respostas rápidas aos requisitos do cliente ou da empresa, sem avaliar e especificar os detalhes da qualidade do próprio sistema. É aqui que se reflete a necessidade de ter em conta a escalabilidade do projeto ao longo do seu ciclo de vida, o que requer competências informáticas centradas na qualidade a partir de uma abordagem *top-down*. Este programa desenvolve os critérios, tarefas e metodologias avançadas para compreender a relevância de um trabalho orientado para a necessidade de implementar políticas de qualidade nas *Software Factories*. O seu estudo será totalmente online e terá a duração de 12 meses, de acordo com a metodologia implementada pela maior universidade digital do mundo.

Especialize-se em Qualidade de Software a partir do ponto de vista técnico e de gestão; certificando-se em 12 meses, e faça a diferença no seu ambiente profissional”

O conceito de Dívida Técnica atualmente a ser aplicado por um grande número de empresas e administrações com os seus fornecedores reflete a forma improvisada como os projetos têm sido desenvolvidos. Gerando um novo custo implícito de ter de refazer um projeto, adotando uma solução rápida e fácil, em oposição ao que deveria ser uma abordagem escalável na evolução do projeto.

Desde há alguns anos, os projetos têm sido desenvolvidos muito rapidamente, com o objetivo de os concluir com o cliente com base no preço e nos prazos, em vez de adotar uma abordagem de qualidade. Estas decisões estão agora a ter o seu preço em muitos fornecedores e clientes.

Este Mestrado Próprio permitirá aos profissionais informáticos analisar os critérios subjacentes à Qualidade de Software a todos os níveis. Critérios tais como a normalização de bases de dados, desacoplamento entre componentes de um sistema de informação, arquiteturas escaláveis, métricas, documentação, tanto funcional como técnica. Para além das metodologias de gestão e desenvolvimento de projetos e outros métodos de garantia de qualidade, tais como técnicas de trabalho em colaboração; mesmo a chamada *Pair Programming,* que permite que os conhecimentos residam na empresa e não em indivíduos.

A grande maioria dos mestrados deste tipo concentra-se numa tecnologia, numa língua ou numa ferramenta. Este programa é único na forma como sensibiliza o profissional para a importância da qualidade do software, reduzindo a dívida técnica dos projetos com um de qualidade em vez de uma abordagem baseada na economia e nos prazos curtos; dota o aluno de conhecimentos especializados para que o orçamento do projeto possa ser justificado.

Para tornar isto possível, a TECH Universidade Tecnológica reuniu um grupo de especialistas na matéria que transmitirão os conhecimentos e experiências mais atualizados. Através de um campus virtual moderno com conteúdos teóricos e práticos, distribuídos em diferentes formatos. Haverá 10 módulos divididos em diferentes tópicos e subtópicos que permitirão a aprendizagem em 12 meses de acordo com a metodologia *Relearning,* que facilita a memorização e a aprendizagem de uma forma ágil e eficiente.

Este **Mestrado Próprio em 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 Desenvolvimento de Software
- Os conteúdos gráficos, esquemáticos e eminentemente práticos com que está concebido fornece informações científicas e práticas sobre as disciplinas que são essenciais para a prática profissional
- Exercícios práticos onde o processo de autoavaliação pode ser levado a cabo a fim de melhorar a aprendizagem
- A sua ênfase especial em metodologias inovadoras
- Lições teóricas, questões ao especialista e trabalhos de reflexão individual
- A disponibilidade de acesso ao conteúdo a partir de qualquer dispositivo fixo ou portátil com ligação à Internet

O Mestrado Próprio em Qualidade do Software analisa os critérios subjacentes ao tema a todos os níveis. Expanda o seu nível de experiência. Inscreva-se já"

Desenvolva os critérios, tarefas e metodologias avançadas para compreender a relevância do trabalho orientado para a qualidade, e fornecer soluções eficazes à sua empresa ou cliente”

Um programa centrado na sensibilização para a importância da qualidade do software e para a necessidade de implementar políticas de qualidade nas software Factories.

Aprenda de uma forma prática e flexível Partilhando o seu dia a dia com esta certificação 100% online exclusiva da TECH Universidade Tecnológica.

O corpo docente do curso inclui profissionais do setor que trazem a sua experiência profissional para esta capacitação, para além de especialistas reconhecidos de sociedades de referência e universidades de prestígio.

Graças ao seu conteúdo multimédia, desenvolvido com a mais recente tecnologia educacional, o profissional terá acesso a uma aprendizagem situada e contextual, ou seja, um ambiente de simulação que proporcionará um programa imersivo programado para se formar em situações reais.

A conceção deste programa baseia-se na Aprendizagem Baseada nos Problemas, através da qual o instrutor deve tentar resolver as diferentes situações da atividade profissional que surgem ao longo do curso académico. Para tal, contará com a ajuda de um sistema inovador de vídeo interativo desenvolvido por especialistas reconhecidos.

02

Objetivos

O Mestrado Próprio em Qualidade do Software fornece aos alunos uma visão clara e especializada da importância da qualidade nos processos de desenvolvimento de software. Bem como as ferramentas mais avançadas para a implementação de processos DevOps e sistemas de garantia de qualidade. Em suma, proporcionará um conhecimento teórico e prático amplo e especializado para que compreendam o desenvolvimento de projetos numa perspetiva moderna e eficiente.

Poderá aceder facilmente a todos os conteúdos sempre que desejar. A partir do seu computador ou dispositivo favorito. Pode também descarregá-los para a sua próxima consulta"

Objetivos gerais

- Desenvolver os critérios, tarefas e metodologias avançadas para compreender a relevância do trabalho orientado para a qualidade
- Analisar os fatores chave na qualidade de um projeto software
- Desenvolver os aspetos normativos relevantes
- Implementar processos de DevOps e de sistemas para a garantia da qualidade
- Reduzir a dívida técnica dos projetos com uma abordagem de qualidade em vez de uma abordagem baseada na economia e em prazos curtos
- Proporcionar ao estudante conhecimentos especializados para poder medir e quantificar a qualidade de um projeto software
- Defender as propostas económicas para projetos com base na qualidade

Objetivos específicos

Módulo 1. Qualidade do Software. Níveis de desenvolvimento TRL

- Desenvolver de uma forma clara e concisa os elementos que compõem a qualidade do software
- Aplicar os modelos e normas em função de sistema, produto e processo software
- Aprofundar o conhecimento das normas de qualidade ISO aplicadas tanto de forma geral como em partes específicas
- Aplicar as regras de acordo com o ambiente (local, nacional, internacional)
- Examinar os níveis de maturidade TRL e adaptá-los às diferentes partes do projeto de software a ser tratado
- Adquirir capacidade de abstração para aplicar um ou vários critérios de elementos e níveis da qualidade do software
- Distinguir os casos de aplicação das normas e níveis de maturidade num projeto simulado de caso real

Módulo 2. Desenvolvimento de Projetos Software. Documentação funcional e técnica

- Determinar a influência da gestão do projeto na qualidade
- Desenvolver as diferentes fases de um projeto
- Diferenciar entre os conceitos de qualidade inerentes à documentação funcional e técnica
- Analisar a fase de levantamento de requisitos, a fase de análise, a gestão da equipa e a fase de construção
- Estabelecer as diferentes metodologias de gestão de projetos de software
- Gerar critérios para decidir sobre a metodologia mais apropriada de acordo com o tipo de projeto

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

- Estabelecer as diferenças entre a qualidade do produto, a qualidade do processo e a qualidade na utilização
- Conhecer a normativa ISO/IEC 15504
- Determinar os detalhes de CMMI
- Aprender as chaves da integração contínua, os repositórios e o impacto que estes têm numa equipa de desenvolvimento de software
- Estabelecer a relevância da incorporação de repositórios por projetos de software. Aprender a criá-los com TFS
- Assimilar a importância da escalabilidade do software na conceção e desenvolvimento de sistemas de informação

Módulo 4 Metodologias de Gestão de Projetos Software. Metodologias *Waterfall* vs. metodologias ágeis

- Determinar em que consiste a metodologia Waterfall
- Aprofundar na Metodologia Scrum
- Estabelecer as Diferenças entre Waterfall e Scrum
- Especificar as diferenças entre as metodologias Waterfall e Scrum e a forma como o vê o cliente
- Examinar o Painel Kanban
- Planear um único projeto com WaterFall e Scrum
- Montar um projeto híbrido

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

- Conhecer a aplicação prática de TDD e as suas possibilidades, para a realização de provas de um projeto de software no futuro.
- Completar casos de simulação real propostos, como aprendizagem contínua deste conceito de TDD
- Analisar, em casos de simulação, até que ponto os testes podem ser bem ou mal-sucedidos, de um ponto de vista construtivo
- Determinar as alternativas à TDD, realizando uma análise comparativa entre elas

Módulo 6. DevOps. Gestão de Qualidade do Software

- Analisar as deficiências de um processo tradicional
- Avaliar as soluções possíveis e escolher a mais adequada
- Compreender as necessidades de negócio e o seu impacto na implementação
- Avaliar os custos das melhorias a implementar
- Desenvolver um ciclo de vida de software evolutivo, adaptado às necessidades reais
- Antecipar possíveis erros e evitá-los desde o processo de conceção
- Fundamentar a utilização de diferentes modelos de implementação

Módulo 7. DevOps e Integração Contínua Soluções práticas avançadas no Desenvolvimento de Software

- Identificar as etapas do ciclo de desenvolvimento e entrega de software adaptados a casos particulares
- Desenhar um processo de entrega de software através de integração contínua
- Construir e implementar integração e implantação contínuas com base no seu desenho prévio
- Estabelecer pontos de controlo de qualidade automáticos em cada entrega de software
- Manter um processo de entrega de software automático e robusto
- Adaptar as necessidades futuras ao processo contínuo de integração e implantação
- Analisar e antecipar vulnerabilidades de segurança durante e após o processo de entrega do software

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

- Avaliar a utilização do Modelo de entidade-relação para a conceção prévia de uma base de dados
- Aplicar uma entidade, um atributo, uma chave, etc., para a melhor integridade dos dados
- Avaliar as dependências, formas e regras de normalização de base de dados
- Especializar-se no funcionamento de um sistema de armazenamento de dados OLAP, desenvolvendo e utilizando tanto a tabela de factos, como a tabela de dimensões
- Determinar os pontos chave para o rendimento da base de dados
- Completar casos de simulação real propostos, como aprendizagem contínua sobre conceção, normalização e desempenho de bases de dados
- Estabelecer, nos casos de simulação, as opções a resolver na criação da base de dados, de um ponto de vista construtivo

Módulo 9. Desenho de Arquiteturas Escaláveis. A Arquitetura no Ciclo de Vida do Software

- Desenvolver o conceito de arquitetura de software e as suas caraterísticas
- Determinar os diferentes tipos de escalabilidade na arquitetura de software
- Analisar os diferentes níveis que podem ocorrer na escalabilidade Web
- Adquirir conhecimentos especializados sobre o conceito de ciclo de vida do software, etapas e modelos
- Determinar o impacto de uma arquitetura no ciclo de vida do software, com as suas vantagens, limitações e ferramentas de apoio
- Completar casos de simulação real propostos, como aprendizagem contínua da arquitetura e ciclo de vida do software
- Avaliar, em casos de simulação, até que ponto o desenho da arquitetura pode ser viável ou desnecessária

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

- Desenvolver o conceito de critérios de qualidade e aspetos relevantes
- Examinar a norma ISO/IEC 9126, principais aspetos e indicadores
- Analisar as diferentes medições para que um projeto de software cumpra as avaliações acordadas
- Examinar os atributos internos e externos a serem abordados na qualidade de um projeto software
- Distinguir as métricas de acordo com o tipo de programação (estruturada, orientada para objetos, por camadas, etc.)
- Completar casos de simulação real, como aprendizagem contínua sobre medição da qualidade
- Ver nos casos de simulação até que ponto é viável ou desnecessário, ou seja, de um ponto de vista construtivo das autoras

Destaque o seu perfil profissional com esta capacitação exclusiva. Obtenha o seu certificado em 12 meses e de uma forma prática com a metodologia que só a TECH Universidade Tecnológica lhe pode oferecer"

03

Competências

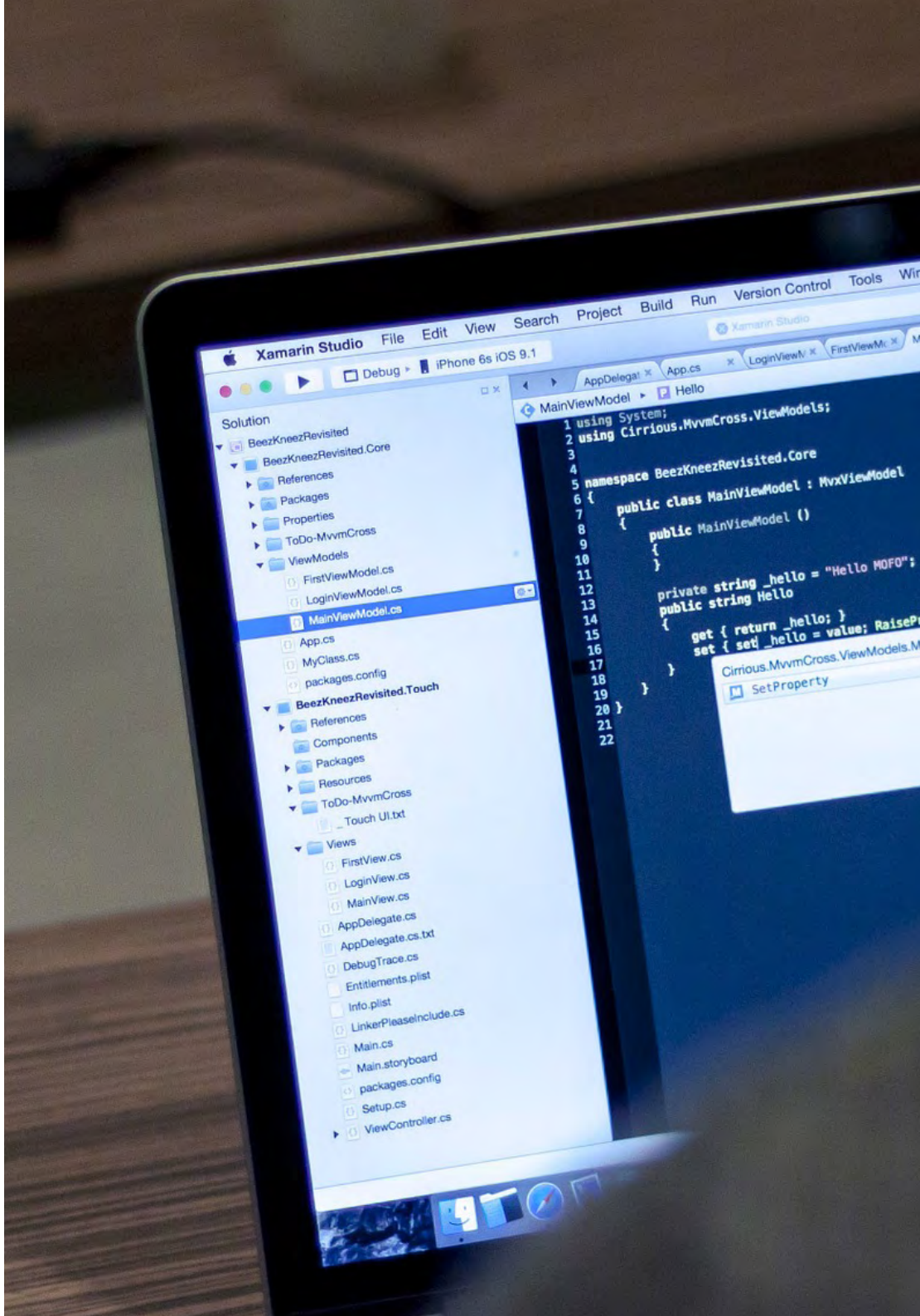
Os alunos deste Mestrado Próprio em Qualidade do Software dominarão o tópico tanto do ponto de vista técnico como de gestão. Podendo desenvolver a abordagem de um projeto, bem como a sua execução e elaborar uma arquitetura sustentável, eficiente e de qualidade para os projetos de software que lhe são apresentados.
Para o efeito, o pessoal docente dedicou toda a sua experiência pessoal à elaboração de uma multiplicidade de casos práticos, que servirão de contextualização e evolução face a essa "dívida técnica" que deixará de estar presente.

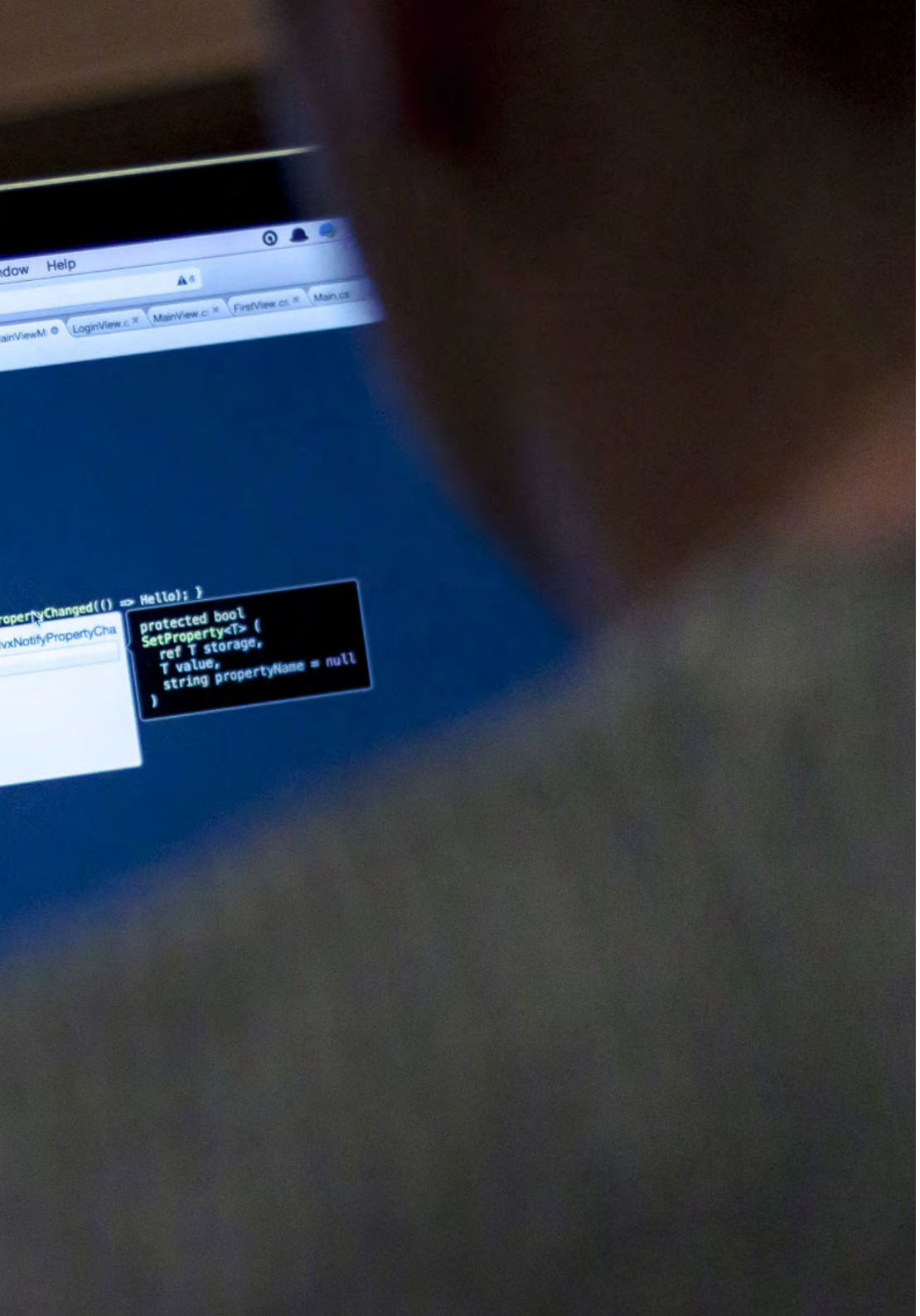
Um profissional informático focado na qualidade é um ativo crescente para consultorias de software e grandes empresas. Inscreva-se agora neste Mestrado Próprio em Qualidade do Software"

Competências gerais

- Reduzir a dívida técnica dos projetos com uma abordagem de qualidade em vez de uma abordagem baseada na economia e em prazos curtos
- Medir e quantificar a qualidade de um projeto de software
- Realizar o TDD corretamente, a fim de elevar os padrões de qualidade do software
- Justificar a orçamentação de projetos orientados para a qualidade
- Desenvolver as normas, modelos e padrões de qualidade
- Examinar diferentes avaliações de maturidade tecnológica
- Reduzir risco e assegurar a manutenção e controlo das versões posteriores
- Dominar as fases em que um projeto é dividido

Melhore as suas competências e descubra as infinitas possibilidades de crescimento profissional que se abrem com esta nova experiência"





Competências específicas

- Avaliar um sistema software em termos do grau de progresso no processo do projeto
- Abordar estas questões de fiabilidade, métrica e garantia em projetos de software de forma correta e estratégica
- Abordar o processo de decisão sobre a metodologia a ser utilizada no projeto
- Dominar os aspetos regulamentares essenciais para a criação de software
- Desenvolver o *Testing* de forma automática
- Estabelecer uma comunicação adequada com o cliente, compreendendo a forma como este percebe o projeto de acordo com a metodologia aplicada
- Elaborar as listas de requisitos das provas
- Realizar a abstração, divisão em provas mais unitárias e eliminar o que não se aplica ao bom desempenho das provas do projeto de software a realizar
- Atualizar a lista de requisitos das provas de uma forma medida e correta
- Adaptar a cultura DevOps às necessidades de negócio
- Desenvolver as mais recentes práticas e ferramentas na integração e implantação contínua
- Refatorizar e enfrentar a gestão e coordenação dos dados

04

Direção do curso

Professores especialistas com um extenso currículo na área das soluções informáticas e desenvolvimento de software e investigação, orientam este Mestrado Próprio para fornecer as ferramentas e conhecimentos necessários ao futuro diplomado centrado na qualidade dos processos de desenvolvimento de software e as ferramentas mais avançadas para implementar processos de DevOps e de sistemas de garantia de qualidade. Esta equipa de profissionais orientará o estudante em todos os momentos, a fim de alcançar os objetivos à distância, uma vez que se trata de um programa puramente online e seguindo a metodologia mais vanguardista implementada pela TECH.

Docentes especializados estão empenhados em fornecer-lhe o melhor conteúdo e em tornar o seu processo de aprendizagem uma experiência ágil e dinâmica. Esclarecendo as suas dúvidas e acompanhando-o ao longo de todo o processo"

Diretor Convidado Internacional

Com uma extensa trajetória profissional de mais de 30 anos no setor tecnológico, Daniel St. John é um prestigiado **Engenheiro Informático** altamente especializado em **Qualidade de Software**. Neste campo, consolidou-se como um verdadeiro líder, devido ao seu enfoque pragmático baseado na melhoria contínua e inovação.

Ao longo da sua carreira, fez parte de instituições de referência internacional como a **General Electric Healthcare** em Illinois. Assim, o seu trabalho focou-se na otimização das **infraestruturas digitais** das organizações, com o objetivo de melhorar significativamente a **experiência dos utilizadores**. Graças a isso, múltiplos pacientes tiveram acesso a um atendimento mais personalizado e ágil, com um acesso mais rápido aos resultados clínicos e aos acompanhamentos de saúde. Além disso, implementou soluções tecnológicas que permitiram aos profissionais melhorar a **tomada de decisões estratégicas** mais informadas, baseadas em grandes volumes de dados.

Paralelamente, Daniel St. John tem desenvolvido projetos tecnológicos vanguardistas para maximizar a eficácia dos processos operacionais nas instituições. Liderou a **transformação digital** de várias empresas de diferentes indústrias, implementando ferramentas emergentes como **Inteligência Artificial**, ***Big Data*** e ***Machine Learning*** para automatizar tarefas diárias complexas. Como resultado, essas organizações conseguiram adaptar-se rapidamente às tendências do mercado e garantir a sua sustentabilidade a longo prazo.

É importante destacar que Daniel St. John tem participado como orador em diversos congressos científicos internacionais. Assim, compartilhou o seu vasto conhecimento em áreas como a adoção de **Metodologias Ágeis**, a realização de **Testes de Aplicações** para garantir a fiabilidade dos sistemas e a implementação de técnicas inovadoras de ***Blockchain*** para garantir a proteção de dados confidenciais.

Sr. St. John, Daniel

- Diretor de Engenharia de Software na General Electric Healthcare, Wisconsin, Estados Unidos
- Chefe de Engenharia de Software na Siemens Healthineers, Illinois
- Diretor de Engenharia de Software na Natus Medical Incorporated, Illinois
- Engenheiro Sénior na WMS Gaming, Chicago
- Engenheiro Superior de Software na Siemens Medical Solutions, Illinois
- Mestrado em Estratégia e Análise de Dados pela Escola de Pós-Graduação em Gestão de Lake Forest
- Licenciatura em Ciências da Computação pela Universidade de Wisconsin-Parkside
- Membro da Junta Consultiva do Instituto de Tecnologia de Illinois
- Certificações em:
  - Python para Ciências de Dados
  - Inteligência Artificial e Desenvolvimento
  - SAFe SCRUM e Gestão de Projetos

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

Direção

Sr. Jerónimo Molina Molina

- IA Engineer & Software Architect. NASSAT - Internet Satélite em Movimento
- Consultor Sr. em 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. Atualmente a investigar as possibilidades de aplicação de Transformers e de Reinforcement Learning em projeto de investigação pessoal
- Especialista Universitário em Criação e Desenvolvimento de Empresas. Bancaixa– FUNDEUN Alicante
- Engenheiro em Informática. Universidade de Alicante
- Mestrado em Inteligência Artificial. Universidade Católica de Ávila
- MBA-Executive. Fórum Europeu Campus Empresarial

Professores

Sr. Oriol Pi Morell

- Product Owner de Hosting e email. CDMON
- Analista Funcional e Engenheiro de Software em diferentes organizações como Fihoca, Atmira, CapGemini
- Professor de diferentes cursos, tais como BPM na CapGemini, ORACLE Forms CapGemini, Processos de negócio Atmira
- Licenciatura em Engenharia Técnica em Informática de Gestão pela Universidade Autónoma
- Mestrado em Inteligência Artificial
- Mestrado em Direção e Administração de empresas. MBA
- Mestrado em Gestão de Sistemas de Informação Experiência Docente
- Pós-graduação, Padrões de desenho. Universitat Oberta de Catalunya

Sr. Marcos Tenrero Morán

- DevOps Engineer– Allot Communications
- Application Lifecycle Management & DevOps– Meta4 Spain. Cegid
- Engenheiro de automatização QA– Meta4 Spain. Cegid
- Licenciado em Engenharia de Computadores pela Universidade Rey Juan Carlos
- Desenvolvimento de aplicações profissionais para Android– Universidade Galileo (Guatemala)
- Desenvolvimento de Serviços na Nuvem (nodeJs, JavaScript, HTML5) - UPM
- Integração Contínua com Jenkins– Meta4. Cegid
- Desenvolvimento Web com Angular-CLI (4), Ionic e nodeJS. Meta4 - Universidade Rey Juan Carlos

Dr. Arturo Peralta Martín-Palomino

- CEO e CTO na Prometeus Global Solutions
- CTO em Korporate Technologies
- CTO em AI Shephers GmbH
- Licenciatura em Engenharia Informática pela Universidade de Castilla la Mancha
- Doutoramento em Economia, Negócios e Finanças pela Universidade Camilo José Cela Prémio Extraordinário de Doutoramento
- Doutoramento em Psicologia pela Universidade de Castilla la Mancha
- Mestrado em Tecnologias Avançadas de Informação da Universidade de Castilla la Mancha
- Mestrado MBA+E (Mestrado em Administração de Empresas e Engenharia Organizacional) pela Universidade de Castilla la Mancha
- Professora associada, docente em cursos de licenciatura e mestrado em Engenharia Informática na Universidade de Castilla la Mancha
- Professor do Mestrado em Big Data e Data Science na Universidade Internacional de Valência
- Professor do Mestrado em Indústria 4.0 e do Mestrado em Design Industrial e Desenvolvimento de Produto
- Membro do Grupo de Investigação SMILe da Universidade de Castilla la Mancha

Sra. Yésica Martínez Cerrato

- Técnica de produtos de segurança eletrónica na Securitas Seguridad Espanha
- Analista de Business Intelligence na Ricopia Technologies (Alcalá de Henares) Licenciatura em Engenharia Eletrónica de Comunicações na Escuela Politécnica Superior, Universidade de Alcalá
- Responsável pela formação de novos funcionários em software de gestão comercial (CRM, ERP, INTRANET), produto e procedimentos na Ricopia Technologies (Alcalá de Henares)
- Responsável pela formação de novos bolseiros incorporados nas salas de aula de informática da Universidade de Alcalá
- Gestora de projeto na área de integração de contas-chave nos Correos y Telégrafos (Madrid)
- Técnica Informático-Responsável pelas salas de aula de informática OTEC, Universidade de Alcalá (Alcalá de Henares)
- Professora de informática na Associação ASALUMA (Alcalá de Henares)
- Bolsa de formação como Técnico de Informática na OTEC, Universidade de Alcalá (Alcalá de Henares)

A nossa equipa pedagógica fornecer-lhe-á todos os seus conhecimentos para que esteja a par das últimas informações sobre a matéria"

05

Estrutura e conteúdo

A estrutura e conteúdo deste Mestrado Próprio foram desenvolvidos para cobrir os tópicos mais importantes para o desenvolvimento de Software de Qualidade. Composto por 10 módulos de ensino, que vão desde o desenvolvimento de projetos de software, documentação funcional e técnica, o *test Driven Developement* e as diferentes metodologias, à implementação de soluções práticas avançadas com DevOps e integração contínua, todas baseadas na obtenção de qualidade de software. O extenso conteúdo multimédia, rigorosamente selecionado por professores especializados, será de grande apoio para aliviar a carga pedagógica e servir como material de referência para consultas futuras.

Os casos práticos, baseados na realidade, servirão para reforçar e contextualizar toda a teoria aprendida durante o programa"

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. Necessidade de aplicação de 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 Seis-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 no uso (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 na 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 de projeto
- 2.2.1. Metodologia na gestão de projetos
- 2.2.2. Metodologias de projetos. Tipologia
- 2.2.3. Metodologia na gestão de projetos. Aplicação

2.3. Fase de identificação de requisitos
- 2.3.1. Identificação dos requisitos de um projeto
- 2.3.2. Gestão das reuniões de 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
- 2.6.2. Riscos no projeto
- 2.6.3. Gestão do risco

2.7. "*MUST*" do projeto
- 2.7.1. *MUST* de 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 para a 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. Aspetos técnicos de um projeto de software
- 2.9.1. Arquiteto do projeto. Aspetos 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 Provas

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. Exemplos
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 arquitetura do 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 *Agile*. Influência na qualidade do software
4.2.3. Metodologia *Agile*. Exemplos

4.3. Metodología Scrum
  4.3.1. Metodología 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 de 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 Comparativa
  4.7.1. Abordagem de um projeto piloto
  4.7.2. Projeto aplicando *Waterfall*. Exemplos
  4.7.3. Projeto aplicando Scrum. Exemplos
4.8. Visão do cliente
  4.8.1. Documentos num *Waterfall*
  4.8.2. Documentos num Scrum
  4.8.3. Comparativo
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 Developement*). 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. Conceção 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 implantação

6.4. Ciclo de entrega de software
- 6.4.1. Métodos de desenho
- 6.4.2. Convénios
- 6.4.3. Roteiro

6.5. Desenvolvimento de código sem 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 do 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 do software. requisitos entre fases

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

7.3. Pipelines declarativos
- 7.3.1. Diferenças entre pipelines tradicionais, como código e declarativos
- 7.3.2. Pipelines declarativos
- 7.3.3. Pipelines declarativos em Jenkins
- 7.3.4. Comparação de 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. Artefactos e ciclos de vida
- 7.5.2. Sistemas de armazenamento e gestão de artefactos
- 7.5.3. Segurança na gestão de Artefactos

7.6. Implantação contínua
- 7.6.1. Implantação contínua como recipientes
- 7.6.2. Implantação contínua com PaaS
- 7.6.3. Implementação contínua de aplicações móveis

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* unitários
- 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 base 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 base 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. Otimização de Índices
8.7.2. Otimização de consultas
8.7.3. Partição de tabelas
8.8. Simulação do 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 BBDD na Qualidade do Software
8.10.1. Otimização do desenho
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 para o domínio
9.2.2. Camadas, partilha de responsabilidade e padrões de desenho
9.2.3. Desacoplamento como base da 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 (Tabelas 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. Etapa 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 o desenho de arquitetura do 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 o desenho de arquitetura do 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. Caraterísticas
  10.2.1. Fiabilidade
  10.2.2. Funcionalidade
  10.2.3. Eficiência
  10.2.4. Usabilidade
  10.2.5. Capacidade de manutenção
  10.2.6. Portabilidade
  10.2.7. Segurança
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. Capacidade de manutenção
10.3.6. Portabilidade
10.3.7. Qualidade em uso
10.3.8. Métrica de qualidade do software
10.3.9. Métricas de qualidade em ISO 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 Caraterísticas
10.5. Métrica de qualidade do software (I). Elementos
10.5.1. Medição
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*

Terá acesso a conteúdos únicos e especializados. Selecionado por professores especialistas para uma certificação que fará transcender o seu perfil profissional"

06

Metodologia

Este programa de capacitação oferece uma forma diferente de aprendizagem.
A nossa metodologia é desenvolvida através de um modo de aprendizagem cíclico: ***o Relearning***.
Este sistema de ensino é utilizado, por exemplo, nas escolas médicas mais prestigiadas do mundo e tem sido considerado um dos mais eficazes pelas principais publicações, tais como a ***New England Journal of Medicine.***

Descubra o Relearning, um sistema que abandona a aprendizagem linear convencional para o levar através de sistemas de ensino cíclicos: uma forma de aprendizagem que provou ser extremamente eficaz, especialmente em disciplinas que requerem memorização"

Estudo de Caso para contextualizar todo o conteúdo

O nosso programa oferece um método revolucionário de desenvolvimento de competências e conhecimentos. O nosso objetivo é reforçar as competências num contexto de mudança, competitivo e altamente exigente.

Com a TECH pode experimentar uma forma de aprendizagem que abala as fundações das universidades tradicionais de todo o mundo"

Terá acesso a um sistema de aprendizagem baseado na repetição, com ensino natural e progressivo ao longo de todo o programa de estudos.

O estudante aprenderá, através de atividades de colaboração e casos reais, a resolução de situações complexas em ambientes empresariais reais.

Um método de aprendizagem inovador e diferente

Este programa da TECH é um programa de ensino intensivo, criado de raiz, que propõe os desafios e decisões mais exigentes neste campo, tanto a nível nacional como internacional. Graças a esta metodologia, o crescimento pessoal e profissional é impulsionado, dando um passo decisivo para o sucesso. O método do caso, a técnica que constitui a base deste conteúdo, assegura que a realidade económica, social e profissional mais atual é seguida.

> *O nosso programa prepara-o para enfrentar novos desafios em ambientes incertos e alcançar o sucesso na sua carreira"*

O método do caso tem sido o sistema de aprendizagem mais amplamente utilizado nas principais escolas de informática do mundo desde que existem. Desenvolvido em 1912 para que os estudantes de direito não só aprendessem o direito com base no conteúdo teórico, o método do caso consistia em apresentar-lhes situações verdadeiramente complexas, a fim de tomarem decisões informadas e valorizarem juízos sobre a forma de as resolver. Em 1924 foi estabelecido como um método de ensino padrão em Harvard.

Numa dada situação, o que deve fazer um profissional? Esta é a questão que enfrentamos no método do caso, um método de aprendizagem orientado para a ação. Ao longo do programa, os estudantes serão confrontados com múltiplos casos da vida real. Terão de integrar todo o seu conhecimento, investigar, argumentar e defender as suas ideias e decisões.

Relearning Methodology

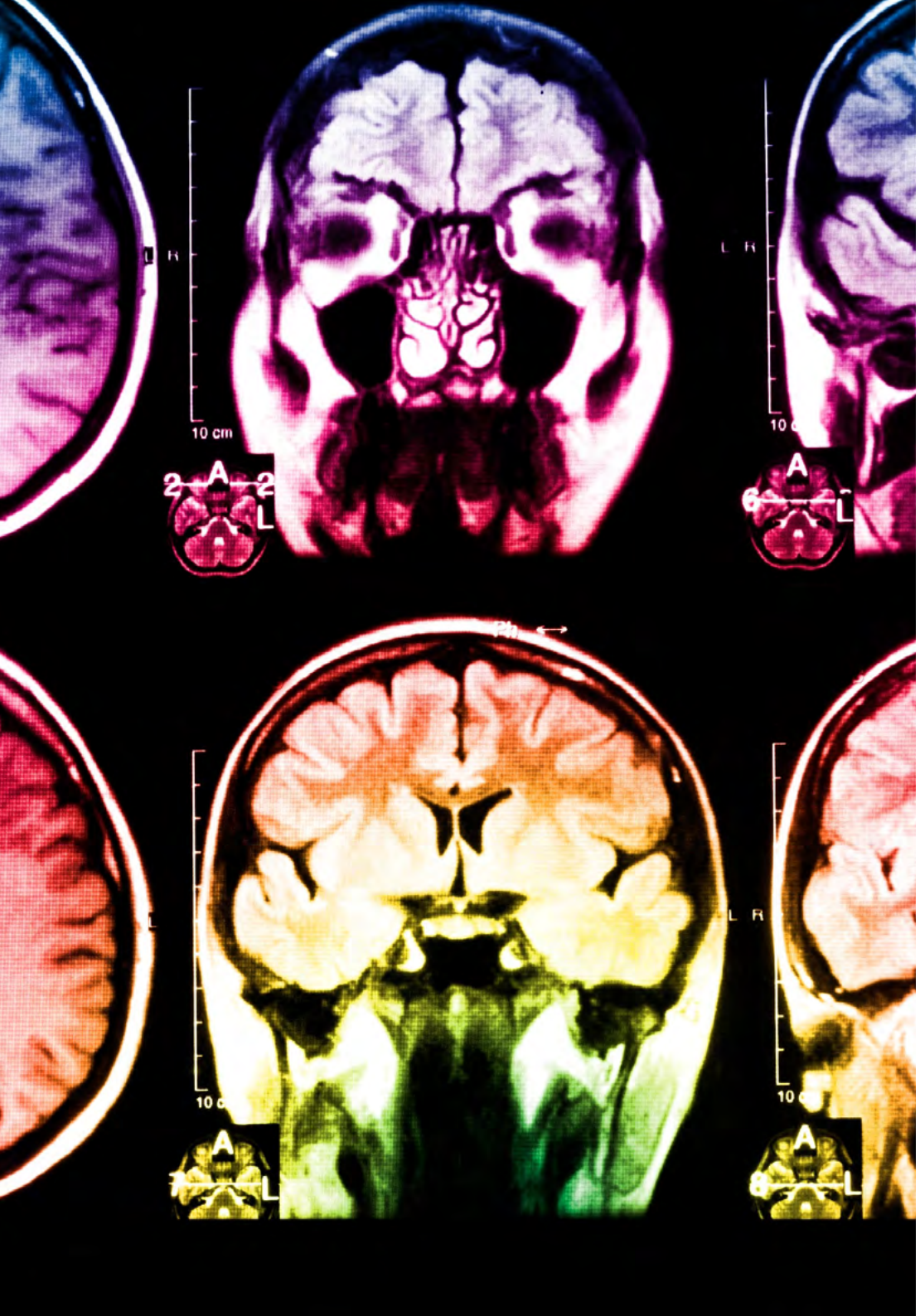
A TECH combina eficazmente a metodologia do Estudo de Caso com um sistema de aprendizagem 100% online baseado na repetição, que combina elementos didáticos diferentes em cada lição.

Melhoramos o Estudo de Caso com o melhor método de ensino 100% online: o Relearning.

Em 2019 obtivemos os melhores resultados de aprendizagem de todas as universidades online do mundo.

Na TECH aprende- com uma metodologia de vanguarda concebida para formar os gestores do futuro. Este método, na vanguarda da pedagogia mundial, chama-se Relearning.

A nossa universidade é a única universidade de língua espanhola licenciada para utilizar este método de sucesso. Em 2019, conseguimos melhorar os níveis globais de satisfação dos nossos estudantes (qualidade de ensino, qualidade dos materiais, estrutura dos cursos, objetivos...) no que diz respeito aos indicadores da melhor universidade online do mundo.



No nosso programa, a aprendizagem não é um processo linear, mas acontece numa espiral (aprender, desaprender, esquecer e reaprender). Portanto, cada um destes elementos é combinado de forma concêntrica. Esta metodologia formou mais de 650.000 licenciados com sucesso sem precedentes em áreas tão diversas como a bioquímica, genética, cirurgia, direito internacional, capacidades de gestão, ciência do desporto, filosofia, direito, engenharia, jornalismo, história, mercados e instrumentos financeiros. Tudo isto num ambiente altamente exigente, com um corpo estudantil universitário com um elevado perfil socioeconómico e uma idade média de 43,5 anos.

O Relearning permitir-lhe-á aprender com menos esforço e mais desempenho, envolvendo-o mais na sua capacitação, desenvolvendo um espírito crítico, defendendo argumentos e opiniões contrastantes: uma equação direta ao sucesso.

A partir das últimas provas científicas no campo da neurociência, não só sabemos como organizar informação, ideias, imagens e memórias, mas sabemos que o lugar e o contexto em que aprendemos algo é fundamental para a nossa capacidade de o recordar e armazenar no hipocampo, para o reter na nossa memória a longo prazo.

Desta forma, e no que se chama Neurocognitive context-dependent e-learning, os diferentes elementos do nosso programa estão ligados ao contexto em que o participante desenvolve a sua prática profissional.

Este programa oferece o melhor material educativo, cuidadosamente preparado para profissionais:

Material de estudo

Todos os conteúdos didáticos são criados pelos especialistas que irão ensinar o curso, especificamente para o curso, para que o desenvolvimento didático seja realmente específico e concreto.

Estes conteúdos são depois aplicados ao formato audiovisual, para criar o método de trabalho online da TECH. Tudo isto, com as mais recentes técnicas que oferecem peças de alta-qualidade em cada um dos materiais que são colocados à disposição do aluno.

Masterclasses

Existem provas científicas sobre a utilidade da observação por terceiros especializada.

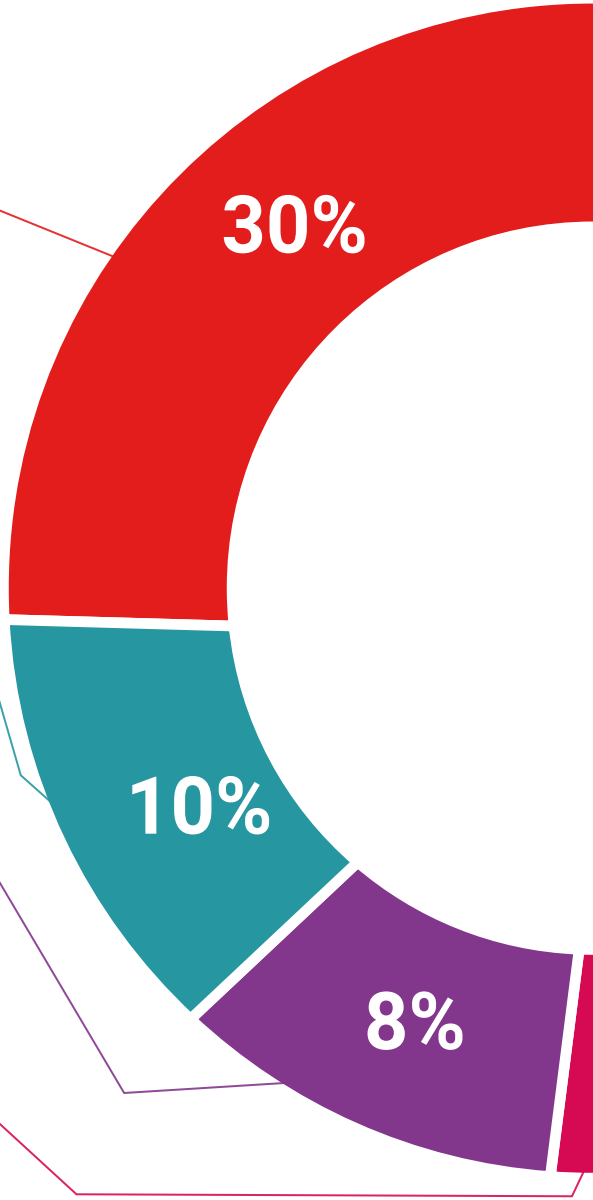
O denominado Learning from an Expert constrói conhecimento e memória, e gera confiança em futuras decisões difíceis.

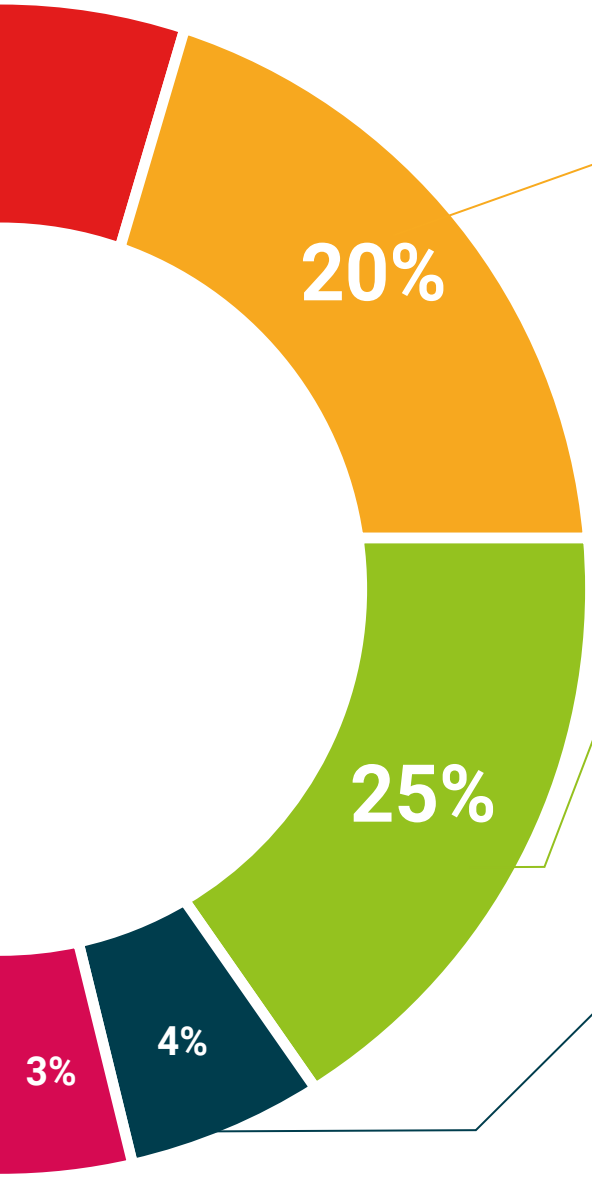
Práticas de aptidões e competências

Realizarão atividades para desenvolver competências e aptidões específicas em cada área temática. Práticas e dinâmicas para adquirir e desenvolver as competências e capacidades que um especialista necessita de desenvolver no quadro da globalização em que vivemos.

Leituras complementares

Artigos recentes, documentos de consenso e diretrizes internacionais, entre outros. Na biblioteca virtual da TECH o aluno terá acesso a tudo o que necessita para completar a sua capacitação





Case studies

Completarão uma seleção dos melhores estudos de casos escolhidos especificamente para esta situação. Casos apresentados, analisados e instruídos pelos melhores especialistas na cena internacional.

Resumos interativos

A equipa da TECH apresenta os conteúdos de uma forma atrativa e dinâmica em comprimidos multimédia que incluem áudios, vídeos, imagens, diagramas e mapas concetuais a fim de reforçar o conhecimento.

Este sistema educativo único para a apresentação de conteúdos multimédia foi premiado pela Microsoft como uma "História de Sucesso Europeu"

Testing & Retesting

Os conhecimentos do aluno são periodicamente avaliados e reavaliados ao longo de todo o programa, através de atividades e exercícios de avaliação e auto-avaliação, para que o aluno possa verificar como está a atingir os seus objetivos.

07

Certificação

O Mestrado Próprio em Qualidade do Software garante, para além de um conteúdo mais rigoroso e atualizado, o acesso a um grau de Mestre emitido pela TECH Universidade Tecnológica.

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

Este **Mestrado Próprio em Qualidade do Software** conta com o conteúdo educacional mais completo e atualizado do mercado.

Uma vez aprovadas as avaliações, o aluno receberá por correio, com aviso de receção, o certificado* correspondente ao título de **Mestrado Próprio** emitido pela **TECH Universidade Tecnológica.**

O certificado emitido pela **TECH Universidade Tecnológica** expressará a qualificação obtida no Mestrado Próprio, atendendo aos requisitos normalmente exigidos pelas bolsas de emprego, concursos públicos e avaliação de carreiras profissionais.

Título: **Mestrado Próprio em Qualidade do Software**

ECTS: **60**

Carga horária: **1.500 horas**


tech universidade tecnológica

Concede o presente

CERTIFICADO

a

Sr./Sra. _____, com o documento de identificação nº_____

Por ter concluído e acreditado com sucesso o

MESTRADO PRÓPRIO

em

Qualidade do Software

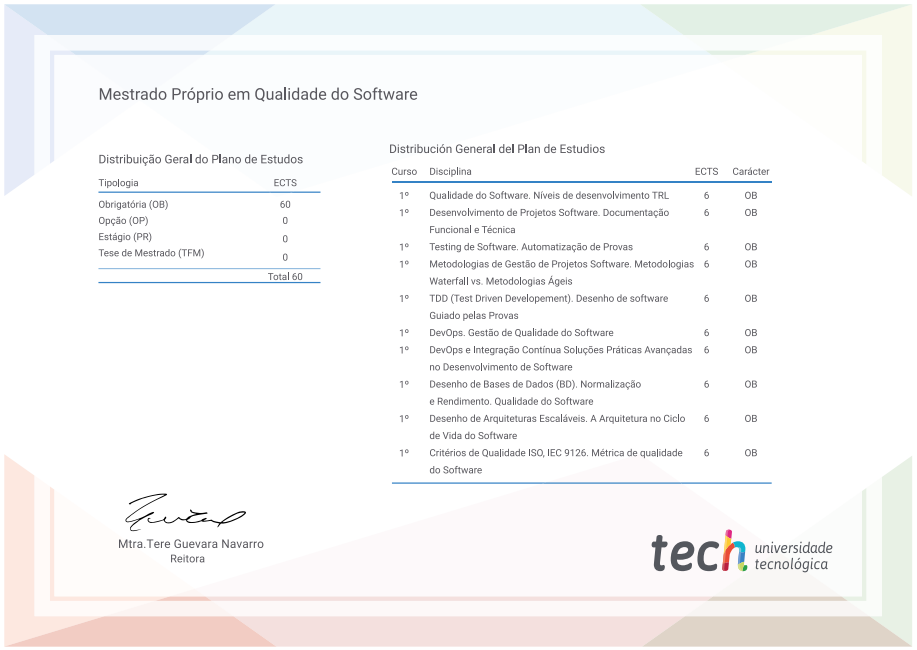
Este é um certificado atribuído por esta Universidade, reconhecido por 60 ECTS e equivalente a 1.500 horas, com data de início a dd/mm/aaaaa e data de conclusão a dd/mm/aaaaa.

A TECH é uma Instituição Privada de Ensino Superior reconhecida pelo Ministério da Educação Pública a partir de 28 de junho de 2018.

A 17 de junho de 2020

Prof.Tere Guevara Navarro
Reitora

código único TECH: AFWOR23S techtitute.com/titulos


Mestrado Próprio em Qualidade do Software

Distribuição Geral do Plano de Estudos

Tipologia	ECTS
Obrigatória (OB)	60
Opção (OP)	0
Estágio (PR)	0
Tese de Mestrado (TFM)	0
	Total 60

Distribución General del Plan de Estudios

Curso	Disciplina	ECTS	Carácter
1º	Qualidade do Software. Níveis de desenvolvimento TRL	6	OB
1º	Desenvolvimento de Projetos Software. Documentação Funcional e Técnica	6	OB
1º	Testing de Software. Automatização de Provas	6	OB
1º	Metodologias de Gestão de Projetos Software. Metodologias Waterfall vs. Metodologias Ágeis	6	OB
1º	TDD (Test Driven Developement). Desenho de software Guiado pelas Provas	6	OB
1º	DevOps. Gestão de Qualidade do Software	6	OB
1º	DevOps e Integração Contínua Soluções Práticas Avançadas no Desenvolvimento de Software	6	OB
1º	Desenho de Bases de Dados (BD). Normalização e Rendimento. Qualidade do Software	6	OB
1º	Desenho de Arquiteturas Escaláveis. A Arquitetura no Ciclo de Vida do Software	6	OB
1º	Critérios de Qualidade ISO, IEC 9126. Métrica de qualidade do Software	6	OB

Mtra.Tere Guevara Navarro
Reitora

tech universidade tecnológica

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

tech universidade tecnológica

Mestrado Próprio
Qualidade do Software

- Modalidade: **online**
- Duração: **12 meses**
- Certificação: **TECH Universidade Tecnológica**
- Créditos: **60 ECTS**
- Tempo Dedicado: **16 horas/semana**
- Horário: **ao seu próprio ritmo**
- Exames: **online**

Mestrado Próprio
Qualidade do Software

tech universidade tecnológica