

Mestrado Próprio

Criação de Interfaces
e Aplicações em Rede



Mestrado Próprio

Criação de Interfaces e Aplicações em Rede

- » Modalidade: online
- » Duração: 12 meses
- » Certificação: TECH Global University
- » Acreditação: 60 ECTS
- » Horário: ao seu próprio ritmo
- » Exames: online

Acesso ao site: www.techtute.com/pt/informatica/mestrado-proprio/mestrado-proprio-criacao-interfaces-aplicacoes-rede

Índice

01

Apresentação

pág. 4

02

Objetivos

pág. 8

03

Competências

pág. 14

04

Estrutura e conteúdo

pág. 18

05

Metodologia de estudo

pág. 32

06

Certificação

pág. 42

01

Apresentação

Intervir de forma competitiva na área da computação requer de conhecimentos sólidos que incluam as novidades e atualizações mais relevantes em engenharia de software, bases matemáticas, estatísticas e outras áreas. Neste programa construímos um percurso intensivo de especialização através do qual, o profissional poderá formar-se científica e tecnologicamente em engenharia informática, com qualidade, segurança e um objetivo final de excelência.



“

O nosso conceito inovador de teleprática dar-lhe-á a oportunidade de aprender por meio de uma experiência imersiva, que lhe proporcionará uma integração mais rápida e uma visão muito mais realista dos conteúdos: “Learning from an Expert”

Esta qualificação desenvolverá os conceitos necessários para trabalhar na criação de Interfaces, com a segurança que o domínio de todas as áreas de conhecimentos transversais a esta matéria, proporcionam ao especialista. Ao longo do programa, serão propostas abordagens didáticas inovadoras para conhecer em profundidade a arquitetura de uma aplicação distribuída. Serão apresentados os aspetos particulares da arquitetura cliente-servidor e tratados os fundamentos e desenvolvimentos imprescindíveis das linguagens de programação mais utilizadas, diferenciando entre linguagens, entre outras múltiplas questões fundamentais para o profissional.

Este conhecimento imprescindível torna-se também o primeiro passo para aceder à capacidade de desenvolvimento deste tipo de tecnologia.

Ao longo desta especialização, é fornecido um cenário real de trabalho para poder avaliar a conveniência da sua aplicação no projeto, avaliando as indicações reais, o modo de desenvolvimento e as expectativas que é possível ter em relação aos resultados.

Mediante a experiência, consegue-se aprender como se desenvolvem os conhecimentos necessários para avançar nesta área de trabalho. Esta aprendizagem, que necessita obrigatoriamente da experiência, consegue, nesta especialização, conciliar o ensino à distância e o ensino prático, oferecendo uma opção única para dar ao seu CV o impulso que procura.



Torne-se um dos profissionais mais procurados do momento: forme-se em engenharia informática através do mais completo e atualizado Mestrado Próprio em Criação de Interfaces e Aplicações em Rede

Este **Mestrado Próprio em Criação de Interfaces e Aplicações em Rede** conta com o conteúdo educativo mais completo e atualizado do mercado. As suas principais características:

- ♦ A mais recente tecnologia em software de ensino online
- ♦ Um sistema de ensino extremamente visual, apoiado por conteúdos gráficos e esquemáticos que são fáceis de assimilar e compreender
- ♦ O desenvolvimento de casos práticos apresentados por especialistas ativos
- ♦ Sistemas de vídeo interativo de última geração
- ♦ Ensino apoiado na teleprática
- ♦ Sistemas de atualização e requalificação contínua
- ♦ Aprendizagem autorregulável: total compatibilidade com outras ocupações
- ♦ Exercícios práticos de autoavaliação e verificação da aprendizagem
- ♦ Grupos de apoio e sinergias educativas: perguntas ao especialista, fóruns de discussão e conhecimento
- ♦ Comunicação com o professor e trabalhos de reflexão individual
- ♦ Disponibilidade dos conteúdos a partir de qualquer dispositivo fixo ou portátil com ligação à Internet
- ♦ Bancos de documentação complementar permanentemente disponíveis, inclusive após o programa



Com um design metodológico que se apoia em técnicas de ensino comprovadas pela sua eficácia, este inovador Mestrado Próprio em Criação de Interfaces e Aplicações em Rede levar-lhe-á através de diferentes abordagens pedagógicas para permitir-lhe aprender de uma forma dinâmica e eficaz”

O corpo docente é composto por profissionais de diferentes áreas relacionadas com esta especialidade. Desta forma, a TECH certifica-se de oferecer o objetivo de atualização educacional que se pretende. Um quadro multidisciplinar de profissionais qualificados e experientes em diferentes áreas, que desenvolverão os conhecimentos teóricos de forma eficiente, mas, sobretudo, que colocarão ao seu serviço os conhecimentos práticos derivados da sua própria experiência: uma das qualidades diferenciais desta qualificação.

Este domínio da matéria é completado pela eficácia do design metodológico. Desenvolvido por uma equipa de especialistas em *e-learning* integra os últimos avanços na tecnologia educacional. Desta forma, poderá estudar com uma gama de ferramentas multimédia cómodas e versáteis que dar-lhe-ão a operatividade de que necessita na sua educação.

O design deste programa baseia-se na Aprendizagem Baseada em Problemas: uma abordagem que concebe a aprendizagem como um processo eminentemente prático. Para o conseguir de forma remota, utilizaremos a teleprática: com a ajuda de um sistema inovador de vídeo interativo, e do Learning From an Expert, poderá adquirir os conhecimentos como se estivesse perante o cenário que está a aprender nesse momento. Um conceito que permitirá que a aprendizagem seja integrada e fundamentada de forma realista e permanente.

Um mestrado que qualificá-lo-á para trabalhar em todos os domínios da Criação de Interfaces e Aplicações em Rede com a solvência de um profissional de alto nível.

Com a experiência de profissionais ativos que leva-lhe-ão a um conhecimento real, imediato e concreto deste campo de trabalho.



02

Objetivos

O objetivo da TECH é fornecer a informação atualizada ao profissional sobre a criação de interfaces e aplicações de rede. Isto tem como objetivo que o engenheiro de sistemas adquira novas competências para a realização de programas informáticos mais completos. Um objetivo que, em somente alguns meses, poderá ser alcançado através de um plano de estudos centrado nas necessidades atuais do setor.



“

Expanda os seus conhecimentos em computação e engenharia de software e prepare-se para competir entre os melhores do setor”



Objetivos gerais

- ♦ Formar científica e tecnologicamente para a prática da engenharia informática
- ♦ Obter conhecimentos alargados no domínio da informática
- ♦ Obter conhecimentos alargados no domínio da estrutura de computadores
- ♦ Adquirir os conhecimentos necessários em engenharia de software
- ♦ Rever os fundamentos matemáticos, estatísticos e físicos essenciais para esta disciplina

“

Uma via de formação e crescimento profissional que o impulsionará para uma maior competitividade no mercado de trabalho”





Objetivos específicos

Módulo 1. Interação pessoa-computador

- ♦ Adquirir conhecimentos sólidos relacionados com a interação pessoa-computador e a criação de interfaces utilizáveis.
- ♦ Compreender a importância da usabilidade das aplicações e o porquê de ser fundamental tê-la em conta ao projetar o nosso software
- ♦ Compreender os diferentes tipos de diversidade humana, as limitações que implicam e a forma de adaptar as interfaces de acordo com as necessidades específicas de cada um deles
- ♦ Aprender o processo do design da interface, desde a análise dos requisitos até à avaliação, passando pelas várias fases intermédias necessárias para realizar uma interface adequada
- ♦ Conhecer as diferentes diretrizes de acessibilidade, os padrões que as estabelecem e as ferramentas que nos permitem avaliá-la
- ♦ Compreender os diferentes métodos de interação com o computador, através de periféricos e dispositivos

Módulo 2. Bases de dados

- ♦ Aprender as diferentes aplicações e finalidades dos sistemas de bases de dados, bem como o seu funcionamento e arquitetura
- ♦ Compreender o modelo relacional, desde a sua estrutura e operações até à álgebra relacional alargada.
- ♦ Aprender em profundidade o que são bases de dados SQL, o seu funcionamento, a definição de dados e a criação de consultas, desde as mais básicas às mais avançadas e complexas

- ♦ Aprender a conceber bases de dados utilizando o modelo entidade-relacionamento, como criar diagramas e as características do modelo E-R alargado
- ♦ Aprofundar o design de bases de dados relacionais, analisando as diferentes formas normais e algoritmos de decomposição
- ♦ Estabelecer as bases para compreender o funcionamento das bases de dados NoSQL, bem como introduzir a base de dados Mongo DB.

Módulo 3. Desenvolvimento de aplicações em rede

- ♦ Conhecer as características da linguagem de marcação HTML e o seu uso na criação de websites, juntamente com as folhas de estilo CSS.
- ♦ Aprender a utilizar a linguagem de programação orientada para o navegador JavaScript e algumas das suas principais características
- ♦ Compreender os conceitos de programação orientada para os componentes e de arquitetura de componentes
- ♦ Aprender a usar o *Framework* para *Front-End Bootstrap* para o design de sítios web
- ♦ Compreender a estrutura do modelo de visualização do controlador no desenvolvimento de sítios web dinâmicos
- ♦ Conhecer a arquitetura orientada para os serviços e as noções básicas do protocolo HTTP

Módulo 4. Software livre e conhecimento aberto

- ♦ Aprender os conceitos de software livre e de conhecimento aberto, bem como os diferentes tipos de licenças associadas
- ♦ Conhecer as principais ferramentas livres disponíveis em diferentes áreas, tais como sistemas operativos, gestão empresarial, gestores de conteúdos e criação de conteúdos multimédia, entre outros
- ♦ Compreender a importância e os benefícios do software livre no mundo empresarial, tanto pelas suas características como pelos seus custos

- ♦ Aprofundar o conhecimento do sistema operativo GNU/Linux, bem como das diferentes distribuições existentes, e como podem ser realizadas adaptações personalizadas delas
- ♦ Aprender sobre o funcionamento e desenvolvimento do WordPress, dado que este CMS é responsável por mais de 35% dos sites ativos no mundo, e mais de 60% no caso particular dos CMS
- ♦ Compreender o funcionamento do sistema operativo para dispositivos móveis Android, bem como as noções básicas de desenvolvimento de aplicações móveis, tanto nativamente como com *Frameworks* multiplataforma

Módulo 5. Bases de dados avançadas

- ♦ Introduzir os diferentes sistemas de bases de dados existentes atualmente no mercado.
- ♦ Aprender a utilizar a XML e as bases de dados para a web
- ♦ Compreender o funcionamento de bases de dados avançadas, tais como bases de dados paralelas e as distribuídas.
- ♦ Compreender a importância da indexação e da associação nos sistemas de bases de dados
- ♦ Compreender o funcionamento dos sistemas de processamento transacional e os sistemas de recuperação
- ♦ Adquirir conhecimentos relacionados com bases de dados não relacionais e extração de dados

Módulo 6. Engenharia do Software

- ♦ Compreender o quadro de referência da engenharia de software e a norma ISO/IEC 12207
- ♦ Aprender as características do processo unificado de desenvolvimento de software e o planeamento no contexto do desenvolvimento de software ágil.
- ♦ Conhecer os diferentes estilos de design de software distribuído e de arquiteturas de software orientadas para serviços

- ♦ Aprender os conceitos essenciais do design de interfaces gráficas de utilizador
- ♦ Compreender os conceitos básicos do desenvolvimento de aplicações web
- ♦ Aprofundar as estratégias e técnicas de teste de software, fatores de qualidade do software e das diferentes métricas utilizadas

Módulo 7. Programação avançada

- ♦ Aprofundar o conhecimento de programação, especialmente em relação à programação orientada a objetos, e os diferentes tipos de relações entre as classes existentes
- ♦ Conhecer os diferentes padrões de conceção para problemas orientados a objetos
- ♦ Aprender sobre a programação orientada por eventos e o desenvolvimento de interfaces de utilizadores com Qt
- ♦ Adquirir os conhecimentos essenciais de programação concorrente, os processos e os fios
- ♦ Aprender a gerir a utilização dos fios e a sincronização, bem como a resolução de problemas comuns no âmbito da programação concorrente
- ♦ Compreender a importância da documentação e das provas no desenvolvimento de software

Módulo 8. Reutilização de software

- ♦ Conhecer o panorama geral da estratégia de reutilização de software
- ♦ Aprender os diferentes padrões relacionados com a reutilização de software, em termos de design, como de criação, estruturais e de comportamento.
- ♦ Introduzir o conceito de *Framework*, bem como os principais tipos como os que se destinam ao design de interfaces gráficas de utilizador, ao desenvolvimento de aplicações web e à gestão da persistência de objetos em bases de dados
- ♦ Compreender o funcionamento do padrão amplamente utilizado atualmente Modelo Vista Controlador (MVC)

Módulo 9. Inteligência artificial e engenharia do conhecimento

- ♦ Estabelecer as bases da inteligência artificial e da engenharia do conhecimento, fazendo um breve percurso pela história da inteligência artificial até à atualidade
- ♦ Compreender os conceitos essenciais da pesquisa em inteligência artificial, tanto a pesquisa informada como a não informada
- ♦ Compreender o funcionamento da inteligência artificial nos jogos
- ♦ Aprender os conceitos fundamentais das redes neuronais e o uso dos algoritmos genéticos
- ♦ Adquirir os mecanismos adequados para representar o conhecimento, especialmente tendo em conta a Web semântica
- ♦ Compreender o funcionamento dos sistemas especializados e dos sistemas de suporte à decisão

Módulo 10. Engenharia do Software avançado

- ♦ Aprender em profundidade as diferentes metodologias ágeis utilizadas na engenharia de software
- ♦ Aprender a desenvolver utilizando as técnicas Scrum, programação extrema e desenvolvimento de software baseado na reutilização
- ♦ Compreender os diferentes padrões de arquitetura de sistema e de conceção de software, bem como a arquitetura das aplicações na nuvem
- ♦ Aprender a fazer testes ao software, com metodologias como *Test Driven Development*, *Acceptance Test Driven Development*, *Behavior Driven Development*, BDD e *Cucumber*
- ♦ Aprofundar a melhoria do processo de desenvolvimento de software e da qualidade do software utilizando as normas ISO/IEC
- ♦ Introduzir o conceito de DevOps e as suas principais práticas.

03

Competências

Este programa em Criação de Interfaces e Aplicações em Rede foi criado como uma ferramenta de elevada qualidade de formação para o profissional. O seu plano de estudos intensivo contribuirá significativamente para o desenvolvimento e trabalho de programadores e portais web, tirando partido dos recursos existentes na Internet e de softwares livres, numa orientação prática e útil.



“

*Com as competências que este Mestrado
Próprio permitir-lhe-á adquirir, poderá começar
a desenvolver aplicações e a trabalhar como
engenheiro de software, com a solvência de
uma formação completa e atualizada”*



Competências gerais

- ♦ Adquirir as competências necessárias para a prática profissional da engenharia com o conhecimento de todos os fatores necessários para a levar a cabo com qualidade e solvência

“

Uma oportunidade criada para profissionais que procuram um curso intensivo e eficaz para progredir significativamente no exercício da sua profissão”





Competências específicas

- ♦ Conhecer em profundidade todas as facetas da interação pessoa-computador e como envolvem os desenvolvimentos informáticos
- ♦ Ser competente no uso de bases de dados
- ♦ Desenvolver diferentes tipos de aplicações em rede
- ♦ Descrever e aproveitar o software livre e o conhecimento aberto existente na rede
- ♦ Trabalhar como engenheiro de software
- ♦ Controlar o uso de bases de dados avançadas
- ♦ Realizar uma programação avançada
- ♦ Saber como se reutiliza o software
- ♦ Criar interfaces e aplicações em rede
- ♦ Dominar os diferentes sistemas de trabalho em engenharia de software avançada

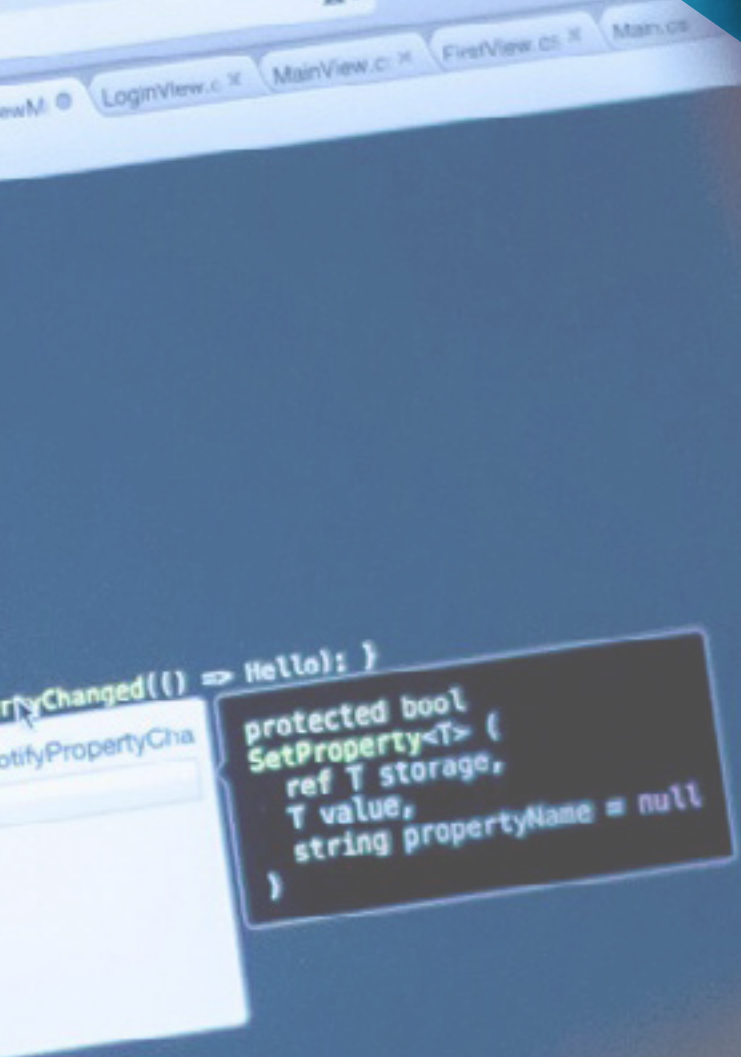
04

Estrutura e conteúdo

O conteúdo deste Mestrado Próprio foi desenvolvido pelos diferentes especialistas deste programa com um objetivo claro: assegurar que os alunos adquiram todas e cada uma das competências necessárias para se tornarem verdadeiros especialistas nesta área.

Um programa abrangente e bem estruturado que o levará aos mais altos padrões de qualidade e sucesso.





“

Um programa de ensino muito completo, estruturado em unidades didáticas completas e atualizadas com base nos últimos avanços do setor, orientado para uma aprendizagem compatível com a sua vida pessoal e profissional”

Módulo 1. Interação pessoa-computador

- 1.1. Introdução à interação pessoa-computador
 - 1.1.1. O que é a interação pessoa-computador
 - 1.1.2. Relação da interação pessoa-computador com outras disciplinas
 - 1.1.3. A interface de utilizador
 - 1.1.4. Usabilidade e acessibilidade
 - 1.1.5. Experiência de utilizador e design centrado no utilizador
- 1.2. O computador e a interação: interface de utilizador e paradigmas de interação
 - 1.2.1. A interação
 - 1.2.2. Paradigmas e estilos de interação
 - 1.2.3. Evolução das interfaces de utilizador
 - 1.2.4. Interfaces de utilizador clássicas: WIMP/GUI, comandos, voz, realidade virtual
 - 1.2.5. Interfaces de utilizador inovadoras: móveis, portáteis, colaborativas, BCI
- 1.3. O fator humano: aspetos psicológicos e cognitivos
 - 1.3.1. A importância do fator humano na interação
 - 1.3.2. O processamento humano de informação
 - 1.3.3. A entrada e saída da informação: visual, auditiva e tátil
 - 1.3.4. Percepção e atenção
 - 1.3.5. Conhecimento e modelos mentais: representação, organização e aquisição
- 1.4. O fator humano: limitações sensoriais e físicas
 - 1.4.1. Diversidade funcional, deficiência e deficiência
 - 1.4.2. Diversidade visual
 - 1.4.3. Diversidade auditiva
 - 1.4.4. Diversidade cognitiva
 - 1.4.5. Diversidade motora
 - 1.4.6. O caso dos imigrantes digitais
- 1.5. O processo de design (I): análise de requisitos para o design da interface de utilizador
 - 1.5.1. Design centrado no utilizador
 - 1.5.2. O que é a análise de requisitos
 - 1.5.3. A recolha de informação
 - 1.5.4. Análise e interpretação da informação
 - 1.5.5. Análise da usabilidade e acessibilidade



- 1.6. O processo de design (II): prototipagem e análise de tarefas
 - 1.6.1. Design concetual
 - 1.6.2. Prototipagem
 - 1.6.3. Análise hierárquica de tarefas
- 1.7. O processo de design (III): a avaliação
 - 1.7.1. Avaliação no processo de design: objetivos e métodos
 - 1.7.2. Métodos de avaliação sem utilizadores
 - 1.7.3. Métodos de avaliação com utilizadores
 - 1.7.4. Padrões e normas de avaliação
- 1.8. Acessibilidade: definição e diretrizes
 - 1.8.1. Acessibilidade e design universal
 - 1.8.2. A iniciativa WAI e as diretrizes WCAG
 - 1.8.3. Diretrizes WCAG 2.0 e 2.1
- 1.9. Acessibilidade: avaliação e diversidade funcional
 - 1.9.1. Ferramentas de avaliação da acessibilidade na web
 - 1.9.2. Acessibilidade e diversidade funcional
- 1.10. O computador e a interação: periféricos e dispositivos
 - 1.10.1. Dispositivos e periféricos tradicionais
 - 1.10.2. Dispositivos e periféricos alternativos
 - 1.10.3. Móveis e tablets
 - 10.1.1. Diversidade funcional, interação e periféricos

Módulo 2. Bases de dados

- 2.1. Aplicações e objetivos dos sistemas de bases de dados
 - 2.1.1. Aplicações de diferentes sistemas de bases de dados
 - 2.1.2. Propósito nos diferentes sistemas de bases de dados
 - 2.1.3. Visão dos dados
- 2.2. Base de dados e arquitetura
 - 2.2.1. Base de dados relacionais
 - 2.2.2. O design de base de dados
 - 2.2.3. Bases de dados baseadas em objetos e semi-estruturadas
 - 2.2.4. Armazenamento de dados e consultas
 - 2.2.5. Gestão de transações
 - 2.2.6. Extração e análise de dados
 - 2.2.7. Arquitetura das bases de dados

- 2.3. O modelo relacional: estrutura, operações e álgebra relacional alargada
 - 2.3.1. A estrutura das BD relacionais
 - 2.3.2. Operações fundamentais na álgebra relacional
 - 2.3.3. Outras operações da álgebra relacional
 - 2.3.4. Operações da álgebra relacional estendida
 - 2.3.5. Valores nulos
 - 2.3.6. Modificação da base de dados
- 2.4. SQL (I)
 - 2.4.1. O que é SQL?
 - 2.4.2. A definição de dados
 - 2.4.3. Estrutura básica das consultas SQL
 - 2.4.4. Operações sobre conjuntos
 - 2.4.5. Funções de agregação
 - 2.4.6. Valores nulos
- 2.5. SQL (II)
 - 2.5.1. Subconsultas aninhadas
 - 2.5.2. Consultas complexas
 - 2.5.3. Visualizações
 - 2.5.4. Cursores
 - 2.5.5. Consultas complexas
 - 2.5.6. Disparadores
- 2.6. Design de base de dados e o modelo E-R
 - 2.6.1. Visão geral do processo de design
 - 2.6.2. O modelo entidade-relacionamento
 - 2.6.3. Restrições
- 2.7. Diagramas entidade-relacionamento
 - 2.7.1. Diagramas entidade-relacionamento
 - 2.7.2. Aspectos do design entidade-relacionamento
 - 2.7.3. Conjuntos de entidades fracas
- 2.8. O modelo entidade-relacionamento alargado
 - 2.8.1. Características do modelo E-R estendido
 - 2.8.2. Design de uma base de dados
 - 2.8.3. Redução a esquemas relacionais

- 2.9. Design de bases de dados relacionais
 - 2.9.1. Características dos bons designs relacionais
 - 2.9.2. Domínios atômicos e a primeira forma normal (1FN)
 - 2.9.3. Decomposição por dependências funcionais
 - 2.9.4. Teoria das dependências funcionais
 - 2.9.5. Algoritmos de decomposição
 - 2.9.6. Decomposição por dependências multivaloradas
 - 2.9.7. Mais formas normais
 - 2.9.8. Processo de design das bases de dados
- 2.10. Bases de dados NoSQL
 - 17.10.1. O que são as bases de dados NoSQL?
 - 17.10.2. Análise das diferentes opções de NoSQL e suas características
 - 17.10.3. Mongo DB

Módulo 3. Desenvolvimento de aplicações em rede

- 3.1. Linguagens de marcação HTML5
 - 3.1.1. Conceitos básicos de HTML
 - 3.1.2. Novos elementos HTML 5
 - 3.1.3. Formulários: novos controles
- 3.2. Introdução a folhas de estilo CSS
 - 3.2.1. Primeiros passos com CSS
 - 3.2.2. Introdução ao CSS3
- 3.3. Linguagem de script de navegador: JavaScript
 - 3.3.1. Conceitos básicos de JavaScript
 - 3.3.2. DOM
 - 3.3.3. Eventos
 - 3.3.4. JQuery
 - 3.3.5. Ajax
- 3.4. Conceito de programação orientada a componentes
 - 3.4.1. Contexto
 - 3.4.2. Componentes e interfaces
 - 3.4.3. Estados de um componente

- 3.5. Arquitetura de componentes
 - 3.5.1. Arquiteturas atuais
 - 3.5.2. Integração e implementação de componentes
- 3.6. *Framework Front-End: Bootstrap*
 - 3.6.1. Design com grelha
 - 3.6.2. Formulários
 - 3.6.3. Componentes
- 3.7. Modelo vista controlador
 - 3.7.1. Métodos de desenvolvimento web
 - 3.7.2. Padrão de design: MVC
- 3.8. Tecnologias *Grid* da informação
 - 3.8.1. Incremento de recursos em computação
 - 3.8.2. Conceito de tecnologia *Grid*
- 3.9. Arquitetura orientada a serviços
 - 3.9.1. SOA e serviços web
 - 3.9.2. Topologia de um serviço web
 - 3.9.3. Plataformas para serviços web
- 3.10. Protocolo HTTP
 - 3.10.1. Mensagens
 - 3.10.2. Sessões persistentes
 - 3.10.3. Sistema criptográfico
 - 3.10.4. Funcionamento do protocolo HTTPS

Módulo 4. Software livre e conhecimento aberto

- 4.1. Introdução ao software livre
 - 4.1.1. História do software livre
 - 4.1.2. "Liberdade" em software
 - 4.1.3. Licenças para a utilização de ferramentas de software
 - 4.1.4. Propriedade intelectual do software
 - 4.1.5. Qual é a motivação para a utilização de software livre?
 - 4.1.6. Mitos do software livre
 - 4.1.7. Top500:

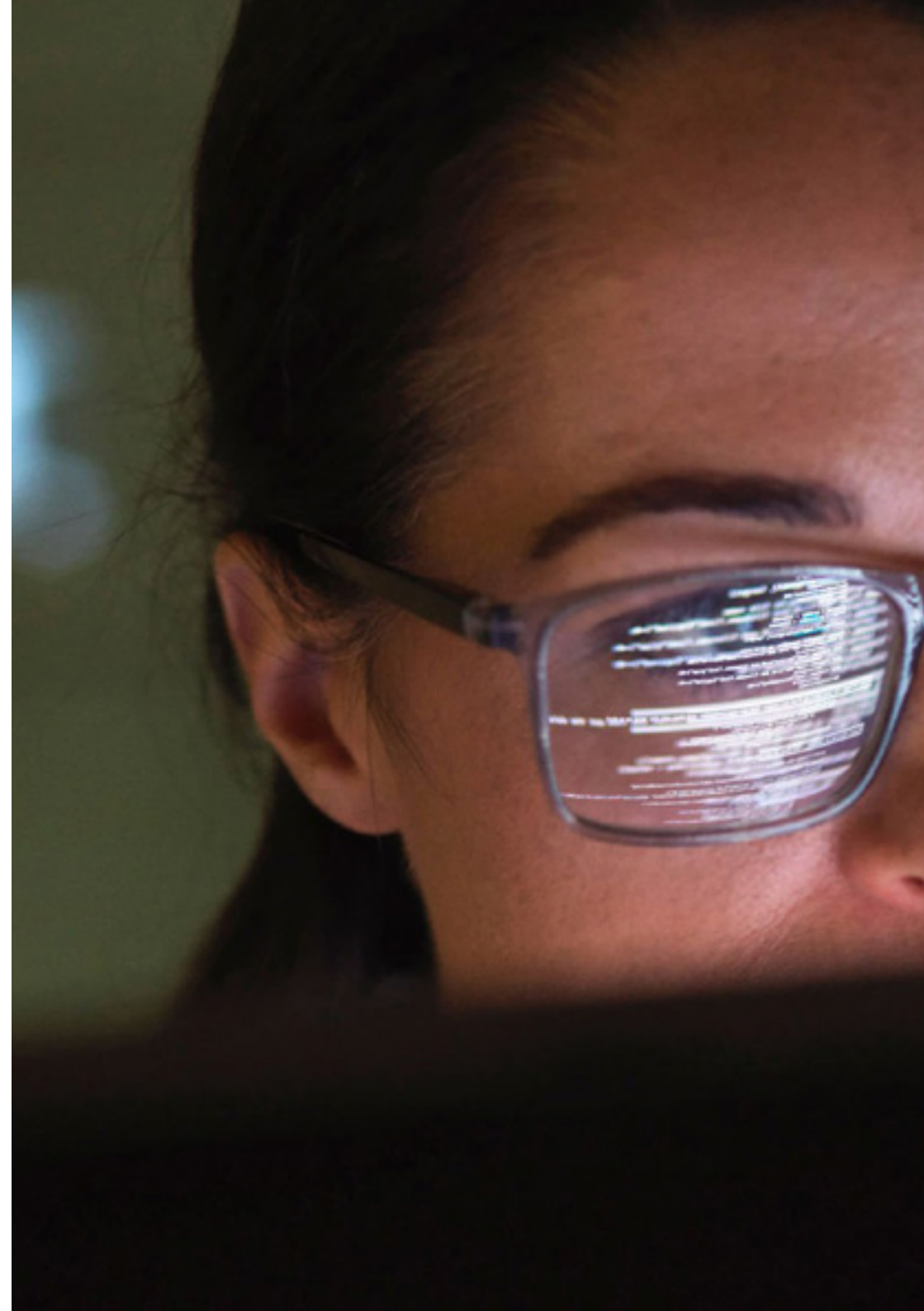
- 4.2. Conhecimento aberto e licenças CC
 - 4.2.1. Conceitos básicos
 - 4.2.2. Licenças *Creative Commons*
 - 4.2.3. Outras licenças de conteúdo
 - 4.2.4. Wikipédia e outros projetos de conhecimento aberto
- 4.3. Principais ferramentas de software livre
 - 4.3.1. Sistemas operativos
 - 4.3.2. Aplicações de escritório
 - 4.3.3. Aplicações de gestão empresarial
 - 4.3.4. Gestores de conteúdo web
 - 4.3.5. Ferramentas de criação de conteúdos multimédia
 - 4.3.6. Outras aplicações
- 4.4. A empresa: o software livre e os seus custos
 - 4.4.1. Software livre: sim ou não?
 - 4.4.2. Verdades e mentiras sobre software livre
 - 4.4.3. Software empresarial baseado em software livre e de código aberto
 - 4.4.4. Custos de software
 - 4.4.5. Modelos de software livre
- 4.5. O sistema operativo GNU/Linux
 - 4.5.1. Arquitetura
 - 4.5.2. Estrutura de diretórios básica
 - 4.5.3. Características e estrutura do sistema de arquivo
 - 4.5.4. Representação interna dos ficheiros
- 4.6. O sistema operativo mobile Android
 - 4.6.1. História
 - 4.6.2. Arquitetura
 - 4.6.3. Forks de Android
 - 4.6.4. Introdução ao desenvolvimento para Android
 - 4.6.5. *Frameworks* para o desenvolvimento de aplicações móveis

- 4.7. Criação de websites com WordPress
 - 4.7.1. Características e estrutura do WordPress
 - 4.7.2. Criação de sites wordpress.com
 - 4.7.3. Instalação e configuração do WordPress no seu próprio servidor
 - 4.7.4. Instalação de plugins e extensão do WordPress
 - 4.7.5. Criação de plugins WordPress
 - 4.7.6. Criação de temas WordPress
- 4.8. Tendências do software livre
 - 4.8.1. Ambientes na nuvem
 - 4.8.2. Ferramentas de monitorização
 - 4.8.3. Sistemas operativos
 - 4.8.4. Big Data e Open Data 2,0
 - 4.8.5. Computação quântica
- 4.9. Controlo de versões
 - 4.9.1. Conceitos básicos
 - 4.9.2. Git
 - 4.9.3. Nuvens e serviços de Git auto-hospedados
 - 4.9.4. Outros sistemas de controlo de versões
- 4.10. Distribuições GNU/Linux personalizadas
 - 4.10.1. Principais distribuições
 - 4.10.2. Distribuições derivadas de Debian
 - 4.10.3. Criação de pacotes deb
 - 4.10.4. Modificação da distribuição
 - 4.10.5. Geração de imagem ISO

Módulo 5. Bases de dados avançadas

- 5.1. Introdução aos diferentes sistemas de bases de dados
 - 5.1.1. Revisão histórica
 - 5.1.2. Bases de dados hierárquicas
 - 5.1.3. Bases de dados de rede
 - 5.1.4. Bases de dados relacionais
 - 5.1.5. Bases de dados não relacionais

- 5.2. XML e bases de dados para a Web
 - 5.2.1. Validação de documentos XML
 - 5.2.2. Transformações de documentos XML
 - 5.2.3. Armazenamento de dados XML
 - 5.2.4. Bases de dados relacionais XML
 - 5.2.5. SQL/XML
 - 5.2.6. Bases de dados nativas XML
- 5.3. Bases de dados paralelas
 - 5.3.1. Sistemas paralelos
 - 5.3.2. Arquiteturas paralelas de bases de dados
 - 5.3.3. Paralelismo nas consultas
 - 5.3.4. Paralelismo entre consultas
 - 5.3.5. Design de sistemas paralelos
 - 5.3.6. Processamento paralelo em SQL
- 5.4. Bases de dados distribuídas
 - 5.4.1. Sistemas distribuídos
 - 5.4.2. Armazenamento distribuído
 - 5.4.3. Disponibilidade
 - 5.4.4. Processamento distribuído de consultas
 - 5.4.5. Fornecedores de bases de dados distribuídas
- 5.5. Indexação e associação
 - 5.5.1. Índices ordenados
 - 5.5.2. Índices densos e dispersos
 - 5.5.3. Índices multinível
 - 5.5.4. Atualização do índice
 - 5.5.5. Associação estática
 - 5.5.6. Como usar índices em bases de dados
- 5.6. Introdução ao processamento transacional
 - 5.6.1. Estados de uma transação
 - 5.6.2. Implementação da atomicidade e durabilidade
 - 5.6.3. Sequencialidade
 - 5.6.4. Recuperabilidade
 - 5.6.5. Implementação do isolamento



- 5.7. Sistemas de recuperação
 - 5.7.1. Classificação de falhas
 - 5.7.2. Estruturas de armazenamento
 - 5.7.3. Recuperação e atomicidade
 - 5.7.4. Recuperação baseada no registro histórico
 - 5.7.5. Transações concorrentes e recuperação
 - 5.7.6. Alta disponibilidade em bases de dados
- 5.8. Execução e processamento de consultas
 - 5.8.1. Custo de uma consulta
 - 5.8.2. Operação de seleção
 - 5.8.3. Ordenação
 - 5.8.4. Introdução à otimização de consultas
 - 5.8.5. Monitorização do desempenho
- 5.9. Bases de dados não relacionais
 - 5.9.1. Bases de dados orientadas a documentos
 - 5.9.2. Bases de dados orientadas a grafos
 - 5.9.3. Bases de dados chave-valor
- 5.10. Data Warehouse, OLAP e extração de dados
 - 5.10.1. Componentes dos armazéns de dados
 - 5.10.2. Arquitetura de um data warehouse
 - 5.10.3. OLAP
 - 5.10.4. Funcionalidades de extração de dados
 - 5.10.5. Outros tipos de extração

Módulo 6. Engenharia do Software

- 6.1. Estrutura de Engenharia de software
 - 6.1.1. Características do Software
 - 6.1.2. Os principais processos na engenharia de software
 - 6.1.3. Modelos de processo de desenvolvimento de software
 - 6.1.4. Quadro de referência padrão para o processo de desenvolvimento de software: a norma ISO/IEC 12207

- 6.2. Processo unificado de desenvolvimento de software
 - 6.2.1. Processo unificado
 - 6.2.2. Dimensões do processo unificado
 - 6.2.3. Processo de desenvolvimento orientado por casos de utilização
 - 6.2.4. Fluxos de trabalho fundamentais de processos unificados
- 6.3. Planeamento no contexto de desenvolvimento de software ágil
 - 6.3.1. Características do desenvolvimento de software ágil
 - 6.3.2. Diferentes horizontes temporais de planeamento no desenvolvimento ágil
 - 6.3.3. Quadro de desenvolvimento ágil Scrum e horizontes temporais de planificação
 - 6.3.4. Histórias de utilizadores como unidade de planeamento e estimativa
 - 6.3.5. Técnicas comuns para a obtenção de uma estimativa
 - 6.3.6. Escalas para a interpretação de estimativas
 - 6.3.7. *Planning poker*
 - 6.3.8. Tipos de planeamento comuns: planeamento de entregas e planeamento de interação
- 6.4. Estilos de desenho de software distribuído e arquitetura de software orientadas para os serviços
 - 6.4.1. Modelos de comunicação em sistemas de software distribuídos
 - 6.4.2. Camada intermédia ou middleware
 - 6.4.3. Padrões de arquitetura para sistemas distribuídos
 - 6.4.4. Processo geral de desenho de serviços de software
 - 6.4.5. Aspectos de desenho de serviços de software
 - 6.4.6. Composição de serviços
 - 6.4.7. Arquitetura de serviços web
 - 6.4.8. Componentes de Infraestrutura e SOA
- 6.5. Introdução ao desenvolvimento de software orientado por modelos
 - 6.5.1. O conceito de modelo
 - 6.5.2. Desenvolvimento de software orientado por modelos
 - 6.5.3. Quadro de referência de desenvolvimento dirigido por modelos MDA
 - 6.5.4. Elementos de um modelo de transformação
- 6.6. Desenho de interfaces gráficas de utilizador
 - 6.6.1. Princípios de desenho de interfaces de utilizador
 - 6.6.2. Padrões de desenho arquitectónico para sistemas interativos: Modelo Vista Controlador (MVC)
 - 6.6.3. Experiência de utilizador (UX *User Experience*)
 - 6.6.4. Design centrado no utilizador
 - 6.6.5. Processo de análise e desenho do interface gráfico de utilizador
 - 6.6.6. Usabilidade de interfaces de utilizador
 - 6.6.7. Acessibilidade em interfaces de utilizador
- 6.7. Desenho de aplicações web
 - 6.7.1. Características das aplicações web
 - 6.7.2. Interface de utilizador de uma aplicação web
 - 6.7.3. Desenho de navegação
 - 6.7.4. Protocolo de interação de base para aplicações web
 - 6.7.5. Estilos de arquitetura para aplicações web
- 6.8. Estratégias e técnicas de provas de software e fatores de qualidade de software
 - 6.8.1. Estratégias de prova
 - 6.8.2. Desenhos de casos de prova
 - 6.8.3. Relação custo qualidade
 - 6.8.4. Modelos de qualidade
 - 6.8.5. Família de normas ISO/IEC 25000 (SQuaRE)
 - 6.8.6. Modelo de qualidade de produto (ISO 2501n)
 - 6.8.7. Modelo de qualidade de dados (ISO 2501n)
 - 6.8.8. Gestão de qualidade de software
- 6.9. Introdução à métrica em engenharia de software
 - 6.9.1. Conceitos básicos: medidas, métricas e indicadores
 - 6.9.2. Tipos de métricas em engenharia de software
 - 6.9.3. O processo de Medição
 - 6.9.4. Módulo 25024. Métricas externas e de qualidade em uso
 - 6.9.5. Métrica orientada a objetos

- 6.10. Manutenção e reengenharia de software
 - 6.10.1. Processos de manutenção
 - 6.10.2. Quadro padrão de processo de manutenção ISO/IEC 14764
 - 6.10.3. Modelo de processo de reengenharia de software
 - 6.10.4. Engenharia reversa

Módulo 7. Programação avançada

- 7.1. Introdução à programação orientada a objetos
 - 7.1.1. Introdução à programação orientada a objetos
 - 7.1.2. Desenho de classes
 - 7.1.3. Introdução à UML para modelação de problemas
- 7.2. Relações entre classes
 - 7.2.1. Abstração e herança
 - 7.2.2. Conceitos avançados de herança
 - 7.2.3. Poliformismo
 - 7.2.4. Composição e agregação
- 7.3. Introdução aos padrões de de desenho para problemas orientados a objetos
 - 7.3.1. O que é um padrão de desenho?
 - 7.3.2. Padrão *Factory*
 - 7.3.3. Padrão *Singleton*
 - 7.3.4. Padrão *Observer*
 - 7.3.5. Padrão *Composite*
- 7.4. Exceções
 - 7.4.1. O que são as exceções
 - 7.4.2. Captura e gestão de exceções
 - 7.4.3. Lançamento de exceções
 - 7.4.4. Criação de exceções
- 7.5. Interfaces de utilizadores
 - 7.5.1. Introdução a Qt
 - 7.5.2. Posicionamento
 - 7.5.3. O que são os eventos?
 - 7.5.4. Eventos: definição e captura
 - 7.5.5. Desenvolvimento de interfaces de utilizador
- 7.6. Introdução à programação concorrente
 - 7.6.1. Introdução à programação concorrente
 - 7.6.2. O conceito de processo e thread
 - 7.6.3. Interação entre processos ou threads
 - 7.6.4. Os threads em C++
 - 7.6.5. Vantagens e desvantagens da programação concorrente
- 7.7. Gestão de threads e sincronização
 - 7.7.1. Ciclo de vida um thread
 - 7.7.2. A classe *Thread*
 - 7.7.3. Planificação de threads
 - 7.7.4. Grupos threads
 - 7.7.5. Threads de tipo demónio
 - 7.7.6. Sincronização
 - 7.7.7. Mecanismos de bloqueio
 - 7.7.8. Mecanismos de comunicação
 - 7.7.9. Monitores
- 7.8. Problemas comuns dentro da programação concorrente
 - 7.8.1. O problema dos produtores consumidores
 - 7.8.2. O problema dos leitores e escritores
 - 7.8.3. O problema do jantar dos filósofos
- 7.9. Documentação e provas de software
 - 7.9.1. Porque é importante documentar o software?
 - 7.9.2. Documentação de desenho
 - 7.9.3. Uso de ferramentas para a documentação
- 7.10. Provas de software
 - 7.10.1. Introdução às provas de software
 - 7.10.2. Tipos de provas
 - 7.10.3. Prova de unidade
 - 7.10.4. Prova de integração
 - 7.10.5. Prova de validação
 - 7.10.6. Prova do sistema

Módulo 8. Reutilização de software

- 8.1. Panorama geral da reutilização de software
 - 8.1.1. Em que consiste a reutilização do software
 - 8.1.2. Vantagens e desvantagens da reutilização de software
 - 8.1.3. Técnicas principais de reutilização de software
- 8.2. Introdução aos padrões de desenho
 - 8.2.1. O que é um padrão de de desenho?
 - 8.2.2. Catálogo dos principais padrões de desenho
 - 8.2.3. Como usar padrões para resolver problemas de desenho
 - 8.2.4. Como selecionar o melhor padrão de desenho
- 8.3. Padrões de criação
 - 8.3.1. Padrões de criação
 - 8.3.2. Padrão *Abstract Factory*
 - 8.3.3. Exemplo de implementação do padrão *Abstract Factory*
 - 8.3.4. Padrão Builder
 - 8.3.5. Exemplo de implementação do Builder
 - 8.3.6. Padrão *Abstract Factory* vs. Builder
- 8.4. Padrões de criação(II)
 - 8.4.1. Padrão *Factory Method*
 - 8.4.2. *Factory Method* vs. *Abstract Factory*
 - 8.4.3. Padrão Singleton
- 8.5. Padrões estruturais
 - 8.5.1. Padrões estruturais
 - 8.5.2. Padrão *Adapter*
 - 8.5.3. Padrão *Bridge*
- 8.6. Padrões estruturais (II)
 - 8.6.1. Padrão *Composite*
 - 8.6.2. Padrão decorador
- 8.7. Padrões estruturais (III)
 - 8.7.1. Padrão Facade
 - 8.7.2. Padrão Proxy

- 8.8. Padrões de comportamento
 - 8.8.1. Conceito dos padrões de comportamento
 - 8.8.2. Padrão de comportamento: Cadeia de responsabilidade
 - 8.8.3. Padrão de comportamento ordem
- 8.9. Padrões de comportamento(II)
 - 8.9.1. Padrão intérprete ou *Interpreter*
 - 8.9.2. Padrão iterador
 - 8.9.3. Padrão observador
 - 8.9.4. Padrão estratégia
- 8.10. *Frameworks*
 - 8.10.1. Conceito de *Frameworks*
 - 8.10.2. Desenvolvimento através de *Frameworks*
 - 8.10.3. Padrão *Model View Controller*
 - 8.10.4. *Framework* para o design da interface gráfica do utilizador
 - 8.10.5. *Frameworks* para o desenvolvimento de aplicações web
 - 8.10.6. *Frameworks* para a gestão da persistência de objetos em bases de dados

Módulo 9. Inteligência artificial e engenharia do conhecimento

- 9.1. Introdução à inteligência artificial e à engenharia do conhecimento
 - 9.1.1. Breve história da inteligência artificial
 - 9.1.2. A inteligência artificial na atualidade
 - 9.1.3. Engenharia do conhecimento
- 9.2. Pesquisa
 - 9.2.1. Conceptos comuns de pesquisa
 - 9.2.2. Pesquisa não informada
 - 9.2.3. Pesquisa informada
- 9.3. Satisfazibilidade booleana, satisfazibilidade de restrições e planejamento automático
 - 9.3.1. Satisfazibilidade booleana
 - 9.3.2. Problemas de satisfatividade de restrições
 - 9.3.3. Planejamento automático e PDDL
 - 9.3.4. Planejamento como busca heurística
 - 9.3.5. Planejamento com SAT

- 9.4. Inteligência artificial nos jogos
 - 9.4.1. Teoria dos jogos
 - 9.4.2. Minimax e poda Alfa-Beta
 - 9.4.3. Simulação: Monte Carlo
- 9.5. Aprendizagem supervisionada e não supervisionada
 - 9.5.1. Introdução à aprendizagem automática
 - 9.5.2. Classificação
 - 9.5.3. Regressão
 - 9.5.4. Validação dos resultados
 - 9.5.5. Agrupamento (*Clustering*)
- 9.6. Redes neurais
 - 9.6.1. Fundamentos teóricos
 - 9.6.2. Modelo computacional
 - 9.6.3. Redes neurais supervisionadas e não supervisionadas
 - 9.6.4. Perceptron simples
 - 9.6.5. Perceptron multicamadas
- 9.7. Algoritmos genéticos
 - 9.7.1. História
 - 9.7.2. Base biológica
 - 9.7.3. Codificação de problemas
 - 9.7.4. Criação da população inicial
 - 9.7.5. Algoritmo principal e operadores genéticos
 - 9.7.6. Avaliação de indivíduos: fitness
- 9.8. Tesouros, vocabulários, taxonomias
 - 9.8.1. Vocabulários
 - 9.8.2. Taxonomias
 - 9.8.3. Tesauro
 - 9.8.4. Ontologias
- 9.9. Representação do conhecimento: web semântica
 - 9.9.1. Web semântica
 - 9.9.2. Especificações: RDF, RDFS e OWL
 - 9.9.3. Inferência/razoabilidade
 - 9.9.4. *Linked Data*

- 9.10. Sistemas periciais e DSS
 - 9.10.1. Sistemas periciais
 - 9.10.2. Sistema de apoio à decisão

Módulo 10. Engenharia do Software avançado

- 10.1. Introdução às metodologias ágeis
 - 10.1.1. Modelo de processo e metodologias
 - 10.1.2. Agilidade e processos ágeis
 - 10.1.3. Manifesto ágil
 - 10.1.4. Algumas metodologias ágeis
 - 10.1.5. Ágil vs. Tradicional
- 10.2. Scrum
 - 10.2.1. Origens e filosofia de Scrum
 - 10.2.2. Valores de Scrum
 - 10.2.3. Fluxo do processo Scrum
 - 10.2.4. Os papéis em Scrum
 - 10.2.5. Os artefactos em Scrum
 - 10.2.6. Os eventos de Scrum
 - 10.2.7. As histórias de utilizador
 - 10.2.8. Extensões de Scrum
 - 10.2.9. Estimativas ágeis
 - 10.2.10. Escalada Scrum
- 10.3. Programação extrema
 - 10.3.1. Justificação e visão geral da XP
 - 10.3.2. O ciclo de vida em XP
 - 10.3.3. Os cinco valores básicos
 - 10.3.4. As doze práticas básicas em XP
 - 10.3.5. Papéis dos participantes
 - 10.3.6. XP industrial
 - 10.3.7. Valoração crítica de XP

- 10.4. Desenvolvimento de software baseado em reutilização
 - 10.4.1. A reutilização do software
 - 10.4.2. Níveis de reutilização de código
 - 10.4.3. Técnicas concretas de reutilização
 - 10.4.4. Desenvolvimento baseado em componentes
 - 10.4.5. Benefícios e problemas da reutilização
 - 10.4.6. Planificação da reutilização
- 10.5. Padrões arquitetura de sistemas e de desenho de software
 - 10.5.1. O Desenho arquitetónico
 - 10.5.2. Padrões arquitetónicos gerais
 - 10.5.3. Arquitetura tolerante a falhas
 - 10.5.4. Arquiteturas de sistema distribuídos
 - 10.5.5. Os padrões de design
 - 10.5.6. Padrões de Gamma
 - 10.5.7. Padrões de design de interação
- 10.6. Arquitetura de aplicações na nuvem
 - 10.6.1. Fundamentos de *Cloud Computing*
 - 10.6.2. Qualidade das aplicações na nuvem
 - 10.6.3. Estilos de arquitetura
 - 10.6.4. Padrões de design
- 10.7. Provas de software: TDD, ATDD e BDD
 - 10.7.1. Verificação e validação do software
 - 10.7.2. As provas de software
 - 10.7.3. *Test Driven Development* (TDD)
 - 10.7.4. *Acceptance Test Driven Development* (ATDD)
 - 10.7.5. *Behavior Driven Development* (BDD)
 - 10.7.6. BDD e *Cucumber*
- 10.8. A melhoria do processo de software
 - 10.8.1. A melhoria do processo de software
 - 10.8.2. O processo de melhoria de processos
 - 10.8.3. Modelos de maturação
 - 10.8.4. O modelo CMMI
 - 10.8.5. CMMI V2.0
 - 10.8.6. CMMI e ágil





- 10.9. A qualidade do produto de software SQuaRE
 - 10.9.1. A Qualidade do Software
 - 10.9.2. Modelo de qualidade do produto de software
 - 10.9.3. Família ISO/IEC 25000
 - 10.9.4. ISO/IEC 25010: modelo e características de qualidade
 - 10.9.5. ISO/IEC 25012: a qualidade dos dados
 - 10.9.6. ISO/IEC 25020: medição da qualidade do software
 - 10.9.7. ISO/IEC 25022, 25023 e 25024: métricas de qualidade do software e dos dados
 - 10.9.8. ISO/IEC 25040: avaliação do software
 - 10.9.9. O processo de certificação
- 10.10. Introdução ao DevOps
 - 10.10.1. Conceito de DevOps
 - 10.10.2. Práticas principais

“*Uma experiência única, fundamental e decisiva para impulsionar o seu desenvolvimento profissional*”

05

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

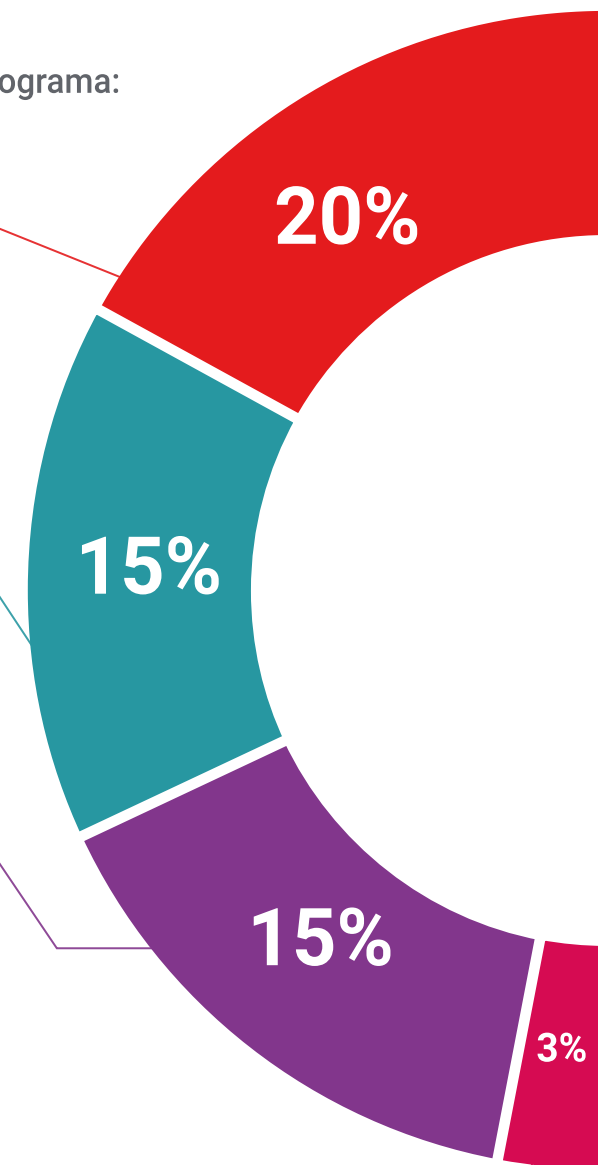
Apresentamos os conteúdos de forma atraente e dinâmica em pílulas multimídia que incluem áudio, vídeos, imagens, diagramas e mapas conceituais com o objetivo de reforçar o conhecimento.

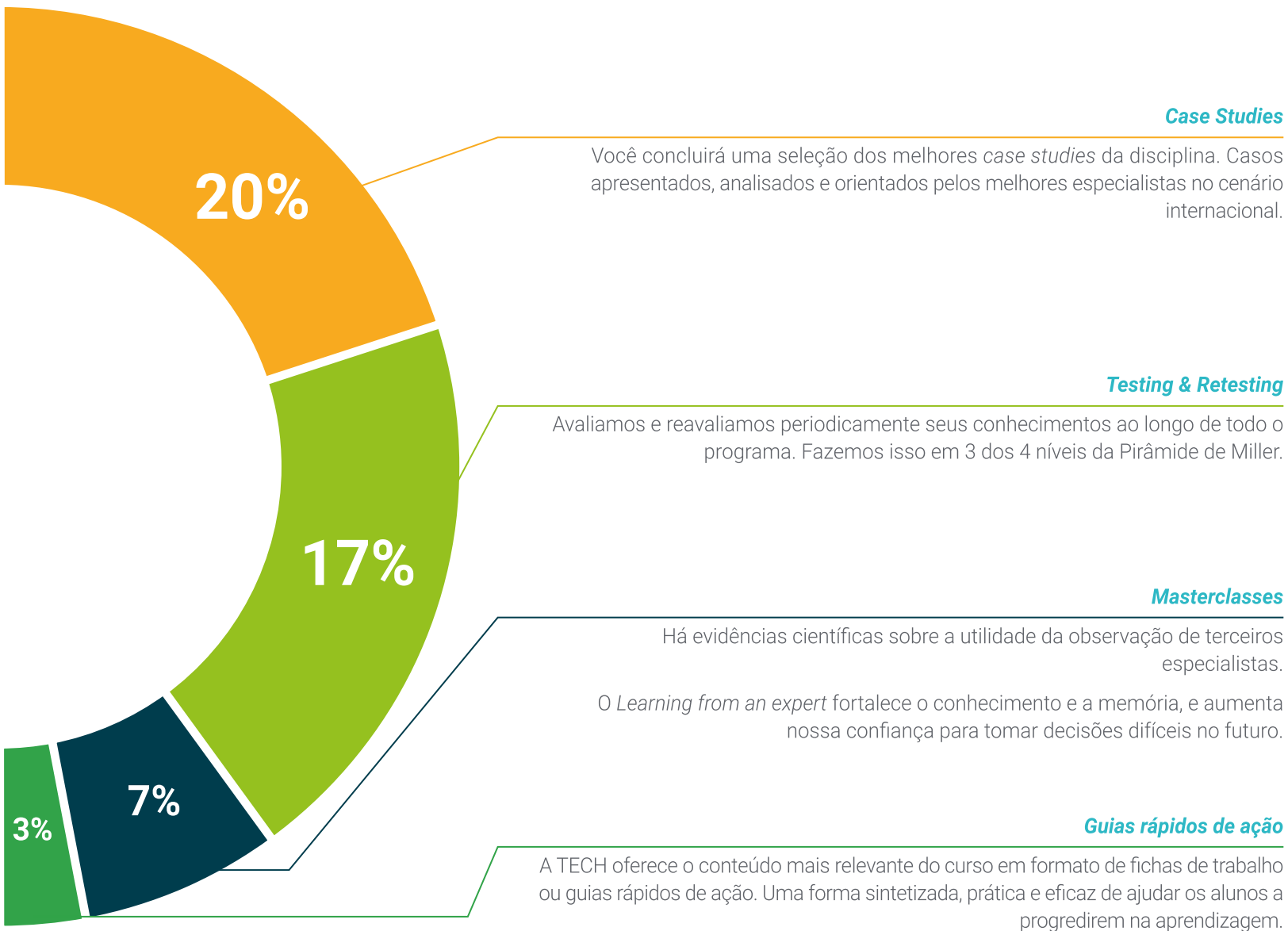
Este sistema exclusivo de capacitação por meio da apresentação de conteúdo multimídia foi premiado pela Microsoft como "Caso de sucesso na Europa"



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



06

Certificação

O Mestrado Próprio em Criação de Interfaces e Aplicações em Rede garante, além da formação mais rigorosa e atualizada, o acesso a um certificado de Curso 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 Próprio em Criação de Interfaces e Aplicações em Rede** reconhecido pela TECH Global University, a maior universidade digital do mundo.

A **TECH Global University**, é uma Universidade Europeia Oficial reconhecida publicamente pelo Governo de Andorra ([bollettino ufficiale](#)). Andorra faz parte do Espaço Europeu de Educação Superior (EEES) desde 2003. O EEES é uma iniciativa promovida pela União Europeia com o objetivo de organizar o modelo de formação internacional e harmonizar os sistemas de ensino superior dos países membros desse espaço. O projeto promove valores comuns, a implementação de ferramentas conjuntas e o fortalecimento dos seus mecanismos de garantia de qualidade para fomentar a colaboração e a mobilidade entre alunos, investigadores e académicos.

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 Próprio em Criação de Interfaces e Aplicações em Rede

Modalidade: online

Duração: 12 meses

Acreditação: 60 ECTS



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



Mestrado Próprio
Criação de Interfaces
e Aplicações em Rede

- » Modalidade: online
- » Duração: 12 meses
- » Certificação: TECH Global University
- » Acreditação: 60 ECTS
- » Horário: ao seu próprio ritmo
- » Exames: online

Mestrado Próprio

Criação de Interfaces
e Aplicações em Rede

