Mestrado Próprio

Desenvolvimento de Software

tech universidade tecnológica

tech universidade tecnológica

Mestrado Próprio
Desenvolvimento de 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-desenvolvimento-software**

Índice

01

Apresentação

Para participar numa das áreas com maior projeção no setor das TI, o profissional deve ser capacitado científica e tecnologicamente, bem como estar preparado para enfrentar eficazmente os desafios que surgem no exercício profissional da engenharia de software. Este Programa está orientado para alcançar um elevado nível de domínio do Desenvolvimento de Software, através dos últimos avanços e desenvolvimentos neste campo, através de uma metodologia de estudo de máximo impacto e extraordinária flexibilidade.

Adquira os conhecimentos mais abrangentes em engenharia de software na capacitação mais atualizada do mercado educacional online e começar a trabalhar em desenvolvimentos neste dinâmico campo profissional"

Com o avanço das novas tecnologias, o Software tornou-se um elemento extremamente importante na atualidade. Nos últimos anos, tornou-se evidente a necessidade de ser capaz de desenvolver produtos de Software com a funcionalidade e qualidade adequadas, dentro do prazo e do orçamento estabelecidos.

Este programa destina-se aos interessados em atingir um nível de conhecimento mais elevado sobre o desenvolvimento de Software. O principal objetivo é a especialização dos estudantes para que possam aplicar os conhecimentos adquiridos neste Mestrado Próprio num ambiente de trabalho que reproduza as condições que possam encontrar no seu futuro, de uma forma rigorosa e realista.

Aproveite a oportunidade de fazer esta capacitação num formato 100% online, sem ter de desistir das suas obrigações, e facilitando o regresso à universidade. Atualize os seus conhecimentos e obtenha um Mestrado Próprio para continuar a crescer pessoal e profissionalmente.

Adquirirá vastos conhecimentos no campo da engenharia de software, mas também no campo da computação e da estrutura informática, incluindo a base matemática, estatística e física essencial na engenharia.

Aproveite a oportunidade e faça esta capacitação num formato 100% online, sem ter de desistir das suas obrigações, e facilitando o regresso à universidade. Atualize os seus conhecimentos e obtenha um Mestrado Próprio para continuar a crescer pessoal e profissionalmente.

Este **Mestrado Próprio em Desenvolvimento de Software** conta com o conteúdo educacional mais completo e atualizado do mercado. As características que mais se destacam são:

- desenvolvimento de 100 cenários simulados apresentados por especialistas em Desenvolvimento de Software
- os seus conteúdos gráficos, esquemáticos e eminentemente práticos com os quais são concebidos, fornecem informações científicas e práticas sobre o Desenvolvimento de Software
- novidades sobre os últimos desenvolvimentos em Desenvolvimento de Software
- exercícios práticos onde o processo de autoavaliação pode ser levado a cabo a fim de melhorar a aprendizagem
- sistema interativo de aprendizagem baseado no método do caso e a sua aplicação à prática real
- lições teóricas, questionamentos ao especialista, fóruns de discussão sobre questões controversas e documentos individuais de reflexão
- disponibilidade de acesso aos conteúdos a partir de qualquer dispositivo fixo ou portátil com ligação à internet

Este programa permitir-lhe-á aprender sobre a estrutura básica de um computador e o seu software, como base para aumentar as suas competências"

Aprenda tudo o que precisa para trabalhar com linguagens de programação em segurança, incorporando ao seus conhecimentos a interpretação e conceção de algoritmos básicos para trabalhar em programação"

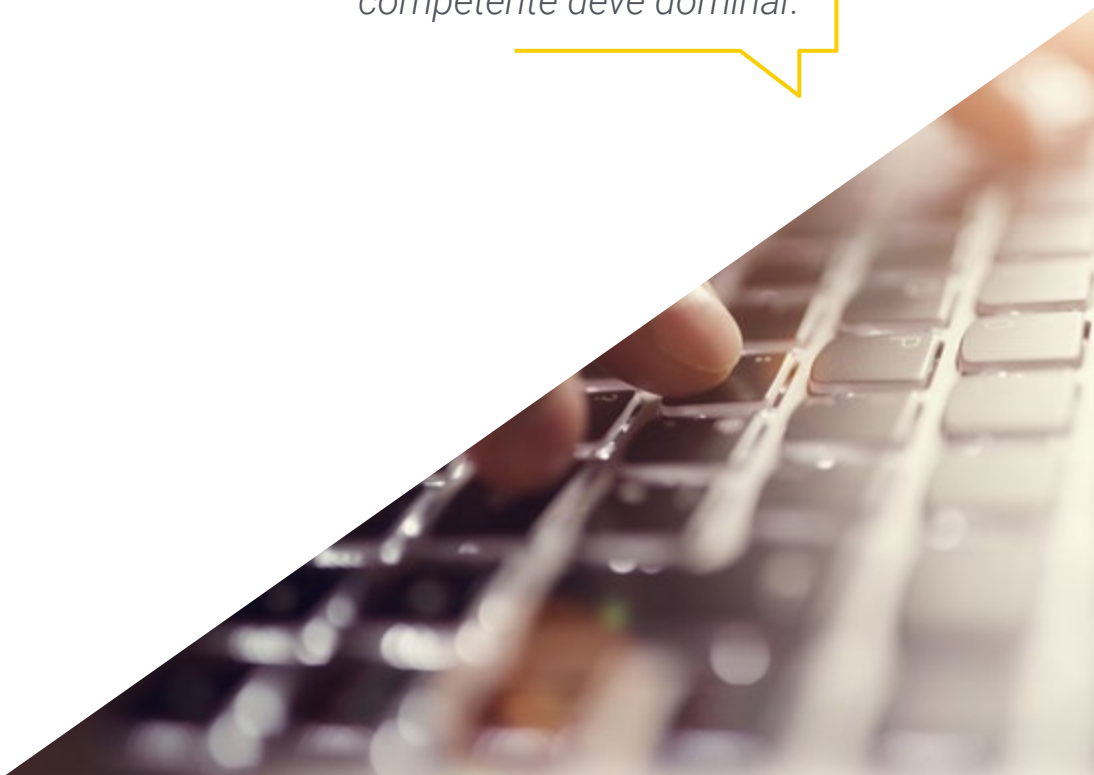
Uma capacitação que lhe permitirá compreender o funcionamento e como intervir sobre todos os elementos essenciais de um programa informático.

Conheça os mais recentes sistemas de dados existentes no mercado, aprenda a conceber algoritmos avançados e todos os aspetos que um profissional altamente competente deve dominar.

O seu corpo docente inclui profissionais do âmbito do Desenvolvimento de Software, que trazem a experiência do seu trabalho para esta capacitação, bem como especialistas reconhecidos das principais 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 uma aprendizagem imersiva programada para se formar em situações reais.

A conceção deste programa centra-se na Aprendizagem Baseada em Problemas, através da qual o Docentes deve tentar resolver as diferentes situações de prática profissional que surgem ao longo do programa. Para o fazer, o profissional terá a ajuda de um sistema inovador de vídeo interativo criado por especialistas reconhecidos em Desenvolvimento de Software, com uma vasta experiência de ensino.



02

Objetivos

O objetivo desta capacitação é fornecer aos profissionais que trabalham em Desenvolvimento de Software, os conhecimento e as habilidades necessárias para realizar a atividade utilizando as técnicas e protocolos mais avançados do momento. Através de uma abordagem de trabalho totalmente adaptável ao aluno, este Mestrado Próprio irá permitir que você adquira progressivamente as competências que o impulsionarão para um nível profissional superior.

Irá mergulhar no campo da computação e da estrutura informática, temas essenciais para qualquer programador de software"

Objetivos gerais

- Capacitar os profissionais científica e tecnologicamente, assim como prepará-los para a prática profissional da Engenharia do Software, tudo isto com uma capacitação transversal e versátil adaptada às novas tecnologias e inovações neste campo
- Obter vastos conhecimentos no campo da engenharia de Software, mas também no campo da computação e da estrutura de computadores, incluindo a base matemática, estatística e física essencial na engenharia

Atinja o nível de conhecimento desejado e domine o Desenvolvimento de Software com esta capacitação de alto nível"

Objetivos específicos

Módulo 1. Fundamentos de programação

- Compreender a estrutura básica de um computador, o Software e das linguagens de Programação de uso geral
- Aprender a conceber e interpretar algoritmos, que são a base necessária para o desenvolvimento de programas informáticos
- Compreender os elementos essenciais de um programa informático, tal como os diferentes tipos de dados, operadores, expressões, sentenças, I/O e sentenças de controlo
- Compreender as diferentes estruturas de dados disponíveis em linguagens de programação de uso geral, tanto estáticas como dinâmicas, assim como adquirir conhecimentos essenciais para a gestão de ficheiros
- Compreender as diferentes técnicas de teste nos programas informáticos e a importância de gerar uma boa documentação juntamente com um bom código fonte
- Aprenda os conceitos básicos da linguagem de Programação C++, uma das linguagens de programação mais utilizadas em todo o mundo

Módulo 2. Estrutura de dados

- Aprender os fundamentos de programação na linguagem C++, incluindo aulas, variáveis, expressões condicionais e objetos
- Compreender os tipos de dados abstractos, tipos de estruturas de dados lineares, estruturas de dados hierárquicos simples e complexas e a sua implementação em C++
- Compreender o funcionamento de estruturas de dados avançadas para além das habituais
- Compreender a teoria e a prática relacionadas com a utilização de montículos e filas de espera prioritárias
- Aprender o funcionamento das tabelas *Hash*, como tipos abstratos de dados e funções
- Compreender a teoria dos Grafos, bem como os algoritmos e conceitos avançados sobre Grafos

Módulo 3. Algoritmia e complexidade

- Aprender as principais estratégias para a conceção de algoritmos, bem como os diferentes métodos e medidas para o cálculo de algoritmos
- Conhecer os principais algoritmos de ordenação utilizados no desenvolvimento de Software
- Compreender o funcionamento dos diferentes algoritmos com árvores, *Heaps* e Grafos
- Compreender o funcionamento dos algoritmos *Greedy*, a sua estratégia e exemplos da sua utilização nos principais problemas conhecidos. Conheceremos também o uso de algoritmos greedy sobre grafos
- Aprenderemos as principais estratégias de descoberta de caminhos mínimos, com a abordagem de problemas essenciais do âmbito e algoritmos para a sua resolução
- Entender a técnica de *Backtracking* e as suas principais utilizações, bem como outras técnicas alternativas

Módulo 4. Bases de dados

- Aprender as diferentes aplicações e objetivos 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 até às mais avançadas e complexas
- Aprender a conceber bases de dados utilizando o modelo de entidade relação, como criar diagramas e as caraterísticas do modelo E-R alargado
- Aprofundar a conceção de bases de dados relacionais, analisando as diferentes formas normais e os algoritmos de decomposição
- Lançar as bases para compreender o funcionamento das bases de dados NoSQL, bem como para introduzir a base de dados MongoDB

Módulo 5. Bases de dados avançadas

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

Módulo 6. Desenho avançado de algoritmos

- Aprofundar na conceção avançada de algoritmos, analisando algoritmos recursivos e de divisão e conquista, bem como realizar análise amortizada
- Compreender os conceitos de Programação dinâmica e os algoritmos para problemas NP
- Entender o funcionamento da optimização combinatória, bem como os diferentes algoritmos de aleatorização e algoritmos paralelos
- Conhecer e compreender o funcionamento dos diferentes métodos de pesquisa locais e com candidatos
- Aprender os mecanismos de verificação formal de programas e de programas iterativos, incluindo a lógica de primeira ordem e o sistema formal de Hoare
- Aprender o funcionamento de alguns dos principais métodos numéricos como o método da bisseção, o método de Newton Raphson e o método das secantes

Módulo 7. Interação Pessoa Computador

- Adquirir sólidos conhecimentos relacionados com a interação pessoa computador e a criação de interfaces utilizáveis
- Entender a importância da usabilidade das aplicações e porque é que é importante tê-las em conta na conceção do nosso Software
- Compreender os diferentes tipos de diversidade humana, os constrangimentos que implicam e como adaptar as interfaces de acordo com as necessidades específicas de cada uma delas
- Aprender o processo de conceção de interfaces, desde a análise de requisitos até à avaliação, através das várias fases intermédias necessárias para realizar uma interface adequada
- Conhecer as diferentes diretrizes de acessibilidade, as normas que as estabelecem e as ferramentas que nos permitem avaliá-las
- Compreender os diferentes métodos de interação com o computador, utilizando periféricos e dispositivos

Módulo 8. Programação avançada

- Aprofundar nos conhecimentos 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 a eventos e o desenvolvimento de interfaces de utilizadores com Qt
- Adquirir os conhecimentos essenciais da Programação concorrente, os processos e os tópicos
- Aprender a gerir a utilização dos tópicos e a sincronização, bem como a resolução dos problemas comuns no âmbito da Programação concorrente
- Entender a importância da documentação e das provas no desenvolvimento de Software

Módulo 9. Desenvolvimento de aplicações em rede

- Conhecer as caraterísticas da linguagem de marcação HTML e a sua utilização na criação de web juntamente com folhas de estilo CSS
- Aprenda a utilizar a linguagem de Programação orientada para o browser JavaScript, e algumas das suas principais caraterísticas
- Compreender os conceitos da Programação orientada para componentes e a arquitetura de componentes
- Aprender a usar o *Framework* para *Frontend* Bootstrap para a conceção de sitios web
- Entender a estrutura do modelo de vista controlador no desenvolvimento de websites dinâmicos
- Conhecer a arquitetura orientada para o serviço e as noções básicas do protocolo HTTP

Módulo 10. Engenharia do Software

- Lançar as bases da engenharia de Software e a modelação, aprendendo os principais processos e conceitos
- Entender o processo de Software e os diferentes modelos para o seu desenvolvimento, incluindo tecnologias ágeis
- Compreender a engenharia de requisitos, o seu desenvolvimento, elaboração, negociação e validação
- Aprender a modelação dos requisitos e os diferentes elementos tais como cenários, informação, aulas de análise, fluxo, comportamento e padrões
- Entender os conceitos e processos da conceção de Software, aprendendo também sobre a conceção da arquitetura e sobre a conceção ao nível de componentes e baseado em padrões
- Conhecer as principais normas relacionadas com a qualidade do Software e a gestão de projetos

03

Competências

Ao passar nas avaliações do Mestrado Próprio em Desenvolvimento de Software, terá adquirido as competências profissionais necessárias para fazer um trabalho de qualidade e adquirirá também novas competências e técnicas que o ajudarão a complementar os conhecimentos informáticos existentes.

Aumente as suas competências em Desenvolvimento de Software e passe para o nível seguinte como profissional neste campo em constante evolução”

Competências gerais

» Responder às necessidades atuais na área do Desenvolvimento de Software

Um programa excepcional em termos da sua densidade, a sua completa atualidade e a forma como é ensinado, o que lhe permitirá progredir rápida e eficazmente"

Competências específicas

- Ser capaz de compreender a estrutura básica de um computador, o software e das linguagens de Programação de uso geral
- Saber aplicar os fundamentos de programação na linguagem C++, incluindo aulas, variáveis, expressões condicionais e objetos
- Conhecer em profundidade as principais estratégias para a conceção de algoritmos, bem como os diferentes métodos e medidas para o cálculo dos mesmos
- Conhecer as diferentes aplicações e objetivos dos sistemas de bases de dados, bem como o seu funcionamento e arquitetura, aplicados no dia a dia
- Ser capaz de Introduzir os diferentes sistemas de bases de dados atualmente disponíveis no mercado
- Saber analisar algoritmos recursivos e de divisão e conquista, bem como realizar análise amortizada
- Utilizar os conhecimentos da interação pessoa computador e a criação de interfaces utilizáveis no exercício diário da profissão
- Gerir em profundidade o conhecimento de Programação
- Conhecer as caraterísticas da linguagem de marcação HTML e a sua utilização na criação web juntamente com as folhas de estilo CSS
- Ser capaz de aplicar os principais processos e conceitos dos fundamentos da engenharia de Software e a modelação

04

Estrutura e conteúdo

A estrutura dos conteúdos foi concebida por uma equipa de profissionais de Engenharia Informática com o objetivo de assegurar que os alunos do Mestrado Próprio possam aprender de forma eficiente e rápida. Para tal, os conteúdos foram organizados de tal forma que a aprendizagem é intensiva e constante, tentando manter a motivação baseada no sentido de progresso do aluno.

Um programa educativo destinado a atingir competências completas de desenvolvimento de software, que o impulsionará para um novo nível profissional"

Módulo 1. Fundamentos de programação

1.1. Introdução à programação
- 1.1.1. Estrutura básica de um ordenador
- 1.1.2. Software
- 1.1.3. Linguagens de programação
- 1.1.4. Ciclo de vida uma aplicação informática

1.2. Desenho de algoritmos
- 1.2.1. Resolução de problemas
- 1.2.2. Técnicas descritivas
- 1.2.3. Elementos e estrutura de um algoritmo

1.3. Elementos de um programa
- 1.3.1. Origem e caraterísticas da linguagem C++
- 1.3.2. O ambiente de desenvolvimento
- 1.3.3. Conceito de programa
- 1.3.4. Tipos de Dados Fundamentais
- 1.3.5. Operadores
- 1.3.6. Expressões
- 1.3.7. Sentenças
- 1.3.8. Entrada e saída de dados

1.4. Sentenças de controlo
- 1.4.1. Sentenças
- 1.4.2. Bifurcações
- 1.4.3. Laços

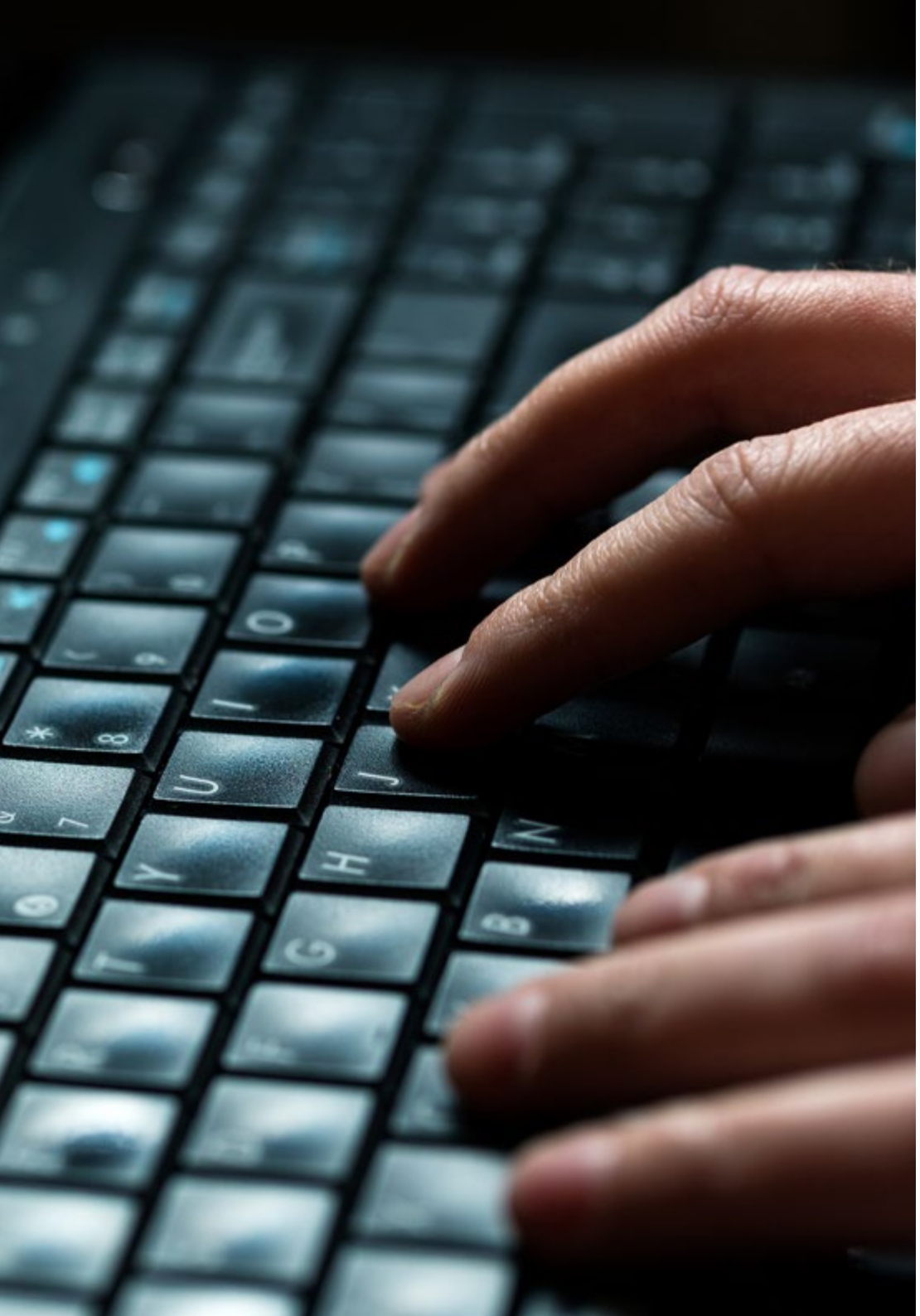
1.5. Abstração e modularidade: funções
- 1.5.1. Desenho modular
- 1.5.2. Conceito de função e utilidade
- 1.5.3. Definição de uma função
- 1.5.4. Fluxo de execução na chamada de uma função
- 1.5.5. Protótipo de uma função
- 1.5.6. Devolução de resultados
- 1.5.7. Chamada a uma função: parâmetros
- 1.5.8. Passagem de parâmetros por referência e por valor
- 1.5.9. Âmbito identificador

1.6. Estruturas de dados estáticas
- 1.6.1. *Arrays*
- 1.6.2. Matrizes. Poliedros
- 1.6.3. Pesquisa e ordenação
- 1.6.4. Cadeias. Funções de E/S para cadeias
- 1.6.5. Estruturas Uniões
- 1.6.6. Novos tipos de dados

1.7. Estruturas de dados dinâmicas: ponteiros
- 1.7.1. Conceito Definição de ponteiro
- 1.7.2. Operadores e operações com ponteiros
- 1.7.3. *Arrays* de ponteiros
- 1.7.4. Ponteiros e *Arrays*
- 1.7.5. Ponteiros a cadeias
- 1.7.6. Ponteiros a estruturas
- 1.7.7. Indireção múltipla
- 1.7.8. Ponteiros a funções
- 1.7.9. Passagem de funções, estruturas e *arrays* como parâmetros de funções

1.8. Ficheiros
- 1.8.1. Conceitos básicos
- 1.8.2. Operações com ficheiros
- 1.8.3. Tipos de ficheiros
- 1.8.4. Organização dos ficheiros
- 1.8.5. Introdução aos ficheiros CC++
- 1.8.6. Gestão de ficheiros

1.9. Recursividade
- 1.9.1. Definição de recursividade
- 1.9.2. Tipos de recursividade
- 1.9.3. Vantagens e desvantagens
- 1.9.4. Considerações
- 1.9.5. Conversão recursivo iterativa
- 1.9.6. A pilha de recorrência



1.10. Prova e documentação
- 1.10.1. Provas de programas
- 1.10.2. Prova da caixa branca
- 1.10.3. Prova da caixa negra
- 1.10.4. Ferramentas para realizar as provas
- 1.10.5. Documentação de programas

Módulo 2. Estrutura de dados

2. 1. Introdução à programação em C++
- 2.1.1. Classes, construtores, métodos e atributos
- 2.1.2. Variáveis
- 2.1.3. Expressões condicionais e loops
- 2.1.4. Objetos

2.2. Tipos abstratos de Dados (TAD)
- 2.2.1. Tipos de dados
- 2.2.2. Estruturas básicas e TAD
- 2.2.3. Vetores e *Arrays*

2.3. Estruturas de Dados Lineares
- 2.3.1. TAD Lista definição
- 2.3.2. Listas ligadas e duplamente ligadas
- 2.3.3. Listas ordenadas
- 2.3.4. Listas em C++
- 2.3.5. TAD Pilha
- 2.3.6. TAD Queue
- 2.3.7. Pilha e Fila em C++

2.4. Estruturas de dados hierárquicas
- 2.4.1. TAD Árvore
- 2.4.2. Caminhos
- 2.4.3. Árvores n-arios
- 2.4.4. Árvores binários
- 2.4.5. Árvores binários de pesquisa

2.5. Estruturas de dados hierárquicas: árvores complexas
  2.5.1. Árvores perfeitamente equilibradas ou de altura mínima
  2.5.2. Árvores multicaminho
  2.5.3. Referências bibliográficas
2.6. Heaps Prioritários e Fila Prioritária
  2.6.1. TAD Heaps
  2.6.2. TAD Fila proprietária
2.7. Tabelas *Hash*
  2.7.1. TAD Tabela *Hash*
  2.7.2. Funções *Hash*
  2.7.3. Função *Hash* em tabelas *Hash*
  2.7.4. Redispersão
  2.7.5. Tabelas *Hash* abertas
2.8. Grafos
  2.8.1. TAD Grafo
  2.8.2. Tipos de Grafo
  2.8.3. Representação gráfica e operações básicas
  2.8.4. Desenho de Grafo
2.9. Algoritmos e conceitos avançados sobre Grafos
  2.9.1. Problemas sobre Grafos
  2.9.2. Algoritmos sobre caminhos
  2.9.3. Algoritmos de pesquisa ou caminhos
  2.9.4. Outros algoritmos
2.10. Outras estruturas de dados
  2.10.1. Conjuntos
  2.10.2. *Arrays* paralelos
  2.10.3. Tabela de símbolos
  2.10.4. *Tries*

Módulo 3. Algoritmia e complexidade

3.1. Introdução às estratégias de desenho do algoritmos
  3.1.1. Recursividade
  3.1.2. Divide e conquista
  3.1.3. Outras estratégias
3.2. Eficiência e análise dos algoritmos
  3.2.1. Medidas de eficiência
  3.2.2. Medir o tamanho da entrada
  3.2.3. Medir o tempo de execução
  3.2.4. Caso pior, melhor e médio
  3.2.5. Notação assintótica
  3.2.6. Critérios de análise matemática de algoritmos não recursivos
  3.2.7. Análise matemática de algoritmos recursivos
  3.2.8. Análise empírica de algoritmos
3.3. Algoritmos de ordenação
  3.3.1. Conceito de ordenação
  3.3.2. Ordenação da bolha
  3.3.3. Ordenação por seleção
  3.3.4. Ordenação por inserção
  3.3.5. Ordenação por mistura (Merge Sort)
  3.3.6. Ordenação rápida (Quicksort)
3.4. Algoritmos com árvores
  3.4.1. Conceito de árvore
  3.4.2. Árvores binários
  3.4.3. Caminhos de árvore
  3.4.4. Representar expressões
  3.4.5. Árvores binários ordenadas
  3.4.6. Árvores binárias equilibradas
3.5. Algoritmos com *Heaps*
  3.5.1. Os *Heaps*
  3.5.2. O algoritmo Heapsort
  3.5.3. As filas de prioridade

3.6. Algoritmos com grafos
  3.6.1. Representação
  3.6.2. Caminho de largura
  3.6.3. Caminho de profundidade
  3.6.4. Ordenação topológica
3.7. Algoritmos *Greedy*
  3.7.1. A estratégia *Greedy*
  3.7.2. Elementos da estratégia *Greedy*
  3.7.3. Câmbio de moedas
  3.7.4. Problema do viajante
  3.7.5. Problema da mochila
3.8. Pesquisa de caminhos mínimos
  3.8.1. O problema do caminho mínimo
  3.8.2. Arcos negativos e ciclos
  3.8.3. Algoritmo de Dijkstra
3.9. Algoritmos *Greedy* sobre Grafos
  3.9.1. A árvore de extensão mínima
  3.9.2. O algoritmo de Prim
  3.9.3. O algoritmo Kruskal
  3.9.4. Análise de complexidade
3.10. *Backtracking*
  3.10.1. O *Backtracking*
  3.10.2. Técnicas alternativas

Módulo 4. Bases de dados

4.1. Aplicações e propósitos dos sistemas de base de dados
  4.1.1. Aplicações dos diferentes sistemas de base de Dados
  4.1.2. Proteção nos diferentes sistemas de base de Dados
  4.1.3. Visão dos dados
4.2. Base de dados e arquitetura
  4.2.1. Bases de dados relacionais
  4.2.2. O desenho de bases de dados
  4.2.3. Bases de dados baseadas em objetos e semiestruturadas
  4.2.4. Armazenamento dos dados e consultas
  4.2.5. Gestão de Transações
  4.2.6. Mineração e análises de dados
  4.2.7. Arquitetura das bases de dados
4.3. O modelo relacional: estrutura, operações e álgebra relacional alargada
  4.3.1. A estrutura das bases de dados relacionais
  4.3.2. Operações fundamentais em álgebra relacional
  4.3.3. Outras operações de álgebra relacional
  4.3.4. Operações de álgebra relacional alargada
  4.3.5. Valores nulos
  4.3.6. Modificação da base de dados
4.4. SQL (I)
  4.4.1. O que é SQL?
  4.4.2. A definição de dados
  4.4.3. Estrutura básica das consultas SQL
  4.4.4. Operações sobre conjuntos
  4.4.5. Funções de agregação
  4.4.6. Valores nulos

4.5. SQL (II)
- 4.5.1. Subconsultas aninhadas
- 4.5.2. Consultas complexas
- 4.5.3. Vistas
- 4.5.4. Cursores
- 4.5.5. Consultas complexas
- 4.5.6. Disparadores

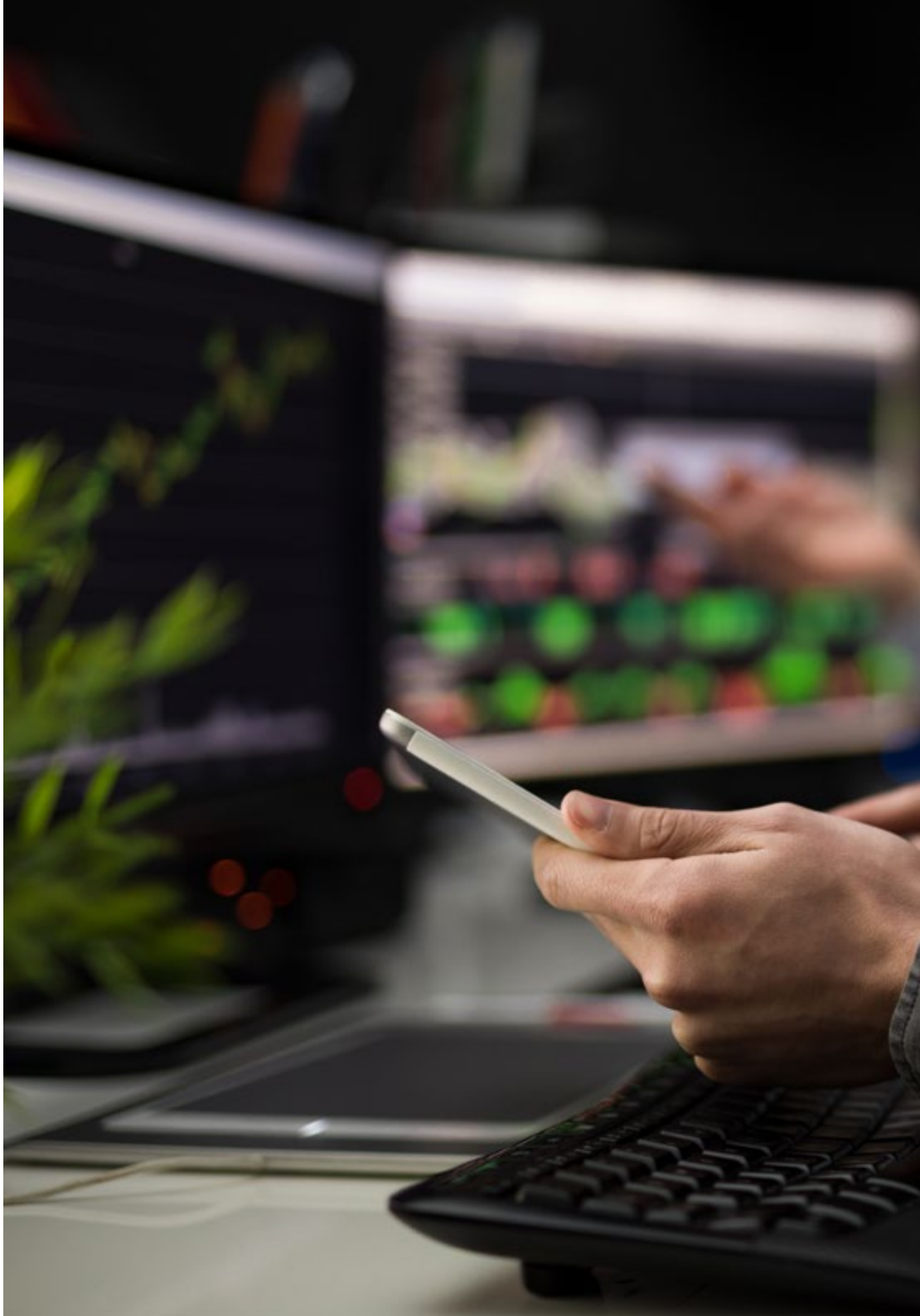
4.6. Desenho de base de dados e do modelo E-R
- 4.6.1. Visão geral do processo de desenho
- 4.6.2. Modelo entidade relação
- 4.6.3. Restrições

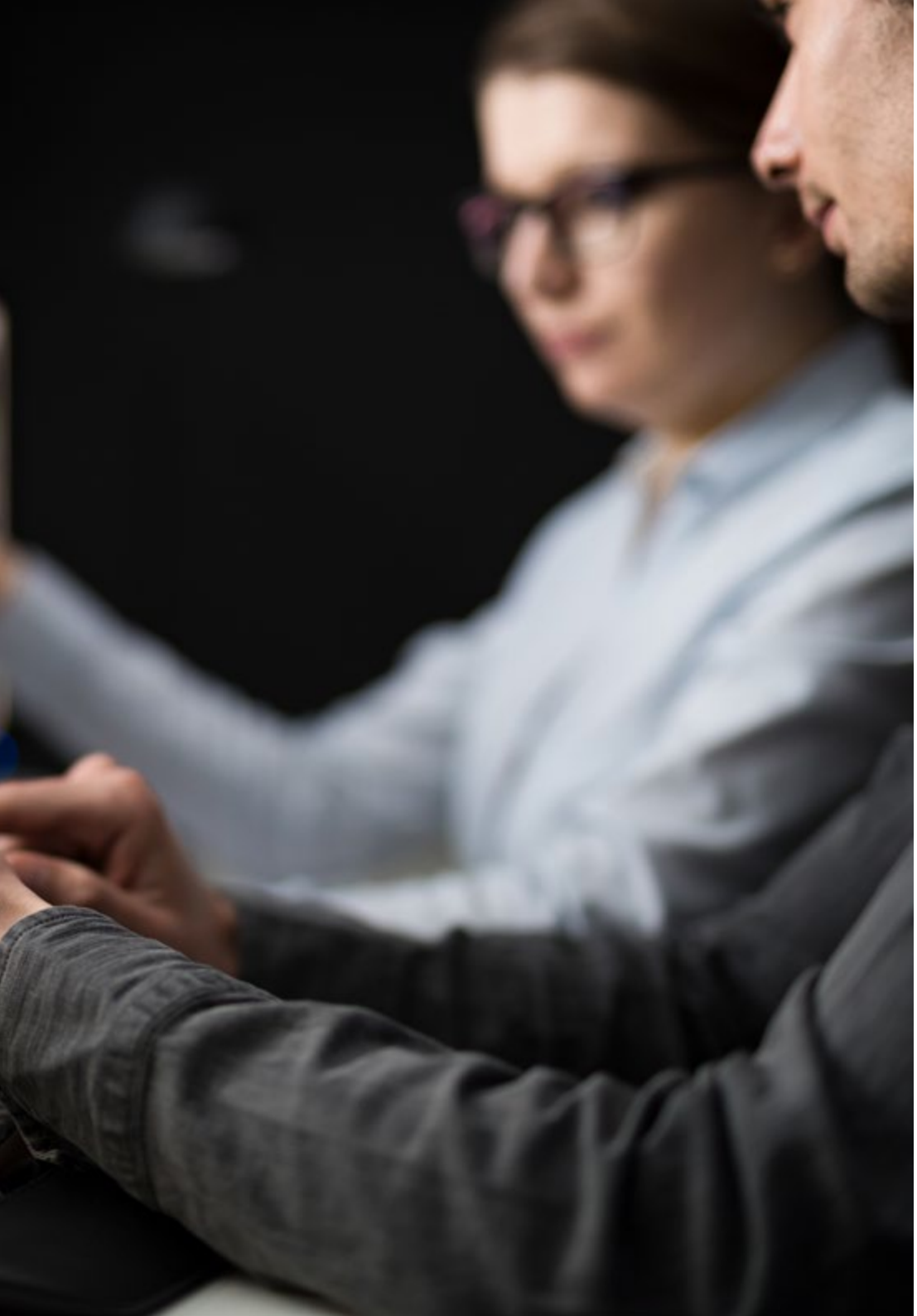
4.7. Diagramas entidade relação
- 4.7.1. Diagramas entidade relação
- 4.7.2. Aspetos do desenho de entidade relação
- 4.7.3. Conjuntos de entidades débeis

4.8. O modelo entidade relação alargado
- 4.8.1. Caraterísticas do modelo E-R alargado
- 4.8.2. Desenho de uma base de dados
- 4.8.3. Redução a esquemas relacionais

4.9. Desenho de bases de dados relacionais
- 4.9.1. Caraterísticas dos bons desenhos relacionais
- 4.9.2. Domínios atómicos e a primeira forma normal (1FN)
- 4.9.3. Decomposição através de dependências funcionais
- 4.9.4. Teoria das dependências funcionais
- 4.9.5. Algoritmos de decomposição
- 4.9.6. Decomposição através de dependências multivalorizadas
- 4.9.7. Mais formas normais
- 4.9.8. Processo de desenho das bases de dados

4.10. Bases de dados NoSQL
- 4.10.1. O que são as bases de dados NoSQL?
- 4.10.2. Análise das diferentes opções de NoSQL e as suas caraterísticas
- 4.10.3. MongoDB





Módulo 5. Bases de dados avançadas

5.1. Introdução aos diferentes sistemas de base de dados
- 5.1.1. Revisão histórica
- 5.1.2. Bases de dados hierárquicas
- 5.1.3. Bases de dados 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 em consultas
- 5.3.4. Paralelismo em consultas
- 5.3.5. Desenho 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 utilizar 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 de 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 em registo 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 rendimento

5.9. Bases de dados não relacionais
- 5.9.1. Bases de dados orientadas para a documentação
- 5.9.2. Bases de dados orientadas para grafos
- 5.9.3. Bases de dados chave-valor

5.10. Data *Warehouse*, OLAP e mineração 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. Funcionalidade da mineração de dados
- 5.10.5. Outros tipos de mineração

Módulo 6. Desenho avançado de algoritmos

6.1. Análise de algoritmos recursivos e de divisão e conquista
- 6.1.1. Posicionar e resolver equações de recorrência homogéneas e não homogéneas
- 6.1.2. Descrição geral da estratégia divisão e conquista

6.2. Análise amortizado
- 6.2.1. A análise agregada
- 6.2.2. O método de contabilidade
- 6.2.3. O método do potencial

6.3. Programação dinâmica e algoritmos para problemas NP
- 6.3.1. Caraterísticas da programação dinâmica
- 6.3.2. Volta atrás: *Backtracking*
- 6.3.3. Ramificação e poda

6.4. Optimização combinatória
- 6.4.1. Representação de problemas
- 6.4.2. Optimizaçáo em 1D

6.5. Algoritmos de aleatorização
- 6.5.1. Exemplos de algoritmos de aleatorização
- 6.5.2. O teorema Buffon
- 6.5.3. Algoritmo de Monte Carlo
- 6.5.4. Algoritmo Las Vegas

6.6. Pesquisa local e com candidatos
- 6.6.1. *Gradient Ascent*
- 6.6.2. *Hill Climbing*
- 6.6.3. *Simulated Annealing*
- 6.6.4. *Tabu Search*
- 6.6.5. Pesquisa com candidatos

6.7. Verificação formal de programas
- 6.7.1. Especificação de abstrações funcionais
- 6.7.2. A linguagem da lógica de primeira ordem
- 6.7.3. O sistema formal de Hoare

6.8. Verificação de programas iterativos
- 6.8.1. Regras do sistema formal de Hoare
- 6.8.2. Conceito de invarante de iterações

6.9. Métodos numéricos
- 6.9.1. O método da bisecção
- 6.9.2. O método de Newton Raphson
- 6.9.3. O método das secantes

6.10. Algoritmos paralelos
- 6.10.1. Operações binárias paralelas
- 6.10.2. Operações paralelas com grafos
- 6.10.3. Paralelismo em divisão e conquista
- 6.10.4. Paralelismo em programação e dinâmica

Módulo 7. Interação Pessoa Computador

7.1. Introdução à Interação pessoa computador
- 7.1.1. Introdução à Interação pessoa computador
- 7.1.2. Relação da interação pessoa computador com outras disciplinas
- 7.1.3. A interface de utilizador
- 7.1.4. Usabilidade e acessibilidade
- 7.1.5. Experiência de usuário e desenho centrado no utilizador

7.2. O computador e a interação: interface do utilizador e paradigmas de interação
- 7.2.1. A interação
- 7.2.2. Paradigmas e estilos de interação
- 7.2.3. Evolução das Interfaces de utilizador
- 7.2.4. Interfaces de utilizador clássicas: WIMP/GUI, comandos, voz, realidade virtual
- 7.2.5. Interfaces de utilizador inovadoras: móveis, portáteis, colaborativas, BCI

7.3. O fator humano: aspetos psicológicos e cognitivos
- 7.3.1. A importância do fator humano na interação
- 7.3.2. Processamento humano de informação
- 7.3.3. A entrada e saída da informação: visual, auditiva e táctil
- 7.3.4. Perceção e atenção
- 7.3.5. Conhecimento e modelos mentais: representação, organização e aquisição

7.4. O fator humano: limitações sensoriais e físicas
- 7.4.1. Diversidade funcional, incapacidade e deficiência
- 7.4.2. Diversidade visual
- 7.4.3. Diversidade auditiva
- 7.4.4. Diversidade cognitiva
- 7.4.5. Diversidade motora
- 7.4.6. O caso dos imigrantes digitais

7. 5. O processo de desenho (I): análise de requisitos para o desenho da interface do utilizador
- 7.5.1. Desenho centrado no utilizador
- 7.5.2. O que é a análise de requisitos
- 7.5.3. A recolha de informação
- 7.5.4. Análise e interpretação da informação
- 7.5.5. Análise da usabilidade e acessibilidade

7.6. O processo de desenho (II): prototipagem e análise de tarefas
- 7.6.1. Desenho concetual
- 7.6.2. Prototipagem
- 7.6.3. Análise hierárquica de tarefas

7.7. O processo de desenho (III): avaliação
- 7.7.1. Avaliação no processo de desenho: objetivos e métodos
- 7.7.2. Métodos de avaliação sem utilizadores
- 7.7.3. Métodos de avaliação com utilizadores
- 7.7.4. Padrões e normas de avaliação

7.8. Acessibilidade: definição e diretrizes
- 7.8.1. Acessibilidade e desenho universal
- 7.8.2. A iniciativa WAI e as diretrizes WCAG
- 7.8.3. Diretrizes WCAG 2.0 e 2.1

7.9. Acessibilidade: avaliação e diversidade funcional
- 7.9.1. Ferramentas de avaliação da acessibilidade na web
- 7.9.2. Acessibilidade e diversidade funcional

7.10. O computador e a interação: periféricos e dispositivos
- 7.10.1. Dispositivos e periféricos tradicionais
- 7.10.2. Dispositivos e periféricos alternativos
- 7.10.3. Telemóveis e tablets
- 7.10.4. Diversidade funcional, interação e periféricos

Módulo 8. Programação avançada

8.1. Introdução à programação orientada a objetos
- 8.1.1. Introdução à programação orientada a objetos
- 8.1.2. Desenho de classes
- 8.1.3. Introdução à UML para modelação de problemas

8.2. Relações entre classes
- 8.2.1. Abstração e herança
- 8.2.2. Conceitos avançados de herança
- 8.2.3. Poliformismo
- 8.2.4. Composição e agregação

8.3. Introdução aos padrões de de desenho para problemas orientados a objetos
- 8.3.1. O que é um padrão de desenho
- 8.3.2. Padrão *Factory*
- 8.3.3. Padrão *Singleton*
- 8.3.4. Padrão *Observer*
- 8.3.5. Padrão *Composite*

8.4. Exceções
- 8.4.1. O que são as exceções
- 8.4.2. Captura e gestão de exceções
- 8.4.3. Lançamento de exceções
- 8.4.4. Criação de exceções

8.5. Interfaces de utilizadores
- 8.5.1. Introdução a Qt
- 8.5.2. Posicionamento
- 8.5.3. O que são os eventos?
- 8.5.4. Eventos: definição e captura
- 8.5.5. Desenvolvimento de interfaces de utilizador

8.6. Introdução à programação concorrente
- 8.6.1. Introdução à programação concorrente
- 8.6.2. O conceito de processo e thread
- 8.6.3. Interação entre processos ou threads
- 8.6.4. Os threads em C++
- 8.6.5. Vantagens e desvantagens da programação concorrente

8.7. Gestão de threads e sincronização
- 8.7.1. Ciclo de vida um thread
- 8.7.2. A classe *Thread*
- 8.7.3. Planificação de threads
- 8.7.4. Grupos threads
- 8.7.5. Threads de tipo demónio
- 8.7.6. Sincronização
- 8.7.7. Mecanismos de bloqueio
- 8.7.8. Mecanismos de comunicação
- 8.7.9. Monitores

8.8. Problemas comuns dentro da programação concorrente
- 8.8.1. O problema dos produtores consumidores
- 8.8.2. O problema dos leitores e escritores
- 8.8.3. O problema do jantar dos filósofos

8.9. Documentação e provas de software
- 8.9.1. Porque é importante documentar o software?
- 8.9.2. Documentação de desenho
- 8.9.3. Uso de ferramentas para a documentação

8.10. Provas de software
- 8.10.1. Introdução às provas de software
- 8.10.2. Tipos de provas
- 8.10.3. Prova de unidade
- 8.10.4. Prova de integração
- 8.10.5. Prova de validação
- 8.10.6. Prova do sistema

Módulo 9. Desenvolvimento de aplicações em rede

9.1. Linguagens de marcação HTML5
- 9.1.1. Conceitos básicos de HTML
- 9.1.2. Novos elementos HTML 5
- 9.1.3. Formulários: novos controlos

9.2. Introdução às folhas de estilo CSS
- 9.2.1. Primeiros passos com CSS
- 9.2.2. Introdução ao CSS3

9.3. Linguagem script de navegador: JavaScript
- 9.3.1. Conceitos básicos de JavaScript
- 9.3.2. DOM
- 9.3.3. Eventos
- 9.3.4. JQuery
- 9.3.5. Ajax

9.4. Conceito de programação orientada a componentes
- 9.4.1. Contexto
- 9.4.2. Componentes e interfaces
- 9.4.3. Estados de um componente

9.5. Arquitetura de componentes
- 9.5.1. Arquiteturas atuais
- 9.5.2. Integração e implantação de componentes

9.6. *Framework Frontend*: Bootstrap
- 9.6.1. Desenho com grelha
- 9.6.2. Formulários
- 9.6.3. Componentes

9.7. Modelo vista controlador
- 9.7.1. Métodos de desenvolvimento Web
- 9.7.2. Padrão de desenho: MVC

9.8. Tecnologias Grid da informação
- 9.8.1. Aumento de recursos informáticos
- 9.8.2. Conceito de tecnologia Grid

9.9. Arquiteturas orientadas para serviços
- 9.9.1. SOA e serviços web
- 9.9.2. Topologia de um serviço web
- 9.9.3. Plataformas para serviços web

9.10. Protocolo HTTP
- 9.10.1. Mensagens
- 9.10.2. Sessões persistentes
- 9.10.3. Sistema criptográfico
- 9.10.4. Funcionamento do protocolo HTTPS

Módulo 10. Engenharia do Software

10.1. Introdução à Engenharia do Software e à modelação
- 10.1.1. A natureza do software
- 10.1.2. A natureza única das WebApps
- 10.1.3. Engenharia do Software
- 10.1.4. O processo do software
- 10.1.5. A prática da Engenharia do Software
- 10.1.6. Mitos do Software
- 10.1.7. Como é que tudo começa?
- 10.1.8. Conceitos orientados a objetos
- 10.1.9. Introdução ao UML

10.2. O processo do software
- 10.2.1. Um modelo geral de processo
- 10.2.2. Modelo de processo prescritivo
- 10.2.3. Modelo de processo especializado
- 10.2.4. O processo unificado
- 10.2.5. Modelos do processo pessoal e da equipa
- 10.2.6. O que é a agilidade?
- 10.2.7. O que é um processo ágil?
- 10.2.8. Scrum
- 10.2.9. Conjunto de ferramentas para o processo ágil

10.3. Princípios que orientam a prática da engenharia do software
- 10.3.1. Princípios que orientam o processo
- 10.3.2. Princípios que orientam a prática
- 10.3.3. Princípios de comunicação
- 10.3.4. Princípios de planificação
- 10.3.5. Princípios de modelação
- 10.3.6. Princípios de construção
- 10.3.7. Princípios de implantação

10.4. Compreensão dos requisitos
- 10.4.1. Engenharia de requisitos
- 10.4.2. Estabelecer as bases
- 10.4.3. Indagação dos requisitos
- 10.4.4. Desenvolvimento de casos de utilização
- 10.4.5. Elaboração do modelo dos requisitos
- 10.4.6. Negociação dos requisitos
- 10.4.7. Validação de requisitos

10.5. Modelação dos requisitos: cenários, informação e classes análise
- 10.5.1. Análise dos requisitos
- 10.5.2. Modelação baseada em cenários
- 10.5.3. Modelos UML que fornecem o caso de utilização
- 10.5.4. Conceitos de modelação de dados
- 10.5.5. Modelação baseada em classes
- 10.5.6. Diagrama de classes

10.6. Modelação de requisitos: fluxo, comportamento e padrões
- 10.6.1. Requisitos que modelam as estratégias
- 10.6.2. Modelação orientada ao fluxo
- 10.6.3. Diagramas de estado
- 10.6.4. Criação de um modelo de comportamento
- 10.6.5. Diagrama de sequência
- 10.6.6. Diagramas de comunicação
- 10.6.7. Padrões para modelação de requisitos

10.7. Conceitos de desenho
- 10.7.1. Desenho no contexto da engenharia do software
- 10.7.2. O processo de desenho
- 10.7.3. Conceitos de desenho
- 10.7.4. Conceitos de desenho orientado a objetos
- 10.7.5. O modelo do desenho

10.8. Desenho da arquitetura
- 10.8.1. Arquitetura do Software
- 10.8.2. Géneros arquitetónicos
- 10.8.3. Estilos arquitetónicos
- 10.8.4. Desenho arquitetónico
- 10.8.5. Evolução dos designs alternativos para a arquitetura
- 10.8.6. Mapeamento da arquitetura com a utilização do fluxo de dados

10.9. Desenho no nível de componentes e baseado em padrões
- 10.9.1. O que é um componente?
- 10.9.2. Desenho de componentes baseados em classe
- 10.9.3. Realização do desenho a nível de componentes
- 10.9.4. Desenho de componentes tradicionais
- 10.9.5. Desenvolvimento baseado em componentes
- 10.9.6. Padrões de desenho
- 10.9.7. Desenho de software baseado em Padrões
- 10.9.8. Padrões arquitetónicos
- 10.9.9. Padrões de desenho a nível de componentes
- 10.9.10. Padrões de desenho do interface de utilizador

10.10. Qualidade de software e administração de projetos
- 10.10.1. Qualidade
- 10.10.2. Qualidade do Software
- 10.10.3. O dilema da qualidade do software
- 10.10.4. Conseguir a qualidade do software
- 10.10.5. Garantia de qualidade de software
- 10.10.6. O espetro administrativo
- 10.10.7. O pessoal
- 10.10.8. O produto
- 10.10.9. O processo

05

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

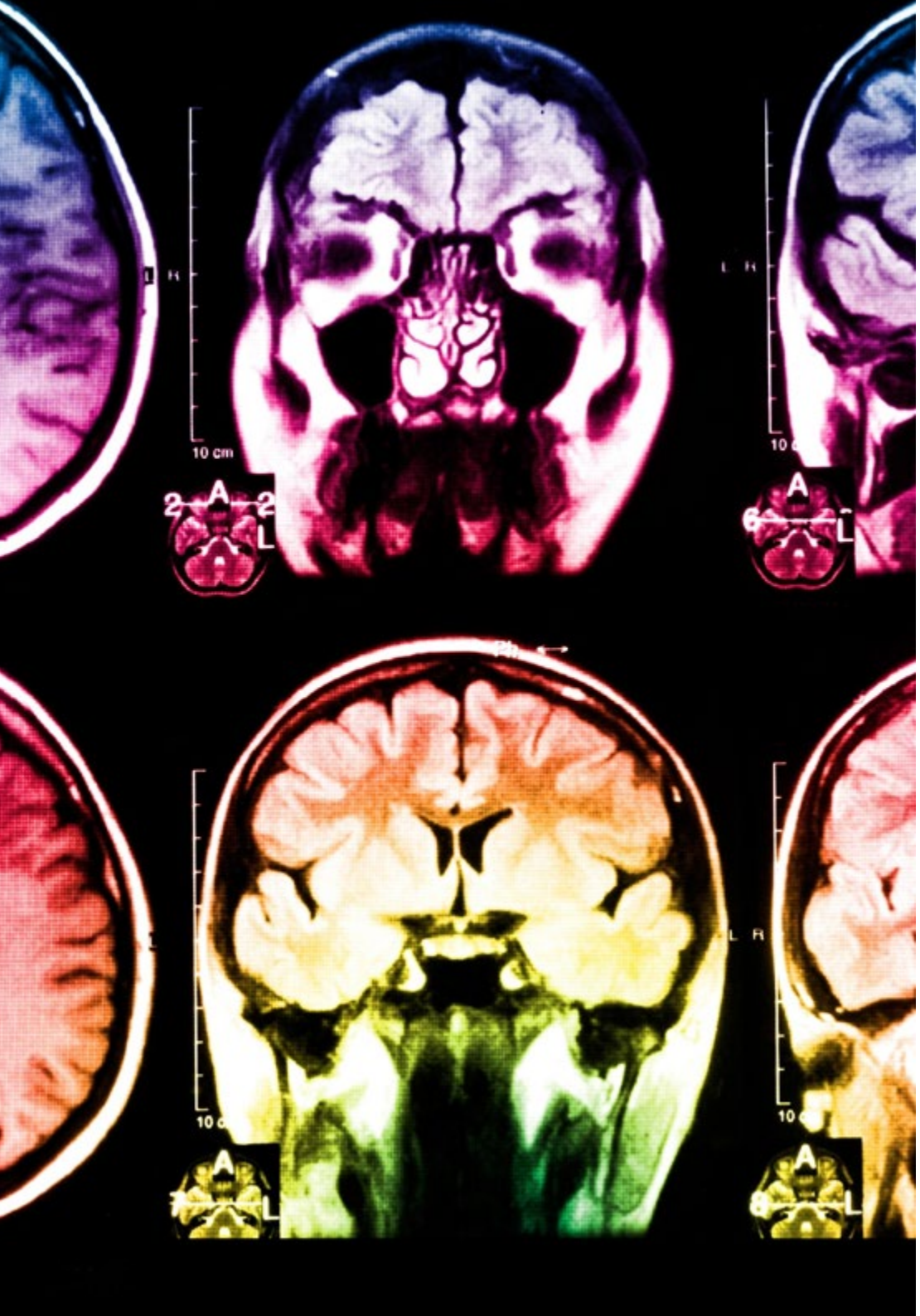
A TECH combina eficazmente a metodologia do Estudo de Caso com um sistema de aprendizagem 100% online baseado na repetição, que combina elementos didáticos diferentes em cada lição.

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.

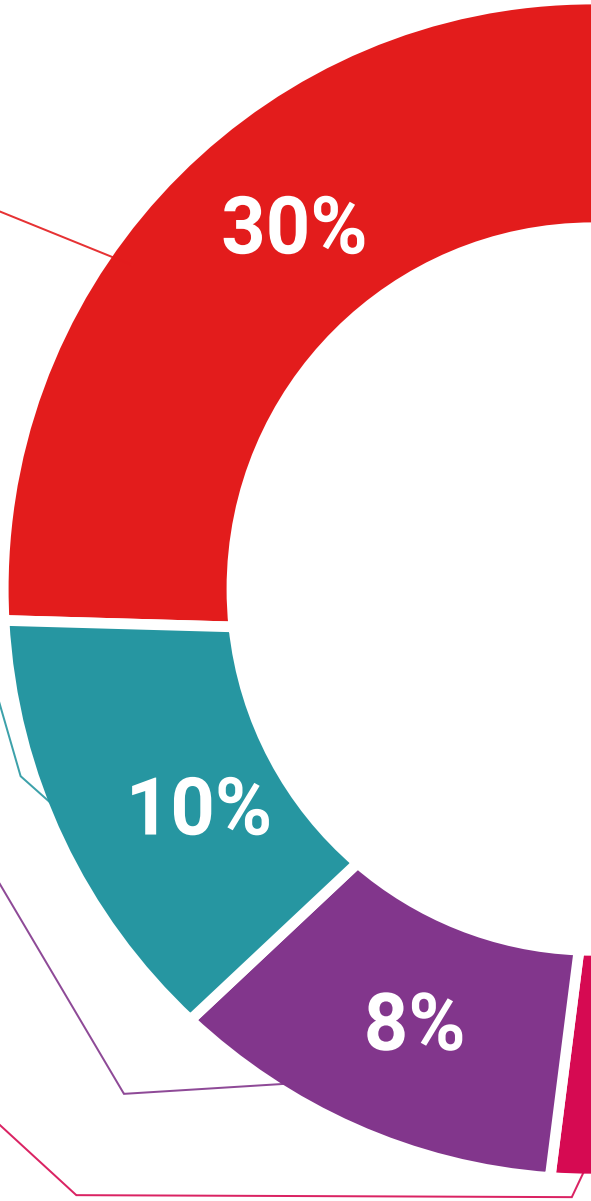
O denominado Learning from an Expert constrói conhecimento e memória, e gera confiança em futuras decisões difíceis.

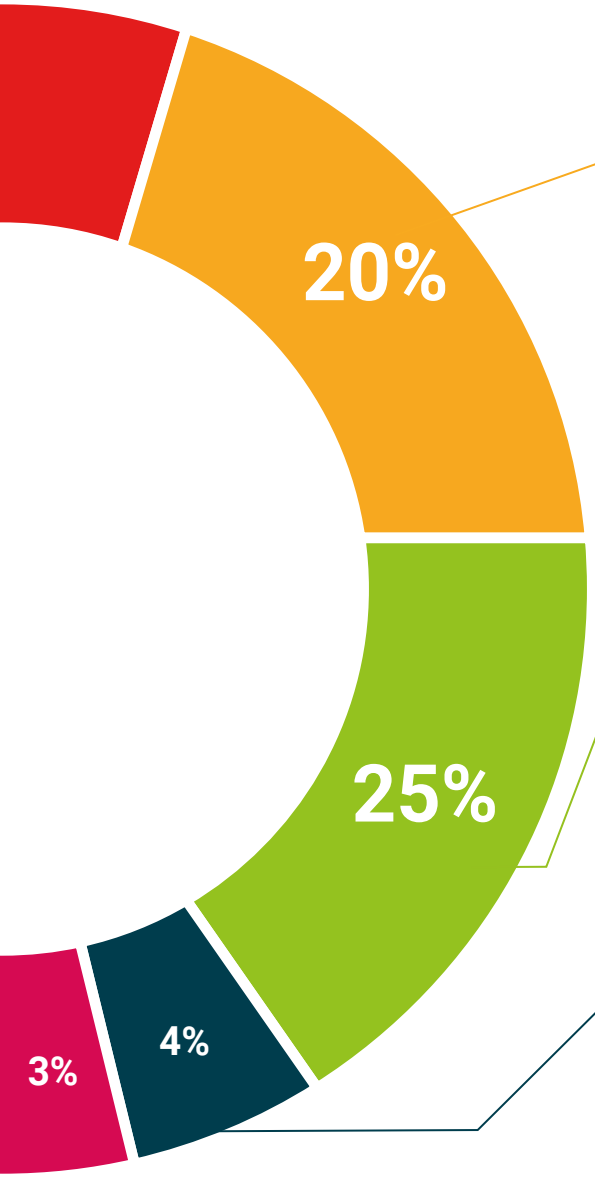
Práticas de aptidões e competências

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.

06

Certificação

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

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

Este **Mestrado Próprio em Desenvolvimento de 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 Desenvolvimento de Software**

ECTS: **60**

Carga horária: **1500 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

Desenvolvimento de Software

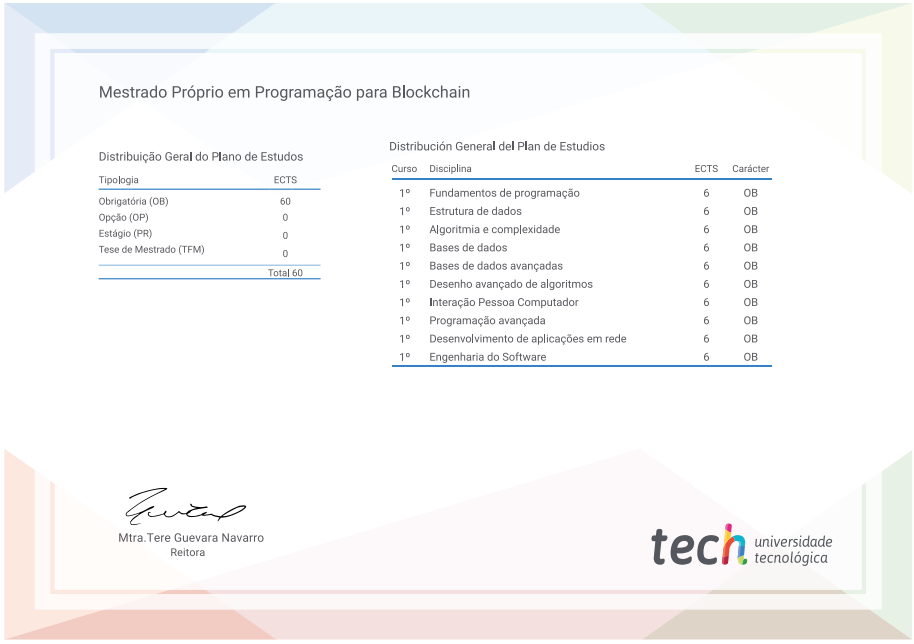
Este é um certificado atribuído por esta Universidade, reconhecido por 60 ECTS e equivalente a 1.500 horas, com data de início a dd/mm/aaaaa e data de conclusão a dd/mm/aaaaa.

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 Programação para Blockchain

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	Fundamentos de programação	6	OB
1º	Estrutura de dados	6	OB
1º	Algoritmia e complexidade	6	OB
1º	Bases de dados	6	OB
1º	Bases de dados avançadas	6	OB
1º	Desenho avançado de algoritmos	6	OB
1º	Interação Pessoa Computador	6	OB
1º	Programação avançada	6	OB
1º	Desenvolvimento de aplicações em rede	6	OB
1º	Engenharia 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
Desenvolvimento de 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
Desenvolvimento de Software

tech universidade tecnológica