

Mestrado

Programação desde Zero



Mestrado Programação desde Zero

- » Modalidade: online
- » Duração: 12 meses
- » Certificação: TECH Global University
- » Créditos: 60 ECTS
- » Horário: no seu próprio ritmo
- » Provas: online

Acesso ao site: www.techtitute.com/br/informatica/mestrado/mestrado-programacao-desde-zero

Índice

01

Apresentação do programa

pág. 4

02

Por que estudar na TECH?

pág. 8

03

Plano de estudos

pág. 12

04

Objetivos de ensino

pág. 24

05

Oportunidades profissionais

pág. 30

06

Metodologia de estudo

pág. 34

07

Equipe de professores

pág. 44

08

Certificação

pág. 48

01

Apresentação do programa

A Programação tornou-se uma habilidade essencial no século XXI, desempenhando um papel fundamental na transformação digital das sociedades e das economias. De acordo com um relatório da OCDE, mais de 60% dos empregos atuais exigem competências digitais avançadas, entre as quais a programação é uma das mais procuradas. Tendo em conta que se trata de um campo essencial na era digital, a TECH elaborou este programa que abordará os fundamentos desta disciplina e a sua aplicação em linguagens de uso global. Através de uma metodologia 100% online, os especialistas descobrirão como estruturar algoritmos, gerenciar dados e desenvolver projetos funcionais, enquanto constroem uma base sólida para continuar avançando no mundo tecnológico.



“

“Aprender a programar nunca foi tão acessível. Você dominará linguagens como Python e JavaScript, em qualquer lugar, graças à metodologia 100% online do Mestrado da TECH. Faça sua matrícula agora!”

A Programação é o núcleo da transformação digital que as sociedades modernas estão experimentando. Desde o desenvolvimento de aplicativos móveis até a automação de processos empresariais, saber programar é uma habilidade fundamental para abrir caminho em um ambiente de trabalho em constante evolução. Nesse sentido, os profissionais que dominam essa área não só encontrarão melhores oportunidades de crescimento, mas também aumentarão sua capacidade de se adaptar e prosperar em um mundo onde a tecnologia está em constante evolução.

Consciente dessa necessidade, a TECH criou o Mestrado em Programação desde Zero, um programa completo que proporcionará os conhecimentos mais atualizados relacionados a essa área. O plano de estudos, elaborado por especialistas do setor, abrangerá desde os fundamentos básicos de algoritmos e estruturas de dados até o uso prático de linguagens como *Python*, *JavaScript* e *HTML*. Além disso, incluirá módulos especializados em design web, desenvolvimento de aplicações e gestão de bases de dados. Desta forma, os especialistas adquirirão conhecimentos técnicos e desenvolverão habilidades analíticas e criativas para resolver problemas através da programação. Tudo isso com um enfoque prático que garantirá a implementação dos conceitos aprendidos.

Ao se especializarem nessa área, os alunos não só encontrarão novas oportunidades no setor tecnológico, mas também se destacarão em indústrias como a financeira, a de saúde ou a logística, onde a programação desempenha um papel fundamental. Além disso, serão capacitados para liderar projetos de desenvolvimento, automatizar processos e otimizar recursos. O objetivo é aumentar seu perfil profissional e facilitar seu acesso a oportunidades altamente competitivas.

A modalidade 100% online deste programa oferecerá a flexibilidade necessária para conciliar o aprendizado com outras responsabilidades pessoais ou profissionais. A TECH disponibilizará aos alunos uma plataforma inovadora acessível 24 horas por dia. Além disso, sua metodologia *Relearning* otimizará a formação através da repetição de conceitos-chave em diferentes contextos, facilitando uma assimilação progressiva e eficiente do conteúdo.

Este **Mestrado em Programação desde Zero** conta com o conteúdo mais completo e atualizado do mercado. Suas principais características são:

- ♦ O desenvolvimento de casos práticos apresentados por especialistas em Programação
- ♦ O conteúdo gráfico, esquemático e extremamente útil fornece informações científicas e práticas sobre aquelas disciplinas indispensáveis para o exercício da profissão
- ♦ Exercícios práticos em que o processo de autoavaliação pode ser usado para melhorar a aprendizagem
- ♦ Destaque especial para as metodologias inovadoras
- ♦ Aulas teóricas, perguntas a especialistas, fóruns de discussão sobre temas controversos e trabalhos de reflexão individual
- ♦ Disponibilidade de acesso a todo o conteúdo a partir de qualquer dispositivo, seja fixo ou móvel, com conexão à Internet



Desde algoritmos básicos até o desenvolvimento de aplicações, este programa levará você do zero ao nível de especialista em pouco tempo. Você se tornará o profissional que a indústria tecnológica precisa!”

“

Entre no mundo da programação com um programa de estudos incomparável. Você dominará Python, JavaScript e desenvolvimento web desde o zero com o Mestrado mais completo do mercado”

Inclui em seu corpo docente profissionais da área de programação, que trazem para este programa a experiência de seu trabalho, além de especialistas reconhecidos de empresas de referência e universidades de prestígio.

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

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

Com o apoio de professores especialistas em tecnologia, este programa irá guiar você passo a passo para dominar as linguagens mais procuradas no mercado. Comece hoje e construa um futuro melhor!

Você descobrirá uma abordagem inovadora que combina teoria e projetos reais neste Mestrado. Inscreva-se agora e dê vida às suas ideias com código!



02

Por que estudar na TECH?

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



“

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

A melhor universidade online do mundo de acordo com a FORBES

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

Forbes
Melhor universidade
online do mundo

Plano
de estudos
mais completo

Os planos de estudos mais completos do panorama universitário

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

A melhor equipe de professores top internacional

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

Equipe de professores
TOP
Internacional

Um método de aprendizado único

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

A metodologia
mais eficaz

A maior universidade digital do mundo

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

Nº.1
Mundial
A maior universidade
online do mundo

A universidade online oficial da NBA

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

Líderes em empregabilidade

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



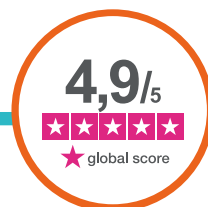
Google Partner Premier

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



A Universidade mais bem avaliada por seus alunos

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



03

Plano de estudos

Ao longo do programa, os profissionais abordarão os fundamentos de algoritmos, estruturas de dados e lógica de programação, para depois se aprofundarem em linguagens como *Python*, *JavaScript* e *HTML*. Além disso, aprofundarão áreas essenciais como web design, desenvolvimento de aplicações e gestão de bases de dados, oferecendo uma visão integral sobre o mundo da programação. Graças a uma abordagem progressiva e estruturada, este plano de estudos não só garantirá a aquisição de conhecimentos técnicos, mas também o desenvolvimento de competências analíticas e criativas, indispensáveis em um ambiente de trabalho competitivo.



“

Este programa é a opção ideal para iniciar sua carreira profissional no setor tecnológico e adquirir as ferramentas necessárias para se destacar em qualquer área da Programação”

Módulo 1. Programação e Desenvolvimento de Software desde o Zero

- 1.1. Software e Hardware. Relação e Diferenças
 - 1.1.1. Software
 - 1.1.2. Diferenças entre Software e Hardware
 - 1.1.3. Relação entre Software, Hardware e Programação
- 1.2. Programação: Aspectos fundamentais
 - 1.2.1. A Programação
 - 1.2.2. Objetivos e Aplicações
 - 1.2.3. Programas, Código Fonte, Compilação e Execução
 - 1.2.4. Erros: Erros de Sintaxe, de Execução, Lógicos
- 1.3. Programação desde Zero
 - 1.3.1. Estrutura de um Programa
 - 1.3.2. Algoritmos: Estrutura e Exemplos
 - 1.3.3. Relação entre algoritmos e programas
 - 1.3.4. Resolução de problemas através de algoritmos
- 1.4. Paradigmas e Tipos de Linguagens de Programação
 - 1.4.1. Paradigmas de programação
 - 1.4.1.1. Paradigmas imperativos
 - 1.4.1.2. Paradigma orientado a objetos
 - 1.4.1.3. Paradigma funcional
 - 1.4.1.4. Paradigma declarativo
 - 1.4.2. Linguagens de baixo nível vs. alto nível
 - 1.4.3. Linguagens compiladas vs. interpretadas
- 1.5. Tradução de Linguagens de Programação
 - 1.5.1. O Compilador. Processo de compilação
 - 1.5.2. O Intérprete: Processo de interpretação
 - 1.5.3. Diferenças entre compilação e interpretação
- 1.6. Bits, Operações Binárias e Portas Lógicas
 - 1.6.1. Bit. Representação binária
 - 1.6.2. Operações básicas com bits: E, OU, XOR, NOT
 - 1.6.3. Conversão entre binário e decimal
 - 1.6.4. Portões lógicos: OR, AND, XOR, NOT, NOR e NAND





- 1.7. Design de Algoritmos desde o Zero
 - 1.7.1. Projeto de um algoritmo a partir do zero
 - 1.7.2. Algoritmos sequenciais, condicionais e repetitivos
 - 1.7.3. Algoritmos recursivos e sua comparação com algoritmos iterativos
- 1.8. Elementos do Programa: Componentes e Estrutura
 - 1.8.1. Entrada e saída de dados
 - 1.8.2. Variáveis e constantes: Uso
 - 1.8.3. Processamento de dados e sua manipulação
 - 1.8.4. Funções e procedimentos básicos com diagramas de fluxo
- 1.9. Estruturas de Controle com Diagramas de Fluxo
 - 1.9.1. Estruturas de controle. Função na programação
 - 1.9.2. Estruturas condicionais: Exemplos com diagramas de fluxo
 - 1.9.3. Estruturas repetitivas: Exemplos com diagramas de fluxo
- 1.10. Ciclo de Vida e Modelos de Desenvolvimento de Software
 - 1.10.1. Ciclo de vida do Software. Fases
 - 1.10.2. Modelos de desenvolvimento: Cascata, iterativo e ágil
 - 1.10.3. Teste e manutenção no desenvolvimento de software

Módulo 2. Desenvolvimento Backend I: Python desde o zero

- 2.1. Python desde o zero. Instalações
 - 2.1.1. Linguagem Python. Características
 - 2.1.2. Instalação do Python no Windows, macOS e Linux
 - 2.1.3. Configuração do ambiente de desenvolvimento: IDEs e editores de código
 - 2.1.4. Primeiro programa em Python: "Hola Mundo"
- 2.2. Sintaxe e Variáveis em Python
 - 2.2.1. Estrutura do código em Python: Identificação
 - 2.2.2. Comentários em Python
 - 2.2.3. Variáveis e tipos de dados em Python
 - 2.2.4. Operações aritméticas e lógicas em Python
- 2.3. Controle de fluxo: Condicionais
 - 2.3.1. Estruturas de controle
 - 2.3.2. Condicionais: if, elif, else
 - 2.3.3. Condicional ternário

- 2.4. Loops em Python
 - 2.4.1. Uso de Loops na Programação
 - 2.4.2. Loops "for" e "while"
 - 2.4.3. Controle de fluxo em loops: break e continue
 - 2.4.4. Loops aninhados
- 2.5. Funções em Python
 - 2.5.1. Função em Python. Utilidade
 - 2.5.2. Parâmetros e Argumentos das Funções
 - 2.5.3. Valores de retorno
 - 2.5.4. Funções predefinidas vs. funções criadas pelo usuário
- 2.6. Listas e Tuplas em Python
 - 2.6.1. Criação e uso de listas em Python
 - 2.6.2. Operações comuns com listas: Adicionar, eliminar, modificar
 - 2.6.3. Tuplas: Diferenças com listas
 - 2.6.4. Iteração sobre listas e tuplas
- 2.7. Dicionários e Sets em Python
 - 2.7.1. Dicionários: Chave-valor
 - 2.7.2. Métodos para manipular dicionários
 - 2.7.3. Sets: Uso
 - 2.7.4. Comparação de dicionários e sets.
- 2.8. Gestão de arquivos em Python desde o zero
 - 2.8.1. Abertura e fechamento de arquivos
 - 2.8.2. Modos de abertura: Leitura, escrita e adição
 - 2.8.3. Leitura e gravação de arquivos de texto
- 2.9. Gestão de erros e exceções
 - 2.9.1. Tipos de Exceções
 - 2.9.2. Uso de Try, Except para lidar com erros
 - 2.9.3. Criação de exceções personalizadas
- 2.10. Boas Práticas e Depuração em Python
 - 2.10.1. Debugging: Propósito.
 - 2.10.2. Técnicas de depuração: Uso de print e breakpoints.
 - 2.10.3. Boas Práticas na Escrita de Código

Módulo 3. Desenvolvimento Backend II - Algoritmos e Estruturas de Dados com Python desde o zero

- 3.1. Algoritmos de Pesquisa em Estruturas de Dados
 - 3.1.1. Finalidade dos Algoritmos de Pesquisa em Estruturas de Dados
 - 3.1.2. Pesquisa linear: Implementação e casos de uso
 - 3.1.3. Pesquisa binária: Exemplos
 - 3.1.4. Comparação de eficiência: Pesquisa linear vs. pesquisa binária
- 3.2. Algoritmos de Ordenação em Estruturas de Dados (I). Técnicas básicas de ordenação Bubble Sort e Insertion Sort
 - 3.2.1. Classificação por bolhas (Bubble Sort): Implementação e análise
 - 3.2.2. Ordenação por inserção (Insertion Sort): Implementação e casos de uso
 - 3.2.3. Comparação entre Bubble Sort e Insertion Sort
- 3.3. Algoritmos de ordenação em estruturas de dados (II). Técnicas avançadas de ordenação: Selection Sort, Merge Sort e Quick Sort
 - 3.3.1. Selection Sort. Implementação e análise
 - 3.3.2. Merge Sort: Implementação
 - 3.3.3. Quick Sort: Implementação
 - 3.3.4. Comparação da eficiência entre algoritmos de ordenação
- 3.4. Algoritmos recursivos de pesquisa em estruturas de dados
 - 3.4.1. Recursão. Uso
 - 3.4.2. Recursão direta e indireta
 - 3.4.3. Algoritmos recursivos: Fatorial e Fibonacci. Exemplos
- 3.5. Complexidade dos algoritmos de pesquisa em estruturas de dados
 - 3.5.1. Complexidade algorítmica. Medição da eficiência
 - 3.5.2. Notação Big-O
 - 3.5.3. Análise da complexidade em algoritmos de pesquisa e ordenação
- 3.6. Estruturas de Dados Avançadas
 - 3.6.1. Árvores: Terminologia
 - 3.6.2. Árvores binárias: Operações
 - 3.6.3. Árvores de pesquisa binária (BST): Pesquisa, inserção e eliminação

- 3.7. Algoritmos de Grafos
 - 3.7.1. Grafos. Representação
 - 3.7.2. Algoritmos de Grafos: DFS e BFS
 - 3.7.3. Comparação entre DFS e BFS
 - 3.8. Programação Dinâmica
 - 3.8.1. Programação Dinâmica. Aplicação
 - 3.8.2. Diferenças entre programação dinâmica e recursão
 - 3.8.3. Otimização por meio de programação dinâmica
 - 3.9. Técnicas de Otimização de Algoritmos de Pesquisa de Dados
 - 3.9.1. Importância da Otimização Algorítmica da Pesquisa de Dados
 - 3.9.2. Técnicas de otimização: Memorização (Memoization)
 - 3.9.3. *Divide and Conquer*: Divide e conquiste
 - 3.10. Outros Algoritmos em Python
 - 3.10.1. Algoritmos de Permutação e Combinação
 - 3.10.2. Algoritmos de Hashing Básico
 - 3.10.3. Algoritmos de Contagem e Geração de Subconjuntos
- Módulo 4. Desenvolvimento Backend III - Flask, criação de APIs e arquitetura básica desde o zero**
- 4.1. Flask como Framework Backend
 - 4.1.1. Framework Backend. Finalidade
 - 4.1.2. Flask. Características
 - 4.1.3. Preparação do ambiente de desenvolvimento e instalação do Flask
 - 4.1.4. Primeiro projeto com Flask: "Hola Mundo"
 - 4.2. Rotas e Solicitações HTTP no Flask
 - 4.2.1. Rota: Funcionamento em uma aplicação web
 - 4.2.2. Métodos HTTP no Flask: GET, POST, PUT e DELETE
 - 4.2.3. Rotas no Flask com parâmetros e dados
 - 4.2.4. Organização de rotas em um projeto
 - 4.3. Controladores e Gestão de Respostas no Flask
 - 4.3.1. Controlador: Função e responsabilidades
 - 4.3.2. Tipos de respostas no Flask: Texto, JSON e HTML
 - 4.3.3. Criação de controladores para APIs no Flask
 - 4.3.4. Operações CRUD em controladores
 - 4.4. APIs RESTful
 - 4.4.1. API RESTful. Princípios
 - 4.4.2. Métodos HTTP e convenções em APIs RESTful
 - 4.4.3. Criação de uma API RESTful no Flask
 - 4.4.4. Projeto de uma API de tarefas com operações CRUD
 - 4.5. Bases de dados e Flask com SQLite
 - 4.5.1. Bases de dados em aplicações web
 - 4.5.2. Conexão com SQLite em projetos Flask
 - 4.5.3. Criação de tabelas e modelos usando SQLAlchemy
 - 4.5.4. Consultas CRUD em SQLite para gestão de dados
 - 4.6. Autenticação e Segurança Básica em APIs
 - 4.6.1. Autenticação e autorização em APIs
 - 4.6.2. Criação de um sistema de Autenticação de Usuário
 - 4.6.3. Uso de tokens para autenticação no Flask
 - 4.6.4. Proteção de rotas e dados do usuário em APIs
 - 4.7. Validação de Dados e Gestão de Erros
 - 4.7.1. Gestão de erros e exceções no Flask
 - 4.7.2. Validação de dados em solicitações à API
 - 4.7.3. Criação de mensagens de erro personalizadas
 - 4.7.4. Estratégias de validação e gestão de erros em CRUD
 - 4.8. Estruturação de APIs escaláveis
 - 4.8.1. Organização e estrutura de um projeto Flask escalável
 - 4.8.2. Modularização e separação de responsabilidades em APIs
 - 4.8.3. Otimização básica de APIs para desempenho e escalabilidade
 - 4.8.4. Estratégia organizacional para grandes projetos

- 4.9. Comunicação em Tempo Real com WebSockets
 - 4.9.1. WebSockets. Aplicações
 - 4.9.2. Implementação de WebSockets no Flask com Flask-SocketIO
 - 4.9.3. Comunicação em tempo real em aplicações Flask
- 4.10. Implementação e Manutenção de Aplicações
 - 4.10.1. Preparação de aplicações Flask para produção
 - 4.10.2. Deploy em plataformas populares como Heroku e Render
 - 4.10.3. Uso do Docker para implantação em contêineres
 - 4.10.4. Monitoramento e manutenção de aplicativos backend

Módulo 5. Programação Orientada a Objetos e Padrões de Design desde o Zero

- 5.1. Programação Orientada a Objetos (POO) desde o Zero
 - 5.1.1. Programação Orientada a Objetos
 - 5.1.2. Diferenças entre POO e programação estruturada
 - 5.1.3. Elementos POO: Classes, objetos, métodos e atributos
- 5.2. Classes e objetos em Python
 - 5.2.1. Criação de classes e objetos em Python
 - 5.2.2. Atributos de instância e de classe
 - 5.2.3. Métodos especiais (init, str, repr, etc.)
 - 5.2.4. Métodos estáticos e de classe: Usos
- 5.3. Encapsulamento e Abstração em Classes
 - 5.3.1. Encapsulamento: Utilidade
 - 5.3.2. Modificadores de acesso em Python
 - 5.3.2.1. Públicos, protegidos e privados
 - 5.3.3. Abstração: Ocultação de detalhes e melhoria da simplicidade
 - 5.3.4. Uso de propriedades (@property) para controle de acesso
- 5.4. Herança em Python. Utilidade em POO
 - 5.4.1. Herança. Utilidade em POO
 - 5.4.2. Criação de classes derivadas e herança múltipla em Python
 - 5.4.3. Métodos e atributos herdados e sobrecarga na herança
 - 5.4.4. Hierarquias de classes e gestão de classes base
- 5.5. Polimorfismo e sobrecarga em Python
 - 5.5.1. Polimorfismo: Duck Typing
 - 5.5.2. Polimorfismo com classes e métodos em Python
 - 5.5.3. Sobrecarga e Sobrescrita de métodos em Python
 - 5.5.4. O polimorfismo no design de software. Aplicações e Vantagens
- 5.6. Relações entre Classes e Design de Estruturas Complexas
 - 5.6.1. Tipos de relações: Associação, agregação e composição
 - 5.6.2. Diferenças entre agregação e composição: Exemplos
 - 5.6.3. Projeto de estruturas complexas utilizando relações entre classes
- 5.7. Padrões de Design e Princípios SOLID
 - 5.7.1. Relevância dos Padrões de Design
 - 5.7.2. Aplicação de padrões de design em projetos POO. Vantagens
 - 5.7.3. Classificação dos padrões de design.
 - 5.7.4. Princípios SOLID e sua importância no design orientado a objetos
- 5.8. Padrões de Design Criativos
 - 5.8.1. Finalidade dos Padrões de Design Criacionais
 - 5.8.2. Padrão Singleton
 - 5.8.3. Padrão Factory e Factory Method
 - 5.8.4. Padrão Builder
- 5.9. Padrões de Design Estrutural
 - 5.9.1. Finalidade dos Padrões de Design Estrutural
 - 5.9.2. Padrão Adapter
 - 5.9.3. Padrão Decorator
 - 5.9.4. Padrão Facade
- 5.10. Padrões de Design de Comportamento
 - 5.9.1. Padrões de comportamento. Aplicações
 - 5.10.2. Padrão Observer
 - 5.10.3. Padrão Strategy

Módulo 6. Frontend I - HTML e CSS desde o zero

- 6.1. HTML desde o zero
 - 6.1.1. HTML Objetivo em Desenvolvimento web
 - 6.1.2. Estrutura de um documento HTML: DOCTYPE, <html>, <head>, <body>
 - 6.1.3. Etiquetas semânticas e de conteúdo: <header>, <nav>, <section>, <footer>
 - 6.1.4. Elementos essenciais: Parágrafos (<p>), listas (,), enlaces (<a>), imagens ()
 - 6.1.5. Boas práticas em HTML
- 6.2. Elementos de Texto e Multimídia em HTML
 - 6.2.1. Etiquetas de texto essenciais: <p>, títulos, listas, negrito e itálico
 - 6.2.2. Inserção de multimídia: Atributos de , <audio>, <video>
 - 6.2.3. Atributos essenciais para acessibilidade (alt, aria-label)
- 6.3. Formulários em HTML
 - 6.3.1. Estrutura e componentes dos formulários: <form>, <input>, <label>, <button>
 - 6.3.2. Tipos de entradas: Texto, e-mail, senha, botões de envio
 - 6.3.3. Validação HTML5: Validação de campos no lado do cliente
 - 6.3.4. Formulários com validação básica. Exemplos
- 6.4. CSS desde o zero
 - 6.4.1. Linguagem CSS desde o zero: Uso e relação com HTML
 - 6.4.2. Sintaxe CSS: Seletores, propriedades e valores
 - 6.4.3. Aplicação de estilos em linha, internos e externos
 - 6.4.4. Seletores avançados: Tipo, classe, ID, pseudoclasses (:hover, :focus)
- 6.5. Modelo de Caixa (Box Model) em CSS
 - 6.5.1. Modelo da caixa: Importância em CSS
 - 6.5.2. Propriedades principais: margin, padding, border, width, height
 - 6.5.3. Uso do box-sizing para controle preciso do box model
 - 6.5.4. O design aplicado ao modelo da caixa. Exemplos
- 6.6. Tipografia e Estilos de Texto em CSS
 - 6.6.1. Propriedades de cores e fontes: color, font-family, font-size
 - 6.6.2. Estilos de texto avançados: Negrito, itálico, alinhamento (text-align)
 - 6.6.3. Espaçamento e separação de texto: line-height, letter-spacing
 - 6.6.4. Unidades de medida em CSS (px, em, rem) e seu uso em tipografia

- 6.7. Design de layouts com CSS - Flexbox
 - 6.7.1. Flexbox: Propósito
 - 6.7.2. Propriedades do Flexbox: justify-content, align-items, flex-direction
 - 6.7.3. Distribuição de elementos e alinhamento no Flexbox
 - 6.7.4. Exemplos de layouts com Flexbox
- 6.8. CSS Grid e Design Responsivo com CSS
 - 6.8.1. CSS Grid: Linhas, colunas e áreas
 - 6.8.2. Media Queries: Estrutura e aplicação em diferentes dispositivos
 - 6.8.3. Design responsivo para celular, tablet e desktop
 - 6.8.4. Ajuste de tipografia e unidades fluidas em design responsivo
- 6.9. Animações e Transições em CSS
 - 6.9.1. Transições: Propriedade transition, efeitos em :hover
 - 6.9.2. Animações com CSS: Uso de @keyframes, animações básicas
 - 6.9.3. Técnicas para suavizar transições e animações na web
- 6.10. Acessibilidade Web no Design
 - 6.10.1. Acessibilidade na web: Importância
 - 6.10.2. Design de sites acessíveis. Boas práticas
 - 6.10.3. Etiquetas aria e ferramentas de validação de acessibilidade

Módulo 7. Frontend II – JavaScript desde o zero

- 7.1. JavaScript desde o zero
 - 7.1.1. Linguagem JavaScript
 - 7.1.2. Integração de JavaScript em HTML
 - 7.1.3. Primeiro programa em JavaScript: "Hola Mundo"
- 7.2. Variáveis e Tipos de Dados em JavaScript
 - 7.2.1. Declaração de variáveis com var, let e const
 - 7.2.2. Tipos de dados: Números, cadeias, booleanos
 - 7.2.3. Conversão entre tipos de dados
- 7.3. Estruturas de Controle em JavaScript
 - 7.3.1. Condicionais: if, else if, else
 - 7.3.2. Bucles: for, while, do...while
 - 7.3.3. Switch-case: Alternativa a múltiplas condicionais
 - 7.3.4. Break e continue em loops

- 7.4. Funções no JavaScript
 - 7.4.1. Declaração de funções
 - 7.4.2. Parâmetros, valores de retorno e escopo
 - 7.4.3. Funções flecha (=>) e funções anônimas
 - 7.4.4. Callbacks e recursividade em funções
- 7.5. Manipulação do DOM (Document Object Model) com JavaScript
 - 7.5.1. DOM: Estrutura do documento HTML
 - 7.5.2. Seleção de elementos do DOM (getElementById, querySelector)
 - 7.5.3. Manipulação de elementos: Alterar texto, estilos e atributos
 - 7.5.4. Eventos: click, input, submit e mais
- 7.6. Arranjos e Objetos em JavaScript
 - 7.6.1. Declaração e uso de arranjos
 - 7.6.2. Métodos comuns de arranjos: push, pop, map, filter
 - 7.6.3. Criação e uso de objetos
 - 7.6.4. Iteração sobre matrizes e objetos
- 7.7. Promessas e Assincronia no JavaScript
 - 7.7.1. Assincronia e uso de callbacks em JavaScript
 - 7.7.2. Promise em JavaScript: Criação e gestão
 - 7.7.3. Uso de async e await em JavaScript
- 7.8. APIs e Fetch em JavaScript
 - 7.8.1. API em JavaScript: Propósito
 - 7.8.2. Consumo de APIs REST com fetch
 - 7.8.3. Gestão de erros e status das solicitações
- 7.9. Armazenamento local em navegadores da web
 - 7.9.1. Local Storage e Session Storage em serviços e aplicações web
 - 7.9.2. Armazenamento e recuperação de dados no Local Storage
 - 7.9.2. IndexedDB como base de dados para o navegador
 - 7.9.3. Gestão de cookies em JavaScript
 - 7.9.4. Armazenamento no navegador: Exemplos
- 7.10. Boas Práticas em JavaScript e Ferramentas para Desenvolvedores
 - 7.10.1. Código em JavaScript: Boas práticas
 - 7.10.2. Uso de ferramentas de desenvolvimento do navegador em JavaScript
 - 7.10.3. Depuração e gestão de erros em JavaScript

Módulo 8. Frontend III - React.js desde o zero

- 8.1. React.js desde o zero
 - 8.1.1. React JS como biblioteca para o desenvolvimento de aplicações web
 - 8.1.2. Componentes e Virtual DOM no React JS: Arquitetura e funcionamento
 - 8.1.3. Instalação e configuração com o framework NextJS
 - 8.1.4. Primeiro componente no React: "Hola Mundo"
- 8.2. JavaScript XML ou JSX e componentes em React
 - 8.2.1. JSX: Sintaxe e características
 - 8.2.2. Criação de componentes funcionais em React.js
 - 8.2.3. Uso de props para a passagem de dados entre componentes
 - 8.2.4. Componentes funcionais versus componentes de classe para o desenvolvimento em React.js
- 8.3. Estado e Eventos em React,Js
 - 8.3.1. Status dos componentes no React
 - 8.3.2. Uso de useState para a gestão do estado
 - 8.3.3. Gestão de eventos em React.js: onClick, onChange, entre outros
 - 8.3.4. Exemplos de gestão de status e eventos no React.js
- 8.4. Ciclo de Vida de Componentes e Efeitos em React
 - 8.4.1. Ciclo de vida dos componentes no React
 - 8.4.2. Usando useEffect para gerenciar efeitos no React
 - 8.4.3. Componentes com montagem, atualização e desmontagem no React
- 8.5. Roteamento com React Router
 - 8.5.1. SPA (Single Page Applications) e roteamento em aplicações web
 - 8.5.2. Instalação e configuração do React Router
 - 8.5.3. Criação de rotas e navegação entre páginas com o React Router
- 8.6. Formulários e validação no React
 - 8.6.1. Criação de formulários interativos no React
 - 8.6.2. Gestão de entradas do usuário e envio de dados no React
 - 8.6.3. Validação de formulários em tempo real no React
- 8.7. Consumo de APIs no React
 - 8.7.1. Consumo de APIs no React
 - 8.7.2. Gestão de estados de carga, sucesso e erro no React
 - 8.7.3. Atualização de componentes com base em dados da API no React

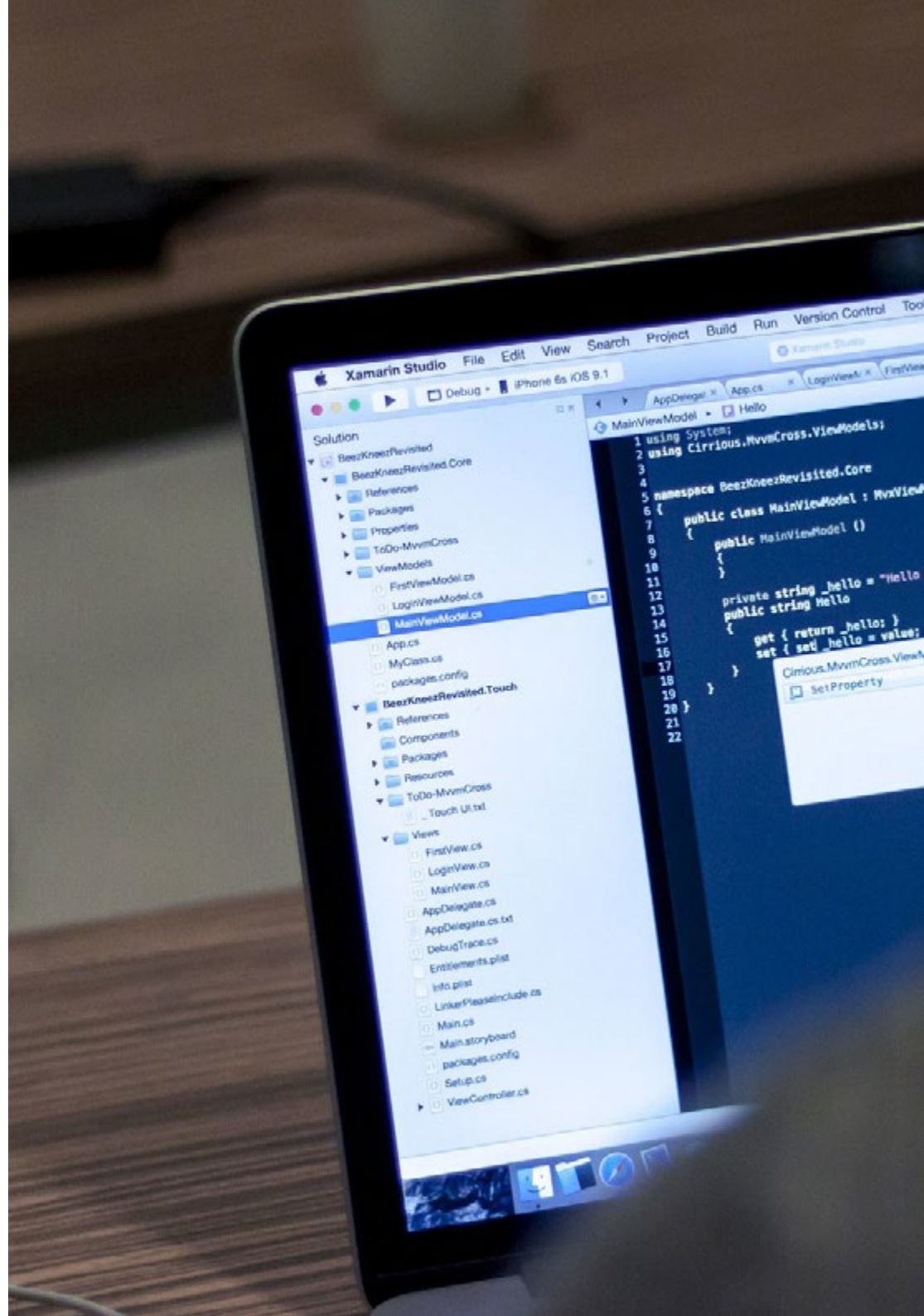
- 8.8. Componentes reutilizáveis e bibliotecas externas no React
 - 8.8.1. Componentes reutilizáveis no React
 - 8.8.2. Criação de componentes reutilizáveis no React
 - 8.8.3. Uso de bibliotecas externas como Material UI e Bootstrap no React
- 8.9. Gestão global do Estado no React
 - 8.9.1. Gestão global do estado com opções nativas: Context API e Hooks personalizados
 - 8.9.2. Bibliotecas externas para a gestão de dados
 - 8.9.3. Comparação de abordagens para a gestão global do estado. Exemplos
- 8.10. Implantação e Otimização de Aplicativos React
 - 8.10.1. Preparação de uma aplicação React para produção
 - 8.10.2. Deploy em plataformas como Netlify e Vercel
 - 8.10.3. Otimização do desempenho: Lazy Loading, memoização, componentes do servidor e code splitting.
 - 8.10.4. Monitoramento e manutenção de aplicações React em produção. Ferramentas e análise de desempenho

Módulo 9. Gerenciamento e Otimização de Bancos de Dados desde o Zero

- 9.1. Base de Dados desde Zero
 - 9.1.1. Bases de dados: Tipos
 - 9.1.2. Bancos de dados relacionais vs. não relacionais
 - 9.1.3. Linguagens de programação SQL e NoSQL
- 9.2. Modelagem de Dados Relacionais
 - 9.2.1. Modelo relacional de Base de Dados
 - 9.2.2. Tabelas, linhas e colunas em um banco de dados relacional
 - 9.2.3. Chaves primárias e estrangeiras: Relações entre tabelas
 - 9.2.4. Normalização: 1NF, 2NF, 3NF
- 9.3. Linguagem SQL DML e DDL
 - 9.3.1. SQL: Linguagem de consulta estruturada
 - 9.3.2. Consultas sobre criação e eliminação: CREATE, DROP
 - 9.3.3. Consultas SELECT, INSERT, UPDATE e DELETE
 - 9.3.4. Filtragem e ordenação de dados com SQL
- 9.4. Consultas Avançadas em SQL
 - 9.4.1. Joins: INNER JOIN e OUTER JOIN
 - 9.4.2. Subconsultas e consultas aninhadas
 - 9.4.3. Funções agregadas no SQL: SUM, AVG, COUNT
- 9.5. Bases de Dados NoSQL e MongoDB
 - 9.5.1. Banco de dados NoSQL
 - 9.5.2. Comparação entre SQL e NoSQL
 - 9.5.3. MongoDB: Banco de dados de documentos
 - 9.5.4. Esquemas flexíveis em NoSQL
- 9.6. Otimização de Bancos de Dados
 - 9.6.1. Importância da otimização das consultas
 - 9.6.2. Uso de índices em bancos de dados relacionais
 - 9.6.3. Otimização de bancos de dados NoSQL
- 9.7. Segurança em Bancos de Dados
 - 9.7.1. Segurança em bancos de dados
 - 9.7.2. Criptografia de dados confidenciais
 - 9.7.3. Gestão de usuários e permissões em bancos de dados
 - 9.7.4. Estratégias de proteção de bancos de dados contra ataques
- 9.8. Escalabilidade de Bases de Dados
 - 9.8.1. Escalabilidade em bancos de dados
 - 9.8.2. Particionamento horizontal e vertical
 - 9.8.3. Replicação e clustering em bancos de dados
- 9.9. Backup e Recuperação de Dados
 - 9.9.1. Importância do backup em bancos de dados
 - 9.9.2. Técnicas de backup automáticas e manuais
 - 9.9.3. Recuperação de dados em bancos de dados relacionais e NoSQL
- 9.10. Implementação de Bancos de Dados em Projetos
 - 9.10.1. Design de bancos de dados para um projeto real
 - 9.10.2. Integração de banco de dados com aplicações backend

Módulo 10. Ferramentas de Desenvolvimento desde o zero: Linux, Controle de Versões, CI/CD, Docker e Metodologias Ágeis

- 10.1. Linux desde o zero
 - 10.1.1. Linux
 - 10.1.2. Diferenças entre o Linux e outros sistemas operacionais
 - 10.1.3. Distribuições populares do Linux para desenvolvedores
 - 10.1.4. Configuração e personalização do ambiente de desenvolvimento
 - 10.1.5. Editores de texto no Linux
- 10.2. Usando o Terminal no Linux desde o Zero
 - 10.2.1. A Terminal. Usos e funcionalidades
 - 10.2.2. Comandos de navegação e gestão de arquivos na terminal
 - 10.2.3. Permissões de arquivos e diretórios no Linux
 - 10.2.4. Redirecionamento de comandos e uso de pipes para otimizar tarefas
- 10.3. Controle de versões com Git desde o zero
 - 10.3.1. Git: Fornecedores CLOUD
 - 10.3.2. Criação e Gestão de Repositórios
 - 10.3.3. Fluxo de trabalho: git init, git add, git commit e git status
 - 10.3.4. Trabalho com galhos: Criação, fusão e resolução de conflitos
- 10.4. Colaboração em equipe com GitHub desde zero
 - 10.4.1. GitHub: Repositórios remotos
 - 10.4.2. Conectando um repositório local ao GitHub: git remote. Configuração inicial
 - 10.4.3. Sincronização com repositórios remotos
 - 10.4.4. Pull requests e revisão colaborativa de código
- 10.5. CI/CD (I) - Integração Contínua (CI) com GitHub Actions desde zero
 - 10.5.1. A Integração Contínua (CI)
 - 10.5.2. Configuração de workflows no GitHub Actions
 - 10.5.3. Automatização de testes e implementações
- 10.6. Docker desde o zero
 - 10.6.1. Docker e Contêineres
 - 10.6.2. Instalação e configuração do Docker
 - 10.6.3. Criação e gestão de contêineres Docker
 - 10.6.4. Dockerfiles: Criação de imagens personalizadas



- 10.7. CI/CD (II) - Entrega Contínua (CD) com Docker e GitHub Actions desde zero
 - 10.7.1. Entrega contínua (CD)
 - 10.7.2. Configuração de pipelines de CD com Docker e GitHub Actions
 - 10.7.3. Implantação automatizada com o Docker Compose
- 10.8. Metodologias Ágeis desde o Zero (I). Princípios e Valores
 - 10.8.1. Metodologias Ágeis Princípios
 - 10.8.2. O Manifesto Ágil: Valores e princípios fundamentais
 - 10.8.3. Comparação com metodologias tradicionais: Cascata vs. Ágil
- 10.9. Metodologias Ágeis (II): Scrum desde o zero
 - 10.9.1. Scrum e sua aplicabilidade
 - 10.9.2. Funções-chave no Scrum: Product Owner, Scrum Master e Development Team
 - 10.9.3. Artefatos no Scrum: Product Backlog, Sprint Backlog e Incremento
 - 10.9.4. Eventos Scrum: Sprint Planning, Daily Scrum, Sprint Review e Retrospective
- 10.10. Metodologias Ágeis (III): Kanban e Métricas desde o Zero
 - 10.10.1. Kanban e sua abordagem visual
 - 10.10.2. Elementos-chave no Kanban: Colunas, cartões e limites WIP
 - 10.10.3. Métricas ágeis: Burnup, burndown charts, velocity e lead tim

“

Você se capacitará no conforto da sua casa e no seu próprio ritmo, com a metodologia online inovadora e a flexibilidade que você precisa. Quer impulsionar sua carreira em tecnologia? Não espere mais! Matricule-se hoje"



04

Objetivos de ensino

Este Mestrado tem como objetivo principal dotar os profissionais das competências necessárias para iniciar a sua carreira no mundo da programação de forma sólida e eficaz. Ao longo do programa, você desenvolverá uma compreensão profunda dos conceitos fundamentais da programação, tais como algoritmos, estruturas de dados e lógica de programação, que constituem a base sobre a qual você construirá habilidades mais avançadas. Além disso, adquirirão domínio das linguagens de programação amplamente utilizadas na indústria (*Python*, *JavaScript* e *HTML*), o que lhes permitirá enfrentar com confiança diversos projetos tecnológicos.

“

Você adquirirá as habilidades necessárias para liderar projetos de desenvolvimento, gerenciar bancos de dados e otimizar processos, aumentando sua competitividade profissional no setor tecnológico”



Objetivos gerais

- ♦ Dominar os fundamentos da programação e do desenvolvimento de software através da identificação dos elementos essenciais de um programa, das estruturas de controle e da relação entre software e hardware
- ♦ Compreender os fundamentos do Python e sua sintaxe básica, desenvolvendo um ambiente adequado para o desenvolvimento backend
- ♦ Adquirir uma sólida compreensão sobre os algoritmos e estruturas de dados fundamentais
- ♦ Dominar os fundamentos do desenvolvimento backend e seu papel dentro de uma arquitetura de software
- ♦ Gerenciar os princípios da programação orientada a objetos e aplicá-los na construção de software flexível, reutilizável e sustentável, utilizando Python como linguagem de programação
- ♦ Desenvolver os conceitos fundamentais de HTML e CSS para estruturar e estilizar páginas web
- ♦ Analisar a linguagem JavaScript, desde os fundamentos até as técnicas avançadas
- ♦ Introduzir os conceitos fundamentais do React.js para construir aplicações web interativas
- ♦ Administrar os tipos de bases de dados e suas linguagens de consulta, tanto em ambientes relacionais como NoSQL
- ♦ Detalhar a automação de testes, integração e implantação de códigos





Objetivos específicos

Módulo 1. Programação e Desenvolvimento de Software desde o Zero

- ♦ Definir e diferenciar entre software e hardware
- ♦ Fundamentar os conceitos essenciais da programação
- ♦ Conhecer a estrutura básica de um programa
- ♦ Explorar e analisar os diferentes paradigmas de programação

Módulo 2. Desenvolvimento Backend I: Python desde o zero

- ♦ Dominar as características do Python
- ♦ Compreender a estrutura e a sintaxe básica do Python
- ♦ Desenvolver habilidades no controle de fluxo por meio de condicionais
- ♦ Aplicar loops para criar ciclos de repetição em Python

Módulo 3. Desenvolvimento Backend II - Algoritmos e Estruturas de Dados com Python desde o zero

- ♦ Implementar e comparar tipos de algoritmos de pesquisa em estruturas de dados
- ♦ Analisar algoritmos de ordenação como bolha, inserção, seleção, merge sort e quick sort
- ♦ Examinar a complexidade algorítmica e a medição da eficiência utilizando a notação Big O
- ♦ Representar grafos e realizar percursos em profundidade (DFS) e em largura (BFS)

Módulo 4. Desenvolvimento Backend III - Flask, criação de APIs e arquitetura básica desde o zero

- ♦ Desenvolver APIs RESTful utilizando Flask
- ♦ Integrar bases de dados em aplicativos Flask
- ♦ Implementar autenticação e segurança em APIs
- ♦ Projetar a arquitetura básica de uma aplicação backend com Flask

Módulo 5. Programação Orientada a Objetos e Padrões de Design desde o Zero

- ♦ Definir os conceitos-chave da Programação Orientada a Objetos, tais como classes, objetos, atributos, métodos, encapsulamento, abstração, herança e polimorfismo
- ♦ Compreender o uso de encapsulamento e abstração em classes através do uso da linguagem de programação Python
- ♦ Examinar o conceito de polimorfismo e sobrecarga na linguagem Python, compreendendo suas aplicações e vantagens
- ♦ Determinar os tipos de relações entre classes, como associação, agregação e composição

Módulo 6. Frontend I - HTML e CSS desde o zero

- ♦ Identificar a estrutura básica de um documento HTML e sua importância no desenvolvimento web
- ♦ Utilizar HTML para organizar e apresentar conteúdo na web de forma semântica e acessível: Arquitetura web
- ♦ Aplicar estilos com CSS para melhorar a apresentação visual dos elementos
- ♦ Use o modelo de caixa CSS para estruturar e distribuir os elementos na interface



Módulo 7. Frontend II – JavaScript desde o zero

- ♦ Compreender a sintaxe e os tipos de dados em JavaScript
- ♦ Aprender a estruturar o código utilizando funções e estruturas de controle
- ♦ Manipular o DOM para interagir com páginas web dinâmicas
- ♦ Trabalhar com APIs e gerenciar assincronia usando Promessas e async/await

Módulo 8. Frontend III - React.js desde o zero

- ♦ Compreender o uso do JSX para criar interfaces declarativas
- ♦ Aprenda a trabalhar com componentes funcionais, props e ciclos de vida
- ♦ Gerenciar estados locais e globais utilizando ferramentas modernas como Context API e Redux Toolkit
- ♦ Implementar roteamento para construir aplicações de página única (SPA)

Módulo 9. Gerenciamento e Otimização de Bancos de Dados desde o Zero

- ♦ Reconhecer os tipos de bancos de dados e suas características
- ♦ Compreender e aplicar o modelo relacional de dados
- ♦ Desenvolver habilidades em SQL para gerenciamento de bancos de dados
- ♦ Utilizar consultas avançadas em SQL

Módulo 10. Ferramentas de Desenvolvimento desde o zero: Linux, Controle de Versões, CI/CD, Docker e Metodologias Ágeis

- ♦ Operar o sistema operativo Linux ao nível da linha de comandos
- ♦ Dominar o uso do Git para controle de versões
- ♦ Implementar pipelines de integração e implantação contínua (CI/CD)
- ♦ Criar e gerenciar contêineres Docker



Quer dominar a linguagem do futuro? Este Mestrado representa o primeiro passo em direção ao futuro tecnológico que você sempre desejou. Matricule-se já!"

05

Oportunidades profissionais

Este programa abre uma ampla gama de saídas profissionais em um setor tecnológico em constante expansão. À medida que as empresas de todos os setores se digitalizam, a demanda por profissionais com habilidades em programação não só aumenta, mas também se diversifica. Dessa forma, surgem oportunidades em áreas tão variadas, como desenvolvimento de software, gerenciamento de bancos de dados, design web e automação de processos. Ao concluir o curso, os alunos estarão perfeitamente capacitados para assumir funções em empresas de tecnologia, *startups* inovadoras e organizações de qualquer setor que busquem integrar soluções digitais em suas operações.





“

Você não só se tornará um especialista de alto nível, mas também terá acesso a uma ampla variedade de oportunidades profissionais, garantindo seu sucesso em um mercado de trabalho em constante crescimento e transformação”

Perfil do aluno

O aluno será um profissional altamente capacitado para enfrentar os desafios tecnológicos do futuro. Nesse sentido, você não só dominará os conceitos fundamentais de algoritmos, estruturas de dados e linguagens de programação, mas também terá desenvolvido uma compreensão profunda sobre as ferramentas e tecnologias mais procuradas na indústria. Com uma base sólida em programação e desenvolvimento de aplicações, o especialista será capaz de abordar projetos tecnológicos de forma autônoma e eficaz. Além disso, o profissional se destacará pela sua capacidade de resolver problemas complexos, aplicar soluções inovadoras e gerenciar sistemas e bancos de dados.

Você se tornará um profissional versátil, com uma sólida preparação técnica e uma visão estratégica, capaz de integrar a programação em soluções tecnológicas que respondam às necessidades das empresas e da sociedade.

- ♦ **Pensamento crítico e resolução de problemas:** Analisar situações complexas e encontrar soluções inovadoras e eficientes através da programação.
- ♦ **Trabalho em equipe e colaboração multidisciplinar:** Trabalhar em equipes multidisciplinares, colaborando de forma eficaz com profissionais de diferentes áreas (designers, engenheiros e analistas de dados), para alcançar objetivos comuns em projetos tecnológicos.
- ♦ **Gestão do tempo e autonomia na aprendizagem:** Gerenciar o tempo de maneira eficaz para lidar com prazos apertados e desenvolver projetos de forma independente e responsável.
- ♦ **Adaptabilidade a novas tecnologias:** Gerenciar novas ferramentas e linguagens de programação, garantindo sua atualização constante em um ambiente tecnológico que evolui em ritmo acelerado.



Após concluir o programa de Mestrado, você poderá usar seus conhecimentos e habilidades nos seguintes cargos:

- 1. Desenvolvedor de Software:** Responsável por projetar, criar e manter aplicativos e programas de computador de acordo com as necessidades do cliente ou da empresa.
- 2. Desenvolvedor Web:** Gestor da criação, design e manutenção de sites, garantindo sua funcionalidade e otimização em diferentes dispositivos.
- 3. Engenheiro de Bases de Dados:** Responsável por projetar, implementar e gerenciar bancos de dados, garantindo sua eficiência, segurança e disponibilidade.
- 4. Analista de Sistemas:** Desenvolvedor dos sistemas informáticos existentes, propondo melhorias para otimizar o desempenho e a eficiência dos processos.
- 5. Programador Front-End:** Responsável pelo desenvolvimento da parte visual de aplicativos e sites, melhorando a experiência do usuário.
- 6. Programador Back-End:** Responsável por desenvolver a lógica do servidor e bases de dados de aplicações e sites, garantindo o seu correto funcionamento.
- 7. Administrador de Redes:** Responsável por gerenciar e manter as infraestruturas de rede dentro de uma empresa, garantindo sua estabilidade e segurança.
- 8. Consultor Tecnológico:** Orientador e consultor em empresas, fornecendo informações sobre a implementação de soluções tecnológicas adequadas às suas necessidades, melhorando seu desempenho e competitividade.
- 9. Desenvolvedor de Aplicativos Móveis:** Gestor do design, desenvolvimento e manutenção de aplicações para dispositivos móveis, garantindo sua usabilidade e desempenho.
- 10. Engenheiro de Software de Automação:** Responsável por projetar e desenvolver soluções automatizadas para melhorar a eficiência dos processos empresariais, utilizando programação e tecnologia avançada.

06

Metodologia de estudo

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

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



“

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

O aluno: a prioridade de todos os programas da TECH

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

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

“

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



Os programas de ensino mais abrangentes do mundo

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

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

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

“

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

Case studies ou Método de caso

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

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

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



Método Relearning

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

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

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

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



Um Campus Virtual 100% online com os melhores recursos didáticos

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

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

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

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



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

A eficácia do método é justificada por quatro conquistas fundamentais:

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

A metodologia universitária mais bem avaliada por seus alunos

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

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

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

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



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



Material de estudo

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

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



Práticas de aptidões e competências

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



Resumos interativos

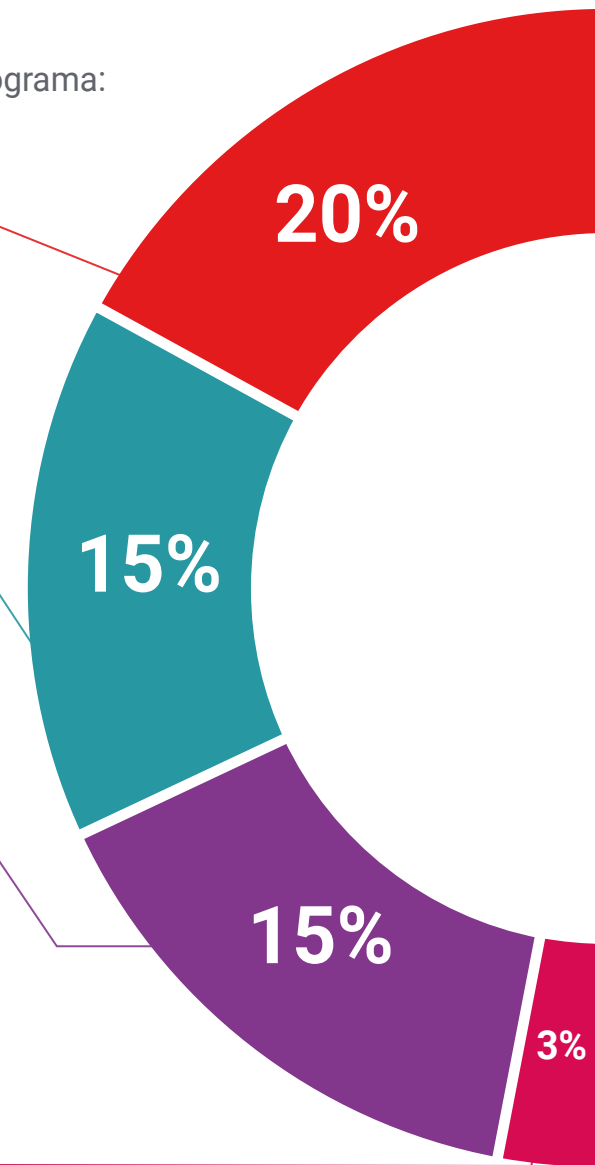
Apresentamos os conteúdos de forma atraente e dinâmica em pílulas multimídia que incluem áudio, vídeos, imagens, diagramas e mapas conceituais com o objetivo de reforçar o conhecimento.

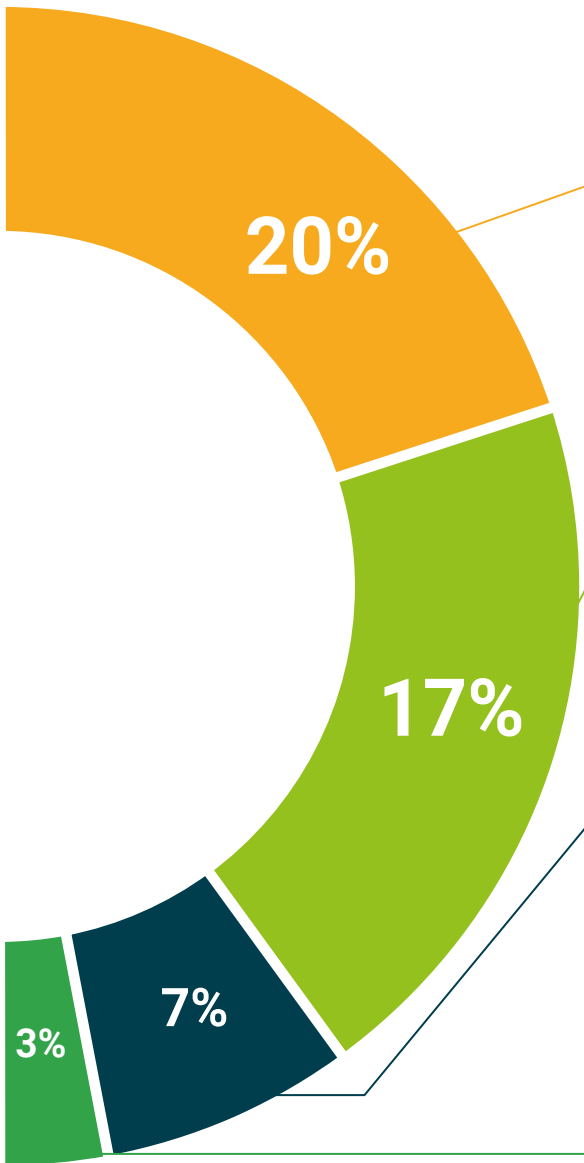
Este sistema exclusivo de capacitação por meio da apresentação de conteúdo multimídia foi premiado pela Microsoft como "Caso de sucesso na Europa"



Leituras complementares

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





Case Studies

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



Testing & Retesting

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



Masterclasses

Há evidências científicas sobre a utilidade da observação de terceiros especialistas.
O *Learning from an expert* fortalece o conhecimento e a memória, e aumenta nossa confiança para tomar decisões difíceis no futuro.



Guias rápidos de ação

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



07

Equipe de professores

O corpo docente é composto por profissionais altamente qualificados e com ampla experiência na área tecnológica. Cada um foi cuidadosamente selecionado para garantir que os alunos recebam uma formação de qualidade, baseada em conhecimentos atualizados e na prática real do setor. Nesse sentido, a experiência profissional de mentores em empresas de renome e a formação acadêmica permitem oferecer uma perspectiva integral e aplicada sobre Programação. Em definitiva, os mentores são especialistas em diversas áreas da tecnologia, o que proporciona uma visão multidisciplinar e enriquecedora.



“

O corpo docente é um pilar fundamental para o sucesso do programa. A qualidade e a experiência dos professores garantirão uma formação integral que o preparará para enfrentar o setor tecnológico com confiança e capacidade”

Direção



Dr. Juan Manuel Lucas Cuesta

- ♦ Engenheiro Sênior de Software e Analista da Indizen - Believe in Talent
- ♦ Engenheiro Sênior de Software e Analista na Krell Consulting e IMAGiNA Artificial Intelligence
- ♦ Engenheiro de Software da Intel Corporation
- ♦ Engenheiro de Software na Intelligent Dialogue Systems
- ♦ Doutor em Engenharia Eletrônica de Sistemas para ambientes inteligentes pela Universidade Politécnica de Madri
- ♦ Formado em Engenharia de Telecomunicações pela Universidade Politécnica de Madri
- ♦ Mestrado em Engenharia Eletrônica de Sistemas para Ambientes Inteligentes pela Universidade Politécnica de Madri



Sr. Juan Antonio Márquez Ruiz de Lacanal

- ♦ Desenvolvedor de software na GTD Defense & Security Solutions
- ♦ Desenvolvedor de software na Solera Inc.
- ♦ Engenheiro de Desenvolvimento e Pesquisa na GRVC Sevilla
- ♦ Co-fundador do Unmute
- ♦ Co-fundador da VR Educa
- ♦ Intercâmbio acadêmico em Engenharia e Empreendedorismo na Universidade Berkeley da Califórnia
- ♦ Formado em Engenharia Industrial pela Universidade de Sevilha

Professores

Dr. Francisco Luna Perejón

- ◆ Especialista em Arquitetura e Tecnologia da Computação
- ◆ Doutor em Engenharia Informática pela Universidade de Sevilha
- ◆ Mestrado em Engenharia Informática pela Universidade de Sevilha
- ◆ Licenciatura em Engenharia da Saúde pela Universidade de Sevilha
- ◆ Licenciatura em Engenharia Informática e Tecnologias da Informação pela Universidade de Sevilha
- ◆ Membro: Grupo de pesquisa em Robótica e Tecnologia da Informação (TEP108)

Sr. Eduardo Péris Millán

- ◆ Diretor da Área Tecnológica de Consultoria
- ◆ Especialista em Engenharia Informática
- ◆ Mestrado em Gestão Estratégica da Informação e do Conhecimento nas Organizações
- ◆ Mestrado em Liderança e Gestão Pública
- ◆ Especialista em Gestão Pública
- ◆ Especialista em Sistemas de Computação para Smart CITIES

Sr. Oriol Pi Morell

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

Sr. José Enrique Grillo Hernández

- ◆ Desenvolvedor de aplicativos e analista tecnológico
- ◆ Desenvolvedor Sênior de Aplicativos Móveis na Globant
- ◆ Desenvolvedor Android na Plexus Tech
- ◆ Desenvolvedor Android Sênior na RoadStr
- ◆ Desenvolvedor Sênior Mobile na Avantgarde IT-Information Technology Services
- ◆ Líder de Projeto na Smartdoss
- ◆ Desenvolvido em Educatablet
- ◆ Analista de Tecnologia em Corporate Mobile Solutions
- ◆ Mestre em Engenharia de Sistemas pela Universidad Simón Bolívar

Sra. Desirée Domínguez Valderrama

- ◆ *Lead Product & Growth Strategist*
- ◆ Mestrado em Design Gráfico e Criatividade pela Escola de Negócios da Câmara de Comércio de Sevilha
- ◆ Especialista em UX/UI Designer pela CoderHouse
- ◆ Especialista em Tecnologia e Empreendedorismo



Aproveite a oportunidade para conhecer os últimos avanços nesta área e aplicá-los em sua prática diária”

08

Certificação

O Mestrado em Programação desde Zero garante, além da capacitação mais rigorosa e atualizada, o acesso a um título de Mestrado 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 **Mestrado em Programação desde Zero** 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 (**boletim oficial**). 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 de seus mecanismos de garantia de qualidade para fomentar a colaboração e a mobilidade entre alunos, pesquisadores e acadêmicos.

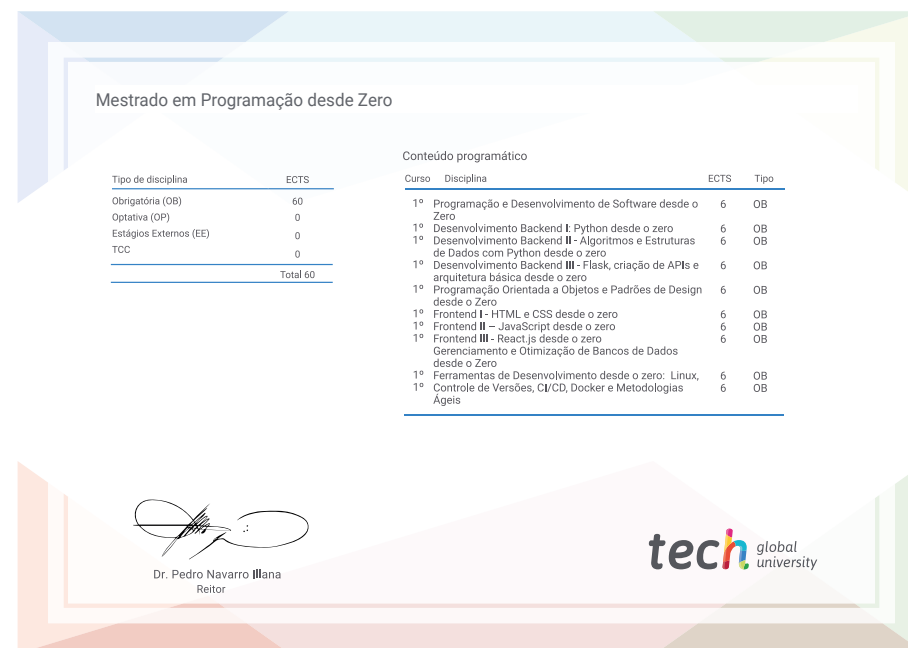
Esse título próprio da **TECH Global University**, é um programa europeu de formação contínua e atualização profissional que garante a aquisição de competências em sua área de conhecimento, conferindo um alto valor curricular ao aluno que conclui o programa.

Título: Mestrado em Programação desde Zero

Modalidade: online

Duração: 12 meses

Créditos: 60 ECTS



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

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

tech global
university

Mestrado

Programação desde Zero

- » Modalidade: Online
- » Duração: 12 meses
- » Certificação: TECH Global University
- » ECTS: 60
- » Horário: no seu próprio ritmo
- » Provas: online

Mestrado

Programação desde Zero

