

Mestrado Próprio

Inteligência Artificial em Programação



tech universidade
tecnológica

Mestrado Próprio Inteligência Artificial em Programação

- » Modalidade: online
- » Duração: 6 meses
- » Certificado: TECH Universidade Tecnológica
- » Horário: no seu próprio ritmo
- » Provas: online

Acesso ao site: www.techtute.com/br/informatica/mestrado-proprio/mestrado-proprio-inteligencia-artificial-programacao

Índice

01

Apresentação

pág. 4

02

Objetivos

pág. 8

03

Competências

pág. 18

04

Direção do curso

pág. 22

05

Estrutura e conteúdo

pág. 26

06

Metodologia

pág. 44

07

Certificado

pág. 52

01

Apresentação

A inteligência artificial (IA) surgiu como um dos pilares do mundo da programação, devido à sua capacidade de automatizar tarefas complexas, tomar decisões baseadas em dados e aprender com padrões. De fato, a IA oferece ferramentas e técnicas que possibilitam a criação de sistemas mais inteligentes e eficientes. Desde os algoritmos de aprendizado de máquina, que melhoram a precisão dos programas, até o desenvolvimento de sistemas autônomos capazes de tomar decisões em tempo real, a IA transformou radicalmente a maneira como o código é projetado e executado. Nesse contexto, a TECH elaborou um programa acadêmico que proporcionará aos alunos a oportunidade de se aprofundar nos mais recentes avanços nessa área, por meio da metodologia revolucionária *Relearning*.



“

Este programa de Inteligência Artificial em Programação lhe proporcionará uma perspectiva integral sobre como a IA afeta e melhora cada estágio do desenvolvimento de software”

A importância da Inteligência Artificial na programação está em sua capacidade de aprimorar e automatizar processos, otimizando o desenvolvimento de software e aumentando a eficiência na solução de problemas complexos. Sua capacidade de analisar grandes volumes de dados e encontrar soluções ideais levou a avanços significativos em áreas como a otimização de algoritmos, a criação de interfaces mais intuitivas e a solução de problemas complexos em diferentes áreas.

É por isso que a TECH desenvolveu este Mestrado Próprio, que surge como uma solução estratégica para ampliar as oportunidades profissionais e o crescimento da carreira dos cientistas da computação. O plano de estudos abordará a melhoria da produtividade no desenvolvimento de software por meio da IA, explorando técnicas e ferramentas para automatizar processos, otimizar códigos e acelerar a criação de aplicativos inteligentes.

Além disso, o programa se concentrará na função crucial da IA no campo do QA *Testing*, implementando algoritmos e métodos de IA para melhorar a qualidade, a precisão e a cobertura dos testes, detectando e corrigindo erros com mais eficiência. Também aprofundará a integração dos recursos de aprendizado de máquina e processamento de linguagem natural no desenvolvimento da Web, criando sites inteligentes que se adaptam e oferecem experiências personalizadas aos usuários.

Além disso, se aprofundará nas técnicas de IA para melhorar a usabilidade, a interação e a funcionalidade dos aplicativos móveis, para criar aplicativos inteligentes e preditivos que se adaptam ao comportamento do usuário. Adicionalmente, a arquitetura do software de IA será analisada em detalhes, incluindo os vários modelos que facilitarão a integração de algoritmos de IA e sua implantação em ambientes de produção.

Com o objetivo de formar especialistas em IA altamente competentes, a TECH desenvolveu um programa abrangente baseado em uma metodologia exclusiva *Relearning*. Essa abordagem permitirá que os alunos consolidem seu aprendizado por meio da repetição de conceitos-chave.

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

- ♦ O desenvolvimento de estudos de caso apresentados por especialistas em Inteligência Artificial na Programação.
- ♦ O conteúdo gráfico, esquemático e eminentemente prático do programa de estudos contém informações sobre as disciplinas que são essenciais para a prática profissional.
- ♦ Contém exercícios práticos onde o processo de autoavaliação é realizado para melhorar o aprendizado
- ♦ Destaque especial para as metodologias inovadoras
- ♦ Lições 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, fixo ou portátil, com conexão à Internet



Você liderará projetos inovadores adaptados às demandas de um mercado de tecnologia em constante evolução. Matricule-se já!

“

Você ficará imerso nos fundamentos da arquitetura de software, incluindo desempenho, escalabilidade e capacidade de manutenção, graças aos recursos multimídia mais inovadores”

O corpo docente do programa conta com profissionais do setor, que transferem toda a experiência adquirida ao longo de suas carreiras para esta capacitação, além de especialistas reconhecidos de sociedades de referência e universidades de prestígio.

O conteúdo multimídia, desenvolvido com a mais recente tecnologia educacional, permitirá ao profissional uma aprendizagem contextualizada, ou seja, realizada através de um ambiente simulado, proporcionando uma capacitação imersiva e programada para praticar diante de situações reais.

A estrutura deste programa se concentra na Aprendizagem Baseada em Problemas, através da qual o profissional deverá resolver as diferentes situações de prática profissional que surgirem ao longo do curso acadêmico. Para isso, contará com a ajuda de um inovador sistema de vídeo interativo realizado por especialistas reconhecidos.

Deseja se especializar em Inteligência Artificial? Com este programa, você dominará a otimização do processo de implantação e a integração da IA na computação em nuvem.

Você se aprofundará na integração de elementos de IA no Visual Studio Code e na otimização de código com o ChatGPT, tudo isso por meio de um programa acadêmico abrangente.



02

Objetivos

O principal objetivo desse programa será fornecer aos profissionais acesso aos conhecimentos mais avançados da área, com uma abordagem que promova sua capacitação abrangente. Assim, eles terão a oportunidade de participar de um programa acadêmico exclusivo e totalmente online. Os alunos estarão equipados com habilidades avançadas e práticas, desde o desenvolvimento de software com IA até o design e a implementação de projetos da Web e aplicativos móveis com inteligência e adaptabilidade. Com este programa, o cientista da computação transcenderá os limites da programação convencional e se tornará um participante ativo da revolução tecnológica.



“

Você abordará o ciclo de vida do teste, desde a criação de casos de teste até a detecção de bugs, graças à TECH”



Objetivos gerais

- ♦ Desenvolver habilidades para configurar e controlar ambientes de desenvolvimento eficientes, garantindo uma base sólida para a implementação de projetos de IA.
- ♦ Adquirir habilidades no planejamento, execução e automação de testes de qualidade, incorporando ferramentas de IA para a detecção e correção de bugs
- ♦ Compreender e aplicar princípios de desempenho, escalabilidade e capacidade de manutenção no projeto de sistemas de computador de grande escala
- ♦ Familiarizar-se com os padrões de design mais importantes e aplicá-los de forma eficaz na arquitetura de software





Objetivos específicos

Módulo 1. Fundamentos da Inteligência Artificial

- ♦ Analisar a evolução histórica da Inteligência Artificial, desde seus primórdios até seu estado atual, identificando os principais marcos e desenvolvimentos
- ♦ Compreender o funcionamento das redes neurais e sua aplicação em modelos de aprendizado em Inteligência Artificial
- ♦ Estudar os princípios e as aplicações dos algoritmos genéticos, analisando sua utilidade na solução de problemas complexos
- ♦ Analisar a importância dos thesauri, vocabulários e taxonomias na estruturação e no processamento de dados para sistemas de IA
- ♦ Explorar o conceito da web semântica e sua influência na organização e compreensão das informações em ambientes digitais

Módulo 2. Tipos e ciclo de vida dos dados

- ♦ Compreender os conceitos fundamentais de estatística e sua aplicação na análise de dados
- ♦ Identificar e classificar diferentes tipos de dados estatísticos, desde dados quantitativos até qualitativos
- ♦ Analisar o ciclo de vida dos dados, desde a geração até o descarte, identificando os principais estágios
- ♦ Explorar os estágios iniciais do ciclo de vida dos dados, destacando a importância do planejamento e da estrutura dos dados
- ♦ Estudar os processos de coleta de dados, incluindo a metodologia, ferramentas e canais de coleta
- ♦ Explorar o conceito de *Datawarehouse* (Armazém de dados), com ênfase nos elementos que o integram e em seu projeto
- ♦ Analisar os aspectos regulatórios relacionados à gestão de dados, cumprindo as normas de privacidade e segurança e as práticas recomendadas

Módulo 3. Os dados na Inteligência Artificial

- ♦ Dominar os fundamentos da ciência de dados, abrangendo ferramentas, tipos e fontes para a análise de informações
- ♦ Explorar o processo de transformação de dados em informações usando técnicas de mineração e visualização de dados
- ♦ Estudar a estrutura e as características dos *datasets*, compreendendo sua importância na preparação e no uso de dados para modelos de Inteligência Artificial.
- ♦ Analisar modelos supervisionados e não supervisionados, incluindo métodos e classificação
- ♦ Usar ferramentas específicas e práticas recomendadas no manejo e processamento de dados, garantindo eficiência e qualidade na implementação da Inteligência Artificial

Módulo 4. Mineração de dados. Seleção, pré-processamento e transformação

- ♦ Dominar as técnicas de inferência estatística para compreender e aplicar métodos estatísticos na mineração de dados
- ♦ Realizar uma análise exploratória detalhada de conjuntos de dados para identificar padrões, anomalias e tendências relevantes
- ♦ Desenvolver habilidades para a preparação de dados, incluindo sua limpeza, integração e formatação para uso na mineração de dados
- ♦ Implementar estratégias eficazes para lidar com valores perdidos em conjuntos de dados, aplicando métodos de imputação ou eliminação conforme o contexto
- ♦ Identificar e mitigar o ruído presente nos dados, utilizando técnicas de filtragem e suavização para melhorar a qualidade do conjunto de dados
- ♦ Abordar o pré-processamento de dados em ambientes de *Big Data*

Módulo 5. Algoritmia e Complexidade em Inteligência Artificial

- ♦ Introduzir estratégias de design de algoritmos, fornecendo uma compreensão sólida dos enfoques fundamentais para a resolução de problemas
- ♦ Analisar a eficiência e complexidade dos algoritmos, aplicando técnicas de análise para avaliar o desempenho em termos de tempo e espaço
- ♦ Estudar e aplicar algoritmos de ordenação, compreendendo seu funcionamento e comparando sua eficiência em diferentes contextos
- ♦ Explorar algoritmos baseados em árvores, compreendendo sua estrutura e aplicações
- ♦ Investigar algoritmos com *Heaps*, analisando sua implementação e utilidade na manipulação eficiente de dados
- ♦ Analisar algoritmos baseados em grafos, explorando sua aplicação na representação e solução de problemas que envolvem relações complexas
- ♦ Estudar algoritmos *Greedy*, entendendo sua lógica e aplicações na resolução de problemas de otimização
- ♦ Investigar e aplicar a técnica de *backtracking* para a resolução sistemática de problemas, analisando sua eficácia em diversos cenários

Módulo 6. Sistemas inteligentes

- ♦ Explorar a teoria de agentes, compreendendo os conceitos fundamentais de seu funcionamento e sua aplicação em Inteligência Artificial e engenharia de Software
- ♦ Estudar a representação do conhecimento, incluindo a análise de ontologias e sua aplicação na organização de informações estruturadas
- ♦ Analisar o conceito da web semântica e seu impacto na organização e recuperação de informações em ambientes digitais

- ♦ Avaliar e comparar diferentes representações do conhecimento, integrando estas para melhorar a eficácia e precisão dos sistemas inteligentes
- ♦ Estudar racionadores semânticos, sistemas baseados em conhecimento e sistemas especialistas, compreendendo sua funcionalidade e aplicações na tomada de decisões inteligentes

Módulo 7: Machine learning e mineração de dados

- ♦ Introduzir os processos de descobrimento do conhecimento e os conceitos fundamentais do machine learning
- ♦ Estudar árvores de decisão como modelos de aprendizado supervisionado, compreendendo sua estrutura e aplicações
- ♦ Avaliar classificadores usando técnicas específicas para medir seu desempenho e precisão na classificação de dados
- ♦ Estudar redes neurais, compreendendo seu funcionamento e arquitetura para resolver problemas complexos de aprendizado de máquina
- ♦ Explorar métodos bayesianos e sua aplicação no aprendizado de máquina, incluindo redes bayesianas e classificadores bayesianos
- ♦ Analisar modelos de regressão e de resposta contínua para a previsão de valores numéricos a partir de dados
- ♦ Estudar técnicas de *agrupamento* para identificar padrões e estruturas em conjuntos de dados não rotulados
- ♦ Explorar a mineração de texto e o processamento de linguagem natural (NLP), compreendendo como técnicas de aprendizado de máquina são aplicadas para analisar e compreender o texto

Módulo 8. Redes Neurais como Base do *Deep Learning*

- ♦ Dominar os fundamentos do Aprendizado Profundo, compreendendo seu papel essencial no *Deep Learning*
- ♦ Explorar as operações fundamentais em redes neurais e compreender sua aplicação na construção de modelos
- ♦ Analisar as diferentes camadas utilizadas em redes neurais e aprender] a selecioná-las adequadamente
- ♦ Compreender a combinação efetiva de camadas e operações para projetar arquiteturas de redes neurais complexas e eficientes
- ♦ Utilizar treinadores e otimizadores para ajustar e melhorar o desempenho das redes neurais
- ♦ Explorar a conexão entre neurônios biológicos e artificiais para uma compreensão mais profunda do design de modelos
- ♦ Ajustar hiperparâmetros para *oFine Tuning* de redes neurais, otimizando seu desempenho em tarefas específicas

Módulo 9. Treinamento de Redes Neurais Profundas

- ♦ Resolver problemas relacionados aos gradientes no treinamento de redes neurais profundas
- ♦ Explorar e aplicar diferentes otimizadores para melhorar a eficiência e convergência dos modelos
- ♦ Programar a taxa de aprendizagem para ajustar dinamicamente a velocidade de convergência do modelo
- ♦ Compreender e abordar o sobreajuste através de estratégias específicas durante o treinamento
- ♦ Aplicar diretrizes práticas para garantir um treinamento eficiente e eficaz de redes neurais profundas

- ♦ Implementar *Transfer Learning* como uma técnica avançada para melhorar o desempenho do modelo em tarefas específicas
- ♦ Explorar e aplicar técnicas de *Data Augmentation* para enriquecer conjuntos de dados e melhorar a generalização do modelo
- ♦ Desenvolver aplicações práticas utilizando *Transfer Learning* para resolver problemas do mundo real
- ♦ Compreender e aplicar técnicas de regularização para melhorar a generalização e evitar o sobreajuste em redes neurais profundas

Módulo 10. Personalização de Modelos e Treinamento com *TensorFlow*

- ♦ Dominar os fundamentos do *TensorFlow* e sua integração com o NumPy para um manejo eficiente de dados e cálculos
- ♦ Personalizar modelos e algoritmos de treinamento utilizando as capacidades avançadas do *TensorFlow*
- ♦ Explorar a API *tf.data* para gerenciar e manipular conjuntos de dados de maneira eficaz
- ♦ Implementar o formato *TFRecord* para armazenar e acessar grandes conjuntos de dados no *TensorFlow*
- ♦ Utilizar camadas de pré-processamento do Keras para facilitar a construção de modelos personalizados
- ♦ Explorar o projeto *TensorFlow Datasets* para acessar conjuntos de dados predefinidos e melhorar a eficiência no desenvolvimento
- ♦ Desenvolver uma aplicação de *Deep Learning* com *TensorFlow*, integrando os conhecimentos adquiridos no módulo
- ♦ Aplicar de maneira prática todos os conceitos aprendidos na construção e treinamento de modelos personalizados com *TensorFlow* em situações do mundo real

Módulo 11. *Deep Computer Vision* com Redes Neurais Convolucionais

- ♦ Compreender a arquitetura do córtex visual e sua relevância no *Deep Computer Vision*
- ♦ Explorar e aplicar camadas convolucionais para extrair características-chave de imagens
- ♦ Implementar camadas de pooling e sua utilização em modelos de *Deep Computer Vision* com Keras
- ♦ Analisar diversas arquiteturas de Redes Neurais Convolucionais (CNN) e sua aplicabilidade em diferentes contextos
- ♦ Desenvolver e implementar uma CNN ResNet utilizando a biblioteca Keras para melhorar a eficiência e desempenho do modelo
- ♦ Utilizar modelos pré-treinados de Keras para aproveitar a aprendizagem por transferência em tarefas específicas
- ♦ Aplicar técnicas de classificação e localização em ambientes de *Deep Computer Vision*
- ♦ Explorar estratégias de detecção e rastreamento de objetos usando Redes Neurais Convolucionais
- ♦ Implementar técnicas de segmentação semântica para compreender e classificar objetos em imagens de maneira detalhada

Módulo 12. Processamento de Linguagem Natural (PLN) com Redes Neurais Recorrentes (RNN) e Atenção

- ♦ Desenvolver habilidades em geração de texto usando Redes Neurais Recorrentes (RNN)
- ♦ Aplicar RNN na classificação de opiniões para análise de sentimentos em textos
- ♦ Compreender e aplicar os mecanismos de atenção em modelos de processamento de linguagem natural
- ♦ Analisar e utilizar modelos *Transformers* em tarefas específicas de NLP
- ♦ Explorar a aplicação de modelos *Transformers* no contexto de processamento de imagens e visão computacional

- ♦ Familiarizar-se com a biblioteca *Transformers* de *Hugging Face* para a implementação eficiente de modelos avançados
- ♦ Comparar diferentes bibliotecas de *Transformers* para avaliar sua adequação em tarefas específicas
- ♦ Desenvolver uma aplicação prática de NLP que integre RNN e mecanismos de atenção para resolver problemas do mundo real

Módulo 13. Autoencoders, GANs, e Modelos de Difusão

- ♦ Desenvolver representações eficientes de dados por meio de *Autoencoders*, *GANs* e Modelos de Difusão
- ♦ Realizar PCA utilizando um codificador automático linear incompleto para otimizar a representação de dados
- ♦ Implementar e compreender o funcionamento de codificadores automáticos empilhados
- ♦ Explorar e aplicar autoencoders convolucionais para representações eficientes de dados visuais
- ♦ Analisar e aplicar a eficácia de autoencoders esparsos na representação de dados
- ♦ Gerar imagens da moda do conjunto de dados MNIST utilizando *Autoencoders*
- ♦ Compreender o conceito de Redes Adversárias Generativas (*GANs*) e Modelos de Difusão
- ♦ Implementar e comparar o desempenho de Modelos de Difusão e *GANs* na geração de dados

Módulo 14. Computação bioinspirada

- ♦ Introduzir os conceitos fundamentais da computação bioinspirada
- ♦ Explorar algoritmos de adaptação social como abordagem-chave na computação bioinspirada
- ♦ Analisar estratégias de exploração-exploração do espaço em algoritmos genéticos
- ♦ Examinar modelos de computação evolutiva no contexto da otimização
- ♦ Continuar a análise detalhada de modelos de computação evolutiva
- ♦ Aplicar programação evolutiva a problemas específicos de aprendizagem
- ♦ Abordar a complexidade de problemas multiobjetivo no contexto da computação bioinspirada
- ♦ Explorar a aplicação de redes neurais no âmbito da computação bioinspirada
- ♦ Analisar a implementação e a utilidade das redes neurais na computação bioinspirada

Módulo 15. Inteligência Artificial: Estratégias e aplicações

- ♦ Desenvolver estratégias de implementação de inteligência artificial em serviços financeiros
- ♦ Analisar as implicações da inteligência artificial na prestação de serviços de saúde
- ♦ Identificar e avaliar os riscos associados ao uso de IA no campo da saúde
- ♦ Avaliar os riscos potenciais vinculados ao uso de IA na indústria
- ♦ Aplicar técnicas de inteligência artificial na indústria para melhorar a produtividade
- ♦ Projetar soluções de inteligência artificial para otimizar processos na administração pública
- ♦ Avaliar a implementação de tecnologias de IA no setor educacional
- ♦ Aplicar técnicas de inteligência artificial na silvicultura e agricultura para melhorar a produtividade
- ♦ Otimizar processos de recursos humanos por meio do uso estratégico da inteligência artificial

Módulo 16. Melhoria da Produtividade no Desenvolvimento de Software com IA

- ♦ Investigar e implementar extensões essenciais de IA no Visual Studio Code para melhorar a produtividade e facilitar o desenvolvimento de *software*
- ♦ Obter uma sólida compreensão dos conceitos básicos de IA e sua aplicação no desenvolvimento de *software*, incluindo algoritmos de aprendizado de máquina, processamento de linguagem natural, redes neurais, etc.
- ♦ Dominar a configuração de ambientes de desenvolvimento otimizados, garantindo que os alunos possam criar ambientes propícios para projetos de IA
- ♦ Aplicar técnicas específicas usando o ChatGPT para identificação e correção automáticas de possíveis melhorias no código, incentivando práticas de programação mais eficientes
- ♦ Promover a colaboração entre diferentes profissionais de programação (de programadores a engenheiros de dados e designers de experiência do usuário) para desenvolver soluções eficientes e éticas de *software* com IA

Módulo 17. Arquitetura do software para QA Testing

- ♦ Desenvolver habilidades para projetar planos de testes sólidos, cobrindo diferentes tipos de *testing* e garantindo a qualidade do *software*
- ♦ Reconhecer e analisar diferentes tipos de estruturas de *software*, como monolíticas, de microsserviços ou orientadas a serviços
- ♦ Obtenha uma visão geral abrangente dos princípios e técnicas para projetar sistemas de computador que sejam escalonáveis e capazes de lidar com grandes volumes de dados
- ♦ Aplicar conhecimentos avançados na implementação de estruturas de dados potencializadas por IA para otimizar o desempenho e a eficiência do *software*
- ♦ Desenvolver práticas de desenvolvimento seguro, concentrando-se em evitar vulnerabilidades para garantir a segurança do *software* a nível arquitetônico

Módulo 18. Projetos Web com IA

- ♦ Desenvolver habilidades abrangentes para a implementação de projetos web, desde o design do *frontend* até a otimização do *backend*, com a inclusão de elementos de IA
- ♦ Otimizar o processo de implantação de sites, incorporando técnicas e ferramentas para melhorar a velocidade e eficiência
- ♦ Integrar a IA à computação em nuvem, permitindo que os alunos criem projetos da Web altamente dimensionáveis e eficientes
- ♦ Adquirir a capacidade de identificar problemas e oportunidades específicos em projetos da Web em que a IA pode ser aplicada com eficácia, como processamento de texto, personalização, recomendação de conteúdo etc.
- ♦ Incentivar os alunos a se manterem atualizados com as últimas tendências e desenvolvimentos em IA para aplicação adequada em projetos da Web

Módulo 19. Aplicativos Móveis com IA

- ♦ Aplicar conceitos avançados de *clean architecture*, *datasources* e *repositories* para garantir uma estrutura robusta e modular em aplicativos móveis com IA
- ♦ Desenvolver habilidades para projetar telas interativas, ícones e recursos gráficos usando IA para aprimorar a experiência do usuário em aplicativos móveis
- ♦ Ampliar a configuração da estrutura do aplicativo móvel e usar o *GitHub Copilot* para acelerar o processo de desenvolvimento
- ♦ Otimizar aplicativos móveis com IA para desempenho eficiente, levando em conta o manejo de recursos e o uso de dados
- ♦ Realizar testes de qualidade em aplicativos móveis com IA, permitindo que os alunos identifiquem problemas e depurem bugs

Módulo 20. IA para QA Testing

- ♦ Dominar princípios e as técnicas para projetar sistemas de computador que sejam escalonáveis e capazes de lidar com grandes volumes de dados
- ♦ Aplicar conhecimentos avançados na implementação de estruturas de dados potencializadas por IA para otimizar o desempenho e a eficiência do *software*
- ♦ Compreender e aplicar práticas de desenvolvimento seguro, focando em evitar vulnerabilidades como a injeção, para garantir a segurança do *software* a nível arquitetônico
- ♦ Gerar testes automatizados, especialmente em ambientes da Web e móveis, integrando ferramentas de IA para melhorar a eficiência do processo
- ♦ Utilizar ferramentas avançadas de controle de qualidade com IA para detecção mais eficiente de *bugs* e melhoria contínua do *software*



Você dominará as tecnologias do futuro com este programa exclusivo 100% online. Exclusivamente com a TECH!"

03

Competências

Este programa proporcionará aos alunos uma vantagem significativa no setor de desenvolvimento de computadores, capacitando-os com habilidades específicas e atualizadas em Inteligência Artificial (IA). Os profissionais poderão não apenas projetar e desenvolver softwares avançados, mas também implementar soluções de IA de forma eficaz em uma variedade de aplicativos, desde projetos da Web e móveis até arquitetura de software em grande escala. Além disso, ao abordar a produtividade no desenvolvimento e nas práticas recomendadas de *QA Testing*, garantirá que os cientistas da computação estejam preparados para enfrentar os desafios do mundo real e se destacar em um campo em constante evolução.



“

Com este programa universitário, você poderá implementar algoritmos de IA em projetos web e aplicativos móveis”



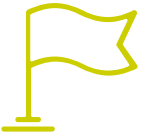
Competências gerais

- ♦ Aplicar extensões de IA em Visual Studio Code e técnicas de criação *no-code* para aumentar a eficiência no desenvolvimento de software
- ♦ Utilizar o ChatGPT para otimizar e melhorar a qualidade do código, aplicando práticas avançadas de programação
- ♦ Implementar projetos Web, desde a criação de *workspaces* até a implementação, integrando a IA, tanto no *frontend* como no *backend*
- ♦ Desenvolver aplicativos móveis com tecnologia de IA, desde a configuração do ambiente até a criação de recursos avançados e a gestão de ativos gráficos
- ♦ Aplicar conceitos avançados de armazenamento e estruturas de dados com tecnologia por IA para melhorar a eficiência e o dimensionamento dos sistemas
- ♦ Incluir práticas de desenvolvimento seguro, evitando vulnerabilidades como injeção, para garantir a integridade e a segurança do software desenvolvido



Você será capaz de projetar experiências de usuário personalizadas e intuitivas por meio da Inteligência Artificial. Matricule-se já!"





Competências específicas

- ♦ Aplicar técnicas e estratégias de IA para melhorar a eficiência no setor *retail*
- ♦ Implementar técnicas de redução de ruído usando codificadores automáticos
- ♦ Crie conjuntos de dados de treinamento de forma eficaz para tarefas de processamento de linguagem natural (NLP)
- ♦ Executar camadas de agrupamento e seu uso em modelos de *Deep Computer Vision* com Keras
- ♦ Usar funções e gráficos de *TensorFlow* para otimizar o desempenho de modelos personalizados
- ♦ Otimizar o desenvolvimento e a implementação de *chatbots* e assistentes virtuais, entendendo como eles funcionam e suas possíveis aplicações
- ♦ Dominar a reutilização de camadas pré-treinadas para otimizar e acelerar o processo de treinamento
- ♦ Construa a primeira rede neural, aplicando os conceitos aprendidos na prática
- ♦ Ativar o Perceptron de múltiplas camadas (MLP) usando a biblioteca Keras
- ♦ Aplicar técnicas de exploração e pré-processamento de dados, identificando e preparando dados para uso efetivo em modelos de aprendizado de máquina
- ♦ Investigar linguagens e softwares para a criação de ontologias, usando ferramentas específicas para o desenvolvimento de modelos semânticos
- ♦ Desenvolver técnicas de limpeza de dados para garantir a qualidade e a precisão das informações usadas em análises subsequentes
- ♦ Dominar a configuração de ambientes de desenvolvimento otimizados, garantindo que os alunos possam criar ambientes propícios para projetos de IA
- ♦ Aplicar técnicas específicas, usando o ChatGPT para identificação e correção automáticas de possíveis melhorias no código, incentivando práticas de programação mais eficientes
- ♦ Criar testes automatizados, especialmente em ambientes Web e móveis, integrando ferramentas de IA para melhorar a eficiência do processo
- ♦ Utilizar ferramentas avançadas de controle de qualidade com IA para detecção mais eficiente de bugs e melhoria contínua do software
- ♦ Integrar a IA à computação em nuvem, permitindo que os alunos criem projetos Web altamente dimensionáveis e eficientes
- ♦ Configure a estrutura do aplicativo móvel e usar o Github Copilot para acelerar o processo de desenvolvimento

04

Direção do curso

Em seu compromisso com o ensino de alto nível, a TECH selecionou cuidadosamente os professores responsáveis pelo desenvolvimento do plano de estudos para este programa. Por esse motivo, este programa acadêmico conta com um corpo docente experiente, com um histórico excepcional na aplicação de Inteligência Artificial sa uma experiência educacional de alto nível, com uma combinação exclusiva de conhecimentos apresentados em várias mídias audiovisuais, para uma integração mais eficaz e dinâmica do conhecimento.



“

Fique por dentro das últimas tendências em Inteligência Artificial aplicada à Programação com os melhores especialistas da área”

Direção



Dr. Arturo Peralta Martín-Palomino

- CEO e CTO em Prometeus Soluções Globais
- CTO em Korporate Technologies
- CTO em AI Shephers GmbH
- Consultor e assessor estratégico de negócios da Alliance Medical
- Diretor de Design e Desenvolvimento na DocPath
- Doutor em Engenharia da Computação pela Universidade de Castilla-La Mancha
- Doutorado em Economia, Negócios e Finanças pela Universidade Camilo José Cela
- Doutor em Psicologia pela Universidade de Castilla-La Mancha
- Mestrado em MBA Executivo pela Universidade Isabel I
- Mestrado em Gestão de Vendas e Marketing pela Universidade Isabel I
- Mestrado especializado em Big Data por formação em Hadoop
- Mestrado em Tecnologias de Informação Avançadas pela Universidade de Castilla-La Mancha
- Membro: Grupo de pesquisa SMILE



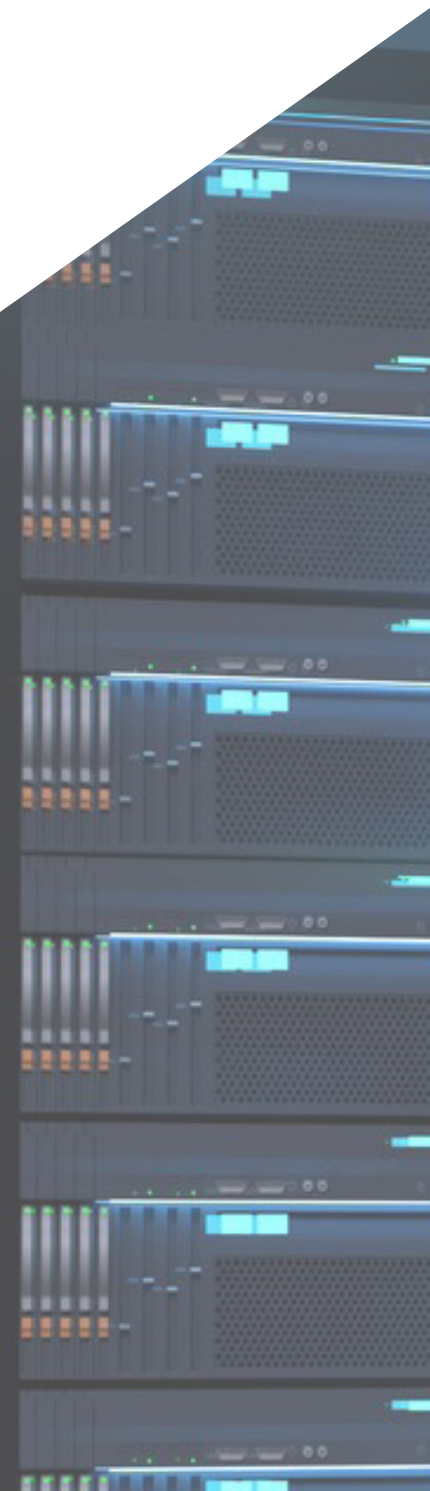
Sr. Ricardo Castellanos Herreros

- Especialista em Engenharia de Sistemas de Computação
- *Chief Technology Officer* em OWQLO
- Consultor Técnico *Freelance*
- Desenvolvedor de aplicativos móveis para eDreams, Fnac, Air Europa, Bankia, Cetelem, Banco Santander, Santillana, Groupón e Grupo Planeta
- Desenvolvedor de sites para Openbank e Banco Santander
- Curso de *Machine Learning Engineer* na Udacity
- Engenheiro Técnico em Sistemas de Computação pela Universidade de Castilla la Mancha

05

Estrutura e conteúdo

Este programa de Inteligência Artificial em Programação se destaca por sua abordagem holística, que aborda não apenas a implementação de algoritmos inteligentes, mas também o aprimoramento da produtividade no desenvolvimento de software e a aplicação de IA em áreas importantes, como *QA Testing*, projetos web, aplicativos móveis e arquitetura de software. A combinação de habilidades técnicas, ferramentas avançadas e a aplicação prática da IA em várias fases de desenvolvimento o posiciona como um programa líder, proporcionando aos profissionais uma compreensão completa e profunda da aplicação da IA na programação.



```
elif .operation == "MIRROR"
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif .operation == "MIRROR"
    mirror_mod.use_x
    mirror_mod.use
    mirror_mod

#selc
mirror
mod
```

“

Você investigará a aplicação prática da IA em projetos Web, incluindo o desenvolvimento de front-end e back-end”

Módulo 1. Fundamentos da Inteligência Artificial

- 1.1. História da inteligência artificial
 - 1.1.1. Quando começamos a falar de inteligência artificial?
 - 1.1.2. Referências no cinema
 - 1.1.3. Importância da inteligência artificial
 - 1.1.4. Tecnologias que habilitam e dão suporte à inteligência artificial
- 1.2. Inteligência Artificial em jogos
 - 1.2.1. Teoria dos jogos
 - 1.2.2. *Minimax* e Poda Alfa-Beta
 - 1.2.3. Simulação: Monte Carlo
- 1.3. Redes de neurônios
 - 1.3.1. Fundamentos biológicos
 - 1.3.2. Modelo computacional
 - 1.3.3. Redes de neurônios supervisionadas e não supervisionadas
 - 1.3.4. Perceptron simples
 - 1.3.5. Perceptrão multicamadas
- 1.4. Algoritmos genéticos
 - 1.4.1. História
 - 1.4.2. Base biológica
 - 1.4.3. Codificação de problemas
 - 1.4.4. Geração da população inicial
 - 1.4.5. Algoritmo principal e operadores genéticos
 - 1.4.6. Avaliação de indivíduos: Fitness
- 1.5. Tesouros, vocabulários, taxonomias
 - 1.5.1. Vocabulários
 - 1.5.2. Taxonomias
 - 1.5.3. Tesouros
 - 1.5.4. Ontologias
 - 1.5.5. Representação do conhecimento: web semântica
- 1.6. Web Semântica
 - 1.6.1. Especificações RDF, RDFS e OWL
 - 1.6.2. Inferência/raciocínio
 - 1.6.3. *Linked Data*

- 1.7. Sistemas especializados e DSS
 - 1.7.1. Sistemas especializados
 - 1.7.2. Sistemas de suporte à decisão
- 1.8. *Chatbots* e assistentes virtuais
 - 1.8.1. Tipos de assistentes: assistentes de voz e texto
 - 1.8.2. Partes fundamentais para o desenvolvimento de um assistente: *Intenções*, entidades e fluxo de diálogo
 - 1.8.3. Integração Web, *Slack*, Whatsapp, Facebook
 - 1.8.4. Ferramentas para o desenvolvimento de assistentes: *Dialog Flow*, *Watson Assistant*
- 1.9. Estratégia de implementação da IA
- 1.10. O futuro da inteligência artificial
 - 1.10.1. Entendemos como detectar emoções através de algoritmos
 - 1.10.2. Criar uma personalidade: linguagem, expressões e conteúdo
 - 1.10.3. Tendências da inteligência artificial
 - 1.10.4. Reflexões

Módulo 2. Tipos e Ciclo de Vida dos Dados

- 2.1. Estatísticas
 - 2.1.1. Estatísticas: descritivas e inferências
 - 2.1.2. População, amostra, individual
 - 2.1.3. Variáveis: definição, escalas de medição
- 2.2. Tipos de dados estatísticos
 - 2.2.1. De acordo com o tipo
 - 2.2.1.1. Quantitativos: dados contínuos e dados discretos
 - 2.2.1.2. Qualitativo: dados binomiais, dados nominais, dados ordinais
 - 2.2.2. De acordo com sua forma
 - 2.2.2.1. Numérico
 - 2.2.2.2. Texto
 - 2.2.2.3. Lógico
 - 2.2.3. De acordo com a fonte
 - 2.2.3.1. Primários
 - 2.2.3.2. Secundários

- 2.3. Ciclo de vida dos dados
 - 2.3.1. Etapas do ciclo
 - 2.3.2. Marcos do ciclo
 - 2.3.3. Princípios FAIR
- 2.4. Etapas iniciais do ciclo
 - 2.4.1. Definição de objetivos
 - 2.4.2. Determinação de recursos necessários
 - 2.4.3. Diagrama de Gantt
 - 2.4.4. Estruturas dos dados
- 2.5. Coleta de dados
 - 2.5.1. Metodologia de coleta
 - 2.5.2. Ferramentas de coleta
 - 2.5.3. Canais de coleta
- 2.6. Limpeza de dados
 - 2.6.1. Fases da limpeza de dados
 - 2.6.2. Qualidade dos dados
 - 2.6.3. Manipulação de dados (com R)
- 2.7. Análise de dados, interpretação e avaliação dos resultados
 - 2.7.1. Medidas estatísticas
 - 2.7.2. Índices de relação
 - 2.7.3. Mineração de dados
- 2.8. Armazém de dados (*datawarehouse*)
 - 2.8.1. Elementos que o compõem
 - 2.8.2. Desenho
 - 2.8.3. Aspectos a considerar
- 2.9. Disponibilidade de dados
 - 2.9.1. Acesso
 - 2.9.2. Utilidade
 - 2.9.3. Segurança
- 2.10. Aspectos regulamentares
 - 2.10.1. Lei Geral de Proteção de Dados
 - 2.10.2. Boas práticas
 - 2.10.3. Outros aspectos regulamentares

Módulo 3. Os dados na Inteligência Artificial

- 3.1. Ciência de dados
 - 3.1.1. Ciência de dados
 - 3.1.2. Ferramentas avançadas para o cientista de dados
- 3.2. Dados, informações e conhecimentos
 - 3.2.1. Dados, informações e conhecimentos
 - 3.2.2. Tipos de dados
 - 3.2.3. Fontes de dados
- 3.3. De dados a informações
 - 3.3.1. Análise de dados
 - 3.3.2. Tipos de análise
 - 3.3.3. Extração de informações de um *Dataset*
- 3.4. Extração de informações através da visualização
 - 3.4.1. A visualização como ferramenta de análise
 - 3.4.2. Métodos de visualização
 - 3.4.3. Visualização de um conjunto de dados
- 3.5. Qualidade dos dados
 - 3.5.1. Dados de qualidade
 - 3.5.2. Limpeza de dados
 - 3.5.3. Pré-processamento básico de dados
- 3.6. *Dataset*
 - 3.6.1. Enriquecimento do *Dataset*
 - 3.6.2. A maldição da dimensionalidade
 - 3.6.3. Modificação de nosso conjunto de dados
- 3.7. Desequilíbrio
 - 3.7.1. Desequilíbrio de classes
 - 3.7.2. Técnicas de mitigação do desequilíbrio
 - 3.7.3. Equilíbrio de um *Dataset*
- 3.8. Modelos não supervisionados
 - 3.8.1. Modelo não supervisionado
 - 3.8.2. Métodos
 - 3.8.3. Classificação com modelos não supervisionados

- 3.9. Modelos supervisionados
 - 3.9.1. Modelo supervisionado
 - 3.9.2. Métodos
 - 3.9.3. Classificação com modelos supervisionados
- 3.10. Ferramentas e práticas recomendadas
 - 3.10.1. Práticas recomendadas para um cientista de dados
 - 3.10.2. O melhor modelo
 - 3.10.3. Ferramentas úteis

Módulo 4. Mineração de dados Seleção, pré-processamento e transformação

- 4.1. Inferência estatística
 - 4.1.1. Estatística descritiva vs. Inferência estatística
 - 4.1.2. Procedimentos paramétricos
 - 4.1.3. Procedimentos paramétricos
- 4.2. Análise exploratória
 - 4.2.1. Análise descritiva
 - 4.2.2. Visualização
 - 4.2.3. Preparação dos dados
- 4.3. Preparação dos dados
 - 4.3.1. Integração e limpeza de dados
 - 4.3.2. Normalização de dados
 - 4.3.3. Transformando atributos
- 4.4. Os Valores Perdidos
 - 4.4.1. Tratamento de valores perdidos
 - 4.4.2. Métodos de imputação de máxima verosimilhança
 - 4.4.3. Imputação de valores perdidos utilizando a aprendizado de máquinas
- 4.5. O ruído nos dados
 - 4.5.1. Classes de ruído e seus atributos
 - 4.5.2. Filtragem de ruídos
 - 4.5.3. O efeito do ruído
- 4.6. A maldição da dimensionalidade
 - 4.6.1. *Oversampling*
 - 4.6.2. *Undersampling*
 - 4.6.3. Redução de dados multidimensionais

- 4.7. De atributos contínuos a discretos
 - 4.7.1. Dados contínuos versus discretos
 - 4.7.2. Processo de discretização
- 4.8. Os dados
 - 4.8.1. Seleção de dados
 - 4.8.2. Perspectivas e critérios de seleção
 - 4.8.3. Métodos de seleção
- 4.9. Seleção de Instâncias
 - 4.9.1. Métodos para seleção de instâncias
 - 4.9.2. Seleção de protótipos
 - 4.9.3. Métodos avançados para seleção de instâncias
- 4.10. Pré-processamento de dados em ambientes Big Data

Módulo 5. Algoritmia e Complexidade em Inteligência Artificial

- 5.1. Introdução às Estratégias de design de algoritmos
 - 5.1.1. Recursividade
 - 5.1.2. Divisão e conquista
 - 5.1.3. Outras estratégias
- 5.2. Eficiência e análise de algoritmos
 - 5.2.1. Medidas de eficiência
 - 5.2.2. Como medir o tamanho da entrada
 - 5.2.3. Como medir o tempo de execução
 - 5.2.4. Melhor, pior e médio caso
 - 5.2.5. Notação assintótica
 - 5.2.6. Critérios de análise matemática para algoritmos não recursivos
 - 5.2.7. Análise matemática de algoritmos recursivos
 - 5.2.8. Análise empírica de algoritmos
- 5.3. Algoritmos de ordenação
 - 5.3.1. Conceito de ordenação
 - 5.3.2. Ordenação bolha (Bubble sort)
 - 5.3.3. Ordenação por seleção (Selection sort)
 - 5.3.4. Ordenação por inserção (Insertion Sort)
 - 5.3.5. Ordenação por mistura (*merge_sort*)
 - 5.3.6. Classificação rápida (*quick_sort*)

- 5.4. Algoritmos com árvores
 - 5.4.1. Conceito de árvore
 - 5.4.2. Árvores binárias
 - 5.4.3. Caminhos de árvores
 - 5.4.4. Representar expressões
 - 5.4.5. Árvores binárias ordenadas
 - 5.4.6. Árvores binárias balanceadas
- 5.5. Algoritmos com *Heaps*
 - 5.5.1. Os *Heaps*
 - 5.5.2. O algoritmo *Heapsort*
 - 5.5.3. As filas de prioridade
- 5.6. Algoritmos com grafos
 - 5.6.1. Representação
 - 5.6.2. Caminho em largura
 - 5.6.3. Caminho em profundidade
 - 5.6.4. Ordenação topológica
- 5.7. Algoritmos *Greedy*
 - 5.7.1. A estratégia *Greedy*
 - 5.7.2. Elementos da estratégia *Greedy*
 - 5.7.3. Conversor de moedas
 - 5.7.4. Problema do Caixeiro Viajante
 - 5.7.5. Problema da mochila
- 5.8. Busca do caminho mínimo
 - 5.8.1. O problema do caminho mínimo
 - 5.8.2. Arco e ciclos negativos
 - 5.8.3. Algoritmo de Dijkstra
- 5.9. Algoritmos *Greedy* sobre Grafos
 - 5.9.1. A árvore de extensão mínima
 - 5.9.2. O algoritmo de Prim (algoritmo guloso)
 - 5.9.3. O algoritmo de Kruskal
 - 5.9.4. Análise de complexidade
- 5.10. *Backtracking*
 - 5.10.1. O *Backtracking*
 - 5.10.2. Técnicas alternativas

Módulo 6. Sistemas inteligentes

- 6.1. Teoria de Agentes
 - 6.1.1. História do conceito
 - 6.1.2. Definição de agente
 - 6.1.3. Agentes em Inteligência Artificial
 - 6.1.4. Agentes em Engenharia de Software
- 6.2. Arquiteturas de agentes
 - 6.2.1. O processo de raciocínio de um agente
 - 6.2.2. Agentes reativos
 - 6.2.3. Agentes dedutivos
 - 6.2.4. Agentes híbridos
 - 6.2.5. Comparativa
- 6.3. Informação e conhecimento
 - 6.3.1. Distinção entre dados, informações e conhecimentos
 - 6.3.2. Avaliação da qualidade dos dados
 - 6.3.3. Métodos de captura de dados
 - 6.3.4. Métodos de aquisição de informações
 - 6.3.5. Métodos de aquisição de conhecimentos
- 6.4. Representação do conhecimento
 - 6.4.1. A importância da representação do conhecimento
 - 6.4.2. Definição da representação do conhecimento através de suas funções
 - 6.4.3. Características de uma representação do conhecimento
- 6.5. Ontologias
 - 6.5.1. Introdução aos metadados
 - 6.5.2. Conceito filosófico de ontologia
 - 6.5.3. Conceito informático de ontologia
 - 6.5.4. Ontologias de domínio e ontologias de nível superior
 - 6.5.5. Como construir uma ontologia?

- 6.6. Linguagens para ontologias e software para criação de ontologias
 - 6.6.1. Tríade RDF, Turtle e N
 - 6.6.2. RDF Schema
 - 6.6.3. OWL
 - 6.6.4. SPARQL
 - 6.6.5. Introdução às diferentes ferramentas para a criação de ontologias
 - 6.6.6. Instalação e uso do *Protégé*
- 6.7. Web Semântica
 - 6.7.1. O estado atual e futuro da segurança ad web semântica
 - 6.7.2. Aplicações da web semântica
- 6.8. Outros modelos de representação do conhecimento
 - 6.8.1. Vocabulários
 - 6.8.2. Visão global
 - 6.8.3. Taxonomias
 - 6.8.4. Tesouros
 - 6.8.5. Folksonomias
 - 6.8.6. Comparativa
 - 6.8.7. Mapas mentais
- 6.9. Avaliação e integração das representações do conhecimento
 - 6.9.1. Lógica de ordem zero
 - 6.9.2. Lógica de primeira ordem
 - 6.9.3. Lógica descritiva
 - 6.9.4. Relação entre diferentes tipos de lógica
 - 6.9.5. *Prolog*: programação baseada em lógica de primeira ordem
- 6.10. Raciocinadores Semânticos, Sistemas Baseados no Conhecimento e Sistemas Especialistas
 - 6.10.1. Conceito de raciocinador
 - 6.10.2. Aplicações de um raciocinador
 - 6.10.3. Sistemas baseados no conhecimento
 - 6.10.4. MYCIN, história dos Sistemas Especialistas
 - 6.10.5. Elementos e Arquitetura de Sistemas Especialistas
 - 6.10.6. Criação de Sistemas Especialistas

Módulo 7. Machine learning e mineração de dados

- 7.1. Introdução aos processos de descoberta de conhecimento e conceitos básicos de machine learning
 - 7.1.1. Conceitos-chave dos processos de descoberta do conhecimento
 - 7.1.2. Perspectiva histórica dos processos de descoberta do conhecimento
 - 7.1.3. Fases dos processos de descoberta do conhecimento
 - 7.1.4. Técnicas utilizadas nos processos de descoberta do conhecimento
 - 7.1.5. Características dos bons modelos de machine learning
 - 7.1.6. Tipos de informações de machine learning
 - 7.1.7. Noções básicas de aprendizagem
 - 7.1.8. Noções básicas de aprendizagem não supervisionada
- 7.2. Exploração e pré-processamento de dados
 - 7.2.1. Processamento de dados
 - 7.2.2. Tratamento de dados no fluxo de análise de dados
 - 7.2.3. Tipos de dados
 - 7.2.4. Transformações de dados
 - 7.2.5. Visualização e exploração de variáveis contínuas
 - 7.2.6. Visualização e exploração de variáveis categóricas
 - 7.2.7. Medidas de correlação
 - 7.2.8. Representações gráficas mais comuns
 - 7.2.9. Introdução à análise multivariada e redução da dimensionalidade
- 7.3. Árvore de decisão
 - 7.3.1. Algoritmo ID
 - 7.3.2. Algoritmo
 - 7.3.3. Overtraining e poda
 - 7.3.4. Análise de resultados
- 7.4. Avaliação de classificadores
 - 7.4.1. Matrizes de confusão
 - 7.4.2. Matrizes de avaliação numérica
 - 7.4.3. Estatístico de Kappa
 - 7.4.4. Curvas Roc

- 7.5. Regras de classificação
 - 7.5.1. Medidas de avaliação de regras
 - 7.5.2. Introdução à representação gráfica
 - 7.5.3. Algoritmo de sobreposição sequencial
- 7.6. Redes Neurais
 - 7.6.1. Conceitos básicos
 - 7.6.2. Redes de neurônios simples
 - 7.6.3. Algoritmo de *backpropagation*
 - 7.6.4. Introdução às redes neurais recorrentes
- 7.7. Métodos bayesianos
 - 7.7.1. Conceitos básicas de probabilidade
 - 7.7.2. Teorema de Bayes
 - 7.7.3. Naive Bayes
 - 7.7.4. Introdução às redes bayesianas
- 7.8. Modelos de regressão e de resposta contínua
 - 7.8.1. Regressão linear simples
 - 7.8.2. Regressão Linear Múltipla
 - 7.8.3. Regressão logística
 - 7.8.4. Árvores de regressão
 - 7.8.5. Introdução às Máquinas de Vetores de Suporte (SVM)
 - 7.8.6. Medidas de bondade do ajuste
- 7.9. *Clustering*
 - 7.9.1. Conceitos básicos
 - 7.9.2. *Clustering* hierárquico
 - 7.9.3. Métodos probabilísticos
 - 7.9.4. Algoritmo EM
 - 7.9.5. Método *B-Cubed*
 - 7.9.6. Métodos implícitos
- 7.10. Mineração de texto e processamento de linguagem natural (PNL)
 - 7.10.1. Conceitos básicos
 - 7.10.2. Criação do corpus
 - 7.10.3. Análise descritiva
 - 7.10.4. Introdução à análise de sentimentos

Módulo 8. Redes Neurais como Base do *Deep Learning*

- 8.1. Aprendizagem profunda
 - 8.1.1. Tipos de aprendizagem profunda
 - 8.1.2. Aplicativos de aprendizagem profunda
 - 8.1.3. Vantagens e desvantagens da aprendizagem profunda
- 8.2. Operações
 - 8.2.1. Soma
 - 8.2.2. Produtos
 - 8.2.3. Transferência
- 8.3. Camadas
 - 8.3.1. Camada de entrada
 - 8.3.2. Camada oculta
 - 8.3.3. Camada de saída
- 8.4. União de Camadas e Operações
 - 8.4.1. Design de arquiteturas
 - 8.4.2. Conexão entre camadas
 - 8.4.3. Propagação para frente
- 8.5. Construção da primeira rede neural
 - 8.5.1. Design da rede
 - 8.5.2. Definição dos pesos
 - 8.5.3. Treinamento da rede
- 8.6. Treinador e Otimizador
 - 8.6.1. Seleção do otimizador
 - 8.6.2. Definição de uma função de perda
 - 8.6.3. Definição de uma métrica
- 8.7. Aplicação dos princípios das redes neurais
 - 8.7.1. Funções de ativação
 - 8.7.2. Retropropagação
 - 8.7.3. Ajuste dos parâmetros
- 8.8. Dos neurônios biológicos para os artificiais
 - 8.8.1. Funcionamento de um neurônio biológico
 - 8.8.2. Transferência de conhecimento para os neurônios artificiais
 - 8.8.3. Estabelecimento de relações entre ambos

- 8.9. Implementação de MLP (Perceptron multicamadas) com Keras
 - 8.9.1. Definição da estrutura da rede
 - 8.9.2. Compilação do modelo
 - 8.9.3. Treinamento do modelo
- 8.10. Hiperparâmetros de *Fine tuning* de Redes Neurais
 - 8.10.1. Seleção da função de ativação
 - 8.10.2. Estabelecer o *learning rate*
 - 8.10.3. Ajuste dos pesos

Módulo 9. Treinamento de Redes Neurais Profundas

- 9.1. Problemas de Gradientes
 - 9.1.1. Técnicas de otimização de gradiente
 - 9.1.2. Gradientes Estocásticos
 - 9.1.3. Técnicas de inicialização de pesos
- 9.2. Reutilização de camadas pré-treinadas
 - 9.2.1. Treinamento de transferência de aprendizagem
 - 9.2.2. Extração de características
 - 9.2.3. Aprendizado profundo
- 9.3. Otimizadores
 - 9.3.1. Otimizadores de descida de gradiente estocástico
 - 9.3.2. Otimizadores Adam e *RMSprop*
 - 9.3.3. Otimizadores de momento
- 9.4. Programação da taxa de aprendizagem
 - 9.4.1. Controle de taxa de aprendizagem automática
 - 9.4.2. Ciclos de aprendizagem
 - 9.4.3. Termos de suavização
- 9.5. Sobreajuste
 - 9.5.1. Validação cruzada
 - 9.5.2. Regularização
 - 9.5.3. Métricas de avaliação
- 9.6. Diretrizes práticas
 - 9.6.1. Design de modelos
 - 9.6.2. Seleção de métricas e parâmetros de avaliação
 - 9.6.3. Testes de hipóteses

- 9.7. *Transfer Learning*
 - 9.7.1. Treinamento de transferência de aprendizagem
 - 9.7.2. Extração de características
 - 9.7.3. Aprendizado profundo
- 9.8. *Data Augmentation*
 - 9.8.1. Transformações de imagem
 - 9.8.2. Geração de dados sintéticos
 - 9.8.3. Transformação de texto
- 9.9. Aplicação prática de *Transfer Learning*
 - 9.9.1. Treinamento de transferência de aprendizagem
 - 9.9.2. Extração de características
 - 9.9.3. Aprendizado profundo
- 9.10. Regularização
 - 9.10.1. L e L
 - 9.10.2. Regularização por máxima entropia
 - 9.10.3. *Dropout*

Módulo 10. Personalização de Modelos e Treinamento com *TensorFlow*

- 10.1. *TensorFlow*
 - 10.1.1. Uso da biblioteca *TensorFlow*
 - 10.1.2. Treinamento de modelos com *TensorFlow*
 - 10.1.3. Operações com gráficos no *TensorFlow*
- 10.2. *TensorFlow* e NumPy
 - 10.2.1. Ambiente computacional NumPy para *TensorFlow*
 - 10.2.2. Utilização de arrays NumPy com *TensorFlow*
 - 10.2.3. Operações NumPy para gráficos do *TensorFlow*
- 10.3. Personalização de modelos e algoritmos de treinamento
 - 10.3.1. Construção de modelos personalizados com *TensorFlow*
 - 10.3.2. Gestão de parâmetros de treinamento
 - 10.3.3. Utilização de técnicas de otimização para treinamento
- 10.4. Funções e gráficos do *TensorFlow*
 - 10.4.1. Funções com *TensorFlow*
 - 10.4.2. Utilização de gráficos para treinamento de modelos
 - 10.4.3. Otimização de gráficos com operações do *TensorFlow*

- 10.5. Carregamento e pré-processamento de dados com *TensorFlow*
 - 10.5.1. Carregamento de conjuntos de dados com *TensorFlow*
 - 10.5.2. Pré-processamento de dados com *TensorFlow*
 - 10.5.3. Utilização de ferramentas do *TensorFlow* para manipulação de dados
 - 10.6. A API *tfdata*
 - 10.6.1. Utilização da API *tfdata* para processamento de dados
 - 10.6.2. Construção de fluxos de dados com *tfdata*
 - 10.6.3. Uso da API *tfdata* para treinamento de modelos
 - 10.7. O formato *TFRecord*
 - 10.7.1. Utilização da API *TFRecord* para serialização de dados
 - 10.7.2. Carregamento de arquivos *TFRecord* com *TensorFlow*
 - 10.7.3. Utilização de arquivos *TFRecord* para treinamento de modelos
 - 10.8. Camadas de pré-processamento do Keras
 - 10.8.1. Utilização da API de pré-processamento do Keras
 - 10.8.2. Construção de *pipelines* de pré-processamento com Keras
 - 10.8.3. Uso da API de pré-processamento do Keras para treinamento de modelos
 - 10.9. Projeto *TensorFlow Datasets*
 - 10.9.1. Utilização do *TensorFlow Datasets* para carregamento de dados
 - 10.9.2. Pré-processamento de dados com o *TensorFlow Datasets*
 - 10.9.3. Uso do *TensorFlow Datasets* para treinamento de modelos
 - 10.10. Construção de uma aplicação de Deep *Learning* com *TensorFlow*
 - 10.10.1. Aplicação Prática
 - 10.10.2. Construção de uma aplicação de Deep *Learning* com *TensorFlow*
 - 10.10.3. Treinamento de um modelo com *TensorFlow*
 - 10.10.4. Utilização da aplicação para previsão de resultados
- Módulo 11. Deep Computer Vision com Redes Neurais Convolucionais**
- 11.1. A Arquitetura *do Visual Cortex*
 - 11.1.1. Funções do córtex visual
 - 11.1.2. Teorias da visão computacional
 - 11.1.3. Modelos de processamento de imagens
 - 11.2. Camadas convolucionais
 - 11.2.1. Reutilização de pesos na convolução
 - 11.2.2. Convolução
 - 11.2.3. Funções de ativação
 - 11.3. Camadas de agrupamento e implementação de camadas de agrupamento com o Keras
 - 11.3.1. *Agrupamento e Deslocamento*
 - 11.3.2. *Flattening*
 - 11.3.3. Tipos de *Pooling*
 - 11.4. Arquiteturas CNN
 - 11.4.1. Arquitetura VGG
 - 11.4.2. Arquitetura *AlexNet*
 - 11.4.3. Arquitetura *ResNet*
 - 11.5. Implementação de uma CNN *ResNet*- usando o Keras
 - 11.5.1. Inicialização de pesos
 - 11.5.2. Definição da camada de entrada
 - 11.5.3. Definição da saída
 - 11.6. Uso de modelos pré-treinados do Keras
 - 11.6.1. Características dos modelos pré-treinados
 - 11.6.2. Usos dos modelos pré-treinados
 - 11.6.3. Vantagens dos modelos pré-treinados
 - 11.7. Modelos pré-treinados para aprendizado por transferência
 - 11.7.1. Aprendizagem por transferência
 - 11.7.2. Processo de aprendizagem por transferência
 - 11.7.3. Vantagens do aprendizagem por transferência
 - 11.8. Classificação e localização em *Deep Computer Vision*
 - 11.8.1. Classificação de imagens
 - 11.8.2. Localização de objetos em imagens
 - 11.8.3. Detecção de objetos
 - 11.9. Detecção e rastreamento de objetos
 - 11.9.1. Métodos de detecção de objetos
 - 11.9.2. Algoritmos de rastreamento de objetos
 - 11.9.3. Técnicas de rastreamento e localização

- 11.10. Segmentação semântica
 - 11.10.1. Aprendizagem profunda para segmentação semântica
 - 11.10.2. Detecção de bordas
 - 11.10.3. Métodos de segmentação baseados em regras

Módulo 12. Processamento de Linguagem Natural (PLN) com Redes Neurais Recorrentes (RNN) e Atenção

- 12.1. Geração de texto usando RNN
 - 12.1.1. Treinamento de uma RNN para geração de texto
 - 12.1.2. Geração de linguagem natural com RNN
 - 12.1.3. Aplicações de geração de texto com RNN
- 12.2. Criação do conjunto de dados de treinamento
 - 12.2.1. Preparação dos dados para treinamento de uma RNN
 - 12.2.2. Armazenamento do conjunto de dados de treinamento
 - 12.2.3. Limpeza e transformação dos dados
 - 12.2.4. Análise de sentimento
- 12.3. Classificação de opiniões com RNN
 - 12.3.1. Detecção de temas nos comentários
 - 12.3.2. Análise de sentimento com algoritmos de aprendizagem profunda
- 12.4. Rede codificador-decodificador para tradução automática neural
 - 12.4.1. Treinamento de uma RNN para tradução automática
 - 12.4.2. Uso de uma rede *encoder-decoder* para tradução automática
 - 12.4.3. Aumento da precisão da tradução automática com RNN
- 12.5. Mecanismos de atenção
 - 12.5.1. Aplicação de mecanismos de atenção em RNN
 - 12.5.2. Uso de mecanismos de atenção para melhorar a precisão dos modelos
 - 12.5.3. Vantagens dos mecanismos de atenção em redes neurais
- 12.6. Modelos *Transformers*
 - 12.6.1. Uso de modelos *Transformers* para processamento de linguagem natural
 - 12.6.2. Aplicação de modelos *Transformers* para visão
 - 12.6.3. Vantagens dos modelos *Transformers*

- 12.7. *Transformers* para visão
 - 12.7.1. Uso de modelos *Transformers* para visão
 - 12.7.2. Processamento de dados Imagem
 - 12.7.3. Treinamento de modelos *Transformers* para visão
- 12.8. Biblioteca *Transformers* de *Hugging Face*
 - 12.8.1. Uso da Biblioteca *Transformers* de *Hugging Face*
 - 12.8.2. Aplicação da Biblioteca *Transformers* de *Hugging Face*
 - 12.8.3. Vantagens da Biblioteca *Transformers* de *Hugging Face*
- 12.9. Outras bibliotecas *Transformers*. Comparativa
 - 12.9.1. Comparação entre as diferentes bibliotecas *Transformers*
 - 12.9.2. Uso das diferentes bibliotecas *Transformers*
 - 12.9.3. Vantagens das diferentes bibliotecas *Transformers*
- 12.10. Desenvolvimento de um aplicativo de PLN com RNN e atenção. Aplicação Prática
 - 12.10.1. Desenvolvimento de um aplicativo de processamento de linguagem natural com RNN e atenção
 - 12.10.2. Uso de RNN, mecanismos de atenção e modelos *Transformers* no aplicativo
 - 12.10.3. Avaliação da aplicação prática

Módulo 13. Autoencoders, GANs e Modelos de Difusão

- 13.1. Representação de dados eficientes
 - 13.1.1. Redução da dimensionalidade
 - 13.1.2. Aprendizado profundo
 - 13.1.3. Representações compactas
- 13.2. Realização de PCA com um codificador automático linear incompleto
 - 13.2.1. Processo de treinamento
 - 13.2.2. Implementação em Python
 - 13.2.3. Utilização de dados de teste
- 13.3. Codificadores automáticos empilhados
 - 13.3.1. Redes neurais profundas
 - 13.3.2. Construção de arquiteturas de codificação
 - 13.3.3. Uso da regularização
- 13.4. Autoencoders convolucionais
 - 13.4.1. Design de modelos convolucionais
 - 13.4.2. Treinamento de modelos convolucionais
 - 13.4.3. Avaliação de resultados

- 13.5. Eliminação de ruído de codificadores automáticos
 - 13.5.1. Aplicação de filtros
 - 13.5.2. Design de modelos de codificação
 - 13.5.3. Uso de técnicas de regularização
- 13.6. Codificadores automáticos dispersos
 - 13.6.1. Aumentando a eficiência da codificação
 - 13.6.2. Minimizando o número de parâmetros
 - 13.6.3. Utilização de técnicas de regularização
- 13.7. Codificadores automáticos variacionais
 - 13.7.1. Utilização de otimização variacional
 - 13.7.2. Aprendizagem profunda não supervisionada
 - 13.7.3. Representações latentes profundas
- 13.8. Geração de imagens MNIST de moda
 - 13.8.1. Reconhecimento de padrões
 - 13.8.2. Geração de imagens
 - 13.8.3. Treinamento de Redes Neurais Profundas
- 13.9. Redes adversárias generativas e modelos de difusão
 - 13.9.1. Geração de conteúdo a partir de imagens
 - 13.9.2. Modelagem de distribuições de dados
 - 13.9.3. Uso de redes adversárias
- 13.10. Implementação dos Modelos
 - 13.10.1. Aplicação Prática
 - 13.10.2. Implementação dos modelos
 - 13.10.3. Uso de dados reais
 - 13.10.4. Avaliação de resultados

Módulo 14. Computação bioinspirada

- 14.1. Introdução à computação bioinspirada
 - 14.1.1. Introdução à computação bioinspirada
- 14.2. Algoritmos de adaptação social
 - 14.2.1. Computação bioinspirada baseada em colônias de formigas
 - 14.2.2. Variantes dos algoritmos das colônias de formigas
 - 14.2.3. Computação baseada em nuvens de partículas

- 14.3. Algoritmos genéticos
 - 14.3.1. Estrutura geral
 - 14.3.2. Implementações dos principais operadores
- 14.4. Estratégias de exploração do espaço para algoritmos genéticos
 - 14.4.1. Algoritmo CHC
 - 14.4.2. Problemas multimodais
- 14.5. Modelos de computação evolutiva (I)
 - 14.5.1. Estratégias evolutivas
 - 14.5.2. Programação evolutiva
 - 14.5.3. Algoritmos baseados na evolução diferencial
- 14.6. Modelos de computação evolutiva (II)
 - 14.6.1. Modelos de evolução baseados em estimativas de distribuições (EDA)
 - 14.6.2. Programação genética
- 14.7. Programação evolutiva aplicada a dificuldades de aprendizagem
 - 14.7.1. Aprendizagem baseada em regras
 - 14.7.2. Métodos evolutivos em problemas de seleção de instâncias
- 14.8. Problemas multiobjetivo
 - 14.8.1. Conceito de dominância
 - 14.8.2. Aplicação de algoritmos evolutivos a problema multiobjetivo
- 14.9. Redes neurais (I)
 - 14.9.1. Introdução às redes neurais
 - 14.9.2. Exemplo prático com redes neurais
- 14.10. Redes neurais (II)
 - 14.10.1. Casos de uso de redes neurais na pesquisa médica
 - 14.10.2. Casos de uso de redes neurais em economia
 - 14.10.3. Casos de uso de redes neurais em visão artificial

Módulo 15. Inteligência Artificial: Estratégias e aplicações

- 15.1. Serviços financeiros
 - 15.1.1. As implicações da Inteligência Artificial (IA) nos serviços financeiros Oportunidades e desafios
 - 15.1.2. Casos de uso
 - 15.1.3. Riscos potenciais relacionados ao uso de IA
 - 15.1.4. Potenciais desenvolvimentos/utilizações futuras da IA

- 15.2. Implicações da Inteligência Artificial no serviço de saúde
 - 15.2.1. Implicações da IA no setor da saúde Oportunidades e desafios
 - 15.2.2. Casos de uso
- 15.3. Riscos relacionados com o uso de IA no serviço de saúde
 - 15.3.1. Riscos potenciais relacionados ao uso de IA
 - 15.3.2. Potenciais desenvolvimentos/utilizações futuras da IA
- 15.4. *Retail*
 - 15.4.1. Participação da IA no *Varejo* Oportunidades e desafios
 - 15.4.2. Casos de uso
 - 15.4.3. Riscos potenciais relacionados ao uso de IA
 - 15.4.4. Potenciais desenvolvimentos/utilizações futuras da IA
- 15.5. Indústria
 - 15.5.1. Participação da IA na Indústria Oportunidades e desafios
 - 15.5.2. Casos de uso
- 15.6. Riscos potenciais relacionados ao uso de IA na indústria
 - 15.6.1. Casos de uso
 - 15.6.2. Riscos potenciais relacionados ao uso de IA
 - 15.6.3. Potenciais desenvolvimentos/utilizações futuras da IA
- 15.7. Administração pública
 - 15.7.1. Participação da IA na administração pública Oportunidades e desafios
 - 15.7.2. Casos de uso
 - 15.7.3. Riscos potenciais relacionados ao uso de IA
 - 15.7.4. Potenciais desenvolvimentos/utilizações futuras da IA
- 15.8. Educação
 - 15.8.1. Participação da IA na educação Oportunidades e desafios
 - 15.8.2. Casos de uso
 - 15.8.3. Riscos potenciais relacionados ao uso de IA
 - 15.8.4. Potenciais desenvolvimentos/utilizações futuras da IA
- 15.9. Silvicultura e agricultura
 - 15.9.1. Participação da IA na silvicultura e na agricultura Oportunidades e desafios
 - 15.9.2. Casos de uso
 - 15.9.3. Riscos potenciais relacionados ao uso de IA
 - 15.9.4. Potenciais desenvolvimentos/utilizações futuras da IA

- 15.10 Recursos Humanos
 - 15.10.1. Implicações da IA para os recursos humanos. Oportunidades e desafios
 - 15.10.2. Casos de uso
 - 15.10.3. Riscos potenciais relacionados ao uso de IA
 - 15.10.4. Potenciais desenvolvimentos/utilizações futuras da IA

Módulo 16. Melhoria da Produtividade no Desenvolvimento de Software com IA

- 16.1. Preparar um ambiente de desenvolvimento adequado
 - 16.1.1. Seleção de ferramentas essenciais para o desenvolvimento de IA
 - 16.1.2. Configuração das ferramentas escolhidas
 - 16.1.3. Implementação de pipelines de CI/CD adaptados a projetos de IA
 - 16.1.4. Gestão eficiente de dependências e versões em ambientes de desenvolvimento
- 16.2. Extensões de IA indispensáveis para o Visual Studio Code
 - 16.2.1. Explorando e selecionando extensões de IA para o Visual Studio Code
 - 16.2.2. Integração de ferramentas de análise estática e dinâmica na IDE
 - 16.2.3. Automação de tarefas repetitivas com extensões específicas
 - 16.2.4. Personalização do ambiente de desenvolvimento para aumentar a eficiência
- 16.3. Projeto *no-code* de interfaces de usuário com elementos de IA
 - 16.3.1. Princípios de design *No-code* e sua aplicação em interfaces de usuário
 - 16.3.2. Incorporação de elementos de IA no design da interface visual
 - 16.3.3. Ferramentas e plataformas para a criação *No-code* de interfaces inteligentes
 - 16.3.4. Avaliação e aprimoramento contínuo de interfaces *No-code* com IA
- 16.4. Otimização de código usando o ChatGPT
 - 16.4.1. Identificar códigos duplicados
 - 16.4.2. Refatoração
 - 16.4.3. Criar códigos legíveis
 - 16.4.4. Entender o que o código faz
 - 16.4.5. Nomes de variáveis e funções aprimorados
 - 16.4.6. Criação de documentação automática
- 16.5. Gestão de repositórios com IA
 - 16.5.1. Automatização de processos de controle de versão com técnicas de IA
 - 16.5.2. Detecção de conflitos e resolução automática em ambientes colaborativos
 - 16.5.3. Análise preditiva de alterações e tendências em repositórios de código
 - 16.5.4. Aprimoramentos na organização e categorização de repositórios usando IA

- 16.6. Integração da IA na gestão de bancos de dados
 - 16.6.1. Otimização de consultas e desempenho usando técnicas de IA
 - 16.6.2. Análise preditiva dos padrões de acesso ao banco de dados
 - 16.6.3. Implementação de sistemas de recomendação para otimizar a estrutura do banco de dados
 - 16.6.4. Monitoramento proativo e detecção de possíveis problemas no banco de dados
- 16.7. Localização de falhas e criação de testes unitários com IA
 - 16.7.1. Geração automática de casos de teste usando técnicas de IA
 - 16.7.2. Detecção antecipada de vulnerabilidades e bugs usando análise estática com IA
 - 16.7.3. Melhoria da cobertura de testes identificando áreas críticas por meio de IA
- 16.8. *Pair Programming* com GitHub Copilot
 - 16.8.1. Integração e uso eficaz do GitHub Copilot em sessões de *Pair Programming*
 - 16.8.2. Integração. Melhoria da comunicação e colaboração do desenvolvedor com o GitHub Copilot
 - 16.8.3. Estratégias de integração para aproveitar ao máximo as dicas de código geradas pelo GitHub Copilot
 - 16.8.4. Integração Estudos de caso e boas práticas em *Pair Programming* assistido pela IA
- 16.9. Tradução automática entre linguagens de programação
 - 16.9.1. Ferramentas e serviços de tradução automática específicos do idioma para linguagens de programação
 - 16.9.2. Adaptação de algoritmos de tradução automática a contextos de desenvolvimento
 - 16.9.3. Aprimoramento da interoperabilidade entre diferentes idiomas por meio da tradução automática
 - 16.9.4. Avaliação e atenuação dos possíveis desafios e limitações da tradução automática
- 16.10. Ferramentas da IA recomendadas para aumentar a produtividade
 - 16.10.1. Análise comparativa de ferramentas de IA para desenvolvimento de software
 - 16.10.2. Integração de ferramentas de IA em fluxos de trabalho
 - 16.10.3. Automação de tarefas rotineiras com ferramentas de IA
 - 16.10.4. Avaliação e seleção de ferramentas com base no contexto e nos requisitos do projeto

Módulo 17. Arquitetura de Software com IA

- 17.1. Otimização e gestão de desempenho em ferramentas de IA
 - 17.1.1. Análise e criação de perfis de desempenho de ferramentas de IA
 - 17.1.2. Estratégias de otimização para algoritmos e modelos de IA
 - 17.1.3. Implementação de técnicas de *caching* e paralelização para melhorar o desempenho
 - 17.1.4. Ferramentas e metodologias para monitoramento contínuo do desempenho em tempo real
- 17.2. Escalabilidade em aplicativos de IA
 - 17.2.1. Projetar arquiteturas dimensionáveis para aplicativos de IA
 - 17.2.2. Implementação de técnicas de particionamento e compartilhamento de carga
 - 17.2.3. Fluxo de trabalho e gestão de carga de trabalho em sistemas dimensionáveis
 - 17.2.4. Estratégias para expansão horizontal e vertical em ambientes de demanda variável
- 17.3. Capacidade de manutenção de aplicativos com IA
 - 17.3.1. Princípios de design para facilitar a capacidade de manutenção em projetos de IA
 - 17.3.2. Estratégias de documentação específicas para modelos e algoritmos de IA
 - 17.3.3. Implementação de testes unitários e de integração para facilitar a manutenção
 - 17.3.4. Métodos para refatoração e melhoria contínua em sistemas com componentes de IA
- 17.4. Projeto de Sistema de grande escala
 - 17.4.1. Princípios de arquitetura para o projeto de sistemas de grande escala
 - 17.4.2. Decomposição de sistemas complexos em microserviços
 - 17.4.3. Implementação de padrões de projeto específicos para sistemas distribuídos
 - 17.4.4. Estratégias de gestão da complexidade para arquiteturas de grande escala com componentes de IA
- 17.5. Armazenamento de dados em grande escala para ferramentas de IA
 - 17.5.1. Seleção de tecnologias de armazenamento de dados dimensionáveis
 - 17.5.2. Projeto de esquema de banco de dados para o manejo eficiente de grandes volumes de dados
 - 17.5.3. Estratégias de particionamento e replicação em ambientes de armazenamento em massa
 - 17.5.4. Implementação de sistemas de gestão de dados para garantir a integridade e a disponibilidade em projetos de IA

- 17.6. Estruturas de dados com IA
 - 17.6.1. Adaptação de estruturas de dados clássicas para uso em algoritmos de IA
 - 17.6.2. Projeto e otimização de estruturas de dados específicas para modelos de aprendizado de máquina
 - 17.6.3. Integração de estruturas de dados eficientes em sistemas com uso intensivo de dados
 - 17.6.4. Estratégias para manipulação e armazenamento de dados em tempo real em estruturas de dados com IA
- 17.7. Algoritmos de programação para produtos de IA
 - 17.7.1. Desenvolvimento e implementação de algoritmos específicos para aplicativos de IA
 - 17.7.2. Estratégias de seleção de algoritmos de acordo com o tipo de problema e os requisitos do produto
 - 17.7.3. Adaptação de algoritmos clássicos para integração em sistemas de inteligência artificial
 - 17.7.4. Avaliação e comparação do desempenho entre diferentes algoritmos em contextos de desenvolvimento de IA
- 17.8. Padrões de design para desenvolvimento de IA
 - 17.8.1. Identificar e aplicar padrões de design comuns em projetos com componentes de IA
 - 17.8.2. Desenvolvimento de padrões específicos para a integração de modelos e algoritmos em sistemas existentes.
 - 17.8.3. Estratégias de implementação de padrões para melhorar a reutilização e a capacidade de manutenção em projetos de IA
 - 17.8.4. Estudos de caso e práticas recomendadas na aplicação de padrões de design em arquiteturas de IA
- 17.9. Implementação de *Clean Architecture*
 - 17.9.1. Princípios e conceitos fundamentais de *Clean Architecture*
 - 17.9.2. Adaptação de *Clean Architecture* a projetos com componentes de IA
 - 17.9.3. Implementação de camadas e dependências em sistemas com arquitetura limpa
 - 17.9.4. Benefícios e desafios da implementação de *Clean Architecture* em desenvolvimento de software de IA
- 17.10. Desenvolvimento seguro de software em aplicativos da Web com IA
 - 17.10.1. Princípios de segurança no desenvolvimento de software com componentes de IA
 - 17.10.2. Identificação e atenuação de possíveis vulnerabilidades em modelos e algoritmos de IA
 - 17.10.3. Implementação de práticas de desenvolvimento seguro em aplicativos da Web com funcionalidades de Inteligência Artificial
 - 17.10.4. Estratégias para proteger dados confidenciais e evitar ataques em projetos de IA

Módulo 18. Projetos Web com IA

- 18.1. Preparação do ambiente de trabalho para o desenvolvimento web com IA
 - 18.1.1. Configuração de ambientes de desenvolvimento web para projetos de inteligência artificial
 - 18.1.2. Seleção e preparação de ferramentas essenciais para o desenvolvimento web com IA
 - 18.1.3. Integração de bibliotecas e *frameworks* específico para projetos da Web com inteligência artificial
 - 18.1.4. Implementação de práticas recomendadas na configuração de ambientes de desenvolvimento colaborativo
- 18.2. Criação de *Workspace* para Projetos de IA
 - 18.2.1. Projeto e organização eficazes de *workspaces* para projetos da Web com componentes de inteligência artificial
 - 18.2.2. Uso de ferramentas de gestão de projetos e controle de versões no *workspace*
 - 18.2.3. Estratégias para colaboração e comunicação eficientes na equipe de desenvolvimento
 - 18.2.4. Adaptação do *workspace* às necessidades específicas dos projetos web de IA
- 18.3. Padrões de design em produtos de IA
 - 18.3.1. Identificação e aplicação de padrões de design comuns em interfaces de usuário com elementos de inteligência artificial
 - 18.3.2. Desenvolvimento de padrões específicos para melhorar a experiência do usuário em projetos da Web com IA
 - 18.3.3. Integração de padrões de design na arquitetura geral de projetos da Web com Inteligência Artificial
 - 18.3.4. Avaliação e seleção de padrões de design apropriados de acordo com o contexto do projeto
- 18.4. Desenvolvimento front-end com IA
 - 18.4.1. Integração de modelos de IA na camada de apresentação de projetos da Web
 - 18.4.2. Desenvolvimento de interfaces de usuário adaptáveis com elementos de inteligência artificial
 - 18.4.3. Implementação de funcionalidades de processamento de linguagem natural (NLP) no front-end
 - 18.4.4. Estratégias para otimização do desempenho no desenvolvimento de front-end com IA

- 18.5. Criação de banco de dados
 - 18.5.1. Seleção de tecnologias de banco de dados para projetos da Web com inteligência artificial
 - 18.5.2. Projeto de esquema de banco de dados para armazenar e gerenciar dados relacionados à IA
 - 18.5.3. Implementação de sistemas de armazenamento eficientes para grandes volumes de dados gerados por modelos de IA
 - 18.5.4. Estratégias para segurança e proteção de dados confidenciais em bancos de dados em projetos da Web com IA
- 18.6. Desenvolvimento de back-end com IA
 - 18.6.1. Integração de serviços e modelos de IA à lógica de negócios de back-end
 - 18.6.2. Desenvolvimento de APIs e pontos de extremidade específicos para comunicação entre os componentes de front-end e IA
 - 18.6.3. Implementação da lógica de processamento de dados e tomada de decisões no back-end com inteligência artificial
 - 18.6.4. Estratégias para escalabilidade e desempenho no desenvolvimento de back-end de projetos da Web com IA
- 18.7. Otimização do processo de implantação na Web
 - 18.7.1. Automatização do processo de criação e implantação de projetos da Web com IA
 - 18.7.2. Implementação de pipelines de CI/CD adaptados a aplicativos da Web com componentes de inteligência artificial
 - 18.7.3. Estratégias para a gestão eficiente de versões e atualizações em implementações contínuas
 - 18.7.4. Monitoramento e análise pós-implantação para melhoria contínua do processo
- 18.8. IA na computação em nuvem
 - 18.8.1. Integração de serviços de inteligência artificial em plataformas de computação em nuvem
 - 18.8.2. Desenvolvimento de soluções dimensionáveis e distribuídas usando serviços de nuvem com recursos de IA
 - 18.8.3. Estratégias para o manejo eficiente de recursos e custos em ambientes de nuvem com aplicativos da Web habilitados para IA
 - 18.8.4. Avaliação e comparação de provedores de serviços em nuvem para projetos da Web habilitados para IA

- 18.9. Criação de um projeto com IA para ambientes LAMP
 - 18.9.1. Adaptação de projetos da Web baseados na pilha LAMP para incluir componentes de IA
 - 18.9.2. Integração de bibliotecas e *frameworks* específicos de IA em ambientes LAMP
 - 18.9.3. Desenvolvimento de funcionalidades de IA que complementam a arquitetura LAMP tradicional
 - 18.9.4. Estratégias para otimização e manutenção em projetos web de IA em ambientes LAMP
- 18.10. Criação de um projeto com IA para ambientes MEVN
 - 18.10.1. Integração de tecnologias e ferramentas da pilha MEVN com componentes de IA
 - 18.10.2. Desenvolvimento de aplicativos da Web modernos e dimensionáveis em ambientes MEVN com recursos de IA
 - 18.10.3. Implementação de funcionalidades de processamento de dados e aprendizado de máquina em projetos MEVN
 - 18.10.4. Estratégias para melhorar o desempenho e a segurança de aplicativos da Web habilitados para IA em ambientes MEVN

Módulo 19. Aplicativos Móveis com IA

- 19.1. Preparação de um ambiente de trabalho para o desenvolvimento móvel com IA
 - 19.1.1. Configuração de ambientes de desenvolvimento móvel para projetos de IA
 - 19.1.2. Seleção e preparação de ferramentas específicas para o desenvolvimento de aplicativos móveis com IA
 - 19.1.3. Integração de bibliotecas e *frameworks* de IA em ambientes de desenvolvimento móvel
 - 19.1.4. Configuração de emuladores e dispositivos reais para testar aplicativos móveis com componentes de inteligência artificial
- 19.2. Criação de um *workspace* com Github copilot
 - 19.2.1. Integração do GitHub Copilot em ambientes de desenvolvimento móvel
 - 19.2.2. Uso eficaz do GitHub Copilot para geração de código em projetos de IA
 - 19.2.3. Estratégias para colaboração do desenvolvedor ao usar o GitHub Copilot no *workspace*
 - 19.2.4. Práticas recomendadas e limitações no uso do GitHub Copilot no desenvolvimento de aplicativos móveis com IA
- 19.3. Configuração do Firebase
 - 19.3.1. Configuração inicial de um projeto Firebase para desenvolvimento móvel
 - 19.3.2. Integração do Firebase em aplicativos móveis com recursos de IA
 - 19.3.3. Uso dos serviços do Firebase como banco de dados, autenticação e notificações em projetos de IA
 - 19.3.4. Estratégias para a gestão de dados e eventos em tempo real em aplicativos móveis com o Firebase

- 19.4. Conceitos de Clean Architecture, DataSources, Repositories
 - 19.4.1. Princípios fundamentais de Clean Architecture no desenvolvimento móvel com IA
 - 19.4.2. Implementação de camadas de DataSources e Repositórios em arquiteturas limpas
 - 19.4.3. Design e estruturação de componentes em projetos móveis com foco em arquitetura limpa
 - 19.4.4. Benefícios e desafios da implementação de *Clean Architecture* em aplicativos móveis com IA
- 19.5. Criação da tela de autenticação
 - 19.5.1. Projeto e desenvolvimento de interfaces de usuário para telas de autenticação em aplicativos móveis habilitados para IA
 - 19.5.2. Integração de serviços de autenticação com o Firebase na tela de login
 - 19.5.3. Uso de técnicas de segurança e proteção de dados na tela de autenticação
 - 19.5.4. Personalização e customização da experiência do usuário na tela de autenticação
- 19.6. Criação de *Dashboard* e Navegação
 - 19.6.1. Projeto e desenvolvimento de *Dashboards* com elementos de Inteligência Artificial
 - 19.6.2. Implementação de sistemas de navegação eficientes em aplicativos móveis com IA
 - 19.6.3. Integração de funcionalidades de IA no *Dashboard* para melhorar a experiência do usuário
- 19.7. Criação de uma tela de listagem
 - 19.7.1. Desenvolvimento de interfaces de usuário para telas de listagem em aplicativos móveis habilitados para IA
 - 19.7.2. Integração de algoritmos de recomendação e filtragem na tela de listagem
 - 19.7.3. Uso de padrões de design para apresentação eficaz de dados na listagem
 - 19.7.4. Estratégias para carregamento eficiente de dados em tempo real na tela com listagem
- 19.8. Criação da tela de detalhes
 - 19.8.1. Projeto e desenvolvimento de interfaces de usuário detalhadas para a apresentação de informações específicas
 - 19.8.2. Integração de funcionalidades de IA para enriquecer a tela de detalhes
 - 19.8.3. Implementação de interações e animações na tela de detalhes
 - 19.8.4. Estratégias para otimizar o desempenho de carregamento e exibição em aplicativos móveis habilitados para IA
- 19.9. Criação da tela de *Settings*
 - 19.9.1. Desenvolvimento de interfaces de usuário para configuração e definições em aplicativos móveis com IA
 - 19.9.2. Integração de configurações personalizadas relacionadas a componentes de inteligência artificial
 - 19.9.3. Implementação de opções e preferências de personalização na tela de configuração
 - 19.9.4. Estratégias de usabilidade e clareza na apresentação de opções na tela de *settings*

- 19.10. Criar ícones, *Splash* recursos gráficos para seu aplicativo com IA
 - 19.10.1. Design e criação de ícones atraentes para representar o aplicativo móvel com IA
 - 19.10.2. Desenvolvimento de telas (*splash*) com recursos visuais impactantes
 - 19.10.3. Seleção e adaptação de recursos gráficos para melhorar a estética do aplicativo móvel
 - 19.10.4. Estratégias para consistência e marca visual em elementos gráficos de aplicativos de IA

Módulo 20. IA para QA Testing

- 20.1. Ciclo de vida de *testing*
 - 20.1.1. Descrição e compreensão do ciclo de vida de *testing* no desenvolvimento de software
 - 20.1.2. Fases do ciclo de vida de *testing* e sua importância na garantia de qualidade
 - 20.1.3. Integração da inteligência artificial em diferentes estágios do ciclo de vida de *testing*
 - 20.1.4. Estratégias para a melhoria contínua do ciclo de vida de *testing* por meio do uso de IA
- 20.2. Test Cases e detecção de bugs
 - 20.2.1. Design e redação eficazes de casos de teste no contexto de QA *Testing*
 - 20.2.2. Identificação de bugs e erros durante a execução do caso de teste
 - 20.2.3. Aplicação de técnicas de detecção antecipada de bugs por meio de análise estática
 - 20.2.4. Uso de ferramentas de inteligência artificial para identificação automática de bugs em test cases
- 20.3. Tipos de *testing*
 - 20.3.1. Exploração de diferentes tipos de *testing* no campo de QA
 - 20.3.2. Testes unitários, de integração, funcionais e de aceitação: recursos e aplicativos
 - 20.3.3. Estratégias para a seleção e combinação apropriada de tipos de *testing* em projetos de IA
 - 20.3.4. Adaptação dos tipos de *testing* convencional para projetos com componentes de inteligência artificial
- 20.4. Criar um plano de teste
 - 20.4.1. Projetar e estruturar um plano de teste abrangente
 - 20.4.2. Identificação de requisitos e cenários de teste em projetos de IA
 - 20.4.3. Estratégias para planejamento de testes manuais e automatizados
 - 20.4.4. Avaliação e ajuste contínuos do plano de teste de acordo com o desenvolvimento do projeto

- 20.5. Detecção e relatório de *Bugs* com IA
 - 20.5.1. Implementação de técnicas de detecção automática de bugs usando algoritmos de aprendizado de máquina
 - 20.5.2. Uso de ferramentas de inteligência artificial para análise dinâmica de código para procurar possíveis bugs
 - 20.5.3. Estratégias para geração automática de relatórios detalhados sobre bugs detectados por IA
 - 20.5.4. Colaboração eficaz entre as equipes de desenvolvimento e de controle de qualidade no tratamento de bugs identificados por IA
- 20.6. Criação de testes automatizados com IA
 - 20.6.1. Desenvolvimento de scripts de teste automatizados para projetos com componentes de inteligência artificial
 - 20.6.2. Integração de ferramentas de automação de testes baseadas em IA
 - 20.6.3. Uso de algoritmos de *machine learning* para geração dinâmica de casos de teste automatizados
 - 20.6.4. Estratégias para execução e manutenção eficientes de testes automatizados em projetos de IA
- 20.7. *API Testing*
 - 20.7.1. Conceitos fundamentais de *API testing* e sua importância no controle de qualidade
 - 20.7.2. Desenvolvimento de testes para a verificação de APIs em ambientes com componentes de Inteligência Artificial
 - 20.7.3. Estratégias para validação de dados e resultados em *API testing* com IA
 - 20.7.4. Uso de ferramentas específicas para *testing* de API em projetos com IA
- 20.8. Ferramentas de IA para *Web Testing*
 - 20.8.1. Exploração de ferramentas de inteligência artificial para automação de testes em ambientes Web
 - 20.8.2. Integração de tecnologias de reconhecimento de elementos e análise visual em *web testing*
 - 20.8.3. Estratégias para detecção automática de alterações e problemas de desempenho em aplicativos da Web usando IA
 - 20.8.4. Avaliação de ferramentas específicas para aumentar a eficiência em *web testing* com IA
- 20.9. *Mobile Testing* por meio de IA
 - 20.9.1. Desenvolvimento de estratégias de *testing* para aplicativos móveis com componentes de inteligência artificial
 - 20.9.2. Integração de ferramentas de *testing* específicas para plataformas móveis baseadas em IA
 - 20.9.3. Uso de algoritmos de aprendizado de máquina para detectar problemas de desempenho em aplicativos móveis
 - 20.9.4. Estratégias para a validação de interfaces e funções específicas de aplicativos móveis usando IA
- 20.10. Ferramentas de controle de qualidade com IA
 - 20.10.1. Exploração de ferramentas e plataformas de controle de qualidade que incorporam funcionalidades de inteligência artificial
 - 20.10.2. Avaliação de ferramentas para gestão e execução eficientes de testes em projetos de IA
 - 20.10.3. Uso de algoritmos de *machine learning* para geração e otimização de casos de teste
 - 20.10.4. Estratégias para a seleção e adoção efetiva de ferramentas de controle de qualidade habilitadas para IA



Posicione-se no mercado de trabalho com um programa 100% online que se adapta às suas necessidades e permite um aprendizado imersivo e sólido”

06

Metodologia

Este curso oferece uma maneira diferente de aprender. Nossa metodologia é desenvolvida através de um modo de aprendizagem cíclico: **o Relearning**. Este sistema de ensino é utilizado, por exemplo, nas faculdades de medicina mais prestigiadas do mundo e foi considerado um dos mais eficazes pelas principais publicações científicas, como o ***New England Journal of Medicine***.



“

Descubra o Relearning, um sistema que abandona a aprendizagem linear convencional para realizá-la através de sistemas de ensino cíclicos: uma forma de aprendizagem que se mostrou extremamente eficaz, especialmente em disciplinas que requerem memorização"

Estudo de caso para contextualizar todo o conteúdo

Nosso programa oferece um método revolucionário para desenvolver as habilidades e o conhecimento. Nosso objetivo é fortalecer as competências em um contexto de mudança, competitivo e altamente exigente.

“

Com a TECH você irá experimentar uma forma de aprender que está revolucionando as bases das universidades tradicionais em todo o mundo”



Você terá acesso a um sistema de aprendizagem baseado na repetição, por meio de um ensino natural e progressivo ao longo de todo o programa.



Um método de aprendizagem inovador e diferente

Este curso da TECH é um programa de ensino intensivo, criado do zero, que propõe os desafios e decisões mais exigentes nesta área, em âmbito nacional ou internacional. Através desta metodologia, o crescimento pessoal e profissional é impulsionado em direção ao sucesso. O método do caso, técnica que constitui a base deste conteúdo, garante que a realidade econômica, social e profissional mais atual seja adotada.

“

Nosso programa prepara você para enfrentar novos desafios em ambientes incertos e alcançar o sucesso na sua carreira”

Através de atividades de colaboração e casos reais, o aluno aprenderá a resolver situações complexas em ambientes reais de negócios.

O método do caso é o sistema de aprendizagem mais utilizado nas principais escolas de Informática do mundo, desde que elas existem. Desenvolvido em 1912 para que os estudantes de Direito não aprendessem a lei apenas com base no conteúdo teórico, o método do caso consistia em apresentar-lhes situações realmente complexas para que tomassem decisões conscientes e julgassem a melhor forma de resolvê-las. Em 1924 foi estabelecido como o método de ensino padrão em Harvard.

Em uma determinada situação, o que um profissional deveria fazer? Esta é a pergunta que abordamos no método do caso, um método de aprendizagem orientado para a ação. Ao longo do curso, os alunos vão se deparar com múltiplos casos reais. Terão que integrar todo o conhecimento, pesquisar, argumentar e defender suas ideias e decisões.

Metodologia Relearning

A TECH utiliza de maneira eficaz a metodologia do estudo de caso com um sistema de aprendizagem 100% online, baseado na repetição, combinando elementos didáticos diferentes em cada aula.

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

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

Na TECH você aprenderá através de uma metodologia de vanguarda, desenvolvida para capacitar os profissionais do futuro. Este método, na vanguarda da pedagogia mundial, se chama Relearning.

Nossa universidade é uma das únicas que possui a licença para usar este método de sucesso. Em 2019 conseguimos melhorar os níveis de satisfação geral dos nossos alunos (qualidade de ensino, qualidade dos materiais, estrutura dos curso, objetivos, entre outros) com relação aos indicadores da melhor universidade online.



No nosso programa, a aprendizagem não é um processo linear, ela acontece em espiral (aprender, desaprender, esquecer e reaprender). Portanto, combinamos cada um desses elementos de forma concêntrica. Esta metodologia já capacitou mais de 650 mil universitários com um sucesso sem precedentes em campos tão diversos como a bioquímica, a genética, a cirurgia, o direito internacional, habilidades administrativas, ciência do esporte, filosofia, direito, engenharia, jornalismo, história, mercados e instrumentos financeiros. Tudo isso em um ambiente altamente exigente, com um corpo discente com um perfil socioeconômico médio-alto e uma média de idade de 43,5 anos.

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

A partir das últimas evidências científicas no campo da neurociência, sabemos como organizar informações, ideias, imagens, memórias, mas sabemos também que o lugar e o contexto onde aprendemos algo é fundamental para nossa capacidade de lembrá-lo e armazená-lo no hipocampo, para mantê-lo em nossa memória a longo prazo.

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



Neste programa, oferecemos o melhor material educacional, preparado especialmente para os profissionais:



Material de estudo

Todo o conteúdo foi criado especialmente para o curso pelos especialistas que irão ministrá-lo, o que faz com que o desenvolvimento didático seja realmente específico e concreto.

Posteriormente, esse conteúdo é adaptado ao formato audiovisual, para criar o método de trabalho online da TECH. Tudo isso, com as técnicas mais inovadoras que proporcionam alta qualidade em todo o material que é colocado à disposição do aluno.



Masterclasses

Há evidências científicas sobre a utilidade da observação de terceiros especialistas.

O "Learning from an expert" fortalece o conhecimento e a memória, além de gerar segurança para a tomada de decisões difíceis no futuro.



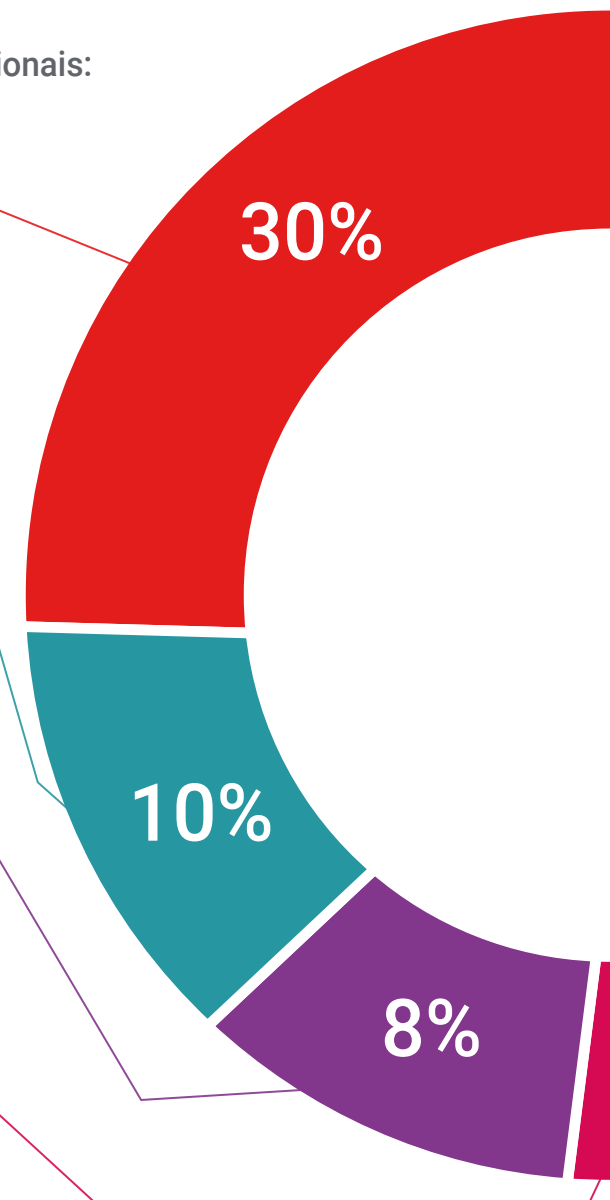
Práticas de habilidades e competências

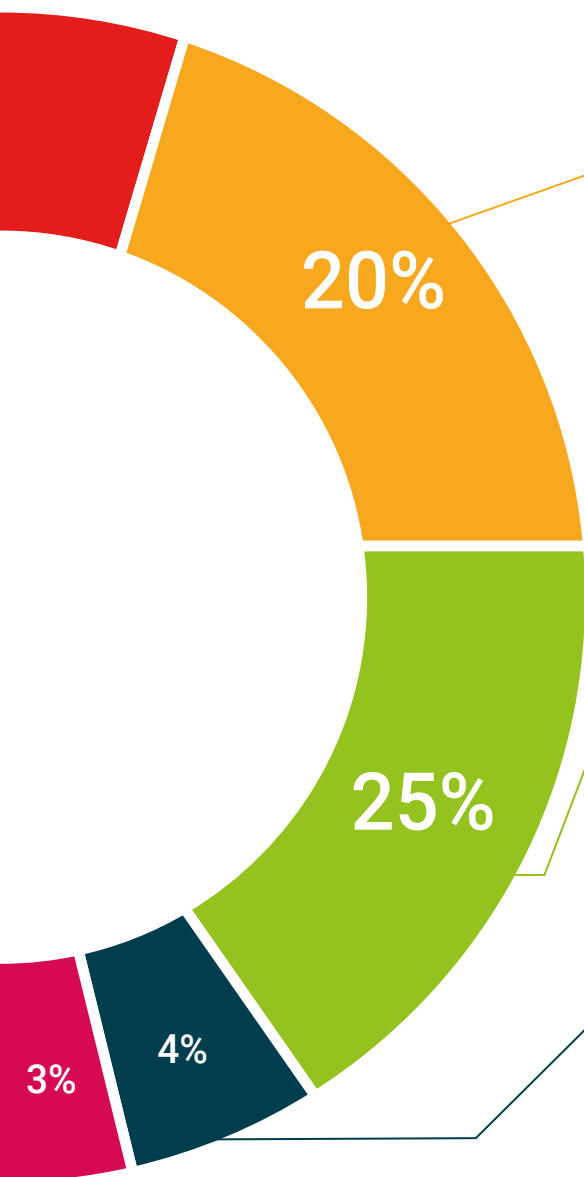
Serão realizadas atividades para desenvolver competências e habilidades específicas em cada área temática. Práticas e dinâmicas para adquirir e ampliar as competências e habilidades que um especialista precisa desenvolver no contexto globalizado em que vivemos.



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 for necessário para complementar a sua capacitação.





Estudos de caso

Os alunos irão completar uma seleção dos melhores estudos de caso escolhidos especialmente para esta capacitação. Casos apresentados, analisados e orientados pelos melhores especialistas do cenário internacional.



Resumos interativos

A equipe da TECH apresenta o conteúdo de forma atraente e dinâmica através de pílulas multimídia que incluem áudios, vídeos, imagens, gráficos e mapas conceituais para consolidar o conhecimento.

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



Testing & Retesting

Avaliamos e reavaliamos periodicamente o conhecimento do aluno ao longo do programa, através de atividades e exercícios de avaliação e autoavaliação, para que possa comprovar que está alcançando seus objetivos.



07

Certificado

O Mestrado Próprio em Inteligência Artificial em Programação garante, além da capacitação mais rigorosa e atualizada, acesso ao certificado do Curso emitido pela TECH Universidade Tecnológica.



“

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

Este **Mestrado Próprio em Inteligência Artificial em Programação** conta com o conteúdo científico mais completo e atualizado do mercado.

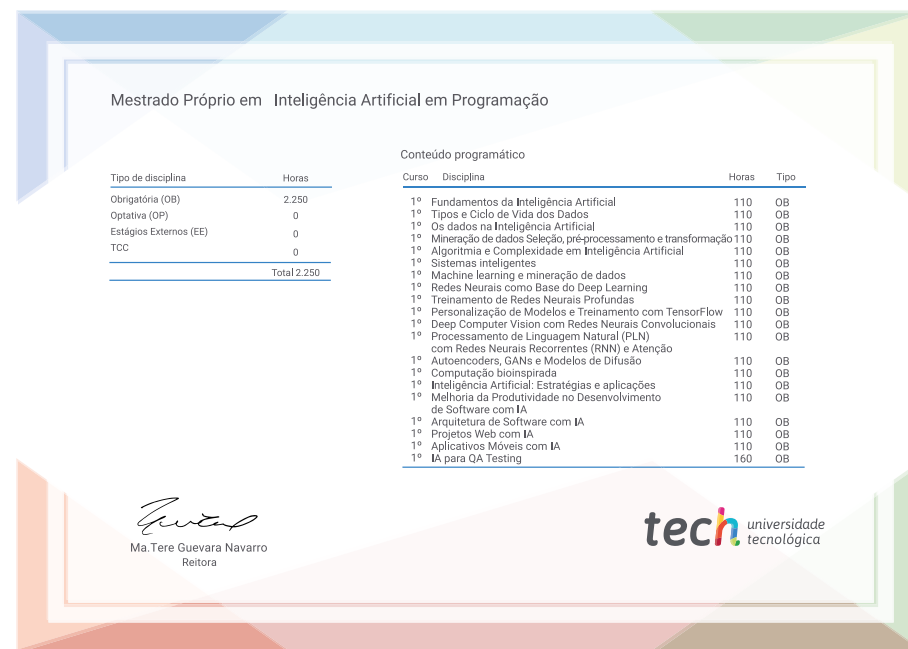
Uma vez aprovadas as avaliações, o aluno receberá por correio o certificado* correspondente ao título de **Mestrado Próprio** emitido pela **TECH Universidade Tecnológica**.

O certificado emitido pela **TECH Universidade Tecnológica** expressará a qualificação obtida no Programa Avançado, atendendo aos requisitos normalmente exigidos pelas bolsas de empregos, concursos públicos e avaliação de carreira profissional.

Título: **Mestrado Próprio em Inteligência Artificial em Programação**

Modalidade: **online**

Duração: **6 meses**



*Apostila de Haia: Caso o aluno solicite que seu certificado seja apostilado, a TECH EDUCATION 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 situação

tech universidade
tecnológica

Mestrado Próprio
Inteligência Artificial
em Programação

- » Modalidade: online
- » Duração: 6 meses
- » Certificado: TECH Universidade Tecnológica
- » Horário: no seu próprio ritmo
- » Provas: online

Mestrado Próprio

Inteligência Artificial em Programação