

Desenvolvimento em Python





Mestrado Desenvolvimento em Python

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

Acesso ao site: www.techtute.com/br/informatica/mestrado/mestrado-desenvolvimento-python

Índice

01

Apresentação

pág. 4

02

Objetivos

pág. 8

03

Competências

pág. 12

04

Direção do curso

pág. 16

05

Estrutura e conteúdo

pág. 20

06

Metodologia

pág. 30

07

Certificado

pág. 38

01

Apresentação

O Python se destaca como uma linguagem de programação versátil e eficiente. Em primeiro lugar, sua sintaxe clara e legível facilita a escrita e a compreensão do código, o que acelera o processo de desenvolvimento e reduz a probabilidade de erros. Além disso, a grande variedade de bibliotecas e *Frameworks* disponíveis fornece aos desenvolvedores acesso a ferramentas poderosas e pré-existentes, permitindo um desenvolvimento mais rápido e eficiente. Sua natureza de código aberto e a comunidade ativa de desenvolvedores também contribuem para o constante aprimoramento e atualização da linguagem, garantindo um ambiente de desenvolvimento dinâmico e moderno. Por essa razão, a TECH desenvolveu um programa abrangente, voltado para a formação de profissionais com os últimos avanços em Desenvolvimento em Python, com uma metodologia 100% online.



“

O Python é conhecido por sua adaptabilidade, sendo usado em aplicações que vão desde o desenvolvimento web até a Inteligência Artificial. O que você está esperando para se matricular?"

Python é uma linguagem de programação de alto nível, muito usada entre os cientistas da computação, porque possui uma ampla variedade de bibliotecas e estruturas que simplificam tarefas comuns, permitindo que os desenvolvedores se concentrem na lógica das aplicações em vez de perder tempo com detalhes técnicos. Além disso, sua versatilidade é outra vantagem significativa, pois pode ser usada em diversos ambientes, desde o desenvolvimento da Web até a análise de dados e a aprendizagem automática.

Assim nasceu este Mestrado em Desenvolvimento em Python, um programa que abordará o manuseio avançado de dados e tipos em Python, explorando identificadores, palavras-chave, tipos integrais e booleanos, bem como a formatação avançada de *strings* e codificações. Além disso, serão examinadas coleções como tuplas, listas e dicionários, além de técnicas de iteração e funções lambda, proporcionando uma base sólida nos fundamentos da linguagem.

Também se aprofundará no desenvolvimento de aplicações em Python, com ênfase nas melhores práticas e metodologias modernas. Desde a arquitetura do aplicativo até a implantação e a manutenção, serão abordados aspectos como design e modelagem avançados, gerenciamento de dependências, padrões de design, *testing* e *debugging*, a otimização do desempenho, a implementação e as estratégias de distribuição.

Da mesma forma, o cientista da computação se aprofundará no desenvolvimento web e mobile com Python, abordando *frameworks*, como Django e Flask, bem como o desenvolvimento de APIs e serviços Web. Além disso, se concentrará no design de interface e na experiência do usuário (UI/UX), desde o uso de ferramentas de design até o aprimoramento da acessibilidade e da usabilidade. E por último, serão abordados o gerenciamento e a análise de dados, usando Python e ferramentas como NumPy, Pandas e Matplotlib.

Assim, essa pós-graduação é apresentada como uma oportunidade única, através de uma proposta acadêmica totalmente online e adaptável. Com essa abordagem, os profissionais terão mais liberdade para gerenciar seu tempo de acesso, o que lhes permitirá harmonizar o trabalho do dia a dia e os compromissos pessoais.

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

- ♦ O desenvolvimento de estudos de caso apresentados por especialistas em desenvolvimento em Python.
- ♦ O conteúdo gráfico, esquemático e extremamente útil, fornece informações teórica e práticas sobre as disciplinas fundamentais para a prática profissional
- ♦ Exercícios práticos onde o processo de autoavaliação é realizado para melhorar a aprendizagem
- ♦ Destaque especial para as metodologias inovadoras
- ♦ Aulas teóricas, perguntas a especialistas, fóruns de discussão sobre temas controversos e trabalhos de reflexão individual
- ♦ Disponibilidade de acesso a todo o conteúdo a partir de qualquer dispositivo, fixo ou portátil, com conexão à Internet



Uma capacitação abrangente e especializada em desenvolvimento em Python, que irá prepará-lo para encarar desafios reais no mundo do desenvolvimento de software"

“

Com este Mestrado, 100% online, você irá aprofundar seu conhecimento em design de Frontend e Backend, gerenciamento de banco de dados e estratégias de publicação em lojas de aplicativos”

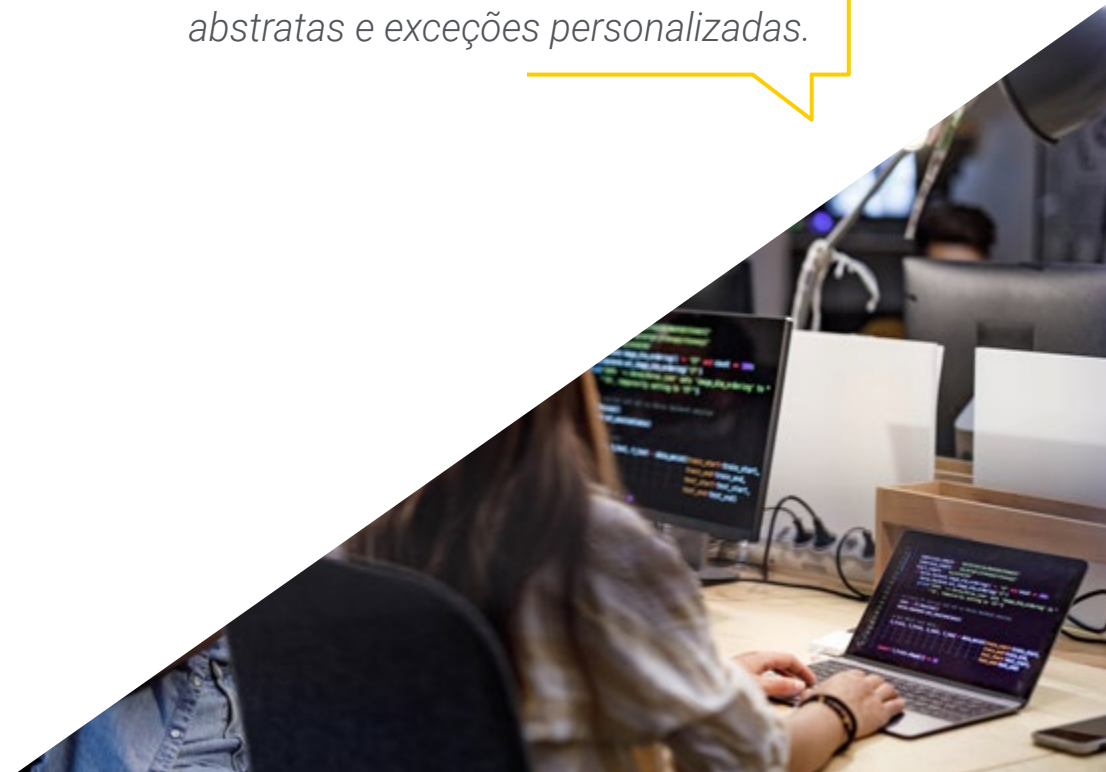
O corpo docente do curso conta com profissionais do setor, que transferem toda a experiência adquirida ao longo de suas carreiras para esta capacitação, além de especialistas reconhecidos de sociedades de referência e universidades de prestígio.

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.

Você abordará desde os conceitos básicos, como variáveis e tipos de dados, até técnicas avançadas de visualização de dados e estratégias de otimização de desempenho e armazenamento. Matricule-se já!

Você analisará a Programação Orientada a Objetos (POO), além de tópicos como classes, herança, polimorfismo, criação de classes abstratas e exceções personalizadas.



02

Objetivos

Este Mestrado foi desenvolvido com o firme propósito de cultivar profissionais altamente competentes e especializados no amplo universo da programação nessa linguagem de alto desempenho. Por meio de módulos abrangentes, os alunos irão se aprofundar na sintaxe e na funcionalidade do Python, desde o manuseio avançado de dados até o domínio da programação orientada a objetos (POO) e o design eficiente de aplicativos web e mobile. Com uma abordagem pedagógica rigorosa e atualizada, o programa garantirá uma experiência educacional exclusiva que posicionará os cientistas da computação como especialistas requisitados no setor de desenvolvimento de software. Prepare-se para superar seus limites e embarcar em uma jornada educacional que não apenas elevará suas habilidades em Python, mas também lhe dará as ferramentas para se destacar e liderar no mundo dinâmico do desenvolvimento tecnológico.



“

Tornar-se um líder em desenvolvimento de software, fornecendo conhecimento avançado e habilidades especializadas para lidar com os desafios tecnológicos contemporâneos"



Objetivos gerais

- Fornecer uma compreensão abrangente de Python
- Capacitar no manejo avançado de dados e tipos em Python
- Aplicar os princípios da programação orientada a objetos (POO) em Python
- Incentivar o uso de boas práticas e metodologias modernas no desenvolvimento de software
- Fornecer uma capacitação abrangente em desenvolvimento web e móvel com Python
- Integração dos princípios de UI/UX ao desenvolvimento de software
- Capacitar sobre a configuração e o uso de ferramentas e ambientes de desenvolvimento de dados
- Aprofundar no uso de estruturas de dados e funções em Python
- Capacitar em técnicas avançadas de visualização de dados com o Matplotlib
- Capacitar em estratégias de otimização de desempenho e armazenamento de dados



Graças à ampla biblioteca oferecida pelo curso, repleta dos recursos multimídia mais inovadores, você obterá as ferramentas para se destacar e liderar no mundo dinâmico do desenvolvimento tecnológico"





Objetivos específicos

Módulo 1. Programação em Python

- ♦ Habilitar a configuração e o uso eficaz do ambiente de desenvolvimento em Python
- ♦ Entender os conceitos avançados de programação

Módulo 2. Dados avançados e controle de fluxo com Python

- ♦ Domine as convenções e práticas para lidar com identificadores e palavras-chave.
- ♦ Aplicar estruturas de dados complexas e suas operações

Módulo 3. Programação orientada a objetos (POO) em Python

- ♦ Dominar a criação e o uso de classes e objetos em Python
- ♦ Aplicar herança e polimorfismo em Python

Módulo 4. Desenvolvimento de aplicativos em Python

- ♦ Especializar-se no design e na modelagem de aplicativos avançados
- ♦ Capacitar-se na otimização, implantação e manutenção de aplicativos

Módulo 5. Desenvolvimento Web e Mobile com Python

- ♦ Usar *Frameworks* web populares em Python
- ♦ Preparar-se para o desenvolvimento e a publicação de aplicativos móveis

Módulo 6. Interface e experiência do usuário com Python

- ♦ Instruir sobre técnicas de design responsivo e adaptativo.
- ♦ Preparar para fazer testes de usabilidade e análise do comportamento do usuário

Módulo 7. Processamento de Dados e *Big Data* com Python

- ♦ Gerenciar técnicas e funções de controle de fluxo para manipulação de dados
- ♦ Promover as melhores práticas de codificação e tratamento de erros em Python

Módulo 8. Estruturas de dados e funções em Python

- ♦ Criar e usar funções avançadas
- ♦ Ler e escrever arquivos e seus respectivos processamentos

Módulo 9. Gerenciamento de dados em Python com NumPy e Pandas

- ♦ Criar e manipular Arrays com NumPy
- ♦ Promover competências em Visualização de Dados com Matplotlib

Módulo 10. Técnicas avançadas e aplicações práticas do NumPy e Pandas

- ♦ Desenvolver o conhecimento especializado no upload e no armazenamento de dados de e para diversas fontes
- ♦ Capacitarse em estratégias avançadas de limpeza e transformação de dados

03

Competências

O programa transmitirá não só conhecimento técnico, mas também se concentrará no estímulo de competências que transformarão os alunos em profissionais de destaque na área de desenvolvimento de software. Incluirá a capacidade de desenvolver soluções inovadoras e eficientes, por meio de um domínio especializado em Python, desde o manuseio avançado de dados até a criação de aplicativos web e mobile de alto desempenho. Além disso, os cientistas da computação estarão capacitados com habilidades especializadas em programação orientada a objetos (POO), design de interface de usuário e experiência do usuário (UI/UX) e análise avançada de dados com bibliotecas como NumPy e Pandas.



“

Incentivaremos o desenvolvimento de habilidades sociais essenciais, como pensamento crítico, resolução de problemas, colaboração em equipe e adaptabilidade.



Competências gerais

- ♦ Desenvolver habilidades práticas de programação
- ♦ Equipar-se com habilidades avançadas em controle de fluxo de programas
- ♦ Gerenciar o projeto e a implementação de software orientado a objetos
- ♦ Capacita-se no desenvolvimento integral de aplicações em Python
- ♦ Domine o design e o gerenciamento de aplicativos web e mobile
- ♦ Controlar o design da interface do usuário e a experiência do usuário usando Python
- ♦ Desenvolver competências em manuseio e análise de dados com Python
- ♦ Adquirir habilidades no manuseio avançado de arquivos e modelagem em Python
- ♦ Desenvolver habilidades avançadas de manipulação de dados com o NumPy e o Pandas
- ♦ Aprofundar no tratamento avançado de dados com NumPy e Pandas



Você irá encarar desafios do mundo real e se destacar em um mercado de trabalho concorrido e em constante evolução. Escolha a TECH!"





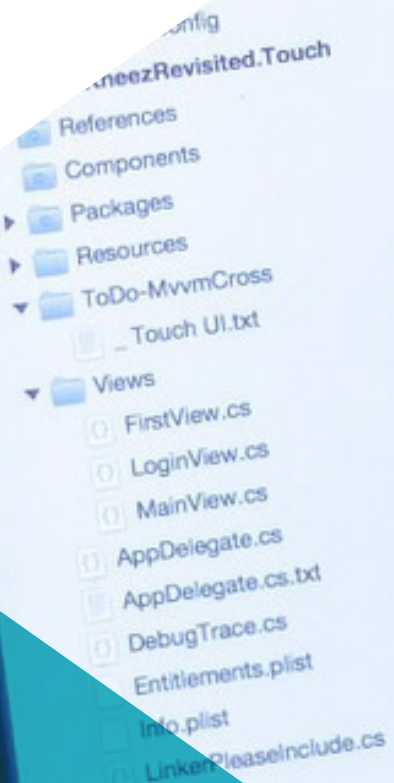
Competências específicas

- ♦ Capacitar-se no manejo avançado de dados em Python
- ♦ Manipular o uso avançado das funções Python
- ♦ Implementar conceitos avançados de programação orientada a objetos (POO), como classes abstratas e exceções personalizadas
- ♦ Controle de testes e *Debugging*
- ♦ Desenvolver APIs e serviços Web
- ♦ Dominar o uso de ferramentas de design de UI/UX com Python
- ♦ Usar bibliotecas essenciais para dados em Python
- ♦ Aplicar diferentes estruturas de dados de forma prática
- ♦ Usar o Pandas para gerenciamento de dados estruturados
- ♦ Promover competências na análise e manipulação de séries temporais e dados complexos e dados complexos

04

Direção do curso

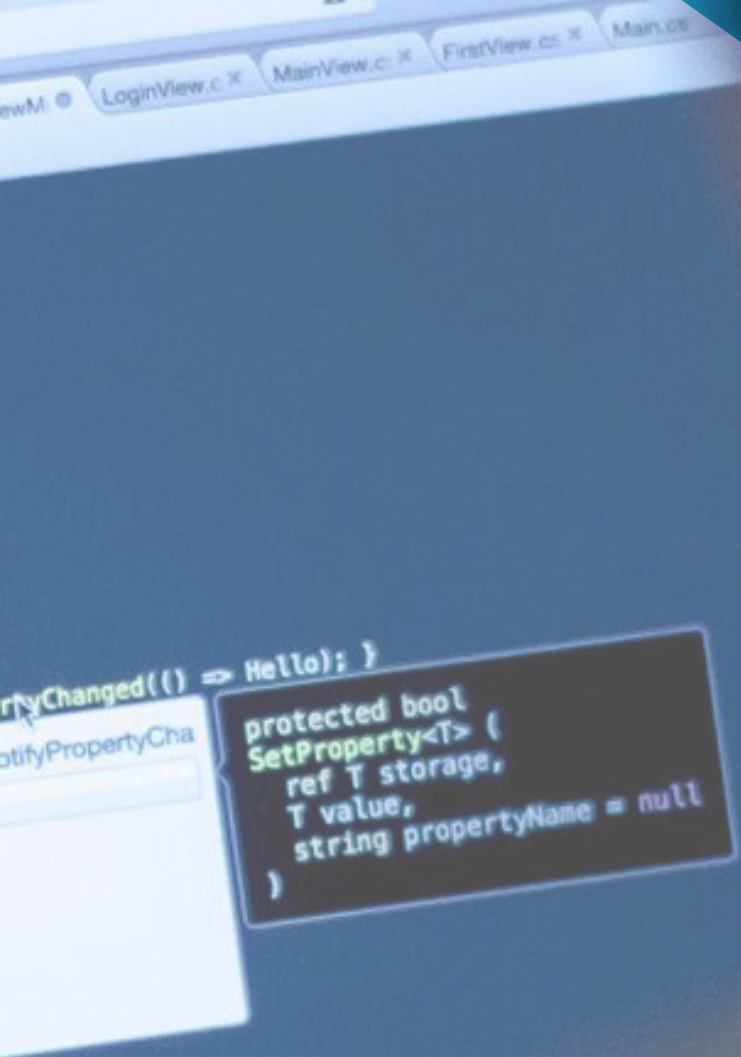
Os professores deste Mestrado são especialistas do setor, com ampla experiência prática em desenvolvimento de software e um histórico comprovado de projetos bem-sucedidos. Ao combinar teoria com aplicação prática, esses mentores não só transmitirão um conhecimento avançado de Python, mas também compartilharão sua experiência no mundo real, proporcionando aos alunos uma visão valiosa das melhores práticas e tendências atuais do setor. Além disso, sua abordagem pedagógica se destacará pelo compromisso de fornecer aos alunos a capacitação técnica mais atualizada para que possam se destacar no campo dinâmico do desenvolvimento de software.



```
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22
```

```
public class MainViewModel : MvxViewModel  
{  
    public MainViewModel ()  
    {  
        private string _hello = "Hello MOFO";  
        public string Hello  
        {  
            get { return _hello; }  
            set { set _hello = value; RaisePrope  
        }  
    }  
}
```

Cirrious.MvvmCross.ViewModels.MvxN
SetProperty



“

Você terá sucesso ao lado dos melhores profissionais e irá adquirir o conhecimento e as habilidades necessárias para ingressar no setor de desenvolvimento em Python"

Direção



Sr. Dionis Matos Rodríguez

- ♦ *Data Engineer* na Wide Agency Sodexo
- ♦ *Data Consultant* na Tokiota
- ♦ *Data Engineer* na Devoteam
- ♦ *BI Developer* na Ibermática
- ♦ *Applications Engineer* na Johnson Controls
- ♦ *Database Developer* na Suncapital España
- ♦ *Senior Web Developer* na Deadlock Solutions
- ♦ *QA Analyst* na Metaconzept
- ♦ Mestrado em *Big Data & Analytics* pela EAE Business School
- ♦ Mestrado em Análise e Projeto de Sistemas
- ♦ Formado em Engenharia da Computação pela Universidade APEC

Professores

Sr. Javier Villar Valor

- ♦ Diretor e Sócio fundador de Impulsa2
- ♦ *Chief Operations Officer* (COO) na Summa Insurance Brokers
- ♦ Diretor de Transformação e Excelência Operacional na Johnson Controls
- ♦ Mestrado em *Coaching* Profissional
- ♦ Executive MBA pela Emlyon Business School, França
- ♦ Mestrado em Gestão da Qualidade pela EOI
- ♦ Engenharia da Computação na University Acción Pro-Education and Culture (UNAPEC)

Sr. Armando Gil Contreras

- ♦ *Lead Big Data Scientist-Big Data* na Jhonson Controls
- ♦ *Data Scientist-Big Data* na Opensistemas S.A.
- ♦ Auditor de Fundos na Criatividade e Tecnologia S.A., (CYTSA)
- ♦ Auditor do setor público na PricewaterhouseCoopers Auditors
- ♦ Mestrado em *Data Science* pelo Centro Universitario de Tecnología y Arte
- ♦ MBA em Relações e Negócios Internacionais pelo Centro de Estudos Financeiros (CEF)
- ♦ Formado em Economia pelo Instituto Tecnológico de Santo Domingo

Sra. Milagros Gil Contreras

- ♦ *Content Creator* na MPCTech LLC
- ♦ Gerente de projetos
- ♦ *Escritora de TI freelancer*
- ♦ MBA pela Universidade Complutense de Madrid
- ♦ Formada em Administração de Empresas pelo Instituto Tecnológico de Santo Domingo

Sra. Benedit Delgado Feliz

- ♦ Assistente administrativa e operadora de vigilância eletrônica na Direção Nacional de Controle de Drogas (DNCD)
- ♦ Atendimento ao cliente em Cáceres e equipes
- ♦ Reclamações e atendimento ao cliente na Express Parcel Services (EPS)
- ♦ Especialista em Microsoft Office pela Escuela Nacional de Informática
- ♦ Comunicadora social da Universidade Católica de Santo Domingo



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

05

Estrutura e conteúdo

O programa de estudos foi elaborado cuidadosamente para proporcionar uma experiência abrangente aos alunos. Desde uma análise aprofundada da sintaxe e das funcionalidades avançadas de Python até a especialização em Programação Orientada a Objetos (POO), design de aplicativos web e mobile e manipulação especializada de dados com bibliotecas como NumPy e Pandas, cada módulo foi cuidadosamente estruturado para fornecer um conhecimento sólido e as habilidades práticas. Além disso, serão explorados tópicos essenciais, como design de interface e experiência do usuário (UI/UX), manipulação avançada de dados, otimização de desempenho e armazenamento.



“

Você irá dominar a linguagem Python, além de estar preparado para encarar com confiança os desafios dinâmicos e tecnológicos do setor”

Módulo 1. Programação em Python

- 1.1. Criação e execução de programas em Python
 - 1.1.1. Configuração do ambiente de desenvolvimento
 - 1.1.2. Execução de *scripts* em Python
 - 1.1.3. Ferramentas de Desenvolvimento Integrado (IDEs)
- 1.2. Dados em Python
 - 1.2.1. Tipos primitivos (*int*, *float*, *str*)
 - 1.2.2. Conversão e *casting* de tipos de dados em Python
 - 1.2.3. Imutabilidade e armazenamento de dados em Python
- 1.3. Referências a objetos em Python
 - 1.3.1. Referências na memória
 - 1.3.2. Identidade X Igualdade
 - 1.3.3. Gestão de referências e coleta de lixo
- 1.4. Coleta de dados em Python
 - 1.4.1. Listas e operações comuns
 - 1.4.2. Tuplas e sua imutabilidade
 - 1.4.3. Dicionários e acesso a dados
- 1.5. Operações lógicas em Python
 - 1.5.1. Operadores booleanos
 - 1.5.2. Expressões condicionais
 - 1.5.3. *Short-Circuit Evaluation*
- 1.6. Operadores aritméticos em Python
 - 1.6.1. Operações aritméticas em Python
 - 1.6.2. Operadores de divisão
 - 1.6.3. Precedência e associatividade
- 1.7. Entrada/saída em Python
 - 1.7.1. Leitura de dados da entrada padrão
 - 1.7.2. Escrita de dados na saída padrão
 - 1.7.3. Gerenciamento de arquivos
- 1.8. Criação e chamada de funções em Python
 - 1.8.1. Sintaxe da função
 - 1.8.2. Parâmetros e argumentos
 - 1.8.3. Valores de retorno e funções anônimas



- 1.9. Uso de *strings* em Python
 - 1.9.1. Manipulação e formatação de *strings*
 - 1.9.2. Métodos comuns de *strings*
 - 1.9.3. Interpolação e *F-strings*
- 1.10. Tratamento de erros e exceções em Python
 - 1.10.1. Tipos comuns de exceções
 - 1.10.2. Blocos *try-except*
 - 1.10.3. Criação de exceções personalizadas

Módulo 2. Dados avançados e controle de fluxo com Python

- 2.1. Identificadores e palavras-chave em Python
 - 2.1.1. Regras para nomes de variáveis
 - 2.1.2. Palavras reservadas em Python
 - 2.1.3. Convenções de nomenclatura
- 2.2. Tipos integrais e booleanos em Python
 - 2.2.1. Tipos integrais
 - 2.2.2. Operações específicas de booleanos
 - 2.2.3. Conversões e representações
- 2.3. Tipos de ponto flutuante e números complexos em Python
 - 2.3.1. Precisão e representação
 - 2.3.2. Operações de ponto flutuante
 - 2.3.3. Uso de números complexos em cálculos
- 2.4. Formatação de *strings* e codificações em Python
 - 2.4.1. Métodos avançados de formatação
 - 2.4.2. Codificações *Unicode* e UTF-8
 - 2.4.3. Trabalho com caracteres especiais
- 2.5. Coleções: Tuplas, listas e dicionários em Python
 - 2.5.1. Comparação e contraste entre tipos
 - 2.5.2. Métodos específicos de cada tipo
 - 2.5.3. Eficiência e seleção do tipo apropriado
- 2.6. *Sets* e *Frozen Sets* em Python
 - 2.6.1. Criação e operações em *Sets*
 - 2.6.2. *Frozen Sets*
 - 2.6.3. Aplicações práticas e desempenho

- 2.7. Iteração e cópia de coleções em Python
 - 2.7.1. Bucles *for* e compreensões de lista
 - 2.7.2. Cópia superficial X Profunda
 - 2.7.3. Iteradores e geradores
- 2.8. Uso de funções *Lambda* em Python
 - 2.8.1. Sintaxe e criação de funções *Lambda*
 - 2.8.2. Aplicações em filtros e mapas
 - 2.8.3. Limitações e boas práticas
- 2.9. Estruturas de controle: Condicionais e loops em Python
 - 2.9.1. Estruturas *if-else* y *elif*
 - 2.9.2. Loops *while* e *for*
 - 2.9.3. Controle de fluxo com *break*, *continue* e *else*
- 2.10. Funções e métodos avançados em Python
 - 2.10.1. Funções recursivas
 - 2.10.2. Funções de ordem superior
 - 2.10.3. Decoradores de função

Módulo 3. Programação orientada a objetos (POO) em Python

- 3.1. Programação orientada a objetos (POO) em Python
 - 3.1.1. Classes e objetos
 - 3.1.2. Encapsulamento e abstração
 - 3.1.3. Programação orientada a objetos (POO) em Python
- 3.2. Criação de classes e objetos em Python
 - 3.2.1. Classes em POO em Python
 - 3.2.2. Instanciação e métodos de inicialização
 - 3.2.3. Atributos e métodos
- 3.3. Atributos e métodos em Python
 - 3.3.1. Atributos de instância X Classe
 - 3.3.2. Métodos de instância, classe e estáticos
 - 3.3.3. Encapsulamento e ocultação de informações
- 3.4. Herança e polimorfismo em Python
 - 3.4.1. Herança simples e múltipla
 - 3.4.2. Sobrescrita e extensão de método
 - 3.4.3. Polimorfismo e *Duck Typing*

- 3.5. Propriedades e acesso a atributos em Python
 - 3.5.1. *Getters* e *Setters*
 - 3.5.2. Decorador *@property*
 - 3.5.3. Controle de acesso e validação
- 3.6. Classes e coleções personalizadas em Python
 - 3.6.1. Criação de tipos de coleção
 - 3.6.2. Métodos especiais (*__len__*, *__getitem__*)
 - 3.6.3. Iteradores personalizados
- 3.7. Agregação e composição de classes em Python
 - 3.7.1. Relações entre classes
 - 3.7.2. Agregação X Composição
 - 3.7.3. Gestão do ciclo de vida de objetos
- 3.8. Uso de decoradores em classes em Python
 - 3.8.1. Decoradores para métodos
 - 3.8.2. Decoradores de classes
 - 3.8.3. Aplicações e casos de uso
- 3.9. Classes abstratas e métodos em Python
 - 3.9.1. Classes abstratas
 - 3.9.2. Métodos abstratos e implementação
 - 3.9.3. Uso de ABC (*Abstract Base Class*)
- 3.10. Exceções e tratamento de erros no POO em Python
 - 3.10.1. Exceções personalizadas em classes
 - 3.10.2. Tratamento de exceções em métodos
 - 3.10.3. Boas práticas em exceção e POO

Módulo 4. Desenvolvimento de aplicativos em Python

- 4.1. Arquitetura de aplicativos em Python
 - 4.1.1. Projeto de Software
 - 4.1.2. Padrões arquitetônicos comuns
 - 4.1.3. Avaliação de requisitos e necessidades
- 4.2. Projeto e modelagem de aplicativos em Python
 - 4.2.1. Uso de UML e diagramas
 - 4.2.2. Modelagem de dados e fluxos de informação
 - 4.2.3. Princípios SOLID e design modular

- 4.3. Gerenciamento de dependências e bibliotecas em Python
 - 4.3.1. Manuseio de pacotes com Pip
 - 4.3.2. Uso de ambientes virtuais
 - 4.3.3. Resolução de conflitos de dependências
- 4.4. Padrões de design no desenvolvimento em Python
 - 4.4.1. Padrões criativos, estruturais e comportamentais
 - 4.4.2. Aplicação prática de padrões
 - 4.4.3. Refatoração e padrões
- 4.5. Testes e *Depuração* em aplicações em Python
 - 4.5.1. Estratégias de *Testing* (Unitário, Integração)
 - 4.5.2. Uso de *Frameworks* de testes
 - 4.5.3. Técnicas de *Depuração* e ferramentas
- 4.6. Segurança e autenticação em Python
 - 4.6.1. Segurança em aplicativos
 - 4.6.2. Implementação de autenticação e autorização
 - 4.6.3. Prevenção de vulnerabilidades
- 4.7. Otimização e performance de aplicativos em Python
 - 4.7.1. Análise de desempenho
 - 4.7.2. Técnicas de otimização de código
 - 4.7.3. Gerenciamento eficiente de recursos e dados
- 4.8. Implantação e distribuição de aplicativos em Python
 - 4.8.1. Estratégias de implementação
 - 4.8.2. Uso de contêineres e orquestradores
 - 4.8.3. Distribuição e atualizações contínuas
- 4.9. Manutenção e atualizações em Python
 - 4.9.1. Gestão do ciclo de vida do software
 - 4.9.2. Estratégias de manutenção e refatoração
 - 4.9.3. Atualizações e migração do sistema
- 4.10. Documentação e suporte em Python
 - 4.10.1. Criação de documentação eficiente
 - 4.10.2. Ferramentas para a documentação
 - 4.10.3. Estratégias de suporte e comunicação com o usuário

Módulo 5. Desenvolvimento Web e Mobile com Python

- 5.1. Desenvolvimento Web com Python
 - 5.1.1. Estrutura e componentes de uma web
 - 5.1.2. Tecnologias em desenvolvimento web
 - 5.1.3. Tendências em desenvolvimento web
- 5.2. *Frameworks* web populares com Python
 - 5.2.1. *Django*, *Flask* e outras opções
 - 5.2.2. Comparação e seleção de *Frameworks*
 - 5.2.3. Integração com *Frontend*
- 5.3. Desenvolvimento *Frontend*: HTML, CSS e JavaScript com Python
 - 5.3.1. HTML e CSS
 - 5.3.2. JavaScript e manipulação de DOM
 - 5.3.3. *Frameworks* e bibliotecas *Frontend*
- 5.4. *Backend* e bancos de dados com Python
 - 5.4.1. Desenvolvimento *Backend* com Python
 - 5.4.2. Gerenciamento de bancos de dados relacionais e não relacionais
 - 5.4.3. Integração *Backend-Frontend*
- 5.5. APIs e serviços web com Python
 - 5.5.1. Projeto de APIs *RESTful*
 - 5.5.2. Implementação e documentação de APIs
 - 5.5.3. Consumo e segurança em APIs
- 5.6. Desenvolvimento Mobile com Python
 - 5.6.1. Plataformas de desenvolvimento mobile (nativas, híbridas)
 - 5.6.2. Ferramentas e ambientes de desenvolvimento
 - 5.6.3. Adaptação de aplicações para dispositivos móveis
- 5.7. Plataformas de desenvolvimento mobile com Python
 - 5.7.1. Android e iOS
 - 5.7.2. *Frameworks* para desenvolvimento cruzado
 - 5.7.3. Provas e *Deployment* em dispositivos móveis
- 5.8. Design e UX em aplicativos móveis com Python
 - 5.8.1. Design de interfaces móveis
 - 5.8.2. Usabilidade e experiência do usuário com Python
 - 5.8.3. Ferramentas de prototipagem e design

- 5.9. Testes e depuração em móveis com Python
 - 5.9.1. Estratégias de *Testing* em dispositivos móveis
 - 5.9.2. Ferramentas de depuração e monitoramento
 - 5.9.3. Automação de Testes
- 5.10. Publicação em lojas de aplicativos com Python
 - 5.10.1. Processo de publicação na App Store e no Google Play
 - 5.10.2. Conformidade e políticas de aplicativos
 - 5.10.3. Estratégias de marketing e promoção

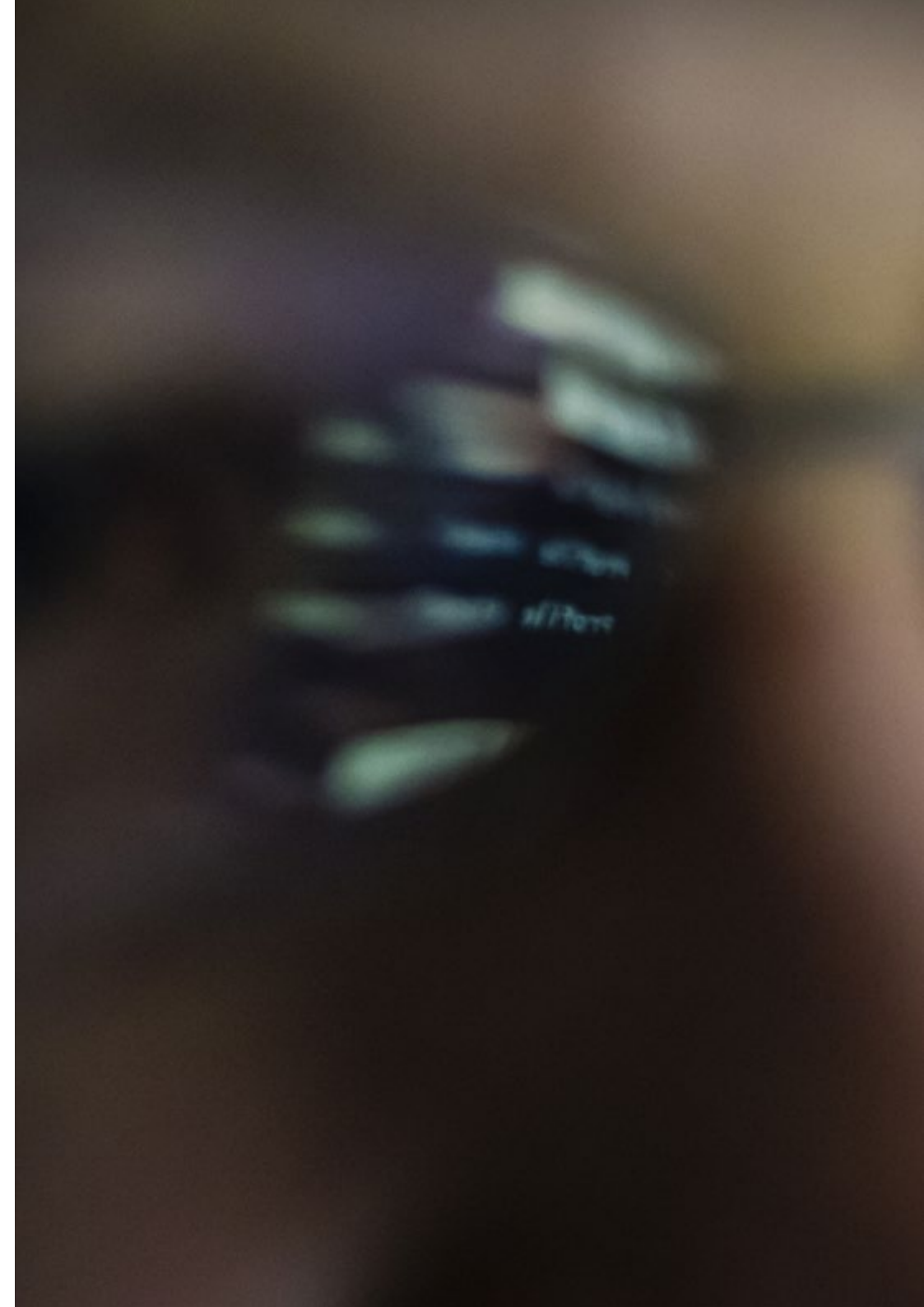
Módulo 6. Interface e experiência do usuário com Python


- 6.1. Design de interface do usuário com Python
 - 6.1.1. Design UI com Python
 - 6.1.2. Interação usuário-computador com Python
 - 6.1.3. Design centrado no usuário com Python
- 6.2. Ferramentas de design UI/UX com Python
 - 6.2.1. Software de design e prototipagem
 - 6.2.2. Ferramentas de colaboração e *Feedback*
 - 6.2.3. Integração do design ao processo de desenvolvimento
- 6.3. Design responsivo e adaptativo com Python
 - 6.3.1. Técnicas de design responsivo
 - 6.3.2. Adaptação a diferentes dispositivos e telas
 - 6.3.3. *Testing* e garantia da qualidade
- 6.4. Animações e transições com Python
 - 6.4.1. Criação de animações eficazes com Python
 - 6.4.2. Ferramentas e bibliotecas de animação
 - 6.4.3. Impacto na UX e no desempenho
- 6.5. Acessibilidade e usabilidade com Python
 - 6.5.1. Acessibilidade Web
 - 6.5.2. Ferramentas e técnicas de avaliação
 - 6.5.3. Implementação de melhores práticas
- 6.6. Prototipagem e *Wireframes* com Python
 - 6.6.1. Criação de *Wireframes* e Mockups
 - 6.6.2. Ferramentas de prototipagem rápida
 - 6.6.3. Testes de usabilidade e *Feedback*

- 6.7. Teste de usabilidade com Python
 - 6.7.1. Métodos e técnicas de teste de usabilidade
 - 6.7.2. Análise e aprimoramento baseados em resultados
 - 6.7.3. Ferramentas de teste de usabilidade
- 6.8. Análise do comportamento do usuário com Python
 - 6.8.1. Técnicas de análise e *Tracking*
 - 6.8.2. Interpretação de dados e métricas
 - 6.8.3. Melhoria contínua orientada por dados
- 6.9. Melhorias baseadas em *Feedback* com Python
 - 6.9.1. Gerenciamento e análise de *Feedback*
 - 6.9.2. Ciclos de *Feedback* e melhoria contínua
 - 6.9.3. Estratégias para implementar mudanças efetivas
- 6.10. Tendências futuras em UI/UX com Python
 - 6.10.1. Inovações e tendências emergentes
 - 6.10.2. Impacto das novas tecnologias na UI/UX
 - 6.10.3. Preparação para o futuro do design

Módulo 7. Processamento de Dados e *Big Data* com Python

- 7.1. Uso de Python em dados
 - 7.1.1. Python em ciência e análise de dados
 - 7.1.2. Bibliotecas de dados essenciais
 - 7.1.3. Aplicações e exemplos
- 7.2. Configuração do ambiente de desenvolvimento com Python
 - 7.2.1. Instalação de Python e ferramentas
 - 7.2.2. Configurações de ambientes virtuais
 - 7.2.3. Ferramentas de Desenvolvimento Integrado (IDE)
- 7.3. Variáveis, tipos de dados e operadores em Python
 - 7.3.1. Variáveis e tipos de dados primitivos
 - 7.3.2. Estruturas de dados
 - 7.3.3. Operadores aritméticos e lógicos
- 7.4. Controle de fluxo: Condicionais e loops
 - 7.4.1. Estruturas de controle condicionais (*if, else, elif*)
 - 7.4.2. Bucles (*for, while*) e controle de fluxo
 - 7.4.3. Compreensões de listas e expressões generativas



- 
- 7.5. Funções e modularidade com Python
 - 7.5.1. Uso de funções
 - 7.5.2. Parâmetros, argumentos e valores de retorno
 - 7.5.3. Modularidade e reutilização de código
 - 7.6. Tratamento de erros e exceções com Python
 - 7.6.1. erros e exceções
 - 7.6.2. Tratamento de exceções com *try-except*
 - 7.6.3. Criação de exceções personalizadas
 - 7.7. Ferramenta *IPython*
 - 7.7.1. Ferramenta *IPython*
 - 7.7.2. Uso do *IPython* para análise de dados
 - 7.7.3. Diferenças com o interpretador padrão do *Python*
 - 7.8. *Jupyter Notebooks*
 - 7.8.1. *Jupyter Notebooks*
 - 7.8.2. Uso de notebooks para análise de dados
 - 7.8.3. Publicação de notebooks *Jupyter*
 - 7.9. Melhores práticas de codificação em Python
 - 7.9.1. Estilo e convenções (PEP 8)
 - 7.9.2. Documentação e comentários
 - 7.9.3. Estratégias de teste e depuração
 - 7.10. Recursos e comunidades Python
 - 7.10.1. Recursos online e documentação
 - 7.10.2. Comunidades e fóruns
 - 7.10.3. Aprendizagem e atualização em Python

Módulo 8. Estruturas de dados e funções em Python

- 8.1. Conjuntos em Python
 - 8.1.1. Operações e métodos
 - 8.1.2. Diferenças e aplicação prática
 - 8.1.3. Iteração e compreensão
- 8.2. Dicionários e seu uso em Python
 - 8.2.1. Criação e manipulação de dicionário
 - 8.2.2. Acesso e gerenciamento de dados
 - 8.2.3. Padrões e técnicas avançadas

- 8.3. Entendimento de listas e dicionários em Python
 - 8.3.1. Sintaxe e exemplos
 - 8.3.2. Eficiência e legibilidade
 - 8.3.3. Aplicações práticas
- 8.4. Funções de dados em Python
 - 8.4.1. Criação de funções
 - 8.4.2. Escopo e espaço de nomes
 - 8.4.3. Funções anônimas e *Lambda*
- 8.5. Argumentos de função e retorno de valores em Python
 - 8.5.1. Argumentos posicionais e nomeados
 - 8.5.2. Valores de retorno múltiplos
 - 8.5.3. Argumentos variáveis e palavras-chave
- 8.6. Funções *Lambda* e funções de ordem superior em Python
 - 8.6.1. Uso das funções *Lambdas*
 - 8.6.2. Funções *Map*, *Filter* e *Reduce*
 - 8.6.3. Aplicações em processamento de dados
- 8.7. Tratamento de arquivos Python
 - 8.7.1. Leitura e escrita de arquivos
 - 8.7.2. Tratamento de arquivos binários e de texto
 - 8.7.3. Boas práticas e tratamento de exceções
- 8.8. Leitura e escrita de arquivos de texto e binários em Python
 - 8.8.1. Formatos de arquivo e codificação
 - 8.8.2. Manuseio de arquivos grandes
 - 8.8.3. Serialização e desserialização (*JSON*, *pickle*)
- 8.9. Contextos e operações com arquivos
 - 8.9.1. Uso do gerenciador de contexto(*with*)
 - 8.9.2. Técnicas de processamento de arquivos
 - 8.9.3. Segurança e tratamento de erros
- 8.10. Bibliotecas de modelagem em Python
 - 8.10.1. *Scikit-learn*
 - 8.10.2. *TensorFlow*
 - 8.10.3. *PyTorch*

Módulo 9. Gerenciamento de dados em Python com NumPy e Pandas

- 9.1. Criação e manipulação de *Arrays* em NumPy
 - 9.1.1. NumPy
 - 9.1.2. Operações básicas com *Arrays*
 - 9.1.3. Manipulação e transformação de *Arrays*
- 9.2. Operações vetorizadas com *Arrays*
 - 9.2.1. Vetorização
 - 9.2.2. Funções universais (*ufunc*)
 - 9.2.3. Eficiência e desempenho
- 9.3. Indexação e segmentação no NumPy
 - 9.3.1. Acesso a elementos e *Slicing*
 - 9.3.2. Indexação avançada e booleana
 - 9.3.3. Reordenação e seleção
- 9.4. Pandas *Series* e *DataFrames*
 - 9.4.1. Pandas
 - 9.4.2. Estruturas de dados no Pandas
 - 9.4.3. Manipulação de *DataFrames*
- 9.5. Indexação e seleção no Pandas
 - 9.5.1. Acesso a dados em série e *DataFrames*
 - 9.5.2. Métodos de seleção e filtragem
 - 9.5.3. Uso de *loc* e *iloc*
- 9.6. Operações com Pandas
 - 9.6.1. Operações aritméticas e alinhamento
 - 9.6.2. Funções de agregação e estatísticas
 - 9.6.3. Transformações e aplicação de funções
- 9.7. Manipulação de dados incompletos no Pandas
 - 9.7.1. Detecção e tratamento de valores nulos
 - 9.7.2. Preenchimento e exclusão de dados incompletos
 - 9.7.3. Estratégias para lidar com dados incompletos
- 9.8. Funções e aplicações do Pandas
 - 9.8.1. Concatenação e mescla de dados
 - 9.8.2. Agrupamento e agregação (*groupby*)
 - 9.8.3. *Pivot Tables* e *Crosstabs*

- 9.9. Visualização com Matplotlib
 - 9.9.1. Matplotlib
 - 9.9.2. Criação e personalização de gráficos
 - 9.9.3. Integração com Pandas
- 9.10. Personalização de gráficos no Matplotlib
 - 9.10.1. Estilos e configurações
 - 9.10.2. Gráficos avançados (*scatter*, *bar*, etc.)
 - 9.10.3. Criação de visualizações complexas

Módulo 10. Técnicas avançadas e aplicações práticas do NumPy e Pandas

- 10.1. Carregamento de dados de diferentes fontes
 - 10.1.1. Importação de CSV, Excel e bancos de dados
 - 10.1.2. Leitura de dados de APIs e Web
 - 10.1.3. Estratégias de gerenciamento de big data
- 10.2. Armazenamento de dados em Python
 - 10.2.1. Exportação para diferentes formatos
 - 10.2.2. Eficiência de armazenamento
 - 10.2.3. Segurança e privacidade de dados
- 10.3. Estratégias de limpeza de dados em Python
 - 10.3.1. Identificação e correção de inconsistências
 - 10.3.2. Normalização e transformação de dados
 - 10.3.3. Automação de processos de limpeza
- 10.4. Transformação avançada de dados no Pandas
 - 10.4.1. Técnicas de manipulação e processamento
 - 10.4.2. Combinação e reestruturação de *DataFrames*
 - 10.4.3. Uso de expressões regulares no Pandas
- 10.5. Combinação de *DataFrames* no Pandas
 - 10.5.1. *Merge*, *Join* e concatenação
 - 10.5.2. Gerenciamento de conflitos e fatores-chave
 - 10.5.3. Estratégias de combinação eficientes
- 10.6. Transformação avançada e pivotamento de dados no Pandas
 - 10.6.1. *Pivot* e *Melt*
 - 10.6.2. Técnicas de *Reshape* e transposição
 - 10.6.3. Aplicações em análise de dados

- 10.7. Séries temporais no Pandas
 - 10.7.1. Gerenciamento de data e hora
 - 10.7.2. *Resampling* e *Window Functions*
 - 10.7.3. Análise de tendências e sazonalidade
- 10.8. Gerenciamento avançado de índices no Pandas
 - 10.8.1. Índices multiníveis e hierárquicos
 - 10.8.2. Seleção e manipulação avançada
 - 10.8.3. Otimização de consultas
- 10.9. Estratégias de otimização de desempenho
 - 10.9.1. Melhorias na velocidade e na eficiência
 - 10.9.2. Uso de Cython e Numba
 - 10.9.3. Paralelização e processamento distribuído
- 10.10. Projetos práticos de manipulação de dados
 - 10.10.1. Desenvolvimento de exemplos de usos reais
 - 10.10.2. Integração de técnicas de Python
 - 10.10.3. Estratégias para resolver problemas complexos de dados



Esse programa inovador se adapta às suas necessidades para que você se mantenha atualizado com as últimas tendências e tecnologias, garantindo assim que você esteja na vanguarda da inovação em desenvolvimento de software"

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.



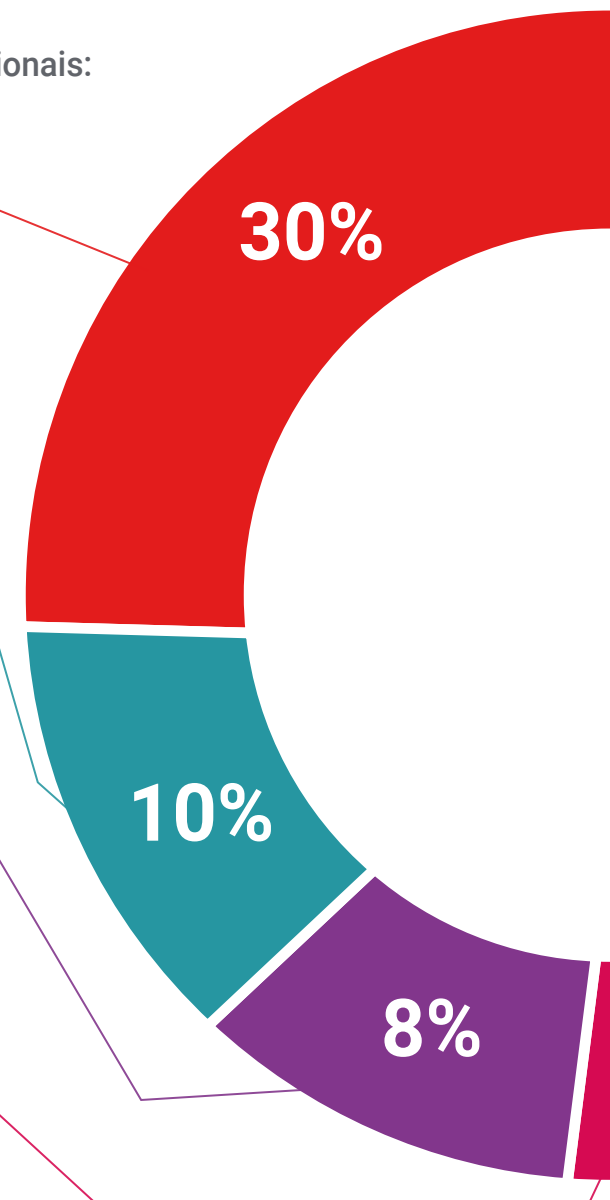
Práticas de habilidades e competências

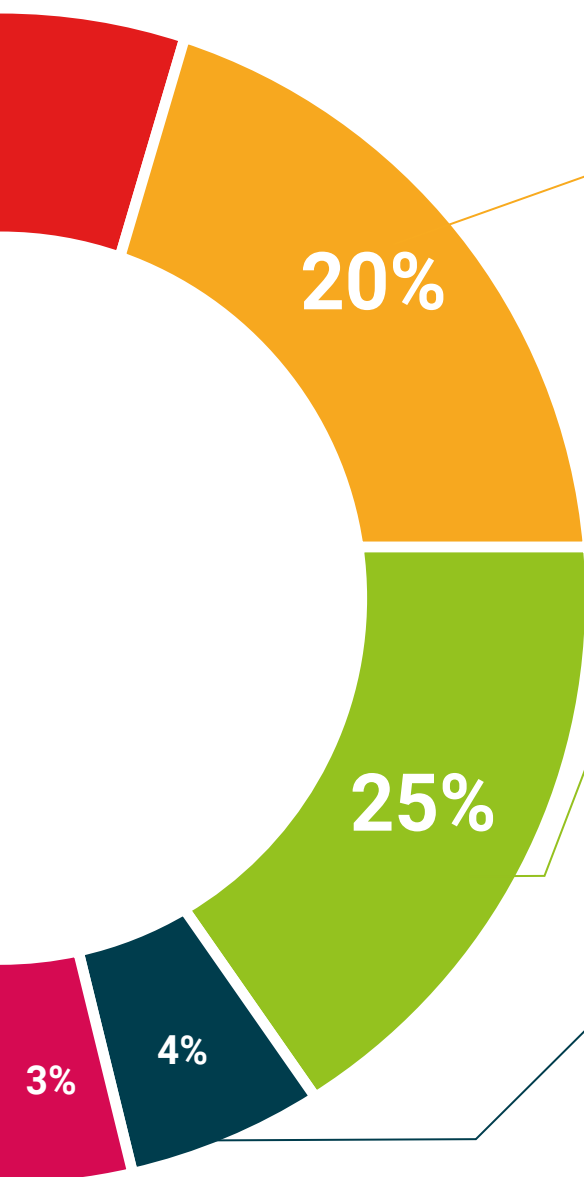
Serão realizadas atividades para desenvolver competências e habilidades específicas em cada área temática. Práticas e dinâmicas para adquirir e ampliar as competências e habilidades que um especialista precisa desenvolver no contexto globalizado em que vivemos.



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 em Desenvolvimento em Python garante, além da capacitação mais rigorosa e atualizada, o acesso a um título de Mestrado emitido pela TECH Global University.



“

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

Este programa permitirá a obtenção do certificado de **Mestrado em Desenvolvimento em Python** reconhecido pela **TECH Global University**, a maior universidade digital do mundo.

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

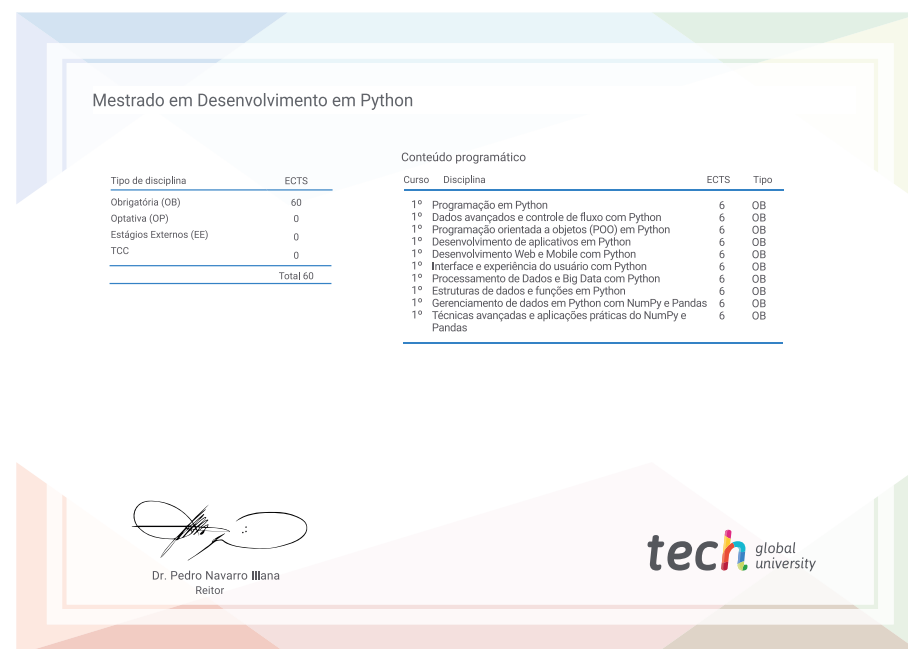
Esse título próprio da **TECH Global University**, é um programa europeu de formação contínua e atualização profissional que garante a aquisição de competências em sua área de conhecimento, conferindo um alto valor curricular ao aluno que conclui o programa.

Título: **Mestrado em Desenvolvimento em Python**

Modalidade: **online**

Duração: **12 meses**

Créditos: **60 ECTS**





Mestrado

Desenvolvimento em Python

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

Mestrado

Desenvolvimento em Python