

CIÊNCIA DE
DADOS

PYTHON AVANÇADO PARA CIÊNCIA DE DADOS

FICHA DO CURSO

Curso: Python Avançado para Ciência de Dados

Modalidade: EAD assíncrono

Carga horária total: 40 horas

Carga horária semanal: 4 horas

Início da oferta: EAD

Fim da oferta: EAD

Pré-requisito: Recomenda-se ter conhecimento básico em Python

Professores: André Ricardo Abed Grégio e Paulo Ricardo Lisboa de Almeida

Objetivos

Este curso é fundamental para profissionais que desejam se destacar na área de ciência de dados utilizando Python como ferramenta principal. Ao adquirir habilidades avançadas em manipulação de dados, organização de código, programação eficiente e interoperabilidade com outras linguagens, os alunos estarão aptos a realizar análises de dados complexas e desenvolver soluções eficazes em diversos contextos profissionais.

Ao final do curso o participante será capaz de desenvolver soluções com a linguagem de forma que sejam eficientes em termos de programação, fáceis de manter e adequar para problemas mais complexos.

Programa

O curso é composto de 3 tópicos descritos a seguir.

Tópico 1: Python para Processamento de Dados Tabulares

- **Importação e Exportação:** Capacitar os alunos a utilizar a biblioteca *Pandas* para importar e exportar dados em diversos formatos, proporcionando habilidades essenciais para a manipulação de conjuntos de dados.
- **Valores Ausentes e Transformação de Valores:** Desenvolver competências para lidar com valores ausentes, realizar transformações em dados e preparar conjuntos de dados para análises subsequentes.
- **Processamento de Variáveis Categóricas e Cronológicas:** Explorar técnicas específicas para o processamento de variáveis categóricas e cronológicas, presente

na grande maioria das situações de análise de dados e, por isso, fundamentais para profissionais da área.

- **Processamento de Texto e Expressões Regulares:** Introduzir o processamento de texto com *Pandas*, incluindo o uso de expressões regulares, para realizar manipulações avançadas em dados de texto, o que permite a organização, padronização e extração de informações de variáveis de texto.
- **Pivotagem, Junções e Concatenações de Tabelas:** Capacitar os alunos a realizar operações avançadas de transformação e combinação de tabelas, como pivotagem para alterar o eixo da tabela que se percorre para examinar as variáveis, juntar dados de tabelas separadas que tenham chave identificadora e concatenar tabelas.
- **Medidas Descritivas e Agregação:** Desenvolver habilidades para calcular medidas descritivas essenciais para compreensão e comunicação dos dados bem como realizar agregações de dados por estratos, fornecendo insights cruciais durante as análises exploratórias.
- **Visualização de Dados:** Introduzir técnicas de visualização de dados com bibliotecas como *Matplotlib* e *Seaborn*, permitindo aos alunos comunicar efetivamente os resultados das análises.
- **Produção de Relatórios com Quarto:** Capacitar os alunos a criar relatórios interativos e dinâmicos utilizando a biblioteca Quarto, promovendo a comunicação eficaz dos resultados da análise.

Tópico 2: Python Organizado e Colaborativo

- **Criação de Módulos:** Introduzir práticas eficazes para a criação e organização de módulos em Python, facilitando a modularidade e reutilização de código.
- **Documentação de Funções:** Desenvolver boas práticas de documentação para funções e módulos, garantindo a clareza e a compreensão do código.
- **Testes de Unidade:** Capacitar os alunos a realizar testes de unidade em seus códigos, garantindo a qualidade e a robustez das implementações.
- **Programação Orientada a Objetos:** Introduzir conceitos de programação orientada a objetos (POO) em Python, permitindo aos alunos estruturar e organizar seus códigos de forma mais eficiente.
- **Técnicas de Debug e Profile:** Desenvolver habilidades avançadas de depuração e perfilamento de código, permitindo aos alunos identificar e corrigir eficientemente problemas de desempenho.

Tópico 3: Programação de Alta Performance com Python

- **Execução Paralelizada:** Explorar técnicas de execução paralela em Python, permitindo aos alunos acelerar a execução de tarefas intensivas computacionalmente.
- **Programação Funcional:** Introduzir os conceitos de programação funcional em Python, incluindo o uso de funções de ordem superior, map, filter e reduce.
- **Conexão com C e C++:** Capacitar os alunos a integrar código em C e C++ em projetos Python, proporcionando melhor desempenho em operações críticas.
- **Interoperabilidade com R:** Explorar estratégias de interoperabilidade entre Python e R, permitindo aos alunos aproveitar as vantagens de ambas as linguagens em suas análises de dados.

Conteúdo Programático

O programa do curso foi dividido em 10 unidades didáticas (UD), sendo que cada UD ocorre durante uma semana. Os temas de cada UD de acordo com os tópicos do curso são apresentados na Tabela 5.

Tabela 5 - Cronograma detalhado do conteúdo das unidades didáticas.

Tópico	UD	Conteúdo programado	Duração
1	1	Importação, Exportação e Manipulação de dados	4 horas
	2	Operações com REGEX, fatores e datas	4 horas
	3	Conciliação e Agregação de Dados	4 horas
	4	Visualização de dados e Relatórios	4 horas
2	5	Criação de Módulos	4 horas
	6	Testes de Unidade e Debug	4 horas
	7	Programação Orientada à Objetos 1	4 horas
	8	Programação Orientada à Objetos 2	4 horas
3	9	Execução Paralelizada e Programação Funcional	4 horas
	10	Interoperabilidade com C/C++ e R.	4 horas
Encerramento do curso			

Procedimentos didáticos e metodológicos

O conteúdo do curso será estruturado em 10 unidades didáticas. A cada unidade didática serão desenvolvidas as seguintes atividades pelos cursistas.

- Assistir ao conteúdo em vídeo da unidade didática.
- Acessar o material em texto e demais arquivos (slides, relatórios, scripts, datasets) que acompanham a unidade didática.
- Fazer o estudo individual com os materiais disponibilizados como: leituras complementares e acesso a vídeos adicionais.
- Realizar as atividades de avaliação do módulo no formato de quiz.

Desempenho no curso

O desempenho no curso será determinado pela nota das atividades avaliativas aplicadas aos cursistas nas 10 unidades didáticas. Requer-se, no mínimo, 70% de aproveitamento para obtenção de certificado. Aproveitamento este calculado como a média das notas obtidas nos quizzes de cada uma das unidades didáticas.

Estratégias de Monitoramento e Retenção

Para monitorar a participação e desempenho dos cursistas têm-se as seguintes estratégias:

- Monitoramento de acesso e consumo das videoaulas e materiais complementares.
- Sistema de monitoramento da conclusão de cada uma das tarefas que compõem o curso dentro da plataforma EaD.
- Monitoramento da realização dos quizzes.

Avaliação de reação dos cursistas

Ao final do curso será realizada uma avaliação de reação dos cursistas.

Referências bibliográficas

1. Beuzen, T., & Timbers, T. (2022). **Python packages**. Chapman & Hall/CRC.
2. Dale, K. (2022). **Data visualization with python and JavaScript 2e**: Scrape, clean, explore, and transform your data (2nd ed.). O'Reilly Media.
3. Gorelick, M., & Ozsvald, I. (2020). **High performance python**: Practical performant programming for humans (2nd ed.). O'Reilly Media.
4. Lott, S. F. (2019). **Mastering Object-Oriented Python**: Build powerful applications with reusable code using OOP design patterns and Python 3.7, 2nd Edition (2nd ed.). Packt Publishing.
5. Lott, S. F., & Banffy, R. (2022). **Functional Python Programming**: Use a functional approach to write succinct, expressive, and efficient Python code (3rd ed.). Packt Publishing.
6. McKinney, W. (2022). **Python for Data Analysis** (3rd ed.). O'Reilly Media.
7. Pajankar, A. (2020). **Practical python data visualization**: A fast track approach to learning data visualization with python (1st ed.). APress.
8. Paskhover, B. (2021). **Pandas in action**. Manning Publications.
9. Wilkes, M. (2020). **Advanced python development**: Using powerful language features in real-world applications (1st ed.). APress.

Informações sobre os conteudistas

André Ricardo Abed Grégio. Bacharel em Ciência da Computação pela Universidade Estadual Paulista Júlio de Mesquita Filho (2004), Mestre em Computação Aplicada pelo Instituto Nacional de Pesquisas Espaciais (2007) e Doutor em Engenharia Elétrica pela Universidade Estadual de Campinas (2012), com período sanduíche na UCSB - University of California, Santa Barbara. Professor Adjunto do Departamento de Informática (DInf) na Universidade Federal do Paraná (UFPR). Pesquisador Colaborador do Laboratório de Segurança e Criptografia no Instituto de Computação da Universidade Estadual de Campinas (UNICAMP). Foi Tecnologista Pleno do Centro de Tecnologia da Informação Renato Archer (CTI/MCTI) em Campinas/SP entre 2009 e 2016. Áreas e interesses de pesquisa: Segurança Computacional em geral; Comportamento de malware; Sistemas para análise de malware; Visualização, análise e mineração de dados de segurança; Segurança de dispositivos móveis.

Paulo Ricardo Lisboa de Almeida. Possui graduação em Engenharia da Computação pela Universidade Estadual de Ponta Grossa (2011), Mestrado em Informática (2013) e

Doutorado em Ciência da Computação (2017), ambos pela Universidade Federal do Paraná. Tem experiência em: Orientação a Objetos, Bancos de Dados Relacionais, Inteligência Computacional, Reconhecimento de Padrões, Processamento de Imagens, Sistemas Operacionais e Arquitetura de Computadores. Principais interesses de pesquisa: Reconhecimento de Padrões, Ensembles de Classificadores, Seleção Dinâmica de Classificadores e Concept Drift.

Requisitos Técnicos

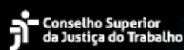
Computador com acesso à internet. Permissão para instalar programas.

CIÊNCIA DE DADOS



JUSTIÇA 4.0

APOIO



JUSTIÇA FEDERAL
Conselho da Justiça Federal



REALIZAÇÃO



CONSELHO
NACIONAL
DE JUSTIÇA