| Curso: **Análise e Desenvolvimento de Sistemas, Engenharia da Computação e Sistemas de Informação** | | |
| --- | --- | --- |
| Nome da Disciplina: **Programação Orientada a Objetos** | | |
| Carga Horária*:* **80 horas** | Aulas: **Teóricas-50% / Práticas-50%** | |
| Docente: **Prof. MSc. Rafael Maximo Carreira Ribeiro** | | |
| Coordenação: **Profa. Dra. Ana Cristina dos Santos** | | |
| **Competências** | | |
| * Desenvolver lógica de raciocínio para organizar os elementos necessários para o desenvolvimento de programas orientado a objetos. * Fornecer as bases para a modelagem e o desenvolvimento de software orientado a objetos. * Aplicar os conceitos da orientação a objetos na linguagem de programação Python. | | |
| **Habilidades** | | |
| * Desenvolver software orientado a objetos. * Persistir dados em arquivos. * Usar um ambiente integrado de desenvolvimento (IDE) Visual Studio Code. * Codificar utilizando boas práticas de programação e segundo o guia de estilo da linguagem. * Utilizar testes de unidade automatizados para aumento de qualidade do software desenvolvido. | | |
| **Disciplinas Relacionadas** | | |
| * **Anteriores:** Lógica de Programação; Linguagem de Programação * **Paralelas:** Desenvolvimento Web; Engenharia de Software; Linguagem SQL * **Posteriores:** Desenvolvimento de APIs e Microsserviços; Práticas e Desenvolvimento e Operações - DevOps | | |

| **Conteúdo Programático** | | |
| --- | --- | --- |
| * Unidade 01 - Introdução, ambiente de desenvolvimento e revisão da estrutura básica de Python:   + Aula 1 - Apresentação da disciplina e preparação do ambiente de desenvolvimento;   + Aula 2 - Revisão de Python básico;   + Aula 3 - Listas e Dicionários em Python. * Unidade 02 - Programação orientada a objetos (POO):   + Aula 4 - Paradigmas de programação e os pilares de POO;   + Aula 5 - Criação de classes em Python e Encapsulamento;   + Aula 6 - Herança e Polimorfismo em Python. * Unidade 03 - Técnicas e recursos adicionais para desenvolvimento de software:   + Aula 7 - Criação de módulos e pacotes;   + Aula 8 - Ambientes virtuais, utilização de bibliotecas de terceiros e gerenciamento de dependências;   + Aula 9 - Introdução ao Desenvolvimento Guiado por Testes (TDD); testes unitários e tratamento de exceções. * Unidade 04 - Tópicos avançados de programação:   + Aula 10 - Manipulação de arquivos com Python;   + Aula 11 - Decoradores e classes abstratas;   + Aula 12 - Introdução a Padrões de Projeto e aos princípios do SOLID. | | |

| **Metodologia de ensino** | | |
| --- | --- | --- |
| * Aulas gravadas nas quais os tópicos da disciplina são apresentados e discutidos. * Encontros regulares ao vivo, realizados por uma plataforma de videoconferência, para aprofundamento do conteúdo e esclarecimento de dúvidas. * Indicação de material extra para leitura. * Atividades contínuas (AC) para acompanhamento do processo de ensino e aprendizagem. | | |

| **Bibliografia Básica** |
| --- |
| * RAMALHO, L. **Python Fluente**. São Paulo: Novatec, 2015. * DIERBACH, C. **Introduction to Computer Science Using Python: A Computational Problem-Solving Focus**. 1st Edition, New York: Wiley, 2012. * GIRIDHAR C. **Aprendendo Padrões de Projeto em Python**. São Paulo: Novatec, 2016. * PHILLIPS, Dusty. **Python 3 Object Oriented Programming: Harness the power of Python 3 objects.** Birmingham: Packt Publishing, 2010. * GOODRICH, M.T. et al. **Data Structures and Algorithms** **in Python.** New York: Wiley, 2013. |

| **Bibliografia Complementar** |
| --- |
| * SHVETS, A. **Mergulho nos Padrões de Projeto**. e-book Refactoring.Guru, 2021. * MENEZES, N.N.C. **Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes**. 2 a. ed. São Paulo: Novatec, 2014. * LUTZ, M. **Programming Python: Powerful Object-Oriented Programming**. 4th ed. Sebastopol CA: O´Reilly, 2011. * GUTTAG, J.V. **Introduction to Computation and Programming Using Python**: With Application to Understanding Data. 2nd ed. Cambrigde MA: The MIT Press, 2016. * ZELLE J. **Python Programming: An Introduction to Computer Science**, 2nd Ed. 2 edition. Sherwood, Or: Franklin, Beedle & Associates Inc.; 2010. |

| **Plano de aulas** | |
| --- | --- |
| **Parte** | **Conteúdo** |
| 1 | Apresentação da disciplina e preparação do ambiente de desenvolvimento. |
| 2 | Revisão de Python básico. |
| 3 | Listas e Dicionários em Python. |
| 4 | Paradigmas de programação e os pilares de POO |
| 5 | Criação de classes e Encapsulamento |
| 6 | Herança e Polimorfismo. |
| 7 | Criação de módulos e pacotes |
| 8 | Ambientes virtuais, utilização de bibliotecas de terceiros e gerenciamento de dependências |
| 9 | Introdução ao Desenvolvimento Guiado por Testes (TDD), testes unitários e tratamento de exceções. |
| 10 | Manipulação de arquivos com Python |
| 11 | Decoradores e classes abstratas |
| 12 | Introdução a Padrões de Projeto e aos princípios do SOLID |