**Plano de Aulas: Programação Python com Orientação a Objetos**

**Módulo 1: Introdução à Programação e Histórico das Linguagens Orientadas a Objetos**

**Carga Horária: 4 horas**

**Conteúdo:**

Breve histórico das linguagens de programação.

Surgimento e evolução das linguagens orientadas a objetos.

Paradigmas de programação: procedural vs. orientado a objetos.

Introdução ao Python: instalação, ambiente de desenvolvimento (IDEs, Jupyter Notebook, etc.).

Primeiros passos em Python: sintaxe básica, tipos de dados e operações simples.

**Módulo 2: Conceitos Básicos de Programação em Python**

**Carga Horária: 8 horas**

**Conteúdo:**

Variáveis e tipos de dados.

Estruturas de controle: condicionais (if, else, elif) e loops (for, while).

Entrada e saída de dados (input, print).

Operadores e expressões (aritméticos, lógicos, relacionais).

Funções básicas: definição, parâmetros e retorno.

**Módulo 3: Introdução à Orientação a Objetos**

**Carga Horária: 8 horas**

**Conteúdo:**

Conceitos de POO: classes, objetos, encapsulamento.

Diferença entre classe e objeto.

Atributos e métodos.

Criação de classes e instanciação de objetos em Python.

Exemplos práticos: modelagem de problemas do mundo real usando POO.

**Módulo 4: Classes e Objetos em Profundidade**

**Carga Horária: 8 horas**

**Conteúdo:**

Construtores (\_\_init\_\_) e destrutores (\_\_del\_\_).

Sobrecarga de métodos (conceito e exemplos).

Métodos especiais em Python (\_\_str\_\_, \_\_repr\_\_, etc.).

Atributos de classe vs. atributos de instância.

Exercícios práticos: criação de classes para modelagem de sistemas.

**Módulo 5: Controle de Acesso e Encapsulamento**

**Carga Horária: 6 horas**

**Conteúdo:**

Modificadores de acesso: público, privado e protegido.

Uso de \_ e \_\_ em Python para controle de acesso.

Encapsulamento e boas práticas.

Propriedades (@property) e métodos getters/setters.

Exercícios: aplicação de encapsulamento em projetos.

**Módulo 6: Herança e Polimorfismo**

**Carga Horária: 8 horas**

**Conteúdo:**

Conceito de herança e sua importância em POO.

Herança simples e múltipla em Python.

Sobrescrita de métodos.

Polimorfismo: conceito e exemplos.

Classes abstratas e interfaces (usando abc module).

Exercícios: criação de hierarquias de classes.

**Módulo 7: Composição e Relacionamento entre Classes**

**Carga Horária: 6 horas**

**Conteúdo:**

Composição vs. herança.

Relacionamentos entre classes: associação, agregação e composição.

Exemplos práticos: modelagem de sistemas com composição.

Boas práticas de design de classes.

**Módulo 8: Projeto Final Integrador**

**Carga Horária: 8 horas**

**Conteúdo:**

Desenvolvimento de um projeto prático que integre todos os conceitos aprendidos.

Exemplos de projetos:

Sistema de gerenciamento de biblioteca.

Simulação de um banco com contas e transações.

Jogo simples com personagens e interações.

Apresentação e revisão do código.