Programação em Python (com Orientação a Objetos)

Pré-requisitos:

- lógica de programação

Carga horária:
120h

Descrição

Após aprenderem os fundamentos de lógica de programação e utilizarem JSON como recurso para simular um banco de dados, os alunos serão desafiados a aplicar os conhecimentos adquiridos em um projeto prático. O recurso consiste no desenvolvimento de um **sistema em Python, no formato de aplicação em console, utilizando os conceitos de Orientação a Objetos (POO)**.

O sistema terá como objetivo o gerenciamento de livros em uma biblioteca. Os alunos deverão estruturar o código em classes e objetos, aplicando encapsulamento, métodos e construtores. A proposta é motivadora porque conecta os conceitos estudados à construção de uma solução funcional que pode ser ampliada e utilizada em situações reais.

Sequência e Descrição da Estratégia de Aprendizagem

Aluno:

- Desenvolver um sistema funcional em console para gerenciamento de livros de uma biblioteca.
- Estruturar o sistema com **classes** (por exemplo: `Livro`, `Biblioteca`, `App`).
- Implementar operações básicas de CRUD (cadastrar, listar, atualizar e excluir livros).
- Armazenar os dados em arquivos JSON, simulando um banco de dados.
- Organizar o código com boas práticas de programação e aplicar princípios básicos de POO.
- Documentar o funcionamento do sistema.

Docente:

- Orientar os alunos na definição das etapas do projeto.
- Esclarecer dúvidas técnicas sobre POO e manipulação de JSON.
- Incentivar a aplicação dos conceitos de encapsulamento, atributos, métodos, objetos e classes.
- Acompanhar o progresso do trabalho, realizando intervenções quando necessário.

Contextualização e Desafio

Criar um sistema de gerenciamento de biblioteca em Python, no formato console, que utilize JSON para simular o armazenamento dos dados.

O sistema deverá permitir que o usuário realize as principais operações de CRUD sobre os livros. O desafio é aplicar **lógica, programação estruturada e POO**, resultando em um projeto funcional e bem organizado.

Estratégia de Avaliação, Resultados Esperados e Entregas

Técnicas de Avaliação:

- observação, autoavaliação, revisão por pares.

Instrumentos de Avaliação:

- código-fonte publicado em repositório público.
- apresentação oral do funcionamento do sistema.

Resultados Esperados:

- aluno capaz de aplicar raciocínio lógico.
- utilização de classes, objetos, atributos e métodos em Python.
- manipulação de dados em JSON para simular persistência.
- organização do código com boas práticas de desenvolvimento.

Entregas:

- código-fonte do sistema com funcionalidades implementadas.
- código limpo e documentado.
- link para repositório público no GitHub.

Recursos e Ambientes Pedagógicos

- **Ambiente de desenvolvimento:** Visual Studio Code.
- **Controle de versão:** Git e GitHub.
- **Computador com Python instalado.**

- **Aulas em laboratório de informática** ou em ambiente remoto com os softwares necessários.

Estratégias de Aprendizagem

- Projeto prático: desenvolvimento de aplicação em console com POO.
- Exercícios intermediários para fixação de POO antes do projeto final (ex: criar classes simples como `Carro`, `Aluno`, `Produto`).
- Desenvolvimento incremental do projeto (cada aula, implementar uma nova funcionalidade).