LIPAI
Onboarding

Projeto 1

Semana 4 e 5 / Projeto 1

Nesta atividade, você vai consolidar tudo o que aprendeu até aqui em um projeto completo em Python. A ideia é sair de exercícios isolados e construir uma aplicação mais estruturada, usando entrada e saída de dados, leitura e escrita de arquivos, estruturas de controle, funções e Programação Orientada a Objetos (POO), organizada em múltiplos módulos e versionada em um repositório no GitHub.

Objetivos

Ao final desta atividade de projetos, você deverá ser capaz de:

- Integrar os conteúdos de:
    - introdução ao Python
    - entrada e saída de dados
    - estruturas de controle
    - funções
    - manipulação de arquivos
    - orientação a objetos
- Organizar um projeto em múltiplos arquivos (módulos), com classes e funções bem definidas.
- Utilizar leitura e escrita de arquivos para persistir dados da aplicação.

- Versionar o código em um repositório dedicado no GitHub, com README descritivo.
- Preparar uma apresentação técnica em PDF, explicando o sistema, o código e suas funcionalidades.

Regras Gerais dos Projetos

- Cada projeto terá um código identificador representado por uma letra maiúscula:
    - Projeto A, Projeto B, Projeto C, …
- Cada aluno será responsável por um projeto específico, de acordo com o código atribuído (A, B, C etc.).
- Cada projeto deve ter um repositório dedicado no GitHub, por exemplo:
    - biblioteca-escolar, financas-pessoais, projeto1-lipai etc.
- O código deve estar modularizado em diferentes arquivos, por exemplo:
    - main.py (ponto de entrada, menu, interação com o usuário)
    - models.py (classes principais)
    - repositorios.py ou services.py (funções de leitura/escrita de arquivos, regras de negócio)
    - outros módulos conforme necessidade.
- Todo projeto deve envolver leitura e escrita de arquivos, por exemplo:
    - arquivos .csv ou .txt dentro de um diretório data/.
- O projeto deve usar, de forma explícita:
    - entrada de dados (input() ou equivalente)
    - saída de dados (impressões formatadas para o usuário)
    - estruturas de controle (if, for, while, etc.)

- funções bem definidas e reutilizáveis
- classes (POO), com atributos e métodos
- pelo menos uma relação entre objetos (por exemplo, um Projeto com várias Participações, um Curso com várias Matrículas, etc.).

Entrega 1 – Código e Repositório (Primeira Semana)

- Na primeira semana, você deverá entregar:
  - Arquivo PDF com:
    - Seu nome completo.
    - O link do repositório no GitHub.
    - Todo o código do seu projeto
  - Repositório no GitHub contendo:
    - Todo o código do projeto.
    - Um arquivo README.md com:
      - Título do projeto e código (ex.: Projeto A – Sistema de Biblioteca Escolar).
      - Descrição breve do problema que o sistema resolve.
      - Principais funcionalidades implementadas.
      - Como executar o projeto (passo a passo, incluindo dependências se houver).
      - Estrutura de diretórios (por exemplo: src/, data/, etc.).
      - o README deve ser claro o suficiente para que outra pessoa consiga clonar o repositório e rodar o projeto apenas seguindo as instruções.

Entrega 2 – Apresentação em PDF (Segunda Semana)

Na segunda semana, você deverá entregar uma apresentação em PDF (slides) sobre o seu projeto e apresentar em um encontro online para todos os participantes do Onboarding.

A apresentação deve conter, pelo menos:

1. Slide de capa
    - Título do projeto.
    - Código do projeto (A, B, C, …).
    - Nome do aluno.
    - Curso/turma/semestre.
2. Descrição do Projeto
    - Problema que o sistema resolve (ex.: gerenciar empréstimos, controlar estoque, registrar participações, etc.).
    - Público ou contexto em que o sistema poderia ser usado.
3. Demonstração das Funcionalidades (com prints de tela)
    - Prints do terminal ou interface mostrando:
        - Menu principal.
        - Cadastro/listagem/edição das entidades principais (ex.: alunos, livros, projetos, transações, etc.).
        - Exemplos de mensagens de erro/validação (quando o usuário digita algo inválido).
    - Cada print deve ter uma legenda curta explicando o que está acontecendo.

4. Organização do Código e Módulos
    - Um slide mostrando a estrutura de pastas e arquivos (por exemplo, um print do VS Code com o explorador de arquivos).
    - Explicar brevemente a função de cada módulo principal:
        - main.py – ponto de entrada, menus, orquestração.
        - models.py – definição das classes e relacionamentos.
        - repositorios.py – leitura/escrita de arquivos e persistência de dados.
        - outros módulos, se houver (por exemplo, relatorios.py, utils.py).
5. Classes e Orientação a Objetos
    - Slides com prints de trechos de código mostrando:
        - Duas ou mais classes principais (atributos e métodos).
        - Uso de propriedades (quando houver) para validar dados.
        - Exemplo de relacionamento entre objetos (por exemplo, um objeto Projeto contendo uma lista de Participacao ou um Curso contendo Matricula).
    - Em cada trecho de código, adicionar comentários na própria apresentação explicando:
        - O que essa classe representa.
        - Por que esse método é importante.
6. Leitura e Escrita de Arquivos.
    - Mostrar:
        - Um exemplo do formato dos arquivos de dados (.csv ou .txt) – com print do arquivo ou do conteúdo.

        - Um trecho de código que lê os dados do arquivo e um que escreve (salva) os dados.
    - Explicar rapidamente:
        - Quando os dados são carregados (início do programa, ao entrar em um menu, etc.).
        - Quando os dados são salvos (ao sair, ao adicionar algo, etc.).
7. Desafios e Aprendizados
    - Um slide com:
        - Principais dificuldades enfrentadas (por exemplo: organizar módulos, tratar erros, fazer leitura de arquivo, etc.).
        - O que você aprendeu com o projeto.
        - O que você faria de forma diferente se pudesse recomeçar.
8. Possíveis Melhorias Futuras
    - Funcionalidades que não foram implementadas, mas seriam interessantes.
    - Ideias de evolução do código (melhor modularização, testes automatizados, interface gráfica, etc.).

Sua apresentação deve ter no máximo 5 minutos.

Essas regras valem para todos os projetos, independentemente do código (A, B, C ..).