

# Atividade Prática 2

---

Você foi contratado para desenvolver um pequeno sistema de cadastro de membros da comunidade escolar. Cada membro possui nome, idade e CPF. No entanto, diferentes tipos de membros possuem informações e comportamentos distintos:

- Aluno: possui matrícula e curso.
- Professor: possui SIAPE (código funcional) e disciplina que leciona.
- Funcionário Técnico-Administrativo: possui setor de trabalho e cargo.

## Etapas de desenvolvimento:

### 1. Classe Base:

- Implemente uma classe abstrata chamada Pessoa, contendo os atributos comuns (nome, idade e cpf) com visibilidade privada.
- Crie os métodos getters e setters para esses atributos.
- Sobrescreva os métodos toString(), equals() e hashCode() na classe Pessoa.

### 2. Classes Derivadas:

- Crie as classes Aluno, Professor e TecnicoAdministrativo, todas herdando da classe Pessoa.
- Cada classe deve conter seus próprios atributos específicos:
  - i. Aluno: matrícula e curso.
  - ii. Professor: SIAPE e disciplina.
  - iii. TecnicoAdministrativo: setor de trabalho e cargo.
- Implemente um método exibirDados() em cada classe, sobrescrevendo o método da classe base, para exibir uma descrição completa do membro.
- Em uma das classes derivadas, adicione um atributo extra e um método específico de sua escolha.

### 3. Encapsulamento e Sobrescrita:

- Todos os atributos devem ser privados.
- Use getters e setters para acesso e modificação dos dados.
- As classes derivadas também devem sobrescrever os métodos toString(), equals() e hashCode().

4. Critério de Armazenamento:

- O aluno desenvolvedor deverá somar os dígitos de sua matrícula e calcular o módulo 3 do valor obtido.
- Com base no resultado, use uma das seguintes coleções para armazenar os objetos:
  - i. Resultado 0: utilize a interface List.
  - ii. Resultado 1: utilize a interface Set para evitar duplicatas.
  - iii. Resultado 2: utilize a interface Map, associando o CPF ao objeto Pessoa.

5. Classe Principal (Main):

- Exiba, no início do programa, o cálculo que determinou a estrutura de dados utilizada.
- Instancie pelo menos um objeto de cada tipo (Aluno, Professor, TecnicoAdministrativo).
- Atribua valores realistas inspirados no contexto da UniFBV.
- Chame os métodos `exibirDados()`, `toString()`, `equals()` e `hashCode()` para cada objeto criado.
- Adicione os objetos à coleção definida pelo critério da matrícula.
- Recupere e exiba os objetos da coleção para validar o funcionamento completo do sistema.

### Entrega da Atividade

O código-fonte desenvolvido deve ser exportado e enviado em um único arquivo no formato PDF. Esse PDF deve conter, obrigatoriamente, o nome completo do aluno, a matrícula e um link para o repositório ou serviço de armazenamento (como Google Drive, GitHub, OneDrive, etc.) onde o código-fonte completo pode ser baixado.

Certifique-se de que o link esteja acessível para avaliação.