4. Associação e Composição

Nesta etapa do projeto, foi desenvolvida a classe ***Turma***, responsável por representar uma turma acadêmica dentro da universidade. Essa classe estabelece relações com diversas outras classes, aplicando de forma prática os conceitos de **associação** e **composição**.

A classe contém os seguintes atributos:

- **código**: identifica a turma de forma única;
- **disciplina**: representa o vínculo direto com a classe Disciplina;
- **professor**: associa a turma ao professor responsável;
- **alunos**: uma lista dinâmica do tipo `ArrayList<Aluno>`, que armazena todos os alunos matriculados.

```java
public class Turma {
    public String codigo;
    public Disciplina disciplina;
    public Professor professor;
    public ArrayList <Aluno> alunos = new ArrayList<>();

    Turma(Disciplina disciplina, Professor professor, String codigo){
        this.disciplina = disciplina;
        this.professor = professor;
        this.codigo = codigo;
    }

    public void adicionarAluno(Aluno aluno){
        alunos.add(aluno);
    }

    public void removerAluno(Aluno aluno){
        alunos.remove(aluno);

    }

    public void listarAluno(){
        System.out.println("Alunos: ");
        for(int i = 0; i < alunos.size(); i++){
            System.out.println(alunos.get(i));
        }
    }

    @Override
    public String toString(){
        return "Turma: " + this.codigo + "\n" + "Professor: " + this.professor + "\n" + "Disciplina: " + this.disciplina;
    }

}
```

Foi implementado um **método construtor** que inicializa os atributos necessários para a criação de uma turma, garantindo que toda turma seja criada já vinculada a uma disciplina e a um professor.

```java

    Turma(Disciplina disciplina, Professor professor, String codigo){
        this.disciplina = disciplina;
        this.professor = professor;
        this.codigo = codigo;
    }

```

Também foram implementados os seguintes métodos:

- ***adicionarAluno(Aluno aluno)***: adiciona um aluno à lista de alunos da turma;
- ***removerAluno(Aluno aluno)***: remove um aluno da turma, caso exista na lista;
- ***listarAluno()***: imprime no console todos os alunos cadastrados na turma, utilizando um laço de repetição para percorrer a lista.

```
12
13      public void adicionarAluno(Aluno aluno){
14          alunos.add(aluno);
15      }
16
17      public void removerAluno(Aluno aluno){
18          alunos.remove(aluno);
19
20      }
21
22      public void listarAluno(){
23          System.out.println("Alunos: ");
24          for(int i = 0; i < alunos.size(); i++){
25              System.out.println(alunos.get(i));
26          }
27      }
28
```

Além disso, o método **toString()** foi sobrescrito para exibir informações completas da turma quando o objeto for impresso, mostrando o código, o professor e a disciplina vinculados, oferecendo uma visualização mais clara e útil do objeto, em vez de exibir apenas seu endereço de memória.

```
28
29      @Override
30      public String toString(){
31          return "Turma: " + this.codigo + "\n" + "Professor: " + this.professor + "\n" + "Disciplina: " + this.disciplina;
32      }
33
34  }
```

Essa implementação evidencia os conceitos de **composição**, pois uma turma é composta pelos alunos nela cadastrados, e de **associação**, já que a turma depende da relação com objetos de outras classes para existir de forma funcional.