MVC: ARRAYLIS, FICHEIRO DE OBJECTOS, E ORDENACAO COM COLLECTIONS.SORT

1. Estuante

```java
import java.io.Serializable;
public class Estudante implements Comparable<Estudante>, Serializable{
    private int codigo;
    private String nome;
    private double test1;
    private double test2;

    public Estudante(int codigo, String nome, double test1, double test2) {
        this.codigo = codigo;
        this.nome = nome;
        this.test1 = test1;
        this.test2 = test2;
    }
    public int getCodigo() {
        return codigo;
    }
    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public double getTest1() {
        return test1;
    }
    public void setTest1(double test1) {
        this.test1 = test1;
    }
    public double getTest2() {
        return test2;
    }
    public void setTest2(double test2) {
        this.test2 = test2;
    }
    public double calculaMedia() {
        return (test1 + test2)/2;
    }
    public int compareTo(Estudante estudante) {
        if(this.codigo>estudante.codigo)return 1;
        if(this.codigo<estudante.codigo)return -1;
        return 0;
    }
    public String toString() {
        return codigo + "-" + nome + "-" + test1 + "-" + test2 + "-" + calculaMedia();
    }
}
```

2. ControllerEtudante

```java
import java.io.FileInputStream;

    public class ControllerEstudante {
        static Scanner k=new Scanner(System.in);
        public static ArrayList<Estudante> listaDeEstudantes(){
        ArrayList<Estudante> estudantes=new ArrayList<Estudante>();

            try{
                FileInputStream abrir=new FileInputStream("estudantes.bin");
                ObjectInputStream ler=new ObjectInputStream(abrir);

                for(int i=0;i<100;i++){
                    estudantes.add((Estudante)ler.readObject());
                }
                ler.close();
                abrir.close();
            }
            catch(Exception e){
                System.out.println(e.getMessage());
            }

            return estudantes;
        }
```

```
    public static void adicionarEstudante()throws Exception{
        ArrayList<Estudante> estudantes=listaDeEstudantes();

        FileOutputStream criar=new FileOutputStream("estudantes.bin");
        ObjectOutputStream escrever=new ObjectOutputStream(criar);

        int i=0;

        while(i<estudantes.size()){
                if(estudantes.get(i)!=null){
                escrever.writeObject(estudantes.get(i));
                i++;
            }
        }
        System.out.print("Codigo:");int codigo=k.nextInt();
        System.out.print("Nome:");String nome=k.next();
        System.out.print("Teste1:");double teste1=k.nextDouble();
        System.out.print("Teste2:");double teste2=k.nextDouble();

        escrever.writeObject(new Estudante(codigo, nome, teste1, teste2));

        escrever.close();
        criar.close();
        System.out.println("ESTUDANTE ADICIONADO COM SUCESSO:");
    }

    public static void listarEstudantes() throws Exception{
        ArrayList<Estudante>estudantes=listaDeEstudantes();


        int i=0;
        while(i<estudantes.size()){
            if(estudantes.get(i)!=null){
                System.out.println(estudantes.get(i).toString());
                i++;
            }
        }
    }

    public static void actualizarEstudante(int codigo) throws Exception  {

        ArrayList<Estudante> estudantes=listaDeEstudantes();
         char opcao;
         for (Estudante estudante : estudantes) {
            if((estudante!=null) && (estudante.getCodigo() == codigo)) {

                do{
                    System.out.println("MENU DE ACTUALIZACAO DO ESTUDANTE");
                    System.out.println("1-Nome: ");
                    System.out.println("2-Teste1: ");
                    System.out.println("3-Teste2: ");
                    System.out.println("0-Sair: ");
                    System.out.println("Digite uma opcao: ");
                    opcao=k.next().charAt(0);
                    switch (opcao) {
                    case '1':
                        System.out.println("Nome:");
                        String nome = k.next();
                        estudante.setNome(nome);
                        estudante.setTest1(estudante.getTest1());
                        estudante.setTest2(estudante.getTest2());
                        break;
                    case '2':
                        System.out.println("Teste 1:");
                        double teste1= k.nextDouble();
                        estudante.setTest1(teste1);
                        estudante.setNome(estudante.getNome());
                        estudante.setTest2(estudante.getTest2());
                        break;
```

```
                        case '3':
                            System.out.println("Teste 2:");
                            double teste2= k.nextDouble();
                            estudante.setTest2(teste2);
                            estudante.setNome(estudante.getNome());
                            estudante.setTest1(estudante.getTest1());
                            break;
                        case '0':
                            System.out.println("Saiu do menu de actualizacao");
                            break;
                        default:
                            System.out.println("Opcao invalida:");
                            break;
                        }
                     }while(opcao!='0');
                    }
                }
                FileOutputStream criar=new FileOutputStream("estudantes.bin");
                ObjectOutputStream escrever=new ObjectOutputStream(criar);

                int i=0;
                while (i<estudantes.size()) {
                    if(estudantes.get(i)!=null){
                        escrever.writeObject(estudantes.get(i));
                        i++;
                    }
                }
                    escrever.close();
                    criar.close();
                    System.out.println("ESTUDANTE ACTUALIZADO COM SUCESSO:");
            }
```

```
        public static void removerEstudante(int codigo) throws Exception  {

            ArrayList<Estudante> estudantes=listaDeEstudantes();

                for (int i=0;i<estudantes.size();i++) {
                    if((estudantes.get(i)!=null) && (estudantes.get(i).getCodigo() == codigo)){
                        estudantes.remove(estudantes.get(i));
                    }
                }
                FileOutputStream criar=new FileOutputStream("estudantes.bin");
                ObjectOutputStream escrever=new ObjectOutputStream(criar);

                int i=0;
                while (i<estudantes.size()) {
                    if(estudantes.get(i)!=null){
                        escrever.writeObject(estudantes.get(i));
                        i++;
                    }
                }
                    escrever.close();
                    criar.close();
                    System.out.println("ESTUDANTE REMOVIDO COM SUCESSO:");
            }
```

```
         public static void ordenarEstudantes () throws IOException{
             ArrayList<Estudante> estudantes=listaDeEstudantes();
             Collections.sort(estudantes);
             FileOutputStream criar=new FileOutputStream("estudantes.bin");
             ObjectOutputStream escrever=new ObjectOutputStream(criar);

             int i=0;
             while (i<estudantes.size()) {
                 if(estudantes.get(i)!=null){
                     escrever.writeObject(estudantes.get(i));
                     i++;
                 }
             }
                 escrever.close();
                 criar.close();
                 System.out.println("ESTUDANTES ORDENADOS COM SUCESSO:");

    }
}
```

3. ViewEstudante

```java
import java.util.Scanner;
public class ViewEstudante {
    static Scanner k=new Scanner(System.in);
    public static void main(String[] args) throws Exception {
        int opcao;
        do{
            System.out.println("+++MENU PRINCIPAL+++");
            System.out.println("1-Adicionar Estudante: ");
            System.out.println("2-Listar Estudante: ");
            System.out.println("3-Actualizar Estudante: ");
            System.out.println("4-Remover Estudante: ");
            System.out.println("5-Ordenar Estudantes: ");
            System.out.println("0-Sair: ");
            System.out.print("Digite uma opcao: ");
            opcao=k.nextInt();

            switch (opcao) {
            case 1:
                ControllerEstudante.adicionarEstudante();
                break;
            case 2:
                ControllerEstudante.listarEstudantes();
                break;
            case 3:
                System.out.println("Digite um codigo que existe na lista: ");
                int codigoActualizacao=k.nextInt();
                ControllerEstudante.actualizarEstudante(codigoActualizacao);
                break;
            case 4:
                System.out.println("Digite um codigo que existe na lista: ");
                int codigoRemocao=k.nextInt();
                ControllerEstudante.removerEstudante(codigoRemocao);
                break;

            case 5:
                ControllerEstudante.ordenarEstudantes();
                break;
            case 0:
                System.out.println("SAIU DO MENU PRINCIPAL");;
                break;
            default:
                System.out.println("OPCAO INVALIDA");
                break;
            }
        }while(opcao!=0);
    }
}

```