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

1. Estuante

```java
public class Estudante implements Comparable<Estudante> {
    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 String toString() {
        return codigo + "-" + nome + "-" + test1 + "-" + test2 + "-" + calculaMedia();
    }
    public int compareTo(Estudante estudante) {
        if(this.codigo>estudante.codigo) return 1;
        if(this.codigo<estudante.codigo) return -1;
        return 0;
    }
}
```

2. ControllerEtudante

```java
import java.io.BufferedReader;

public class ControllerEstudante {
    static Scanner k = new Scanner(System.in);
    public static void adicionarEstudante() throws IOException {

        FileWriter criar = new FileWriter("notas.txt", true);
        BufferedWriter adicionar = new BufferedWriter(criar);
        System.out.print("Codigo: ");
        int codigo = k.nextInt();
        System.out.print("Nome: ");
        String nome = k.next();
        System.out.print("Teste 1: ");
        double test1 = k.nextDouble();
        System.out.print("Teste 2: ");
        double test2 = k.nextDouble();
        Estudante estudante = new Estudante(codigo, nome, test1, test2);
        adicionar.write(estudante.toString());
        adicionar.newLine();

        adicionar.close();
        criar.close();
        System.out.println("ADICIONADO COM SUCESSO:");
    }
```

```java
    public static ArrayList<Estudante> listaDeEstudantes() throws IOException {
        ArrayList<Estudante> estudantes = new ArrayList<Estudante>();
        FileReader abrir = new FileReader("notas.txt");
        BufferedReader ler = new BufferedReader(abrir);
        String linha = ler.readLine();

        String[] elementosDaLinha;

        while(linha != null && !linha.isEmpty()) {
            elementosDaLinha = linha.split("-");
            int codigo = Integer.parseInt(elementosDaLinha[0]);
            String nome = elementosDaLinha[1];
            double test1 = Double.parseDouble(elementosDaLinha[2]);
            double test2 = Double.parseDouble(elementosDaLinha[3]);
            estudantes.add(new Estudante(codigo, nome, test1, test2));
            linha=ler.readLine();
        }

        ler.close();
        abrir.close();
        return estudantes;
    }

    public static void listarEstudantes()throws IOException {
        ArrayList<Estudante> estudantes=listaDeEstudantes();
        for (int i = 0; i < estudantes.size(); i++) {
            System.out.println(estudantes.get(i).toString());
        }
    }
```

```java
static void actualizarEstudante(int codigo) throws IOException  {
        ArrayList<Estudante> estudantes=listaDeEstudantes();
         char opcao;
        for (Estudante estudante : estudantes) {
            if(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:");
                    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');
                }
            }
                FileWriter criar = new FileWriter("notas.txt");
                BufferedWriter adicionar = new BufferedWriter(criar);
                for (int i=0;i<estudantes.size();i++) {
                    adicionar.write(estudantes.get(i).toString());
                    adicionar.newLine();
                }
                adicionar.close();
                criar.close();
                System.out.println("ACTUALIZADO COM SUCESSO:");
    }

          public static void removerEstudante(int codigo) throws IOException {
              ArrayList<Estudante> estudantes=listaDeEstudantes();
              for (int i=0;i<estudantes.size();i++) {
                  if(estudantes.get(i).getCodigo() == codigo) {
                      estudantes.remove(estudantes.get(i));
                  }
              }

              FileWriter criar = new FileWriter("notas.txt");
              BufferedWriter adicionar = new BufferedWriter(criar);
              for (int i=0;i<estudantes.size();i++) {
                  adicionar.write(estudantes.get(i).toString());
                  adicionar.newLine();
              }
          adicionar.close();
          criar.close();
          System.out.println("REMOVIDO COM SUCESSO:");
      }

        public static void ordenarEstudantes() throws IOException{
            ArrayList<Estudante> estudantes=listaDeEstudantes();
            Collections.sort(estudantes);
            FileWriter criar = new FileWriter("notas.txt");
            BufferedWriter adicionar = new BufferedWriter(criar);
            for (int i=0;i<estudantes.size();i++) {
                adicionar.write(estudantes.get(i).toString());
                adicionar.newLine();
            }
        adicionar.close();
        criar.close();
        System.out.println("ESTUDANSTES ORDENADOS COM SUCESSO:");

        }
}
```

3. ViewEstudante

```java
import java.io.IOException;
public class ViewEstudante {
    public static void main(String[] args) throws IOException {
        Scanner k = new Scanner(System.in);
        int opcao;
        do {
            System.out.println("MENU DE PRINCIPAL DO SISTEMA");
            System.out.println("1 - Adicionar Estudante: ");
            System.out.println("2 - Listar Estudantes: ");
            System.out.println("3 - Actualizar Estudante: ");
            System.out.println("4 - Remover Estudante: ");
            System.out.println("5 - Listar Estudantes Ordenados: ");
            System.out.println("0 - Sair: ");
            System.out.print("Opcao: "); opcao = k.nextInt();
            switch (opcao) {
            case 1:
                ControllerEstudante.adicionarEstudante();
                break;
            case 2:
                ControllerEstudante.listarEstudantes();
                break;
            case 3:
                System.out.print("Digite um codigo de estudante que existe na lista: "); int codigoActualizacao = k.nex
                ControllerEstudante.actualizarEstudante(codigoActualizacao);
                break;
            case 4:
                System.out.print("Digite um codigo de estudante 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);
   }
}
```