MANUAL TÉCNICO

Presentado por
Grupo8

Marcos Arnoldo Itzep Ixmay 201907156
Josué Noel Mazariegos Gramajo 201900092
Byron Enrique Rumpich Sal 201907769
Carlos Alberto Rodas Galvez 201700420

Índice

GENERAL

En este manual se observan y se describen las funciones y procedimientos realizados para el funcionamiento de este programa.

Clases del Paquete

- Alumnos
- Profesores
- Cursos
- Notas
- Usuarios
- Lista Usuarios
- Asignar Alumnos
- Asignar Profesores
- Main

Alumnos

Variables

```
4
5       public class ALUMNOS {
6           private int id_alumno;
7           private int carne_alumno;
8           private String nombre_alumno;
9           private String fechaNac_alumno;
10          private String genero_alumno;
```

Estas variables son los atributos de la clase alumnos.

Asignaciones

```
11
12      public ALUMNOS(int id_alumno, int carne_alumno, String nombre_alumno, String fechaNac_alumno, String genero_alumno)
13
14      {
15          this.id_alumno = id_alumno;
16          this.carne_alumno = carne_alumno;
17          this.nombre_alumno = nombre_alumno;
18          this.fechaNac_alumno = fechaNac_alumno;
19          this.genero_alumno = genero_alumno;
20      }
```

Métodos

Estos métodos se utilizan para mostrar o modificar los atributos.

****Estos bloques se replican en los demás objetos con sus respectivos atributos.**

```
23      public String getnombre() {
24          return this.nombre;
25      }
26
27      public void setnombre(String nombre) {
28          this.nombre = nombre;
29      }
30
31      public int getid_prof() {
32          return this.id_prof;
33      }
34
35      public void setid_prof(int id_prof) {
36          this.id_prof = id_prof;
        }
```

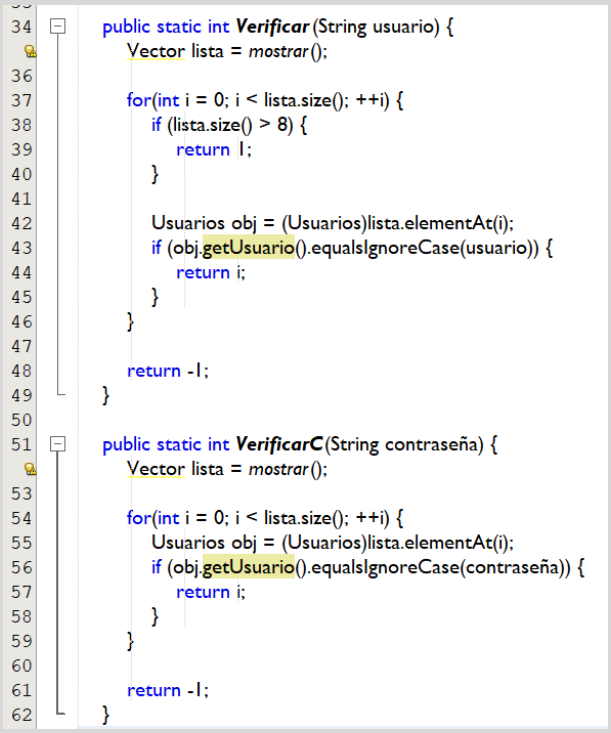
Usuarios

```
import java.util.Vector;

public class Usuarios {


    private String usuario;
    private String contraseña;

    public Usuarios() {
    }

    public String getUsuario() {...3 lines }

    public void setUsuario(String usuario) {...3 lines }

    public String getContraseña() {...3 lines }

    public void setContraseña(String contraseña) {...3 lines }

    public static Vector mostrar() {...3 lines }

    public static int Verificar(String usuario) {...16 lines }

    public static int VerificarC(String contraseña) {...12 lines }

    public static int VerificarL(String usuario, String contraseña) {...12 lines }

}
```

En esta clase se encuentran sus atributos y su métodos para mostrarlos y modificarlos

Los métodos ***Verificar***, ***VerificarC*** y ***VerifiarL***

se encargan de comprobar la existencia de los usuarios registrados



```
34  public static int Verificar(String usuario) {
    Vector lista = mostrar();
36
37      for(int i = 0; i < lista.size(); ++i) {
38          if (lista.size() > 8) {
39              return 1;
40          }
41
42          Usuarios obj = (Usuarios)lista.elementAt(i);
43          if (obj.getUsuario().equalsIgnoreCase(usuario)) {
44              return i;
45          }
46      }
47
48      return -1;
49  }
50
51  public static int VerificarC(String contraseña) {
    Vector lista = mostrar();
53
54      for(int i = 0; i < lista.size(); ++i) {
55          Usuarios obj = (Usuarios)lista.elementAt(i);
56          if (obj.getUsuario().equalsIgnoreCase(contraseña)) {
57              return i;
58          }
59      }
60
61      return -1;
62  }
```

Verificaciones como el número de usuarios registrados o verificar la igualdad de la contraseña ingresada con la contraseña almacenada.

Lista Usuarios

```
5   public class ListaUsuario {
6
7       private static java.util.Vector<practica22.Usuarios> datos;
8
9       public ListaUsuario() { /* compiled code */ }
10
11      public static void agregar(practica22.Usuarios obj) { /* compiled code */ }
12
13      public static void eliminar(int pos) { /* compiled code */ }
14
15      public static java.util.Vector mostrar() {return null;
16  /* compiled code */ }
17
18  }
```

Esta clase esta relacionada con la clase Usuarios y es un vector con una función para guardar los usuarios y las contraseñas.

Main

Dentro de esta clase están los métodos encargados del funcionamiento del programa.

```
10  public class Practica22 {
11
12      Notas notas[]  = new Notas[100];
13      Usuarios rusuario [] = new Usuarios[5];
14      int NotasC;
15      static int contador_alumnos=0;
16      static ALUMNOS alumnos[] = new ALUMNOS[100];
17      static int contador_curso = 0;
18      static Cursos[] curso = new Cursos[15];
19      static AsignarAlumnos[] Asig_Al = new AsignarAlumnos[200];
20      static AsignarProfesores[] Asig_Pf = new AsignarProfesores[30];
21      static int contadorasignarAlumnos,contadornotas,contadorprofesores = 1;
22      static Profesores[] profesores = new Profesores[20];
23      static int Nprof = 0;
24      static boolean asignacion;
25      static int añonuevo;
26      static int posAlmuno, posCurso,posNotaA,posNotaC,posProfe;
27
28
29
30      Scanner lec =new Scanner (System.in);
31      Scanner lect =new Scanner (System.in);
32      Scanner lectu =new Scanner (System.in);
```

InicioSesión

```
90
91      void InicioSesion(){
92          System.out.println("===Seleccione una opción===");
93          System.out.println("1. Ingresar como Administrador");
94          System.out.println("2. Ingresar como Usuario ");
95          System.out.println("====================================");
96          System.out.print(">>>Escribir el número de opción: ");
97
99          Scanner lec = new Scanner(System.in);
100         int opccion = lec.nextInt();
101
102         switch(opccion){
103             case 1:
104                 Inicio();
105                 break;
106             case 2:
107                 InicioUsuario();
108                 break;
109
110         }
111
112     }
```

El siguiente método permite a los usuarios ingresar como un administrador o un usuario normal. La opción seleccionada por el usuario ingresa a un condicional *swich-case* que nos dirige al inicio de sesión correspondiente.

Inicio

```
113
114     void Inicio(){
115         System.out.println("===Ininciar Sesión===");
116         System.out.println("Introduzca su Usuario");
117         System.out.print(">>>");
            Scanner lec = new Scanner(System.in);
119         String usuario = lec.nextLine();
120         if(usuario.equals("admin")){
121             System.out.println("Introduzca su Contraseña");
122             System.out.print(">>>");
                Scanner lect = new Scanner(System.in);
124             String contraseña = lec.nextLine();
125             if (contraseña.equals("admin")){
126                 System.out.println("Bienvenido al Menu");
127                 Menu();
128             }else{
129                 System.out.println("***Contraseña incorrecta***");
130                 Inicio();
131             }
132
133         }else{
134             System.out.println("***Usuario Incorrecto***");
135             Inicio();
136         }
137
138     }
```

Este método permite el ingreso al menú del administrador confirmando el usuario y contraseña con un condicional *if – else.*

Menu

```
40      void Menu(){
41          boolean entrar_menu = true;
42          while (entrar_menu == true) {
43              System.out.println("===Seleccione lo que desea hacer===");
44              System.out.println("1. Cargar Alumnos");
45              System.out.println("2. Cargar Profesores ");
46              System.out.println("3. Cargar Cursos");
47              System.out.println("4. Asignar Alumnos");
48              System.out.println("5. Asignar Profesores");
49              System.out.println("6. Cargar Notas");
50              System.out.println("7. Registrar Usuario");
51              System.out.println("====================================");
52              System.out.print(">>>Escribir el número de opción: ");
53
                Scanner lec = new Scanner(System.in);
55              int opccion = lec.nextInt();
56
57              switch (opccion) {
58
59                  case 1:
60                      CargaAlumnos();
61                      break;
62                  case 2:
63                      CargarPersonas();
64                      break;
65                  case 3:
66                      CargaCursos();
67                      break;
```

Si el usuario ingresa como administrador este es el menú que imprimirá el método ***Menú***, seguido de un condicional *swich* encargado de operar el método seleccionado a través de cada uno de los case y la selección del administrador.

CargaAlumnos

```
233
234         public static void CargaAlumnos(){
235             Scanner nose = new Scanner(System.in);//Instanciando scanner
236             Scanner leer = new Scanner(System.in);//Instanciando scanner
                File archivo = null;//abrir archivo
238
239             FileReader fr = null;//almacenar texto de archivo
                BufferedReader br = null;//leer el texto almacenado
241             try {
242                 System.out.println(">>>Ingrese la ruta del archivo que desea leer (>>Alumnos<<): ");
243                 System.out.print(">");
244                 String R1 = leer.nextLine();
245                 R1 = R1.replaceAll("\"", "");
246                 int saltolinea = 0;
247
248                 // LECTURA DEL ARCHIVO
249                 System.out.println(".....");
250                 System.out.println(">> Su mensaje carga de Alumnos es <<");
251                 archivo = new File(R1);
252                 fr = new FileReader(archivo);
253                 br = new BufferedReader(fr);
254                 // Variable para guardar la linea que almacena el BufferedReader.
255                 String pok = ",";
256                 String linea;
257                 int Linea_alumno = 1;
258                 // String pok = "";// lectura fichero
259                 //(int id_alumno, int carne_alumno, String nombre_alumno, int fechaNac_alumno, String genero_alumno)
```

Este método se opera dentro de un *try – catch* para el manejo de excepciones y se ejecuta carga de alumnos al programa.

Se inicia abriendo un archivo para guardar y leer el contenido del archivo.

```
260                 while ((linea = br.readLine()) != null) {
261                     if (saltolinea != 0) {
262                         System.out.println(linea);
263                         String datos[] = linea.split(",");
264                         int id_al = Integer.parseInt(datos[0]);
265                         int carne_al = Integer.parseInt(datos[1]);
266                         String nombre_al = datos[2].toUpperCase();
                            String fecha_al = datos[3].toUpperCase();;
268                         String genero_al= datos[4].toUpperCase();
269                         if (alumnos[0] == null) {
270                             alumnos[0] = new ALUMNOS(id_al,carne_al,nombre_al,fecha_al,genero_al);
271                         } else {
272                             boolean asigID = true;
273                             for (int i = 0; i < contador_alumnos; i++) {
274                                 if (alumnos[i].getId_alumno() == id_al) {
275                                     asigID = false;
276                                     System.out.println("no se puede asignar el alumno de la linea " + Linea_alumno + " \t ya existe una ID Igual asi que se descarta este ID");
277                                 }
278                             }
```

El texto del archivo se organiza separando por.

comas los valores que entraran en el arreglo de alumnos.

Este proceso se itera siempre y cuando no se repita ningún usuario o se alcance el máximo de alumnos permitidos.

```
279                     if (asigID == true && contador_alumnos <= 100) {
280                         int aux = 0;
281                         Boolean SincOcup = false;
282                         while (SincOcup == false) {
283                             if (alumnos[aux] == null) {
284                                 alumnos[aux] = new ALUMNOS(id_al,carne_al,nombre_al,fecha_al,genero_al);
285                                 SincOcup = true;
286                                 contador_alumnos++;
287                                 Linea_alumno++;
288                             } else {
289                                 aux++;
290                             }
291                         }}
292                     else {
293                         System.out.println("Llegaste al limite de Alumnos ");
294                     }
295                 }}
296                 else{
297                     saltolinea++;
298                 }}
299             contador_alumnos++;
300             System.out.println(".....");
301             System.out.println(">> Carga terminada cargaste " + contador_alumnos + "  Alumnos <<");
302             System.out.println(".....");
303
```

```
305             } catch (Exception e) {
                    // System.out.println("Error al ingresar el archivo");
                    e.printStackTrace();
307             } finally {
308                 // En el finally cerramos el fichero, para asegurarnos
309                 // que se cierra tanto si todo va bien como si salta
310                 // una excepcion.
311                 try {
312                     if (null != fr) {
313                         //  System.out.println("Error al leer el archivo");
314                         fr.close();
315                     }
                    } catch (Exception e2) {
                        e2.printStackTrace();
318                 }
319             }
320         }
```

****Estos bloques de código se utilizan en la carga de Profesores, Notas y Cursos**

AsignarAlumno

```
588     public static void AsignarAlunmo() {
589     Scanner lector = new Scanner(System.in);
        File archivo = null;
591     FileReader fr = null;
        BufferedReader br = null;
593
594     try {
595
596         System.out.print("Ingresar la ruta del archivo csv a leer: ");
597         String ruta;
598         ruta = lector.nextLine();
599         ruta = ruta.replaceAll("\"", "");
600         archivo = new File(ruta);
601         fr = new FileReader(archivo);
602         br = new BufferedReader(fr);
603         int lineaA = 1;
        String linea = br.readLine();
605     int l_prof = 1;
606     while ((linea = br.readLine()) != null) {
607         System.out.println(linea);
608         String datos4[] = linea.split(",");
609         int idAlu = Integer.parseInt(datos4[0]);
610         int idCurso = Integer.parseInt(datos4[1]);
611         posAlmuno = BuscarAlumno(idAlu);//profesores[3]=3
612         posCurso = BuscarCurso(idCurso);
613         if (posAlmuno != 0101 && posCurso != 0101) {//si el alumno si existe se procede
614             if (Asig_Al[0] == null) {
615                 Asig_Al[0] = new AsignarAlumnos(alumnos[posAlmuno], curso[posCurso]);
616                 //AsignarAlu nuevo = new AsignarAlumnos(alumnos[posAlmuno], curso[posCurso]);
```

En este método se realiza la lectura de un archivo, para que los valores sean relacionados con el objeto A*lumnos* y *Cursos.*

```
616                 //AsignarAlu nuevo = new AsignarAlumnos(alumnos[posAlmuno], curso[posCurso]);
617
618                 System.out.println("Se asigno el Alumno con ID: " + alumnos[posAlmuno].getId_alumno());
619                 System.out.println("Al curso: " + curso[posCurso].getId());
620                 System.out.println("-----------------------------------------------");
621                 contadorasignarAlumnos++;
622                 lineaA++;
623             } else {
624                 boolean asigID = true;
625                 for (int i = 0; i < contadorasignarAlumnos; i++) {
626                     if (Asig_Al[i].getIdAlumnos() == alumnos[posAlmuno]) {
627                         asigID = false;
628                         System.out.println(" no se puede asignar el Alumno de la linea " + lineaA + "\t ya esta asignado");
629
630                     }
631                 }
632                 if (asigID == true && contadorasignarAlumnos <= 200) {
633                     int aux1 = 0;
634                     Boolean SincOcup = false;
635                     while (SincOcup == false) {
636                         if (Asig_Al[aux1] == null) {
637                             Asig_Al[aux1] = new AsignarAlumnos(alumnos[posAlmuno], curso[posCurso]);
638                             //AsignarAlu nuevo = new AsignarAlumnos(alumnos[posAlmuno], curso[posCurso]);
639                             //AsignandoPok[contadorasignarAlumnos]=nuevo;
640                             SincOcup = true;
641                             System.out.println("Se asigno el Alumno con ID: " + alumnos[posAlmuno].getId_alumno());
                            System.out.println("Al curso " + curso[posCurso].getId());
643                             System.out.println("-----------------------------------------------");
644
```

Con condicionales *if – else* se comprueba que el id del alumno y del curso existan, luego se almacenan en un arreglo.

```
644                             contadorasignarAlumnos++;
645                             lineaA++;
646                         } else {
647                             auxI++;
648                         }
649                     }
650                 } else {
651                     System.out.println("Llegaste al limite de Alumnos");
652                 }
653             }//////////////////////////////////////////////
654         }
655     }
656     //     MostrarDatosAsigPok(AsignandoPok);
657     //AsignarPokes[contadorasignarPok]=nuevo;
658     contadorasignarAlumnos++;
659     System.out.println("..........");
660     System.out.println(">> Carga terminada cargaste " + contadorasignarAlumnos + " Asignacion Alumnos <<");
661     System.out.println("..........");
    } catch (Exception e) {
663     // System.out.println("Error al ingresar el archivo");
        e.printStackTrace();
665 } finally {
666     // En el finally cerramos el fichero, para asegurarnos
667     // que se cierra tanto si todo va bien como si salta
668     // una excepcion.
669     try {
670         if (null != fr) {
671             //  System.out.println("Error al leer el archivo");
672             fr.close();
673         }
        } catch (Exception e2) {
            e2.printStackTrace();
676     }
677
678 }
```

Si el alumno ya aparece asignado a un curso, no se permite asignar de nuevo a ese curso.

RUsuarios

```
140        void RUsuarios(){
141
142            System.out.println("===Registrar Usuario===");
143            System.out.println("Introduzca su Usuario");
144            System.out.print(">>>");
               Scanner lec = new Scanner(System.in);
146            String usuario = lec.nextLine();
147            Usuarios obj = new Usuarios();
148            if (Usuarios.Verificar(usuario)==-1) {
149
150                System.out.println("Introduzca su Contraseña");
151                System.out.print(">>>");
                   Scanner lect = new Scanner(System.in);
153                String contraseña = lect.nextLine();
154
155                System.out.println("Introduzca de nuevo su Contraseña");
156                System.out.print(">>>");
                   Scanner lectu = new Scanner(System.in);
158                String contraseña2 = lect.nextLine();
```

Este método se utiliza para registrar nuevos usuarios, solo se puede acceder a el usuario administrador. Con el uso de un lector se ingresan el usuario y contraseña nuevos.

```
160            if(contraseña.equals(contraseña2)){
161                obj.setUsuario(usuario);
162                ListaUsuario.agregar(obj);
163
164                obj.setContraseña(contraseña);
165                ListaUsuario.agregar(obj);
166
167                System.out.println("");
168                System.out.println("------Usuario Registrado con Exito------");
169                System.out.println("");
170                InicioSesion();
171            }else{
172                System.out.println("Las contraseñas no coinciden");
173                Menu();
174            }
175
176        }else{
177            System.out.println("El usuario ya ha sido ingresado o limite de usuarios alcanzados");
178            InicioSesion();
179        }
180
181    }
```

Seguido de condicionales *if – else* para confirmar la contraseña y proceder a guardar al nuevo usuario o evitar usuarios duplicados.

InicioUsuario

```
182
183     void InicioUsuario(){
184         System.out.println("===Inicio de Sesión de Usuario===");
185         System.out.println("Introduzca su Usuario");
186         System.out.print(">>>");
            Scanner lec = new Scanner(System.in);
188         String usuario = lec.nextLine();
189
190         System.out.println("Introduzca su Contraseña");
191         System.out.print(">>>");
            Scanner lect = new Scanner(System.in);
193         String contraseña = lec.nextLine();
194
195         int pos = Usuarios.VerificarL(usuario,contraseña);
196         if (pos == -1){
197             System.out.println("");
198             System.out.println("----Usuario o Contraseña Incorrecta----");
199             System.out.println("");
200             InicioSesion();
201         }else{
202             System.out.println("");
203             System.out.println("=======Bienvenido al Menu=======");
204             System.out.println("");
205             MUsuarios();
206
207         }
208
209
```

Este método se encarga del inicio de sesión de los usuarios existentes validando los datos con el condicional *if – else*.

MUsuarios

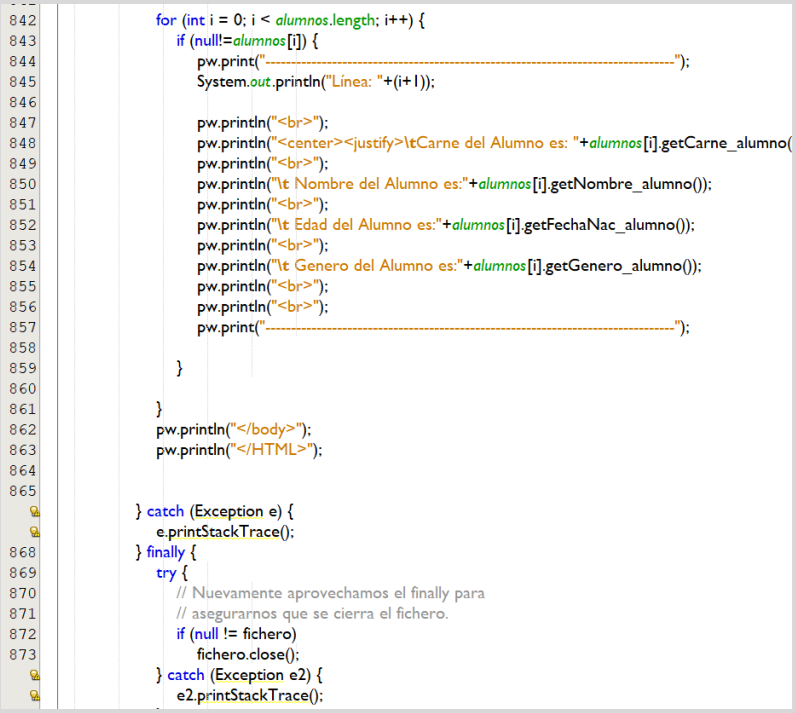
```
214        void MUsuarios(){
215            System.out.println("===Seleccione lo que desea hacer===");
216            System.out.println("1. Generar Reportes");
217            System.out.println("2. Salir");
218            System.out.println("====================================");
219            System.out.print(">>>Escribir el número de opción: ");
               Scanner lec = new Scanner(System.in);
221            int opccion = lec.nextInt();
222            switch(opccion){
223                case 1:
224                    System.out.println("Imprimir Reportes");
225                    MUsuarios();
226                    break;
227                case 2:
228                    InicioSesion();
229                    break;
230
231            }
232        }
```

Si el usuario ingresa como Usuario normal este es el menú que imprimirá el método ***MUsuarios***, este opera de la misma manera que el menú usuarios, pero posee menos permisos.

ReporteA

```
820        public static void ReporteA(){
821            //Creando archivo
822            Date date = new Date();
823            DateFormat hourdateFormat = new SimpleDateFormat("HH:mm:ss dd/MM/yyyy");
824            FileWriter fichero = null;
               PrintWriter pw = null;
826            try{
827                fichero = new FileWriter("R_Alumnos.html");//crear archivo
828                pw = new PrintWriter(fichero);//este es nuestro lapiz
829                pw.println(">HTML<");
830                //<center> </center>
831                pw.println("");
832                pw.println("<center> Practica 2 (hora y fecha: " + hourdateFormat.format(date) + ")" + " </center>");
833                pw.println("<center>Reporte Practica 2 </center>" );
834                pw.println("<head>");
835                pw.println("<title>Reporte Alumnos</title>");
836                pw.println("</head>");
837                ////////////////////////////////////////////////////////////////////////
838                pw.println("<body text =#FFFFFF, background= Imagenes/fondo.jpg");
839                //INICIANDO CONTENIDO
840                //"C:\Users\lorena\Desktop\cargaAlumnos.txt"
841
```

Se crea un archivo de tipo *HTML* para editar y generar los reportes.



```
842                 for (int i = 0; i < alumnos.length; i++) {
843                     if (null!=alumnos[i]) {
844                         pw.print("-----------------------------------------------------------------------");
845                         System.out.println("Línea: "+(i+1));
846 
847                         pw.println("<br>");
848                         pw.println("<center><justify>\tCarne del Alumno es: "+alumnos[i].getCarne_alumno(
849                         pw.println("<br>");
850                         pw.println("\t Nombre del Alumno es:"+alumnos[i].getNombre_alumno());
851                         pw.println("<br>");
852                         pw.println("\t Edad del Alumno es:"+alumnos[i].getFechaNac_alumno());
853                         pw.println("<br>");
854                         pw.println("\t Genero del Alumno es:"+alumnos[i].getGenero_alumno());
855                         pw.println("<br>");
856                         pw.println("<br>");
857                         pw.print("-----------------------------------------------------------------------");
858 
859                     }
860 
861                 }
862                 pw.println("</body>");
863                 pw.println("</HTML>");
864 
865 
                } catch (Exception e) {
                    e.printStackTrace();
868             } finally {
869                 try {
870                     // Nuevamente aprovechamos el finally para
871                     // asegurarnos que se cierra el fichero.
872                     if (null != fichero)
873                         fichero.close();
                    } catch (Exception e2) {
                        e2.printStackTrace();
```

Con un ciclo *for* se recorre el arreglo de alumnos y obtener los valores que se desean imprimir en el reporte.