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
- 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.**

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

Usuarios

```java
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

```java
public static int Verificar(String usuario) {
    Vector lista = mostrar();

    for(int i = 0; i < lista.size(); ++i) {
        if (lista.size() > 8) {
            return 1;
        }

        Usuarios obj = (Usuarios)lista.elementAt(i);
        if (obj.getUsuario().equalsIgnoreCase(usuario)) {
            return i;
        }
    }

    return -1;
}

public static int VerificarC(String contraseña) {
    Vector lista = mostrar();

    for(int i = 0; i < lista.size(); ++i) {
        Usuarios obj = (Usuarios)lista.elementAt(i);
        if (obj.getUsuario().equalsIgnoreCase(contraseña)) {
            return i;
        }
    }

    return -1;
}
```

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);
          int opccion = lec.nextInt();
100
101         switch(opccion){
102             case 1:
103                 Inicio();
104                 break;
105             case 2:
106                 InicioUsuario();
107                 break;
108
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.

Interfaz de Usuario

```java
public class ventana extends JFrame {

    static String ruta;
    static int contador_alumnos, contador_cursos = 0;
    static Alumnos Alumnos[] = new Alumnos[300];
    static Cursos Cursos[] = new Cursos[300];
    public static Asignacion Asignacion[] = new Asignacion[300];
    public static int contAsignacionAlumnos = 0;
    DateTimeFormatter fmt = DateTimeFormatter.ofPattern("dd/MM/yyyy");
    int contealu;


    public JPanel panel;//declarar panel
    private JTextField cajatexto1;
    private JButton boton;
    private JLabel etiqueta;


    public ventana() {
        this.setSize(950, 500);//tamaño de la ventana
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setTitle("[IPC] practica3");//titulo de la ventana
        setLocationRelativeTo(null);//localizacion de la venta
        setMinimumSize(new Dimension(200, 200));//tamaño minimo de la ventana
        this.getContentPane().setBackground(Color.BLUE);
        IniciarComponentes();
        texto1();//caja de textos
        BOTONES();
```

Para realizar la interfaz se debe importar de la librería Jframe para crear las respectivas ventanas.

```java
private void IniciarComponentes() {

    panel = new JPanel(); //instanciando panel
    JLabel etiqueta = new JLabel();

    panel.setLayout(null);// desactivando el diseño
    panel.setBackground(Color.yellow);
    this.getContentPane().add(panel); //agrega panel a ventana


    etiqueta.setText("ingrese la ruta para cargar alumnos:");
    etiqueta.setBounds(20, 10, 350, 30);//stablece posicion de etiqueta
    panel.add(etiqueta);//añadir etiqueta a panel


}

private void texto1() {
    //caja de texto1
    cajatexto1 = new JTextField();
    cajatexto1.setBackground(Color.green);//color de caja de texto
    cajatexto1.setForeground(Color.black);//color de letra de caja de texto
    cajatexto1.setBounds(350, 10, 450, 25);//establece posicion de caja de texto
    panel.add(cajatexto1); //agregar caja de texto a panel


}
```

Se trabaja dentro de un panel para poder editar y colocar componentes de manera más sencilla.

Las cajas de texto están serán utilizadas para el ingreso de las rutas para la carga de archivos.

Graficas

Barras

```
public Grafica_barras(JPanel grafica) { this.grafica = grafica; }

public void run() {
    try {
        if (ventana.Cursos != null) {
            for (int i = 0; i < ventana.Cursos.length; i++) {
                if (ventana.Cursos[i] != null) {
                    if (ventana.Cursos[i].getId() == ventana2.id_curso) {
                    int alum = 0;
                    Notas temp[] = ventana.Cursos[i].getAluNotas();
                    try {

                        for (int j = 0; j < temp.length; j++) {
                            LocalDate edad = temp[j].getAlumno().getFecha();
                            int nuevaedad = 2021 - edad.getYear() ;
                            if (temp[j] != null) {
                                //for (int i = 0; i < alumnos.length; i++) {
                                sleep(500);
                                //   System.out.println(alumnos[i].getId_alumno());
                                System.out.println(nuevaedad);
                                if (temp[i] != null) {
                                    if (nuevaedad > 0 && nuevaedad <= 11) {
                                        intervalo_10++;
                                    } else if (nuevaedad > 9 && nuevaedad <= 21) {
                                        intervalo_20++;
                                    } else if (nuevaedad > 19 && nuevaedad <= 31) {
                                        intervalo_30++;
                                    } else if (nuevaedad > 29 && nuevaedad <= 41) {
                                        intervalo_40++;
```

Dentro de un ciclo *for* se condicionan las entradas de datos para la barra.

```
}
DefaultCategoryDataset datos_barras = new DefaultCategoryDataset();// default
datos_barras.setValue(intervalo_10, "[0-10]", "[0-10]");
datos_barras.setValue(intervalo_20, "[10-20]", "[10-20]");
datos_barras.setValue(intervalo_30, "[20-30]", "[20-30]");
datos_barras.setValue(intervalo_40, "nose", "[30-40]");
datos_barras.setValue(intervalo_50, "nose", "[40-50]");
datos_barras.setValue(intervalo_60, "nose", "[50-60]");
datos_barras.setValue(intervalo_70, "nose", "[60-70]");
datos_barras.setValue(intervalo_80, "nose", "[70-80]");
datos_barras.setValue(intervalo_90, "nose", "[80-90]");
datos_barras.setValue(intervalo_100, "nose", "[90-100]");
//datos_barras es para añadir los datos que queremos
// //datos_barras.setvalue("Cantidad",nombre eje x,nombre eje y )
//creamos la grafica con JFreechart  de barras
JFreeChart grafico_barras = ChartFactory.createBarChart(
        title: "Edad de los Alumnos en el Curso",    ///nommbre del curso
        categoryAxisLabel: "Edades",   /// nombre eje x
        valueAxisLabel: "Cantidad",   /// nombre eje y
        datos_barras,             // datos o cantidades a ingresar
        PlotOrientation.HORIZONTAL,/// horizontal PlotOrientation.HORIZONTAL  o verticcal  PlotOrientation.VERTICAL
        legend: true,//legenda el dato de abajo >=v de los colores graficados
        tooltips: true,
        urls: false);

java.awt.image.BufferedImage image = grafico_barras.createBufferedImage( width: 600,  height: 500);
muestra = new JLabel(new ImageIcon(image));
muestra.setBounds(10, 20, 800, 500);
grafica.removeAll();//REMOVER LO DEL PANEL
grafica.add(muestra);//añadir
grafica.repaint();//repintar
```

La grafica se etiqueta con las divisiones por parámetros en este caso por edades.

Pie

```
            contadorgenero++;
        } else {
            System.out.println("nel prro");
        }
        double hH = (H / contadorgenero) * 100;
        double mM = (M / contadorgenero) * 100;
        DefaultPieDataset datosGrafica = new DefaultPieDataset();
        double redondeoH = Math.round(H * 100);
        redondeoH = redondeoH / 100;
        double redondeoM = Math.round(M * 100);
        redondeoM = redondeoM / 100;
        datosGrafica.setValue("MUJERES " + Math.round(hH * 0.01 * contadorgenero), redondeoM);
        datosGrafica.setValue("HOMBRES " + Math.round(mM * 0.01 * contadorgenero), redondeoH);
        JFreeChart grafico = ChartFactory.createPieChart( title: "Género de Alumnos en el Curso", datosGrafica, legend: true, tooltips: true, urls: false);
        java.awt.image.BufferedImage image = grafico.createBufferedImage( width: 450, height: 290);
        muestra = new JLabel(new ImageIcon(image));
        muestra.setBounds(0, 0, 400, 400);
        grafica.removeAll();
        grafica.add(muestra);
        grafica.repaint();
        //} else {
        //  System.out.println("no hay nadaaaa");
        //}
        //} else {
        //  System.out.println("No entra aquí ----");
        //}
    } catch (Exception e) {
            System.out.println("Lista de alumnos del curso terminada");
        }
```

La grafica se etiqueta con las divisiones por parámetros en este caso por sexo.