Don Mariano, profesor de literatura, desea realizar una aplicación para facilitar la entrega de trabajos de los alumnos . Antes de embarcarse en un proyecto más ambicioso, desea conocer la acogida que tiene entre sus alumnos y sus compañeros de departamento, por eso decide simplificar mucho la aplicación centrándola únicamente en la entrega de un trabajo que consiste en un comentario de texto. Los usuarios de dicha aplicación van a ser los profesores del departamento y los alumnos. De todos los usuarios vamos a almacenar su dni, nombre, apellidos y contraseña. Los trabajos serán almacenados en el disco duro y serán un fichero de texto cuyo nombre es el dni y la extensión .txt Un profesor podrá leer el trabajo de cualquier alumno, pero un alumno solo podrá leer su trabajo. Los únicos que podrán entregar un trabajo serán los alumnos. El funcionamiento de la aplicación será el que se detalla a continuación: Cuando se inicia por primera vez, se avisará de que se está iniciando por primera vez y se pedirán los datos de usuario, dicho usuario será un profesor. A continuación se mostrará el menú del profesor. Si no es la primera vez, se leerá de un fichero binario la información de los usuarios y se pedirá un dni y una contraseña. Si no son válidos, se vuelven a pedir, si son válidos, se muestra el menú correspondiente al usuario. El menú de profesor consta de las siguientes opciones: - Crear usuario: pedirá los datos para crear un nuevo usuario que podrá ser otro profesor o un alumno - Listar usuarios: mostrará un listado de todos los usuarios, mostrando un usuario por línea que, en el caso de los profesores serán dni, nombre y apellidos y, en el caso de los alumnos, dni, nombre, apellidos, si ha entregado o no el trabajo, y, en caso de que lo haya entregado, si el trabajo es válido o no (se considerará un trabajo válido si tiene entre 10 y 20 líneas) - Mostrar trabajo: se le pedirá el dni del alumno y mostrará por pantalla el fichero de texto de dicho alumno - Cambiar de usuario: pedirá el dni y si está registrado en la lista de usuarios, pedirá la contraseña. Si ésta coincide con la almacenada, éste pasará a ser el usuario actual y mostrará el menú correspondiente. - Salir: sale de la aplicación guardando previamente los datos de los usuarios El menú del alumno tiene las siguientes opciones: - Entregar trabajo: Se le pedirá al alumno que vaya escribiendo línea a línea el documento a entregar y será guardando en el documento de texto correspondiente. Para terminar, el alumno escribirá una línea con el texto FIN, que podrá estar escrito en mayúsculas, minúsculas o una combinación de ambos y no deberá guardarse. - Mostrar trabajo: mostrará por pantalla el fichero de texto de dicho alumno (no será necesario pedir el dni ya que será el del alumno que hace la petición. - Cambiar de usuario: igual que en el menú del profesor - Salir: igual que en el menú del profesor Para aprobar el examen será imprescindible que éste no tenga errores de compilación.

```
package tercera2021rojos;

import java.io.*;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

/**
 *
 * @author anusk
 */
public class Tercera2021Rojos {

    public static ArrayList<Usuario> usuarios = new ArrayList<Usuario>();
    public static Usuario logueado;
    public static final String FDATOS = "usuarios.bin";

    public static int leerInt()
    {
        Scanner sc = new Scanner(System.in);
        while (true)
        {
            try
            {
                return sc.nextInt();
            }
            catch (Exception e)
            {
                System.out.println("NÃºmero no vÃ¡lido");
                sc.nextLine();
            }
```

```
    }
}

public static int menuProfesor()
{
    int op = 0;
    while (op<1 || op >5)
    {
        System.out.println("1. Crear usuario");
        System.out.println("2. Listar usuarios");
        System.out.println("3. Mostrar trabajo");
        System.out.println("4. Cambiar de usuario");
        System.out.println("5. Salir");
        op = leerInt();
    }
    return op;
}

public static int menuAlumno()
{
    int op = 0;
    while (op<1 || op >4)
    {
        System.out.println("1. Entregar trabajo");
        System.out.println("2. Mostrar trabajo");
        System.out.println("3. Cambiar de usuario");
        System.out.println("4. Salir");
        op = leerInt();
    }
    return op;
}
```

```
public static void crearUsuario()
{
   Scanner sc = new Scanner(System.in);
   System.out.print("DNI: ");
   String dni = sc.nextLine();
   System.out.print("Nombre y apelliodos: ");
   String nombre = sc.nextLine();
   System.out.print("Password: ");
   String pass = sc.nextLine();
   int op = 0;
   while (op!=1 && op!=2)
   {
      System.out.println("Tipo de usuario:\n1. Alumno\n2. Profesor");
      op = leerInt();
   }
   if (op==1)
      usuarios.add(new Alumno(dni,nombre,pass));
   else
      usuarios.add(new Profesor(dni,nombre,pass));
}

public static void crearProfesor()
{
   System.out.println("Se estÃ¡ iniciando la aplicaciÃ³n por primera vez.");
   System.out.println("Dando de alta a un profesor...");
   Scanner sc = new Scanner(System.in);
   System.out.print("DNI: ");
   String dni = sc.nextLine();
   System.out.print("Nombre y apelliodos: ");
   String nombre = sc.nextLine();
```

```
    System.out.print("Password: ");
    String pass = sc.nextLine();
    Profesor p = new Profesor(dni,nombre,pass);
    usuarios.add(p);
    logueado = p;

}

public static void listarUsuarios()
{
    Iterator<Usuario>it = usuarios.iterator();
    while (it.hasNext())
        System.out.println(it.next());
}

public static void mostrarTrabajo()
{
    logueado.mostrarTrabajo();
}

public static Usuario buscar(String dni)
{
    Iterator<Usuario>it = usuarios.iterator();
    while (it.hasNext())
    {
        Usuario u = it.next();
        if (u.getDni().equals(dni))
            return u;
    }
    return null;
}
```

```
public static void cambiarUsuario()
{
   logueado = null;
   Scanner sc = new Scanner(System.in);

   while (logueado==null)
   {
      Usuario u = null;
      String dni="";
      while (u==null)
      {
         System.out.print("DNI: ");
         dni = sc.nextLine();
         u = buscar(dni);
         if (u==null)
            System.out.println("Usuario invÃ¡lido");
      }

      System.out.print("Password: ");
      String pass = sc.nextLine();
      if (u.getPass().equals(pass))
      {
         logueado = u;
         System.out.println("Se ha logueado el usuario "+dni);
      }
      else
      {
         System.out.println("Password incorrecto");
      }
   }
```

```
}

public static void guardarDatos()
{
   ObjectOutputStream oos = null;
   try
   {
      oos = new ObjectOutputStream(new FileOutputStream(FDATOS));
      oos.writeObject(usuarios);
   }
   catch (IOException e)
   {
      System.out.println("Error guardando datos");
   }
   finally
   {
      try
      {
         if (oos!=null)
            oos.close();
      }
      catch (IOException e)
      {
         System.out.println("Error cerrando el fichero");
      }
   }

}

public static void recuperarDatos()
```

```
{
   ObjectInputStream ois = null;
   try
   {
      ois = new ObjectInputStream(new FileInputStream(FDATOS));
      usuarios = (ArrayList<Usuario>) ois.readObject();
      cambiarUsuario();
   }
   catch (ClassNotFoundException e)
   {
      System.out.println("Error en los datos");
   }
   catch (FileNotFoundException e)
   {
      crearProfesor();
   }
   catch (IOException e)
   {
      System.out.println("Error leyendo el fichero");
      crearProfesor();
   }
    catch (Exception e)
   {
      System.out.println("Error en los datos");
      crearProfesor();
   }
   finally
   {
      try
      {
         if (ois!=null)
```

```
            ois.close();
        }
        catch (IOException e)
        {
            System.out.println("Error cerrando el fichero");
        }
    }
}

public static void main(String[] args) {

    recuperarDatos();
    boolean salir = false;
    while (!salir)
    {
        if (logueado instanceof Profesor)
        {
            int op = menuProfesor();
            switch (op)
            {
                case 1: crearUsuario(); break;
                case 2: listarUsuarios(); break;
                case 3: mostrarTrabajo(); break;
                case 4: cambiarUsuario(); break;
                case 5: salir = true; break;
            }
        }
        else
        {
            int op = menuAlumno();
            switch (op)
```

```
            {
                case 1: ((Alumno)logueado).entregarTrabajo();break;
                case 2: mostrarTrabajo(); break;
                case 3: cambiarUsuario(); break;
                case 4: salir = true; break;
            }
          }
      }
      guardarDatos();
   }

}
```

USUSARIO.JAVA

```java
package tercera2021rojos;

import java.io.Serializable;

/**
 *
 * @author anusk
 */
public abstract class Usuario implements Serializable {

    protected String dni;
    protected String nombre;
    protected String pass;

    public Usuario(String d, String n, String p)
    {
        dni = d;
        nombre = n;
        pass = p;
    }

    public String getDni() {
        return dni;
    }

    public String getPass() {
        return pass;
    }

    public abstract void mostrarTrabajo();

    @Override
    public String toString()
    {
        return dni+"\t"+nombre;
    }

}
```

PROIFESIR

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

/**
 *
 * @author anusk
 */
public class Profesor extends Usuario{

    public Profesor(String d, String n, String p)
    {
        super(d,n,p);
    }

    @Override
    public void mostrarTrabajo()
    {

        Scanner sc = new Scanner(System.in);
        System.out.println("DNI del alumno: ");
        String dniA = sc.nextLine();
        String f = dniA+".txt";
        BufferedReader br = null;
        try
        {
            br = new BufferedReader(new FileReader(f));
```

```
            String linea = br.readLine();
            while (linea!=null)
            {
               System.out.println(linea);
               linea = br.readLine();
            }
         }
         catch (FileNotFoundException e)
         {
            System.out.println("El trabajo no se ha entregado");
         }
         catch (IOException e)
         {
            System.out.println("Error leytendo el fichero");
         }
         finally
         {
            try
            {
               if (br!=null)
                  br.close();
            }
            catch (IOException e)
            {
               System.out.println("Error cerrando el fichero");
            }
         }
      }
   }
```

ALUMNO

```
package tercera2021rojos;

import java.io.*;
import java.util.Scanner;

/**
 *
 * @author anusk
 */
public class Alumno extends Usuario{

    public Alumno(String d, String n, String p)
    {
        super(d,n,p);
    }

    public void entregarTrabajo()
    {
        String f = dni+".txt";
        Scanner sc = new Scanner(System.in);
        BufferedWriter bw = null;
        System.out.println("Escribe el documento. Termina con una lÃ-nea con el texto FIN");
        try
        {
            bw = new BufferedWriter(new FileWriter(f));
            String linea = sc.nextLine();
            while (!linea.equalsIgnoreCase("FIN"))
            {
                bw.write(linea);
```

```
            bw.newLine();
            linea = sc.nextLine();
        }
    }
    catch (IOException e)
    {
        System.out.println("Error escribiendo el fichero de texto");
    }
    finally
    {
        try
        {
            if (bw!=null)
                bw.close();
        }
        catch (IOException e)
        {
            System.out.println("Error cerrando el fichero de texto");
        }
    }
}

public boolean entregado()
{
    File f = new File(dni+".txt");
    return f.exists();
}

public boolean esValido()
{
    String f = dni+".txt";
```

```
BufferedReader br = null;
try
{
   br = new BufferedReader(new FileReader(f));
   int cont = 0;
   while (br.readLine()!=null)
      cont++;
   return cont>=10 && cont<=20;
}
catch (FileNotFoundException e)
{
   System.out.println("El trabajo no se ha entregado");
   return false;
}
catch (IOException e)
{
   System.out.println("Error leytendo el fichero");
   return false;
}
finally
{
   try
   {
      if (br!=null)
         br.close();
   }
   catch (IOException e)
   {
      System.out.println("Error cerrando el fichero");
   }
}
```

```
}

@Override
public void mostrarTrabajo()
{
   String f = dni+".txt";
   BufferedReader br = null;
   try
   {
      br = new BufferedReader(new FileReader(f));
      String linea = br.readLine();
      while (linea!=null)
      {
         System.out.println(linea);
         linea = br.readLine();
      }
   }
   catch (FileNotFoundException e)
   {
      System.out.println("El trabajo no se ha entregado");
   }
   catch (IOException e)
   {
      System.out.println("Error leytendo el fichero");
   }
   finally
   {
      try
      {
         if (br!=null)
```

```
                br.close();
            }
            catch (IOException e)
            {
               System.out.println("Error cerrando el fichero");
            }
        }
    }


    public String toString()
    {
        return super.toString()+"\t"+(entregado()?"Entregado":"No
entregado")+"\t"+(entregado()&& esValido()?"VÃ¡lido":(entregado()?"No valido":""));
    }
}
```