**EJERCICIO 2:**

Hacer una aplicación JAVA que gestione los alumnos de un centro escolar. Los datos se almacenan en un fichero serializado, de nombre **alumnos.dat**.

Los datos que se almacenan de cada alumno son:

* Nº de expediente
* Nombre
* Nota 1º parcial.
* Nota 2º parcial.
* Nota 3º parcial.

Mediante un menú se podrán realizar determinadas operaciones:

* Añadir Alumno. Esta opción pedirá los datos del cliente y añadirá el registro correspondiente en el fichero.
* Listar todos los alumnos. Obtener un listado en el que aparezcan los datos del estudiante más la media de sus tres notas.
* Buscar Alumno. Pedirá al usuario el número de expediente a buscar, y si existe, lo muestra.
* Borrar Alumno. Pedirá al usuario el número de expediente a buscar, y si existe, lo borrará del fichero.
* Borrar un fichero, es decir eliminarlo del disco.
* Salir de la aplicación.

1. Empezar creando la Java Clas Library **MiClaseAlumno** en la que se incluirá la clase **Alumno** (repasa el ejercicio **G33** de los ejercicios guiados).Esta Librería se incluirá en la Biblioteca de los proyectos que se diseñen para esta aplicación.

import java.io.\*;

import java.util.Scanner;

public class **Alumno** implements Serializable{

private int num\_expediente;

private String nombre;

private int []notas=new int[3];

public **Alumno** ()

{}

public **Alumno** (int num\_expediente, String nombre, int []notas ) {

this.num\_expediente = num\_expediente;

this.nombre = nombre;

this.notas=notas;

}

public int getNum\_expediente() {

return num\_expediente;

}

public void setNum\_expediente(int num\_expediente) {

this.num\_expediente = num\_expediente;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public int[] getNotas() {

return notas;

}

public void setNotas(int[] notas) {

this.notas = notas;

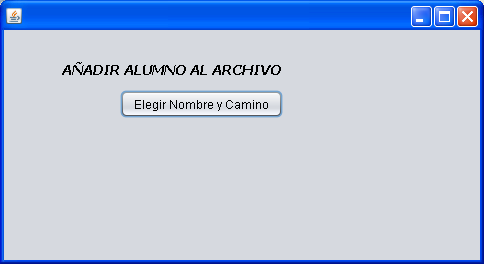
}

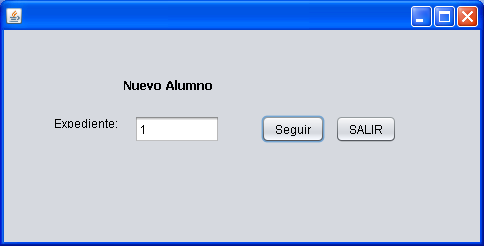
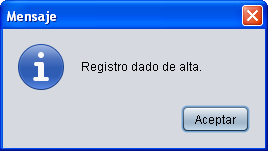
}

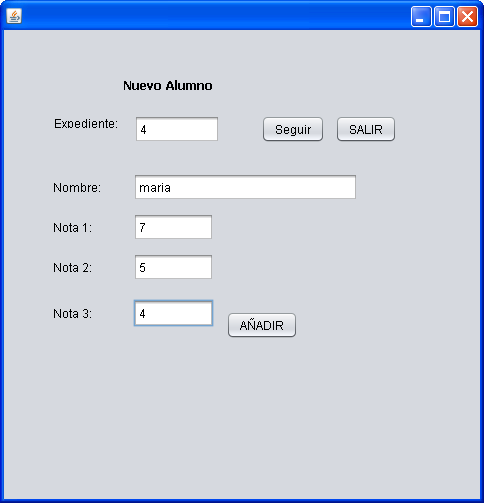
1. Añadir alumno al archivo introduciendo los datos por consola.

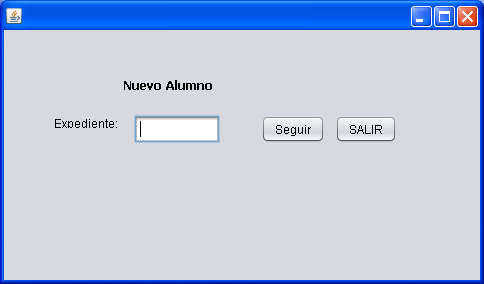
Crea el proyecto **AddAlumno.**

El registro sólo se dará de alta en el caso de no estarlo ya.









Opción para salir de la aplicación

Implementa las clases:

* Principal: AñadirAlumno.java
* ControlArchivo.java.

Esta clase contiene el método:

**public boolean existeAlumno(String nomfich,int expediente)**

Este método recorre todo el fichero y comprueba si un número de expediente está o no dado de alta.

Se ejecuta cuando pulsamos **seguir** y si devuelve **true** avisa y se queda en este panel.

Si devuelve **false** pasamos al panel siguiente.

También tiene el método:

**public void añadir(String nomfich,int expediente,String nombre, int []notas)**

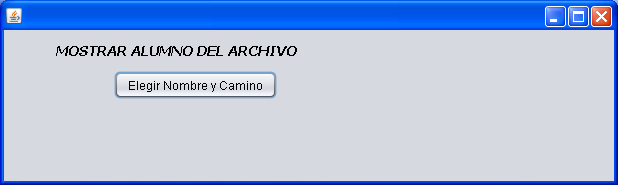
Este método abre el archivo para añadir, crea un objeto y lo escribe.

1. Obtener un listado de todos los alumnos.

Crea el proyecto **ListarAlumno.**

Para esta opción necesitaremos un objeto JTable. Para aprender su manejo se practicará el ejercicio guiado **G49**.

Se mostrará también la nota media de cadaalumno.





Implementa las clases:

* Principal: **listaAlumno.java**.

Necesitaremos declarar un:

DefaultTableModel m;

Esta clase contiene los métodos:

private void **PrepararTabla()** y private void **LlenarTabla()** similares a los diseñados en la aplicación **Clientes.**

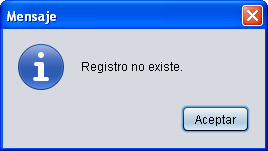
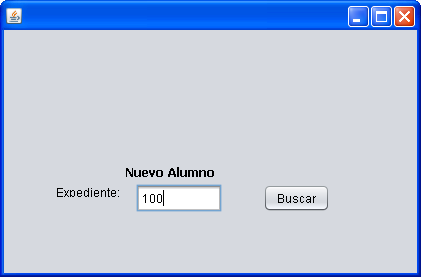
* ControlArchivo.java. Añade a esta clase el método:

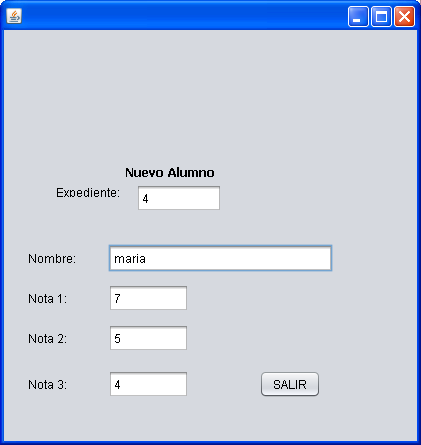
**public ArrayList mostrar(String nomfich)**

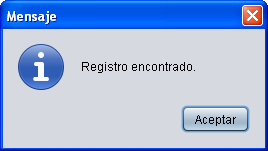
Este método devuelve una lista con el contenido del archivo.

1. Buscar alumno por número de expediente. Recuerda que no hay números de expedientes repetidos.







Implementa las clases:

* Principal: BuscarAlumno.java
* ControlArchivo.java.

Añade a esta clase el método:

**public Alumno buscarAlumno(String nomfich,int expediente)**

Este método recorre todo el fichero y comprueba si un número de expediente está o no dado de alta, si es así devuelve un Alumno en caso contrario devuelve null.

1. Borrar un alumno que buscamos por número de expediente.

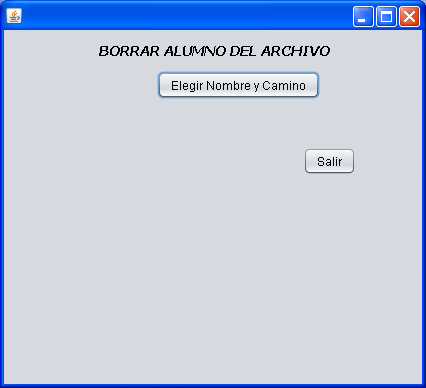
Es necesario crear un archivo auxiliar de nombre **temporal.dat** que guarda los alumnos que vamos leyendo excepto el que coincida con el que queremos borrar.

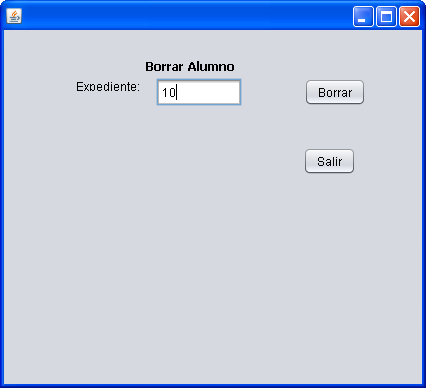
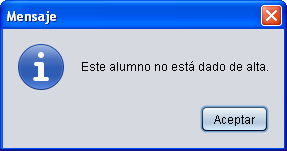
Posteriormente se borrará el archivo de alumnos y se renombrará con ese nombre el temporal.dat.

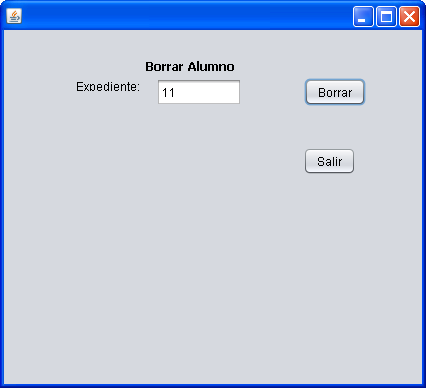
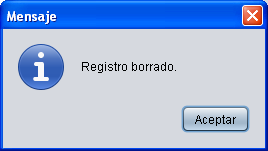
Supongamos que tenemos un archivo con esta información:



El programa hará lo siguiente:







Implementa las clases:

* Principal: BorrarAlumno.java
* ControlArchivo.java.

Añade a esta clase el método:

**public void borrarAlumno(String nomfich,int expediente)**

Este método toma un archivo origen dónde está el registro a borrar y hace una copia en un archivo temporal de todos los registros menos el que se va a borrar, luego se borra el archivo origen y se renombra el temporal con el nombre del archivo origen.

Puedes utilizar este método que extrae de un nombre de archivo la subcadena que va desde el principio hasta el último **\**.

public static String camino(String nomfich)

{

int i=nomfich.length()-1;

boolean b=true;

while(i>=0 && b )

{

if (nomfich.charAt(i)=='\\')

{

b=false;

}

else

{

i--;

}

}

String camino=nomfich.substring(0, i+1);

return camino;

}

1. Borrar un fichero, es decir eliminarlo del disco.

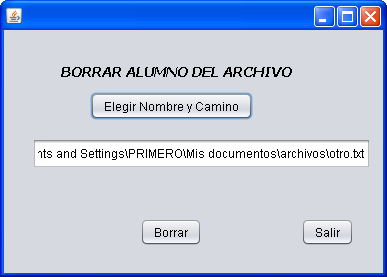
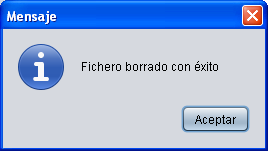
Implementa las clases:

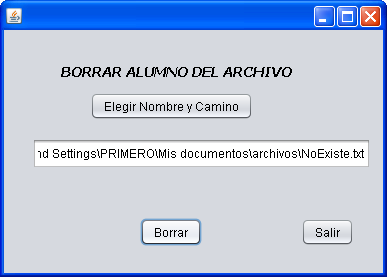
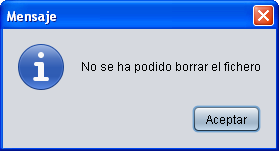
* Principal: BorrarArchivo.java
* ControlArchivo.java.

Añade a esta clase el método:

**public static void borrarArchivo(String nomfich)**







1. Diseña una aplicación con un menú que incluya las opciones anteriores.

