

DOCUMENTACIÓN

ACTIVIDAD 1.2:

GESTIÓN DE NOTAS

CON ACCESO

SECUENCIAL Y

ALEATORIO

Rocío Mora García.
2ºDAM-DUAL

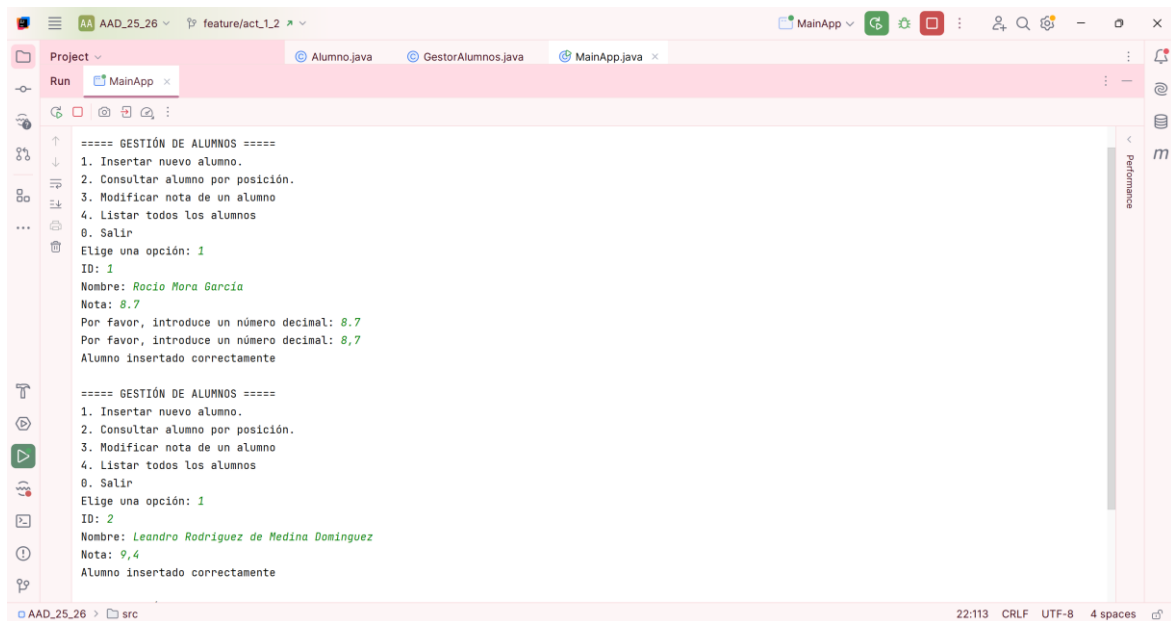
El objetivo de esta actividad es desarrollar una aplicación Java que gestione un fichero binario con registros de alumnos mediante acceso secuencial y aleatorio, utilizando la clase `RandomAccessFile`.

La aplicación está compuesta por tres clases, cada una de ellas cumple un propósito:

- **Alumno.** Define el formato binario de cada registro (52 bytes fijos).
- **GestorAlumnos.** Gestiona la lectura, escritura y modificación en el fichero.
- **MainApp.** Ofrece un menú de consola para interactuar con los datos.

Para comprobar su funcionamiento, he realizado diferentes pruebas:

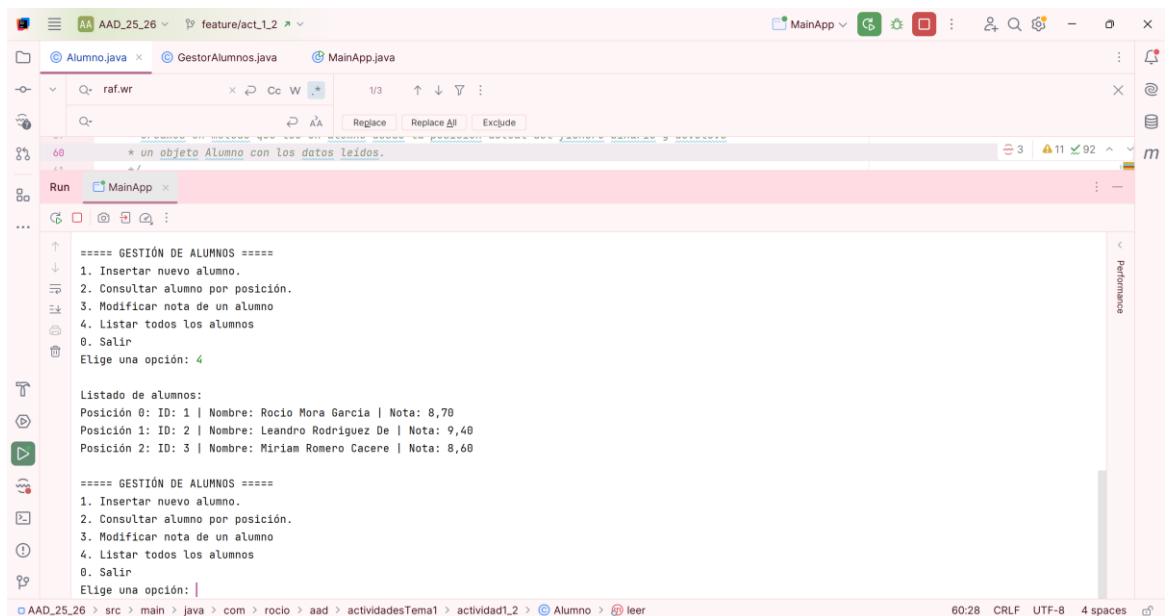
1. Inserción de alumnos.



```
==== GESTIÓN DE ALUMNOS ====
1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno
4. Listar todos los alumnos
0. Salir
Elige una opción: 1
ID: 1
Nombre: Rocio Mora Garcia
Nota: 8,7
Por favor, introduce un número decimal: 8,7
Por favor, introduce un número decimal: 8,7
Alumno insertado correctamente

==== GESTIÓN DE ALUMNOS ====
1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno
4. Listar todos los alumnos
0. Salir
Elige una opción: 1
ID: 2
Nombre: Leandro Rodriguez de Medina Dominguez
Nota: 9,4
Alumno insertado correctamente
```

2. Listar alumnos.



```
==== GESTIÓN DE ALUMNOS ====
1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno
4. Listar todos los alumnos
0. Salir
Elige una opción: 4

Listado de alumnos:
Posición 0: ID: 1 | Nombre: Rocio Mora Garcia | Nota: 8,70
Posición 1: ID: 2 | Nombre: Leandro Rodriguez De | Nota: 9,40
Posición 2: ID: 3 | Nombre: Miriam Romero Cacerre | Nota: 8,60

==== GESTIÓN DE ALUMNOS ====
1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno
4. Listar todos los alumnos
0. Salir
Elige una opción: |
```

3. Consultar posición.

```
60 /* un objeto Alumno con los datos leídos. */
61
62
```

Run MainApp

3. Modificar nota de un alumno
4. Listar todos los alumnos
0. Salir
Elige una opción: 2
Introduce la posición del alumno: 2
Datos del alumno -> ID: 3 | Nombre: Miriam Romero Cacere | Nota: 8,60

===== GESTIÓN DE ALUMNOS =====
1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno
4. Listar todos los alumnos
0. Salir
Elige una opción: 2
Introduce la posición del alumno: 0
Datos del alumno -> ID: 1 | Nombre: Rocío Mora García | Nota: 8,70

===== GESTIÓN DE ALUMNOS =====
1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno

4. Modificar nota.

1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno
4. Listar todos los alumnos
0. Salir
Elige una opción: 2
Introduce la posición del alumno: 0
Datos del alumno -> ID: 1 | Nombre: Rocío Mora García | Nota: 8,70

===== GESTIÓN DE ALUMNOS =====
1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno
4. Listar todos los alumnos
0. Salir
Elige una opción: 3
Posición del alumno: 2
Nueva nota: 9,8
Nota modificada correctamente.

5. Listar tras modificar la nota.

Nueva nota: 9,8
Nota modificada correctamente.

===== GESTIÓN DE ALUMNOS =====
1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno
4. Listar todos los alumnos
0. Salir
Elige una opción: 4

Listado de alumnos:
Posición 0: ID: 1 | Nombre: Rocío Mora García | Nota: 8,70
Posición 1: ID: 2 | Nombre: Leandro Rodríguez De | Nota: 9,40
Posición 2: ID: 3 | Nombre: Miriam Romero Cacere | Nota: 9,80

===== GESTIÓN DE ALUMNOS =====
1. Insertar nuevo alumno.
2. Consultar alumno por posición.
3. Modificar nota de un alumno

Las inserciones realizadas se guardan en un fichero *.dat* que se va a generar dentro de la carpeta del proyecto. En el caso de que el programa se ejecute en otro ordenador y la carpeta no existe, este la crea automáticamente para evitar errores.