

# Documentación Técnica

## a) Descripción del sistema

**Objetivo del programa:** El programa **ActualitzarNota** tiene como objetivo principal permitir la modificación de las notas de los alumnos almacenadas en un archivo de texto llamado **NotasDAM2.txt**. El programa busca el archivo de manera recursiva en el sistema de archivos y actualiza la nota de un alumno especificado por el usuario.

## b) Diagrama de clases y secuencias

### Diagrama de Clases

```
• +-----+
• | ActualitzarNota |
• +-----+
• | + findFile()      |
• | + modificarNota()|
• | + main()          |
• +-----+
```

### Diagrama de Secuencias

1. El usuario inicia el programa.
2. Se solicita el nombre del alumno y la nueva nota.
3. Se llama al método **findFile()** para buscar **NotasDAM2.txt**.
4. Si se encuentra el archivo, se llama al método **modificarNota()** para actualizar la nota del alumno.
5. Se muestra un mensaje indicando si la nota se ha modificado correctamente o si ha ocurrido un error.

## c) Detalle del código

### 1. Clase: **ActualitzarNota**

#### Objetivo:

Gestionar la búsqueda y modificación de notas de alumnos en un archivo.

#### Métodos:

- **findFile(String startDir, String fileName)**
  - **Objetivo:** Buscar el archivo especificado de manera recursiva.
  - **Parámetros:**
    - **startDir:** Directorio de inicio para la búsqueda.
    - **fileName:** Nombre del archivo a buscar.
  - **Valor de retorno:** Ruta absoluta del archivo encontrado o **null** si no se ha encontrado.
  - **Excepciones gestionadas:** **IOException**.
- **modificarNota(String filePath, String alumne, char novaNota)**
  - **Objetivo:** Modificar la nota de un alumno en el archivo especificado.
  - **Parámetros:**
    - **filePath:** Ruta del archivo donde se realizará la modificación.
    - **alumne:** Nombre del alumno a modificar.
    - **novaNota:** Nueva nota a establecer.
  - **Valor de retorno:** No retorna ningún valor.
  - **Excepciones gestionadas:** **IOException**.

### 2. Comportamiento esperado del código

- El programa debe buscar el archivo **NotasDAM2.txt** a partir del directorio especificado.
- Si encuentra el archivo, lee línea por línea y busca al alumno especificado. Si lo encuentra, actualiza la nota y vuelve a escribir el archivo con la nueva información.
- En caso de no encontrar al alumno o si hay un error al leer o escribir el archivo, el programa informa al usuario.

### 3. Posibles excepciones que se gestionan

- **IOException:** Gestión de errores durante la lectura o la escritura del archivo.
- **Otros errores:** Validaciones de entrada (como el formato del nombre o de la nota) que pueden desencadenar errores.

## d) Entorno de desarrollo

- **Versión del lenguaje Java utilizado:** Java JDK 23 o superior.
- **Bibliotecas dependientes:**
  - `java.io.*`
  - `java.nio.file.*`
  - `java.util.*`

## e) Comentarios en el código

Los comentarios dentro del código son una parte esencial de la documentación. Cada método y función debe estar comentada con:

- **Finalidad:** Explicación breve de lo que hace el método.
- **Parámetros de entrada:** Descripción de los parámetros que recibe el método.
- **Valores de retorno:** Descripción de lo que retorna, si aplica.
- **Posibles excepciones:** Enumeración de las excepciones que puede lanzar el método.

```
/**  
  
 * Busca un archivo de manera recursiva a partir de un directorio inicial.  
  
 *  
  
 * @param startDir Directorio de inicio para la búsqueda.  
  
 * @param fileName Nombre del archivo a buscar.  
  
 * @return Ruta absoluta del archivo encontrado o null si no se ha encontrado.  
  
 * @throws IOException si hay un error de entrada/salida.  
  
 */  
  
public static String findFile(String startDir, String fileName) throws IOException {  
  
    // Implementación del método...  
  
}
```