

CASO PRÁCTICO 7

- **TÍTULO: Programación orientada a objetos con Java**

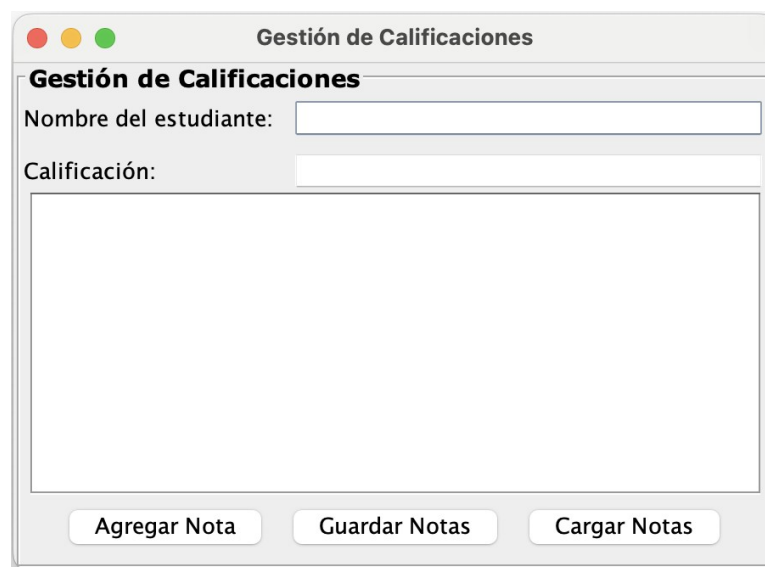
- **SITUACIÓN**

Tenemos que resolver los siguientes problemas para la empresa de programación para la que trabajamos.

- **INSTRUCCIONES**

Crea una aplicación en Java que permita gestionar las notas de los estudiantes utilizando arrays, con una interfaz gráfica con Swing que contenga lo siguiente:

- Un campo de texto para escribir el nombre del estudiante.
- Un campo de texto para escribir la nota del estudiante.
- Un área de texto (JTextArea) no editable donde se muestren las notas de los estudiantes.
- Botones para agregar nuevas notas, guardar notas en un archivo y cargar notas desde un archivo.



The image shows a Java Swing window titled "Gestión de Calificaciones". The window has a title bar with three colored buttons (red, yellow, green) on the left. Inside the window, there is a header section with the title "Gestión de Calificaciones". Below the header, there are two labels: "Nombre del estudiante:" followed by a text input field, and "Calificación:" followed by another text input field. Below these input fields is a large, empty text area (JTextArea). At the bottom of the window, there are three buttons: "Agregar Nota", "Guardar Notas", and "Cargar Notas".

- Al hacer clic en el botón "Agregar Nota", la aplicación debe agregar la nota del estudiante a los arrays correspondientes (String[] nombres y double[] notas). Antes de agregar una nueva nota, el programa debe validar si el nombre y la nota son correctos (el nombre no puede estar vacío, y la nota debe ser un número entre 0 y 10). Si el estudiante ya existe en los arrays, el programa debe preguntar al usuario si desea modificar la nota existente. Si el usuario acepta, se actualizará la nota del estudiante; si no, no se hará ningún cambio.

The screenshot shows a window titled "Gestión de Calificaciones". It contains two input fields: "Nombre del estudiante:" with the value "Pepe" and "Calificación:" with the value "6.75". Below these fields is a text area displaying "Pepe - 5.25". At the bottom of the window are three buttons: "Agregar Nota", "Guardar Notas", and "Cargar Notas".

The screenshot shows a dialog box titled "Modificar Nota" overlaid on the "Gestión de Calificaciones" window. The dialog box contains a small icon of a character with a red nose and a speech bubble. The text inside the dialog box reads: "El estudiante Pepe ya tiene una nota. ¿Deseas modificarla?". At the bottom of the dialog box are two buttons: "No" and "Sí".



- Al hacer clic en el botón "Guardar Notas", las notas deben guardarse en un archivo llamado notas.txt. El archivo debe contener una línea para cada estudiante con el formato: Nombre Estudiante - Nota. Si el archivo ya existe, debe sobrescribirse con las nuevas notas.
- Al hacer clic en el botón "Cargar Notas", la aplicación debe cargar las notas desde el archivo notas.txt si este existe. Las notas deben ser cargadas y mostradas en el área de texto de la aplicación.

Por otra parte, si el nombre del estudiante ya existe al intentar agregar una nota, se debe mostrar un mensaje con las opciones "Sí" y "No". Si el usuario selecciona "Sí", la nota se modificará. Si selecciona "No", no se realizará ninguna modificación. Si el estudiante no existe, se agrega una nueva entrada con la nota.

Las notas de los estudiantes deben mostrarse en el área de texto (JTextArea) en tiempo real cada vez que se agregue una nueva nota o se carguen desde el archivo.

Los arrays nombres y notas deben tener un tamaño máximo de 30 elementos (puedes ajustar este tamaño según sea necesario).

El archivo notas.txt debe estar en el mismo directorio que el programa o en una ruta accesible.

La aplicación debe ser capaz de manejar errores como entradas incorrectas o problemas al leer/escribir el archivo.