

# Programa de Singleton: Archivo de registro

## Resumen

Este programa simula un sistema monitoreado que guarda registros de cuando un usuario inicia sesión, crea y borra su archivo personal y cierra sesión, se creo la clase del archivo de registros como un Singleton para que los usuarios escriban en el mismo registro y que un administrador pueda ver todos los registros a la vez.

## Clases

### AlmacenDeRegistros.java

Clase con patrón Singleton que contiene una instancia de sí misma y puede asignársela a los clientes con el método estático getRegistro(), contiene un Array de Strings al que se le pueden añadir registros con el método addRegistro() que guarda la hora y el contenido del mensaje en el Array, finalmente un administrador puede activar el método verRegistros() para desplegar todo el contenido almacenado en el Array.

### Cliente.java

Son las entidades que pueden generar registros para el almacén, tienen 5 métodos, uno genera el registro de inicio de sesión, otro genera el archivo personal si este no existe y genera un registro de esta acción, otro borra el archivo personal si este existe y genera un registro de esta acción, otro trata de acceder al almacén de registros, pero solo funciona con el Override de la clase Administrador, de lo contrario despliega un mensaje de error, el último genera un registro del cierre de sesión, el archivo personal es simulado por un booleano que sirve para indicar si se pueden realizar las acciones de crear y borrar.

### Administraor.java

Hereda la clase cliente y reemplaza el método de verRegistro por uno que activa la función de mostrar registros del archivo.

## Simulación.

Inicia creando la instancia de un cliente y despliega un menú con las posibles acciones, se añade al menos un registro de inicio de sesión y de pueden realizar las acciones hasta el cierre de sesión que también genera un registro.

Se crea una segunda instancia que también abre su menú, inicia la sesión y puede realizar las acciones establecidas en el menú hasta que cierre su sesión.

La tercera instancia es de tipo Administrador y además de realizar lo mismo de las instancias anteriores, puede invocar el archivo de registros y mostrar en consola todos los registros generados en la única instancia de archivo por los usuarios.

## Conclusión

Este es un caso de uso del patrón Singleton que permite que una sola instancia sea usada por todos los elementos del programa, esto nos permite llevar un control general de los movimientos de los usuarios en el sistema.