Programación de

Servicios y Procesos

CLIENTE-SERVIDOR

Diego González Martínez | 13/01/2023

# Ejercicio 6

**¿De qué trata el Ejercicio 6B?**

**Actividad 6.2.** Utilizando la aplicación desarrollada en la actividad anterior, configura las políticas de acceso para:

* Firmar digitalmente la aplicación.
* Que sólo pueda leer los datos del directorio c:/datos.

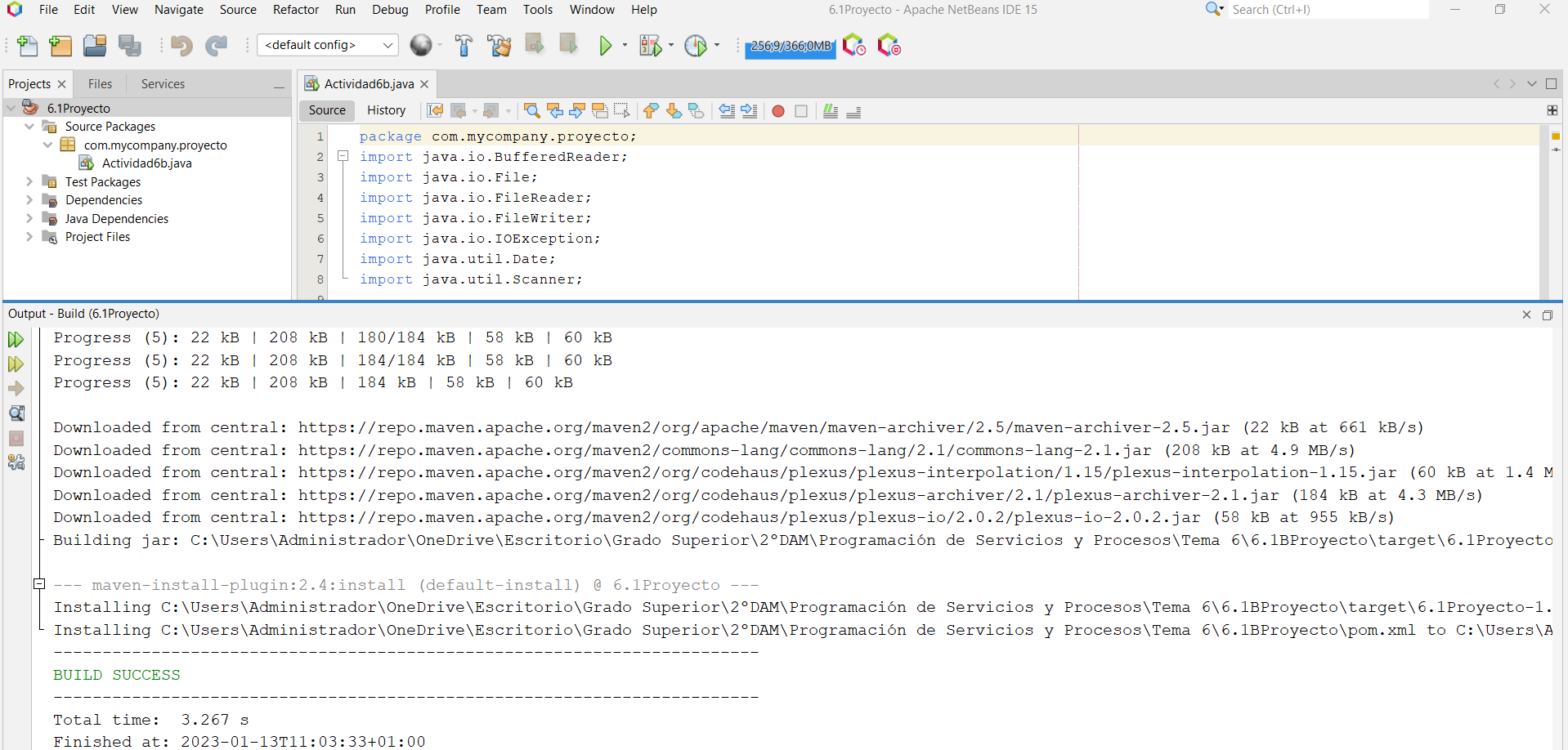
**¿Cuál es la funcionalidad?**

Este es un programa en Java que solicita al usuario un nombre de usuario y un nombre de archivo, los valida y luego lee el contenido del archivo especificado si se encuentra en el directorio "c:\datos". El programa también registra la actividad del usuario al abrir el archivo agregando una línea a un archivo de texto llamado "activity\_log.txt". El nombre de usuario debe tener 8 caracteres y solo letras minúsculas, y el nombre de archivo debe tener 8 caracteres y una extensión de 3 caracteres. Si el usuario ingresa una entrada inválida, el programa les pedirá que lo intenten de nuevo.

**Firma Digital**

Para firmar digitalmente la aplicación he tenido que realizar estos pasos:

1. Crear el jar de la aplicación (tienes que darle al martillo de arriba del todo):



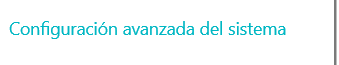
1. Nos vamos al cmd y ejecutamos este comando:

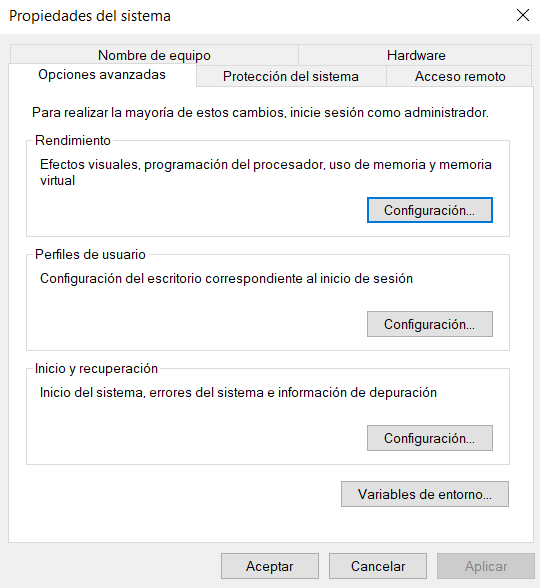


A mi me ha dado este error:

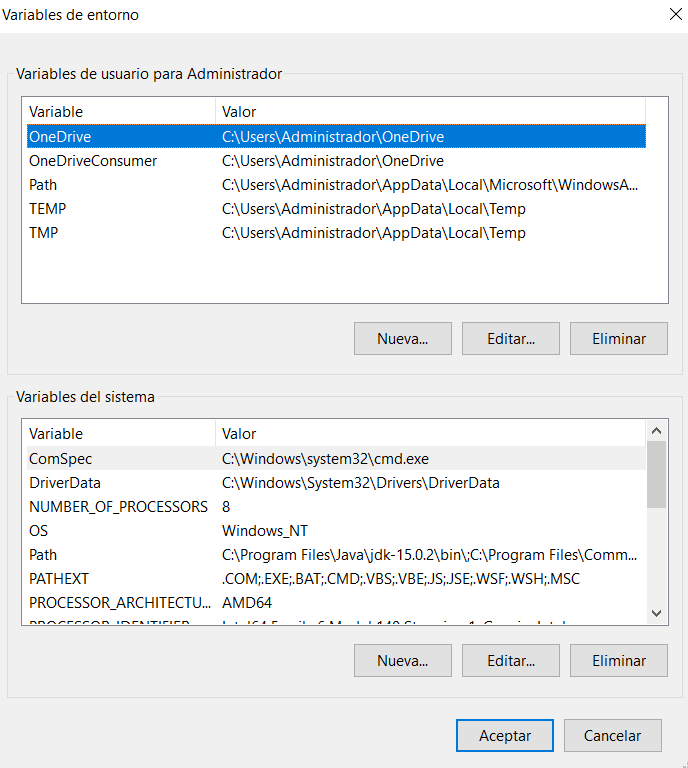
keytool no se reconoce como un comando interno o externo.

Para arreglarlo hay que hacer:

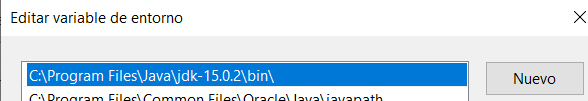




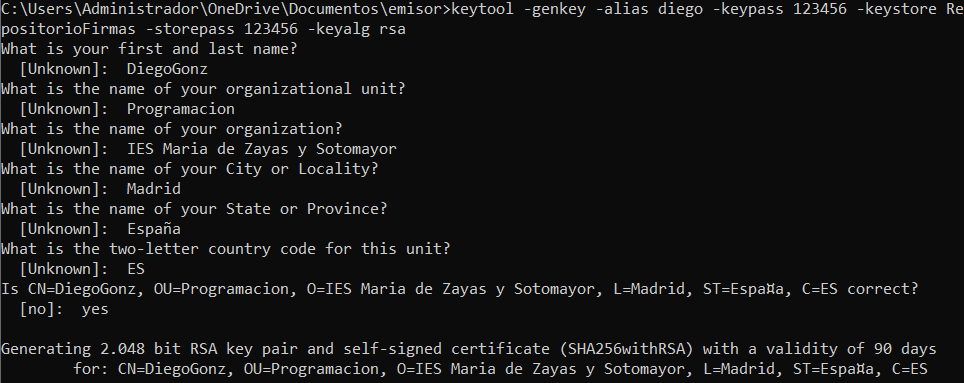
Nos vamos a variables de entorno



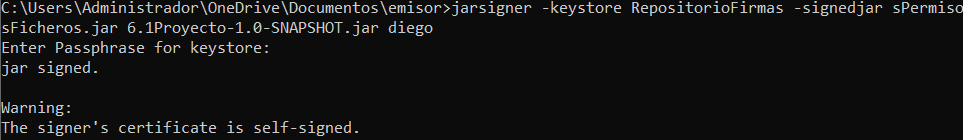
En variables del sistema le damos a editar y ponemos una nueva ruta donde se encuentra la carpeta bin, en la cual está el keytool.exe



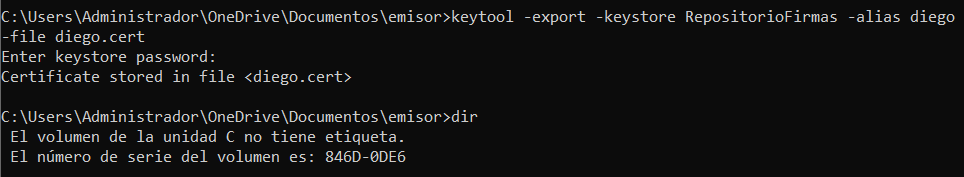
1. El código debería de funcionar, rellenamos todos los datos:



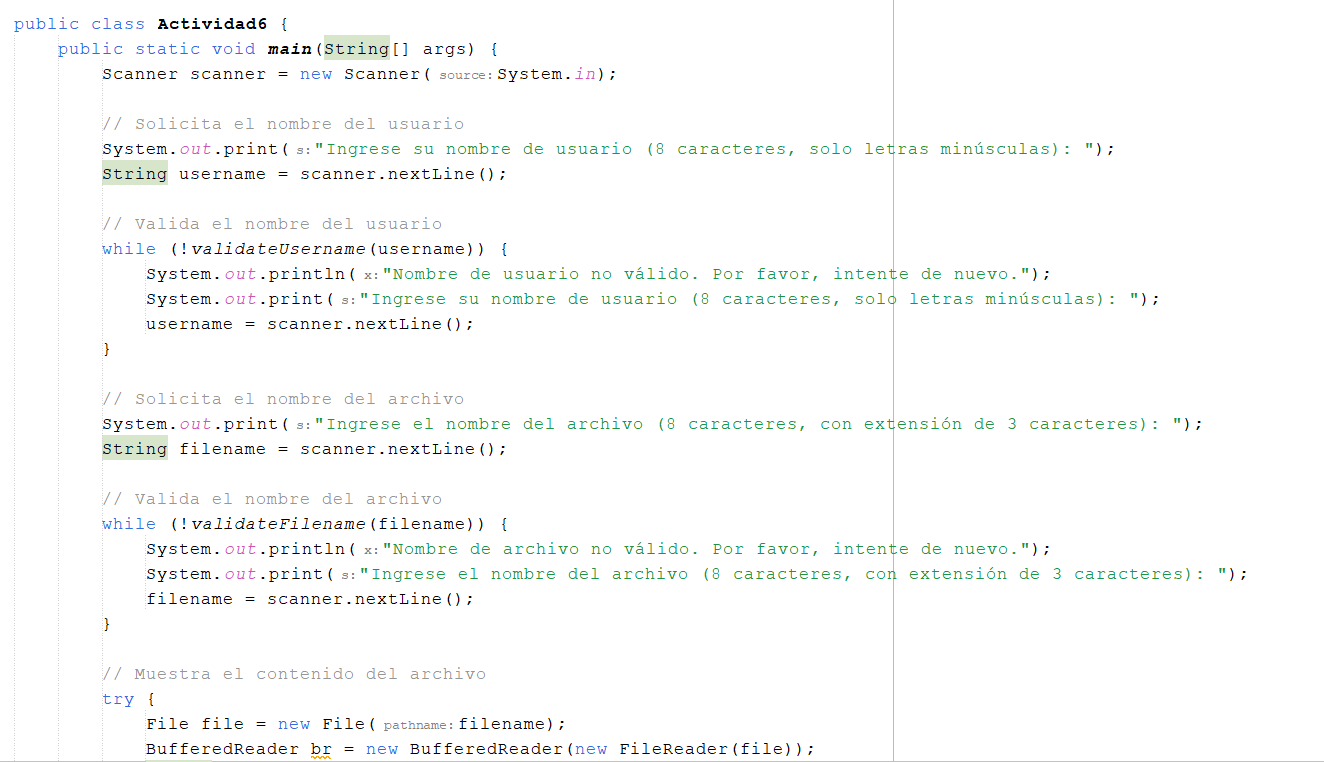
1. Firmamos el fichero donde se encuentra el proyecto, creando un documento.

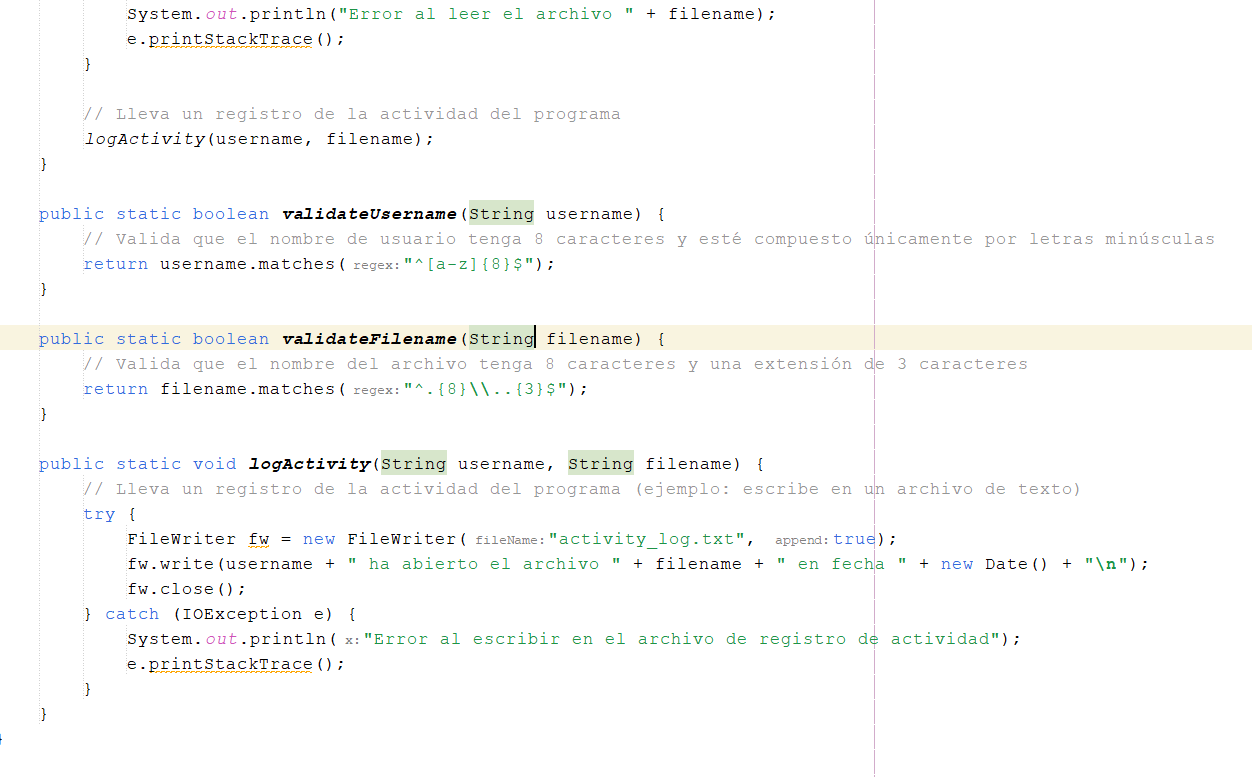


1. Si queremos mandarle el documento firmado a alguien, tenemos que pasar la clave pública, para validar la firma.



**Clase Actividad6B:**





Define la clase principal, "Actividad6B", que contiene el método principal.

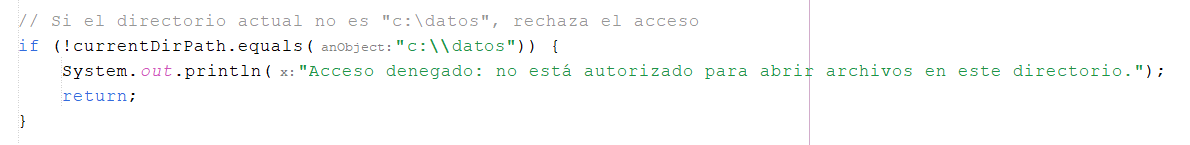
Crea un objeto Scanner para obtener la entrada del usuario.Pide al usuario que ingrese su nombre de usuario (8 caracteres, solo letras minúsculas) y asigna la entrada a una variable.

Valida el nombre de usuario ingresado usando un método separado, "validateUsername", que comprueba si tiene 8 caracteres y solo letras minúsculas. Si la validación falla, solicita al usuario que ingrese el nombre de usuario de nuevo. Pide al usuario que ingrese el nombre de un archivo (8 caracteres, con una extensión de 3 caracteres) y asigna la entrada a una variable.

Valida el nombre de archivo ingresado usando un método separado, "validateFilename", que comprueba si tiene 8 caracteres, un punto y 3 caracteres después del punto. Si la validación falla, solicita al usuario que ingrese el nombre de archivo de nuevo.

Intenta leer el archivo y imprimir su contenido línea por línea usando un objeto BufferedReader. Si ocurre un error, captura la excepción IOException e imprime un mensaje de error.

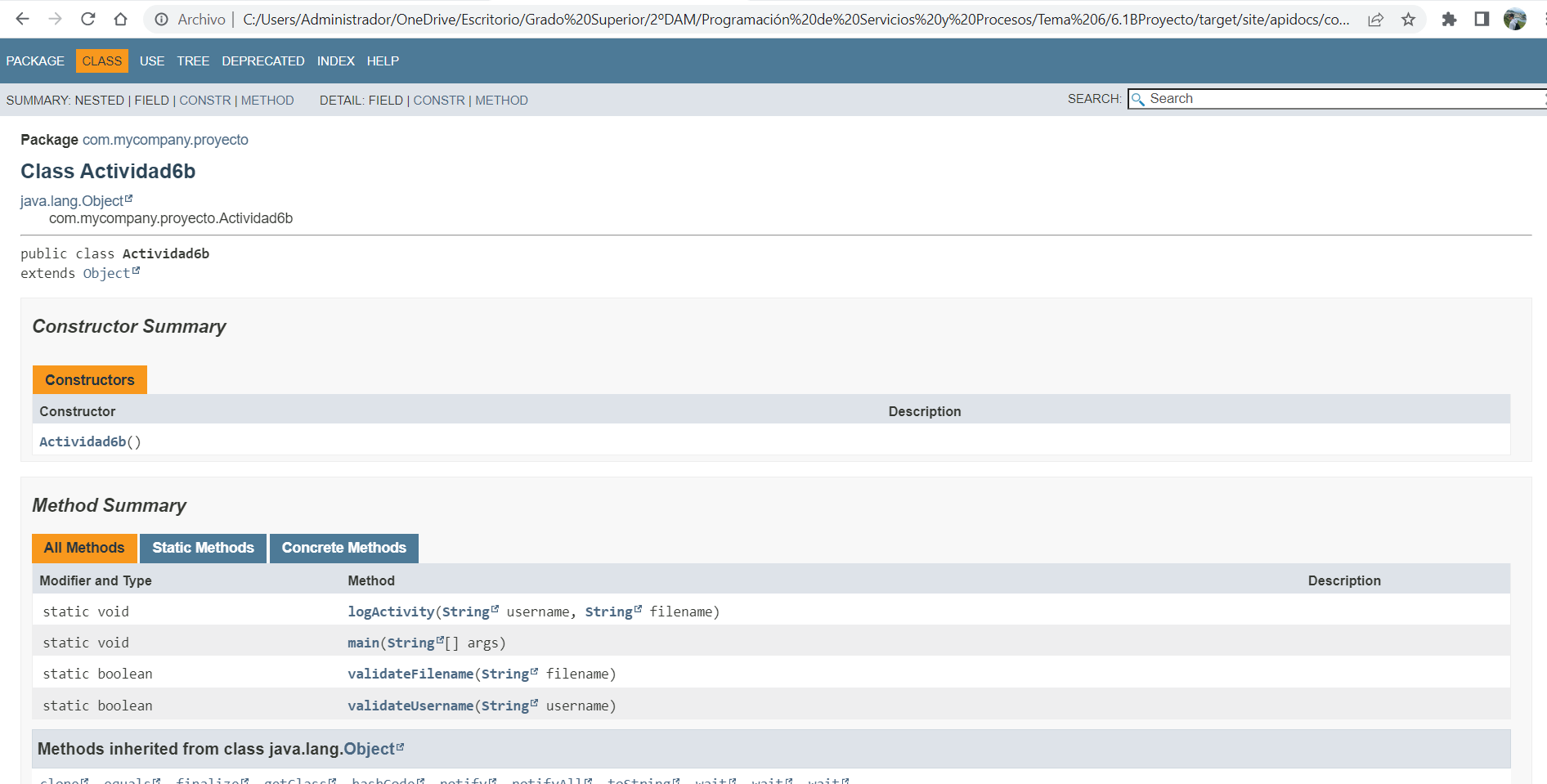
Llama a un método separado, "logActivity", para registrar la actividad del usuario (ejemplo: escribe en un archivo de texto) con el nombre de usuario y el nombre de archivo ingresados junto con la fecha actual. Si ocurre un error, captura la excepción IOException e imprime un mensaje de error.



Este bloque de código está verificando si la ruta del directorio actual no es igual a "c:\datos". Si no lo es, imprime un mensaje "Acceso denegado: no está autorizado para abrir archivos en este directorio." y finaliza la ejecución del programa con la instrucción "return".

**JAVADOC:**

**C:\Tema 6\6.1BProyecto\target\site\apidocs\**

****