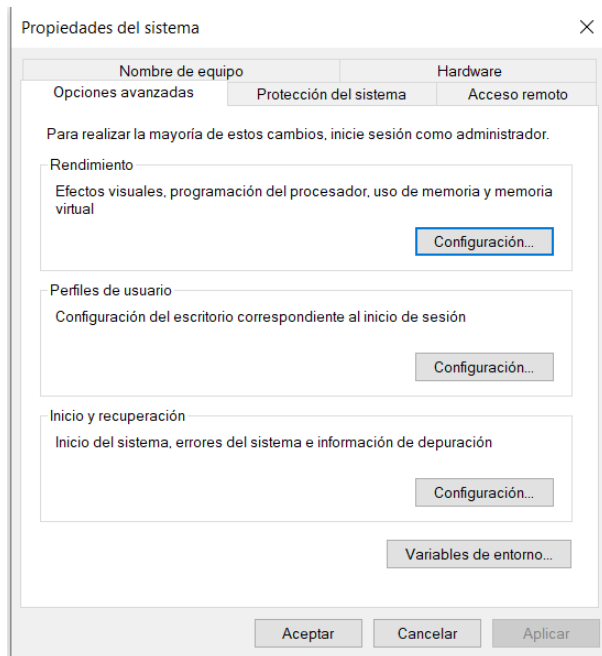


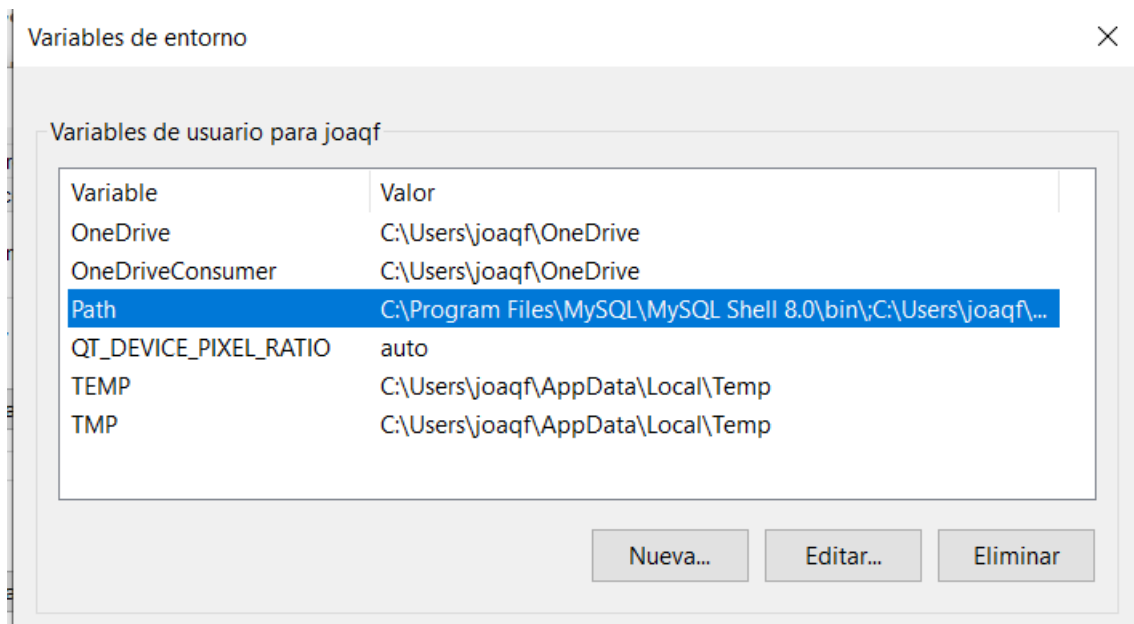
# FICHEROS DE POLICIES

## 1. Configurar variable PATH para Java

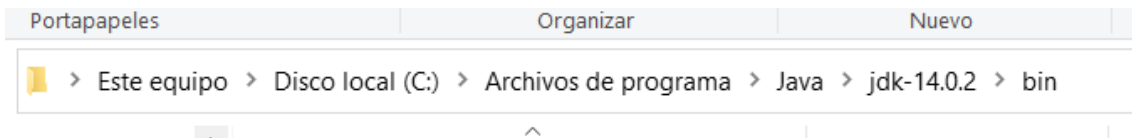
En configuración avanzada del sistema tenemos:



Hay que pinchar en variables de entorno....



Copiar la ruta hasta el directorio bin



### Editar la variable PATH y añadir la ruta

C:\Users\joaqf\AppData\Local\atom\bin
C:\Users\joaqf\AppData\Local\Programs\Microsoft VS Code\bin
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Program Files\Java\jdk-14.0.2\bin

Le damos a Aceptar, cerramos el cmd y volvemos a abrir....

Ahora si escribimos java o keytool desde cualquier directorio te tiene que detectar el comando.

```
Microsoft Windows [Versión 10.0.19041.746]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\joaqf>java
Usage: java [options] <mainclass> [args...]
        (to execute a class)
  or  java [options] -jar <jarfile> [args...]
        (to execute a jar file)
  or  java [options] -m <module>[/<mainclass>] [args...]
        java [options] --module <module>[/<mainclass>] [args...]
        (to execute the main class in a module)
  or  java [options] <sourcefile> [args]
        (to execute a single source-file program)

Arguments following the main class, source file, -jar <jarfile>,
-m or --module <module>/<mainclass> are passed as the arguments to
main class.

where options include:
```

## 2. EJECUCIÓN DEL EJEMPLO GETPROPS

Es **muy importante**, si compiláis con NetBeans el ejemplo de GetProps (ejemplo 0 en el apartado de recursos) os coloquéis en el directorio *clases* y hacer java nombrePaquete.nombreClase **sin .class**. Como se puede observar, se imprimen todas las características.

```
C:\Users\joaof\Desktop\Clases2020-2021\PSP\PSP2020-2021\PSP06_Tecnicas_Programacion_Segura\Ejemplos\Ejemplo0_props\build\classes>java ejemplo0_props.Ejemplo0_props
* java.class.path = .;E:\misProgramas;
* java.home = C:\Program Files\Java\jdk-14.0.2
* java.vendor = Oracle Corporation
* java.version = 14.0.2
* os.name = Windows 10
* os.version = 10.0
* user.dir = C:\Users\joaof\Desktop\Clases2020-2021\PSP\PSP2020-2021\PSP06_Tecnicas_Programacion_Segura\Ejemplos\Ejemplo0_props\build\classes
* user.home = C:\Users\joaof
* user.name = joaof
```

Hay que saber que si la clase está dentro de un paquete,

```
package ejemplo0_props;
```

Hay que ponerse en la ruta antes de la carpeta que tiene el nombre del paquete para hacer el comando **java**.

Si ejecutamos con la opción

- Con -Djava.security.manager (gestor de seguridad).

```
C:\Users\joaof\Desktop\Clases2020-2021\PSP\PSP2020-2021\PSP06_Tecnicas_Programacion_Segura\Ejemplos\Ejemplo0_props\build\classes>java -Djava.security.manager ejemplo0_props.Ejemplo0_props
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.class.path" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.home" "read")
* java.vendor = Oracle Corporation
* java.version = 14.0.2
* os.name = Windows 10
* os.version = 10.0
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.dir" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.home" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.name" "read")
C:\Users\joaof\Desktop\Clases2020-2021\PSP\PSP2020-2021\PSP06_Tecnicas_Programacion_Segura\Ejemplos\Ejemplo0_props\build\classes>
```

Esto se podrá hacer también con el jar generado en la carpeta dist pero con el comando java -jar:

```
C:\Users\joaof\Desktop\Clases2020-2021\PSP\PSP2020-2021\PSP06_Tecnicas_Programacion_Segura\Ejemplos\Ejemplo0_props\dist>java -jar Ejemplo0_props.jar
* java.class.path = Ejemplo0_props.jar
* java.home = C:\Program Files\Java\jdk-14.0.2
* java.vendor = Oracle Corporation
* java.version = 14.0.2
* os.name = Windows 10
* os.version = 10.0
* user.dir = C:\Users\joaof\Desktop\Clases2020-2021\PSP\PSP2020-2021\PSP06_Tecnicas_Programacion_Segura\Ejemplos\Ejemplo0_props\dist
* user.home = C:\Users\joaof
* user.name = joaof
```

```
C:\Users\joaql\Desktop\Clases2020-2021\PSP\PSP2020-2021\PSP06_Tecnicas_Programacion_Segura\Ejemplos\Ejemplo0_props\dist>java -jar -Djava.security.manager Ejemplo0_props.jar
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.class.path" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.home" "read")
* java.vendor = Oracle Corporation
* java.version = 14.0.2
* os.name = Windows 10
* os.version = 10.0
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.dir" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.home" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.name" "read")
```

Algunas características no tenemos permisos y no se mostrarán saltando esa excepción.

Existen muchas características:

Propiedad	Uso
file.separator	Separador de direcciones: / en Linux y \ en Windows
java.class.path	Ruta para encontrar directorios y ficheros JAR que contienen los archivos de clases.
java.home	Directorio para JRE
java.vendor	Nombre del proveedor
java.vendor.url	URL del proveedor
java.version	Versión de JRE
line.separator	Fin de línea
os.arch	Arquitectura del SO
os.name	Nombre del SO
os.version	Versión del SO
path.separator	Carácter separador usado en java.class.path (Linux = : y en win = ;)
user.dir	Directorio donde se está ejecutando el programa
user.home	Directorio del usuario (por defecto)
user.name	Nombre del usuario

Cuando se lanza el programa Java se cargan por defecto un fichero de políticas predeterminado que da todos los permisos para acceder a las propiedades comunes útiles (os.name o java.version). Se trata de propiedades no sensibles a la seguridad.

Sin embargo, user.home o java.home no suelen incluirse en la política de seguridad por defecto. Esto hace que si aplico la política, cuando el sistema trata de acceder eleva una excepción.

### 3. JAVA POLICY. Fichero de políticas en Java

El fichero java.policy se puede encontrar en las siguientes rutas:

- Windows: java.home\lib\security\java.policy  
java.home = C:\Program Files\Java\jdk1.8.0\_71\jre
- Linux: java.home/lib/security/java.policy

Lo más normal es tener un solo fichero java.policy en todo el Sistema y un fichero java.policy en el directorio del usuario user.home. Las localizaciones de estos ficheros podemos verlas en la carpeta java.home\lib\security:

- Policy.url.1 = file: C:\Program Files\Java\jdk-14.0.2\lib\security\default.policy
- Policy.url.2 = file :user.home/.java.policy

El administrador del Sistema puede modificar el fichero java.security y especificar distintas URL de políticas que están en otro servidor y no pueden ser editados por los usuarios. Una práctica habitual es tener que especificar un fichero de políticas para cada aplicación, la ejecución de la aplicación se haría así:

Java -Djava.security.policy=fichero.policy aplicación

Las políticas son una secuencia de entradas con esta forma

```
grant codeBase "URL" {  
    permission Nombre_clase "destino", "acción";  
};  
  
grant codeBase "file:/D:/Curso 21_22/DISTANCIA_PSP/PSP06/GetProps/dist/GetProps.jar" {  
    permission java.util.PropertyPermission "user.home", "read";  
};
```

En el ejemplo de la plataforma, he definido un archivo de policy llamado policy0.policy en la raíz del proyecto con este contenido:

```
grant {  
    permission java.util.PropertyPermission "user.home", "read";  
    permission java.util.PropertyPermission "user.name", "read";  
    permission java.util.PropertyPermission "user.dir", "read";  
};
```

Aquí estamos dando permiso para user.home, user.name y user.dir.

Si queremos que se aplique ese fichero habrá que hacer...

```
\Ejemplo0_props\dist>java -jar -Djava.security.manager -Djava.security.policy=ejemplo0.policy Ejemplo0_props.jar
```

Como se ve, estoy añadiendo también la opción de -

Djava.security.policy=nombreficheroPolíticas. En este caso se encuentra en el mismo directorio que el .jar. Si no, habría que poner la ruta completa hacia el fichero de políticas.

```
02/09/2020 18:11 <DIR> .  
02/09/2020 18:11 <DIR> ..  
03/02/2020 13:15      200 ejemplo0.policy  
03/02/2020 13:19     2.080 Ejemplo0_props.jar  
03/02/2020 13:19     1.330 README.TXT  
      3 archivos      3.610 bytes  
      2 dirs 379.774.836.736 bytes libres
```

Ahora donde dimos permisos, ya podemos ver la información.

```

Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.class.path" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.home" "read")
* java.vendor = Oracle Corporation
* java.version = 14.0.2
* os.name = Windows 10
* os.version = 10.0
* user.dir = C:\Users\joaqf\Desktop\Clases2020-2021\PSP\PSP2020-2021\PSP06_Tecnicas_Programacion_Segura\Ejemplos\Ejemplo0_props\dist
* user.home = C:\Users\joaqf
* user.name = joaqf

```

Otra forma de hacerlo es desde código. Así no tenemos que ejecutarlo desde línea de comandos. Del ejemplo descomentar estas instrucciones:

```

// System.setSecurityManager(new SecurityManager());
public static void main(String[] args) {
    System.setProperty("java.security.policy",
                       "./ejemplo0.policy");
    System.setSecurityManager(new SecurityManager());
}

```

La primera instrucción elige el fichero de políticas y la segunda lanza el securityManager. Es IMPORTANTE, que se haga en ese orden.

Si sólo descomento securityManager()

```

/* System.setProperty("java.security.policy",
                      "./ejemplo0.policy"); */
System.setSecurityManager(new SecurityManager());

```

Me salen las excepciones en Netbeans.

```

* os.name = Windows 10
* os.version = 10.0
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.class.path" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.home" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.dir" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.home" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "user.name" "read")
BUILD SUCCESSFUL (total time: 0 seconds)

```

Con fichero de políticas:

```
run:

* java.vendor = Oracle Corporation

* java.version = 14.0.2

* os.name = Windows 10

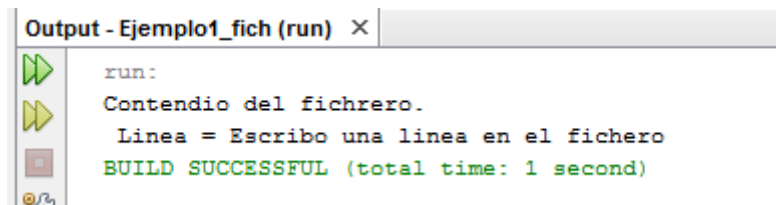
* os.version = 10.0

Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.class.path" "read")
Caught exception java.security.AccessControlException: access denied ("java.util.PropertyPermission" "java.home" "read")
* user.dir = C:\Users\joaqf\Desktop\Clases2020-2021\PSP\PSP2020-2021\PSP06_Tecnicas_Programacion_Segura\Ejemplos\Ejemplo0_props

* user.home = C:\Users\joaqf

* user.name = joaqf
BUILD SUCCESSFUL (total time: 0 seconds)
```

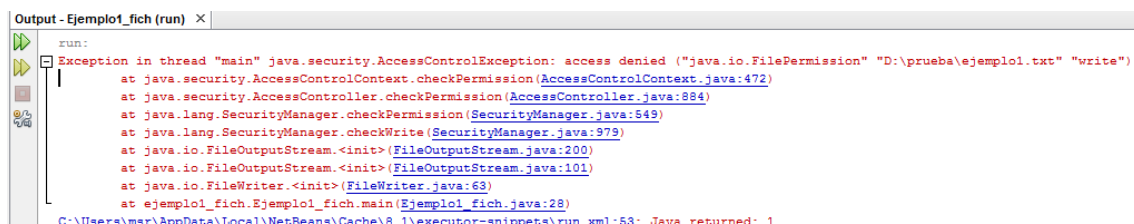
Ejemplo 1: Comparar comportamiento de programa con y sin política de seguridad. Ejemplo que escribe y lee una línea en un fichero en una localización específica.



Añadiendo el gestor de seguridad:

```
public static void main(String[] args) {
    String nombreFichero = "D:\\prueba\\ejemplo1.txt";
    //añadiendo el gestor de seguridad
    System.setSecurityManager(new SecurityManager());
    try {
```

La salida pasa a ser:



**Ejemplo 2:** Añadir política de seguridad para permitir escribir en el fichero. Crea el archivo con la política de seguridad *ejemplo2.policy*, que permita a cualquier programa que esté en *d:\prueba* escribir en el directorio *d:\prueba\datos*. La salida será algo así:

```
D:\prueba>java -jar -Djava.security.policy=ejemplo2.policy Ejemplo2.jar
Fin proceso escritura
Exception in thread "main" java.security.AccessControlException: access denied <
"java.io.FilePermission" "D:\prueba\ejemplo2.txt" "read">
    at java.security.AccessControlContext.checkPermission(Unknown Source)
    at java.security.AccessController.checkPermission(Unknown Source)
    at java.lang.SecurityManager.checkPermission(Unknown Source)
    at java.lang.SecurityManager.checkRead(Unknown Source)
    at java.io.FileInputStream.<init>(Unknown Source)
    at java.io.FileInputStream.<init>(Unknown Source)
    at java.io.FileReader.<init>(Unknown Source)
    at ejemplo2.Ejemplo2.LeerFichero(Ejemplo2.java:41)
    at ejemplo2.Ejemplo2.main(Ejemplo2.java:33)
```

Modifica el archivo de política de la aplicación para que deje también leer.

**Analizar el archivo ejemplo2.policy:**

```
grant codeBase "file:/D:/prueba/*"{
    permission java.io.FilePermission "D:\\prueba\\*", "read, write";
};
```

**Averigua:**

1- ¿Qué pasa si eliminamos el codeBase?

El permiso se aplicará a cualquier programa que ejecutemos independientemente de donde esté localizado.

2- La ruta destino "D:\\prueba\\\*", tiene caracteres "/" y "\*", ¿qué significado tienen?

3- Si para dar permisos de lectura y escritura de ficheros en el directorio en el que se está ejecutando el programa puedo poner:

```
grant{
    permission java.io.FilePermission "${user.dir}/*", "read, write";
};
```

¿Qué tendría que poner para dar el permiso en el directorio personal del usuario?

## Trabajando con Sockets.

La clase `java.net.SocketPermission` permite establecer política de seguridad cuando trabajamos con Sockets y puertos.

Entonces, en el fichero de políticas, el "nombre\_Destino" tendrá que tener un nombre de host y un puerto. El conjunto de acciones que especifican la forma de conexión al host serán:

- accept
- listen
- connect
- resolve

Host = (host) [:puerto]



Numero\_puerto = numpuerto | - numpuerto | numpuerto-[numero de puerto]

El nombre del host puede ser un nombre o una IP seguido por el número del puerto.

El número de puerto es opcional.

La forma de especificar un rango para el puerto es:

- la forma "NP – " → rango de puertos desde el puerto NP en adelante.
- La forma "- NP" → rango de puertos desde el puerto NP hacia atrás.

**Ejemplo 3:** Programa servidor que acepta peticiones de clientes en el puerto 6600. El cliente cuando se conecta ve un mensaje de bienvenida y cuando se va uno de despedida.

El cliente aplicará la política de seguridad: `System.setSecurityManager(new SecurityManager());`

Observamos que la salida no da permiso:

```
java.security.AccessControlException: access denied ("java.net.SocketPermission"
"127.0.0.1:6600" "connect,resolve")
```

Ficheros de políticas de acceso:

Servidor: `servidor3.policy` → permiso que permita a los programas de la carpeta `d:\prueba\ejemplo3` aceptar conexiones en localhost

```
grant codeBase "file:/D:/prueba/ejemplo3/" {
    permission java.net.SocketPermission "localhost", "accept";
};
```

Cliente: `cliente3.policy` → permiso que permita a los programas de la carpeta `d:\prueba\ejemplo3` conectarse al puerto 6600 de localhost.

```
grant codeBase "file:/D:/prueba/ejemplo3/" {
    permission java.net.SocketPermission "localhost:6600", "connect";
};
```

Al ejecutar ambos, servidor y cliente, aplicando sus respectivas políticas de seguridad se obtiene un excepción `Access denied`.

Podemos verificar la sintaxis de nuestros ficheros de configuración de las políticas de seguridad usando la herramienta "policyTool" de java.