Ejercicios

1. Crea una aplicación que valide los siguientes datos de entrada:
    - Una matrícula con formato cuatro dígitos y tres letras, separado por un guión y las letras en mayúsculas.
    - Verificar el DNI, ocho dígitos y una letra, esta vez la letra puede estar tanto en mayúsculas como en minúsculas.

    Probar de varias formas la validación de los datos.

2. Crear un proyecto en el que se controlen todos las acciones que se están llevando a cabo a lo largo de la ejecución de la aplicación.

    La aplicación en si, pide al usuario que introduzca un nombre de usuario valido, se pide que tenga formato email, una vez asegurada esta parte se le pide que introduzca el nombre del fichero que quiere visualizar, el nombre del fichero es como máximo de 8 caracteres y tiene una extensión de 3 caracteres. Se visualizara el fichero en si.

    Es importante tener en cuenta que se tiene que realizar una validación de los datos de entrada y llevar un registro de la actividad del programa.

3. Crea un programa en Java que simule un sistema de gestión de contraseñas seguro. El programa debe permitir a un usuario registrarse con una contraseña, almacenar esa contraseña de forma segura (utilizando una función hash(por ejemplo, SHA-256) y permitir el inicio de sesión verificando la contraseña mediante la función hash.

    Si el usuario intenta iniciar sesión con un nombre de usuario o contraseña incorrectos, el sistema debe mostrar un mensaje de error adecuado.

```
Sistema de gestión de contraseñas
1. Registrarse
2. Iniciar sesión
3. Salir
Seleccionar una opción: 1
Nombre de usuario: Eider
Contraseña: 12345
La cuenta de Eider ha sido creada.
```

4. Implementar un sistema cliente/servidor en el que el cliente le envíe un mensaje al servidor, utilizando un modelo de clave privada/simétrica.
5. Implementar un sistema cliente/servidor en el que el servidor le envíe una información al cliente, utilizando un modelo de clave pública/asimétrica.
6. Implementa una aplicación distribuida formada por un cliente y un servidor que se comuniquen mediante sockets TCP. El cliente deberá generar un par de claves RSA y usar su clave privada para firmar cualquier mensaje que envíe al servidor. La clave pública deberá estar disponible para que el servidor pueda verificar la firma.

   En cada envío, el cliente mandará al servidor el mensaje original y su firma digital (usando, por ejemplo, *SHA256withRSA*). El servidor recibirá ambos elementos, verificará la firma y mostrará si el mensaje es auténtico o ha sido modificado.

   En el servidor:

```
Servidor en espera...
Cliente conectado.
Mensaje recibido: prueba de firma
Firma válida: true
```

   En el cliente:

```
Introduce el mensaje a enviar: prueba de firma
Clave pública, mensaje y firma enviados.
```

7. Crea una aplicación cliente- servidor utilizando sockets seguros(SSL). Se transmitirán mensajes cifrados utilizando el cifrado simétrico. Recuerda la importancia de disponer de un certificado de servidor válido para que esta comunicación se pueda establecer.