

I.E.S. Pablo Picasso

Programación 1DAM - Curso 2023/2024

Actividad 6.11 – Prueba Práctica Bloque 6

Se nos pide que implementemos para una aplicación un sistema de autenticación basado en una tarjeta de claves. Una tarjeta de claves es una tarjeta física que tiene la persona a autenticar y que contiene una tabla cuyas casillas contienen números de 3 cifras. El sistema proporciona unas coordenadas (fila y columna) al usuario y éste tiene que responder con la clave que aparece en la tarjeta en dicha coordenada. Si la clave coincide con lo que tiene almacenado el sistema (que conoce las tarjetas de coordenadas de todos los usuarios), se permite el paso al usuario.

Cada usuario tiene un nombre de usuario único compuesto por 2 a 8 caracteres alfanuméricos, debiendo ser el primero una letra. Al ser único no pueden haber dos usuarios con el mismo nombre de usuario.

La información de todas las tarjetas de claves están almacenadas en un archivo XML (de éste hablaremos más adelante).

El sistema deberá cargar la información de las tarjetas y proporcionar un método para la autenticación de un usuario.

Una vez autenticado y dentro de la aplicación el usuario podrá acceder a sus clientes. Se podrán listar los clientes del usuario, así como buscarlos por nombre, apellidos, dni o edad. Los clientes de cada usuario se incluyen en el mismo archivo XML.

Ya se ha realizado un diseño de las clases a utilizar que se proporciona en el archivo `Actividad_6.11_doc.zip`. También se proporcionan pruebas en el archivo `Actividad_6.11_pruebas.zip`

Por último se proporciona un pequeño programa de prueba (incompleto, ver más abajo) (`prog.ud06.actividad611.main.ColeccionApp`) con un método `main()` que hace la interfaz con el usuario. Asimismo sirve para probar que las clases realizadas hacen su tarea como se espera.

La aplicación (ColeccionApp)

La clase `ColeccionApp` utilizará para obtener los datos necesarios para funcionar (usuarios, clientes, tarjetas) un archivo XML llamado `biblioteca.xml` que debe estar localizado en la carpeta raíz del proyecto y cuya estructura se detalla más abajo

La clase `ColeccionApp` no está completa ya que tiene métodos sin implementar. Implementa los métodos que restan en `ColeccionApp` y añade más si lo crees necesario.

El funcionamiento de la aplicación, a grandes rasgos, es el siguiente:

- Autentica al usuario solicitando un nombre de usuario y, si el usuario existe, el contenido de una casilla de la tarjeta de claves del usuario. Las coordenadas de esta casilla se elegirán al azar teniendo en cuenta las dimensiones de la tabla de claves de la tarjeta del usuario.
- Si el usuario NO autentica se mostrará un mensaje y se saldrá inmediatamente del programa
- Si el usuario autentica correctamente (la clave proporcionada coincide con la almacenada), se continúa con lo siguiente, teniendo en cuenta que de aquí en adelante cuando hablamos de usuario estamos refiriéndonos al usuario que se ha autenticado.
- Muestra un menú repetidamente con las opciones:
 - Listar clientes: Lista todos los clientes del usuario, ordenados por apellido.
 - Buscar clientes por nombre: Solicita al usuario dos datos: Primero se solicita como buscar. Las dos opciones son nombre completo (representado por la letra c) o parte del nombre (p). La primera opción busca clientes cuyo nombre coincida completamente con el texto de búsqueda. La segunda busca clientes cuyo nombre coincida en parte con el texto de búsqueda. Después se solicita el texto a buscar, que no puede ser vacío o sólo blancos. A continuación se hace la búsqueda. Si se obtienen resultados, se listan como en Listar clientes. Si no se muestra un mensaje al estilo "No hay clientes".
 - Buscar clientes por apellidos. Igual que el anterior pero busca el texto en los apellidos de los clientes, en lugar de en los nombres
 - Buscar clientes por DNI. Se solicita el DNI y se muestra el cliente con dicho DNI o el mensaje "Cliente no encontrado"
 - Buscar clientes por edad. Solicita al usuario dos edades. El primero será una edad mínima y el segundo una edad máxima. El usuario debe poder proporcionar una u otra, ambas o ninguna admitiendo valores vacíos. Si el usuario introduce ambas, la primera debe ser menor o igual que la segunda y se localizan los clientes cuya edad esté comprendida entre la primera y la última, ambas inclusive (si sólo se quiere buscar por una edad en concreto se puede hacer que la mínima y la máxima sean iguales). Si sólo se especifica la mínima se localizarán todos los clientes cuya edad sea mayor o igual a dicha edad mínima. Si sólo se especifica la máxima se localizarán todos los clientes cuya edad sea menor o igual a dicha edad mínima. Si no se especifica ninguna se localizarán todos los clientes.
 - Salir de programa. Se termina el programa

Ejemplo de uso:

```
Autenticación
Introduzca el nombre de usuario: paco
Introduzca la clave en las coordenadas (3, 1): 161
Autenticación correcta.
```

MENU PRINCIPAL

```
-----
1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa
Elija una opción (0-5): 1
```

LISTAR CLIENTES

Los clientes del usuario son:

Contreras Gomez, Juan Carlos. DNI: 11294752G. Edad: 36

Herrera Iglesias, Maria del Carmen. DNI: 48129735G. Edad: 65

Jimenez Lopez, Diego. DNI: 76211879E. Edad: 21

MENU PRINCIPAL

1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa

Elija una opción (0-5): 2

BUSCAR CLIENTE POR NOMBRE DE PILA

¿Buscar por nombre completo o parte del nombre? (c=nombre completo, p=parte del nombre): j

La opción elegida no es válida. Debe ser una de c, p

MENU PRINCIPAL

1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa

Elija una opción (0-5): 2

BUSCAR CLIENTE POR NOMBRE DE PILA

¿Buscar por nombre completo o parte del nombre? (c=nombre completo, p=parte del nombre): c

Introduzca el texto a buscar en el nombre del cliente: Maria del Carmen

Los clientes del usuario que contienen "Maria del Carmen" en el nombre de pila son:

Herrera Iglesias, Maria del Carmen. DNI: 48129735G. Edad: 65

MENU PRINCIPAL

1. Listar clientes
2. Buscar clientes por nombre de pila

```
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa
Elija una opción (0-5): 2
```

BUSCAR CLIENTE POR NOMBRE DE PILA

```
-----
¿Buscar por nombre completo o parte del nombre? (c=nombre
completo, p=parte del nombre): p
Introduzca el texto a buscar en el nombre del cliente: Carmen
Los clientes del usuario que contienen "Carmen" en el nombre de
pila son:
Herrera Iglesias, Maria del Carmen. DNI: 48129735G. Edad: 65
```

MENU PRINCIPAL

```
-----
1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa
```

Elija una opción (0-5): 3

BUSCAR CLIENTE POR APELLIDOS

¿Buscar por apellidos completos o parte de los apellidos?

(c=apellidos completos, p=parte de los apellidos): j

La opción elegida no es válida. Debe ser una de c, p

MENU PRINCIPAL

1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa

Elija una opción (0-5): 3

BUSCAR CLIENTE POR APELLIDOS

¿Buscar por apellidos completos o parte de los apellidos?

(c=apellidos completos, p=parte de los apellidos): c

Introduzca el texto a buscar en los apellidos de los clientes del usuario: Herrera Iglesias

Los clientes del usuario que contienen "Herrera Iglesias" en los apellidos son:

Herrera Iglesias, Maria del Carmen. DNI: 48129735G. Edad: 65

MENU PRINCIPAL

1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa

Elija una opción (0-5): 3

BUSCAR CLIENTE POR APELLIDOS

¿Buscar por apellidos completos o parte de los apellidos?

(c=apellidos completos, p=parte de los apellidos): p

Introduzca el texto a buscar en los apellidos de los clientes del usuario: Iglesias

Los clientes del usuario que contienen "Iglesias" en los apellidos son:

Herrera Iglesias, Maria del Carmen. DNI: 48129735G. Edad: 65

MENU PRINCIPAL

1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa

Elija una opción (0-5): 4

BUSCAR CLIENTE POR DNI

Introduzca el DNI a buscar (DNI completo): 11111111A

No se encontró cliente con el DNI especificado

MENU PRINCIPAL

1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad

0. Salir del programa
Elija una opción (0-5): 4

BUSCAR CLIENTE POR DNI

Introduzca el DNI a buscar (DNI completo): 48129735G

El cliente con el DNI "48129735G" es:

Herrera Iglesias, Maria del Carmen. DNI: 48129735G. Edad: 65

MENU PRINCIPAL

1. Listar clientes
 2. Buscar clientes por nombre de pila
 3. Buscar clientes por apellidos
 4. Buscar clientes por DNI
 5. Buscar clientes por edad
 0. Salir del programa
- Elija una opción (0-5): 5

BUSCAR CLIENTE POR AÑO

¿Buscar clientes cuya edad sera mayor o igual a? (vacío para cualquiera):

¿Buscar clientes cuya edad sea menor o igual a? (vacío para cualquiera):

Los clientes encontrados son:

Contreras Gomez, Juan Carlos. DNI: 11294752G. Edad: 36

Herrera Iglesias, Maria del Carmen. DNI: 48129735G. Edad: 65

Jimenez Lopez, Diego. DNI: 76211879E. Edad: 21

MENU PRINCIPAL

1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa

Elija una opción (0-5): 5

BUSCAR CLIENTE POR AÑO

¿Buscar clientes cuya edad sera mayor o igual a? (vacío para cualquiera): 30

¿Buscar clientes cuya edad sea menor o igual a? (vacío para cualquiera):

Los clientes encontrados son:

Contreras Gomez, Juan Carlos. DNI: 11294752G. Edad: 36

Herrera Iglesias, Maria del Carmen. DNI: 48129735G. Edad: 65

MENU PRINCIPAL

1. Listar clientes
2. Buscar clientes por nombre de pila
3. Buscar clientes por apellidos
4. Buscar clientes por DNI
5. Buscar clientes por edad
0. Salir del programa

Elija una opción (0-5): 5

BUSCAR CLIENTE POR AÑO

¿Buscar clientes cuya edad sera mayor o igual a? (vacío para cualquiera):

¿Buscar clientes cuya edad sea menor o igual a? (vacío para cualquiera): 60

Los clientes encontrados son:

Contreras Gomez, Juan Carlos. DNI: 11294752G. Edad: 36

Jimenez Lopez, Diego. DNI: 76211879E. Edad: 21

MENU PRINCIPAL

1. Listar clientes
 2. Buscar clientes por nombre de pila
 3. Buscar clientes por apellidos
 4. Buscar clientes por DNI
 5. Buscar clientes por edad
 0. Salir del programa
- Elija una opción (0-5): 5

BUSCAR CLIENTE POR AÑO

¿Buscar clientes cuya edad sera mayor o igual a? (vacío para cualquiera): 30

¿Buscar clientes cuya edad sea menor o igual a? (vacío para cualquiera): 39

Los clientes encontrados son:

Contreras Gomez, Juan Carlos. DNI: 11294752G. Edad: 36

MENU PRINCIPAL

1. Listar clientes
 2. Buscar clientes por nombre de pila
 3. Buscar clientes por apellidos
 4. Buscar clientes por DNI
 5. Buscar clientes por edad
 0. Salir del programa
- Elija una opción (0-5): 0

Ejemplo de uso en caso de autenticación incorrecta:

Autenticación

Introduzca el nombre de usuario: pacos

Autenticación incorrecta

Error de autenticación. Terminando programa

Otro ejemplo de uso en caso de autenticación incorrecta:

Autenticación

Introduzca el nombre de usuario: paco

Introduzca la clave en las coordenadas (3, 1): 53

Autenticación incorrecta

Error de autenticación. Terminando programa

Archivo XML

Se proporciona el archivo xml con la información. Se llama `biblioteca.xml` y tiene la siguiente estructura:

- El elemento raíz se llama `biblioteca`.
- Dentro de `biblioteca` hay tres elementos: `usuarios`, `tarjetas` y `clientes`
- `usuarios` contiene uno o mas elementos `usuario`. Cada elemento `usuario` tiene dos atributos obligatorios: `nombre`, que es el nombre y apellidos del usuario e `id`, que contiene el nombre de usuario, que debe ser único. El elemento no tiene ningún contenido.

- `tarjetas` contiene uno o mas elementos `tarjeta`. Cada tarjeta contiene elementos `fila`, cada elemento `fila` contiene varios elementos `celda`. Cada elemento `celda` contiene el valor de la casilla correspondiente. Todos los elementos `fila` de una tarjeta determinada deben tener el mismo número de elementos `celda`. Cada tarjeta puede tener un número distinto de filas.
- El elemento `clientes` contiene cero o más elementos `cliente`.
- Cada elemento `cliente` contiene los elementos `nombre`, `apellidos`, `dni` y `edad`. `nombre` es el nombre de pila del cliente, `apellidos` son los apellidos del cliente, `dni` es el DNI del cliente y `edad` es la edad del cliente. El nombre no puede estar vacío ni consistir sólo de espacios, al igual que los apellidos. El dni debe constar obligatoriamente de 8 números seguidos de una letra mayúscula. Esta letra está relacionada con el número según un algoritmo que deberás localizar en Internet. La edad debe ser un número entero igual o superior a cero.

Se proporciona:

- Archivo `biblioteca.xml` con la información de los usuarios , sus tarjetas de claves y sus usuarios.
- Archivo con la documentación de las clases de la librería `Actividad_6.11_doc.zip`
- Batería de casos de prueba JUnit para algunas clases en el archivo `Actividad_6.11_pruebas.zip`. No es obligatorio utilizarlas pero pueden ser útiles para localizar errores en la librería.
- Clase `ColeccionApp` con punto de entrada `main()` para probar el funcionamiento real del sistema.

Se deberá entregar:

- La carpeta COMPLETA del proyecto comprimida en el formato ZIP

Criterios de evaluación y calificación:

Se proporcionará una calificación por cada uno de los siguientes criterios de evaluación:

- Criterio 6.a) Se han escrito programas que utilicen arrays. Se valorará el uso de arrays bidimensionales para almacenar las claves de cada tarjeta.
- Criterio 6.c) Se han utilizado listas para almacenar y procesar información. Se valorará el uso de listas.
- Criterio 6.d) Se han empleado iteradores para recorrer las listas. Se valorará el uso de iteradores para recorrer la colección de clientes de cada usuario para realizar las búsquedas
- Criterio 6.f) Se han creado clases y métodos genéricos. Se valorará la creación y uso de genéricos (clases o métodos o ambos), especialmente la clase `Diccionario`.
- Criterio 6.g) Se han utilizado expresiones regulares en las búsquedas de patrones en cadenas de texto. Se valorará el uso de expresiones regulares para validar datos leídos desde el archivo XML y para cualquier otro uso que creas conveniente.
- Criterio 6.h) Se han identificado las clases relacionadas con el tratamiento de documentos XML. Se valorará el uso correcto de las clases relacionadas con el tratamiento de documentos XML

- Criterio 6.i) Se han realizado programas que realicen manipulaciones sobre documentos XML. Se valorará la implementación correcta de la extracción de datos de ficheros XML.
- Criterio 3.g) Se ha comentado y documentado el código. Se valorará la existencia de comentarios en clases, métodos, atributos y bloques, así como la adecuación de los mismos, es decir, que los comentarios sean adecuados a lo que se está comentando.
- Criterio 3.h) Se han seguido unas reglas de estilo de código. Se valorará el seguimiento más o menos estricto de las reglas de estilo de código

En caso de que la aplicación no compile o termine de forma prematura debido a errores de programación la puntuación de **TODOS** los criterios anteriores será de 0 puntos.

La implementación de la clase **TarjetaClaves** debe emplear obligatoriamente arrays para almacenar y recuperar información. Si no se cumple esta obligación la puntuación en el criterio 6.a será de 0 puntos.

La implementación de la clase **Usuario** debe emplear obligatoriamente listas para almacenar los clientes del usuario. Si no se sigue esta obligación la puntuación en el criterio 6.c será de 0 puntos.

La implementación de la clase **Usuarios** debe utilizar obligatoriamente uno o más objetos de la clase **Diccionario** (la cantidad de objetos debes decidirla tu). El alumno que no siga este requisito obtendrá 0 puntos en el criterio 6.f

Se deberán entregar un total de 4 entregas parciales y una entrega final. Si no se realiza la entrega final la práctica se considera como no realizada. Si no se realiza alguna de las entregas parciales se rebajarán las calificaciones un 10% por cada entrega no realizada en caso de no estar debidamente justificada.