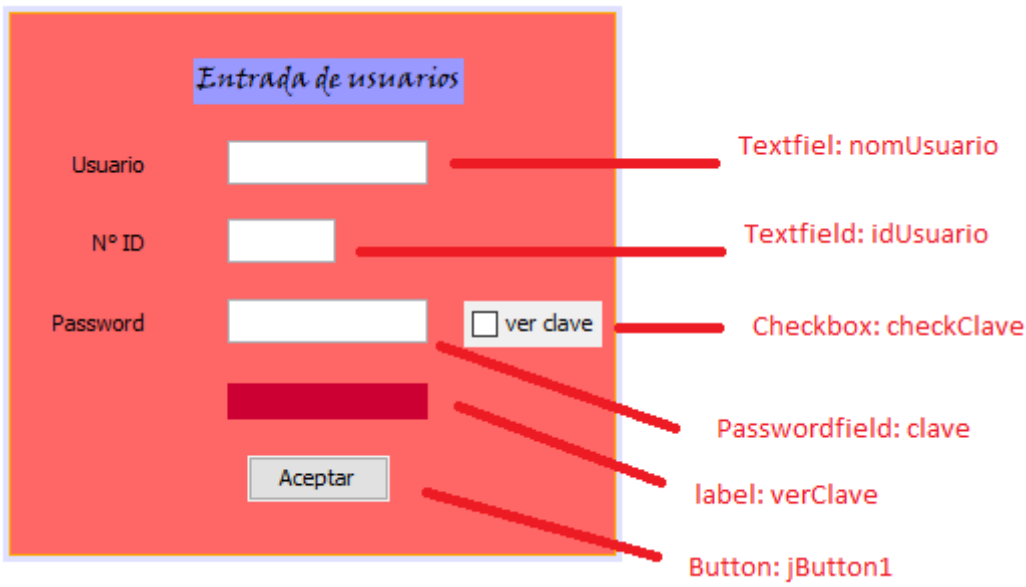
Eventos y métodos

Eventos son las acciones que ejecutan los usuarios dentro del programa y disparan el código al momento de realizar dicho evento. Aquí entran por ejemplo el clic del mouse, apretar un botón, escribir, mover el mouse. Etc.

Para practicar este concepto vamos a realizar una pequeña pantalla de inicio de sesión que se puede utilizar en cualquier programa X.

Una vez realizada la carga de datos el programa nos devolverá un mensaje donde conste si los datos cargados son correctos o no. Se le añade a la ventana la opción de revelar el password al momento de ingresarla.

Frame:



Notas:

- El password field es similar al text field con la única diferencia que al momento de ingresar caracteres estos aparecerán con el símbolo ****
- Para que un label tenga color de fondo hay que tildar la opción opaque en opciones (properties)
- Recuerde que para elegir color de fondo se realiza en la opción background y para elegir fuente y tamaño se encuentra la opción Font.
- Recuerde colocar todos los elementos en un panel

Código:

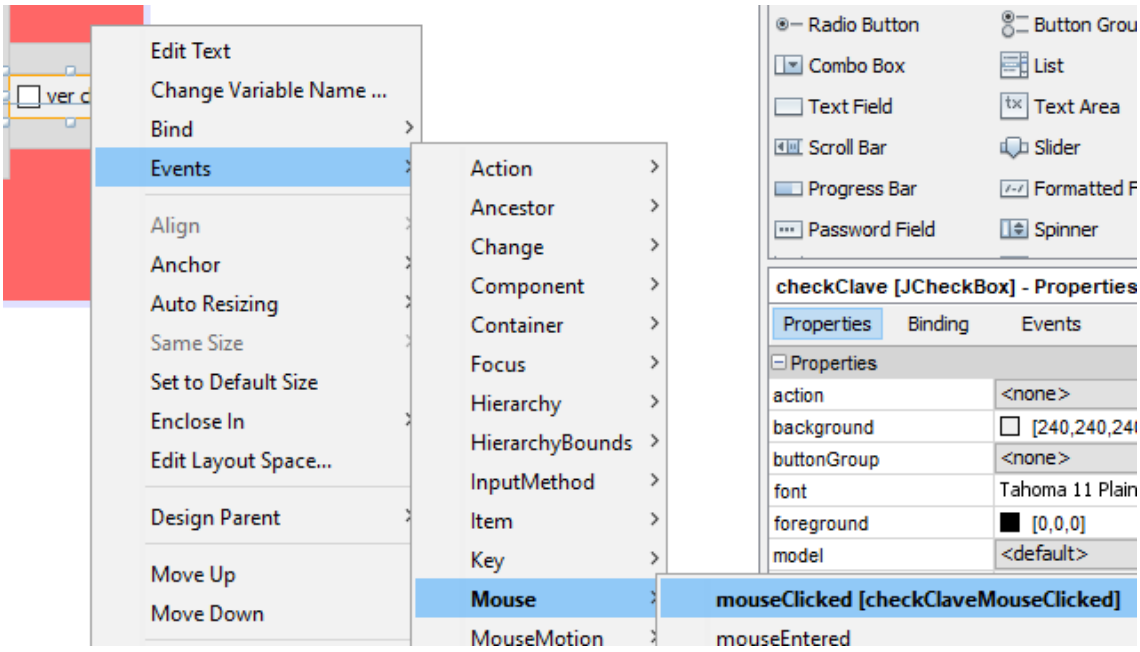
Lo primero que haremos es buscar el constructor del frame para añadirle una línea de código.

Recuerde que el constructor se encuentra debajo de la declaración de la clase. En este caso se llama ingreseClave porque así se llama el proyecto de ejemplo pero recuerde que tomará el nombre que elijan al momento de crear su frame.

El código que añadiremos es verClave.setVisible(false); para que al momento de ejecutar el programa el label que revela la clave no sea visible para el usuario.

```
public class ingreseClave extends javax.swing.JFrame {
    public ingreseClave() {
        initComponents(); //código por defecto
        verClave.setVisible(false);//código que añadimos
    }
```

Ahora vamos utilizar el evento mouse clicked en el **check box** para que se dispare al hacer clic con el mouse su código. Para programarlo debemos de hacer clic derecho en el **checkbox/events/mouse/mouseclicked**



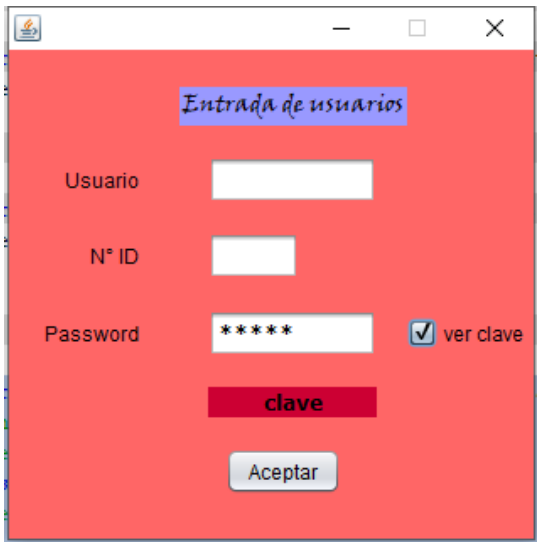
Aquí encontraremos todos los posibles eventos que puede disparar el mouse.

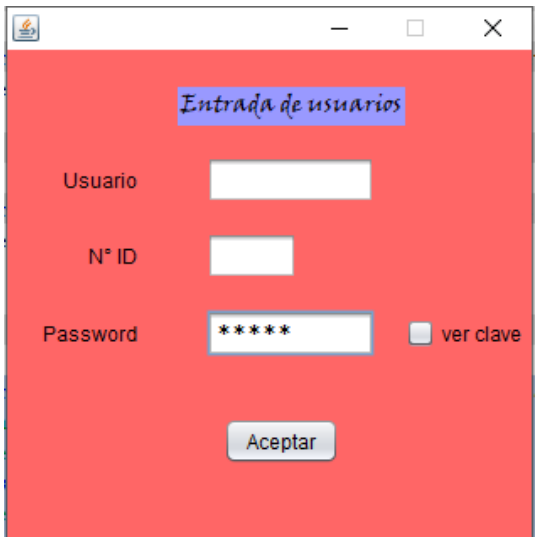
Una vez ingresemos a la pantalla del código redactaremos las siguientes líneas:

```
private void checkClaveMouseClicked(java.awt.event.MouseEvent evt) {
if (checkClave.isSelected()){ //comprueba si el checkbox está tildado o no
```

```
    verClave.setVisible(true);
} else{
    verClave.setVisible(false);
}
    }
```

Con el código hasta ahora lo que logramos el ocultar y visibilizar el label de la clave





El siguiente paso será limitar la cantidad de caracteres que el usuario puede ingresar. Esto es necesario para que los usuarios tengan un número limitado de caracteres y establecer un rango o criterio de caracteres. Para nuestro programa tanto usuario como password tendrán un límite de 8 caracteres mientras que N°ID será de 4.

Para ahorrar código y en vista que el proceso es prácticamente el mismo para los tres casos crearemos un **método; recordemos que un método es una función que se ejecutará cada vez que sea llamada. Hasta ahora hemos utilizado métodos por defecto de Java pero vale recordar que ya hemos realizado algunos ejemplos en las primeras actividades de este curso.**

La estructura de un método es :

[public/private] [tipo de método: void/int/String/etc] [nombre del método] [(parámetros)] {código y **return si no fuera void**}

Nuestro método ya terminado es el siguiente, el mismo debe de insertarse debajo del constructor del frame donde se trabajó en el punto anterior.

```
public void limiteCharter(java.awt.event.KeyEvent evt,int cajaText,int limite){ //es un método void ya que no devuelve ningún valor. limiteCharter es el nombre que le asignamos, el nombre es a gusto del programador. Este método requiere 3 parámetros; un evento y dos int.
    if (cajaText==limite){
        evt.consume();//finaliza el evento
        Toolkit.getDefaultToolkit().beep();//sonido del sistema
    }
   }
```

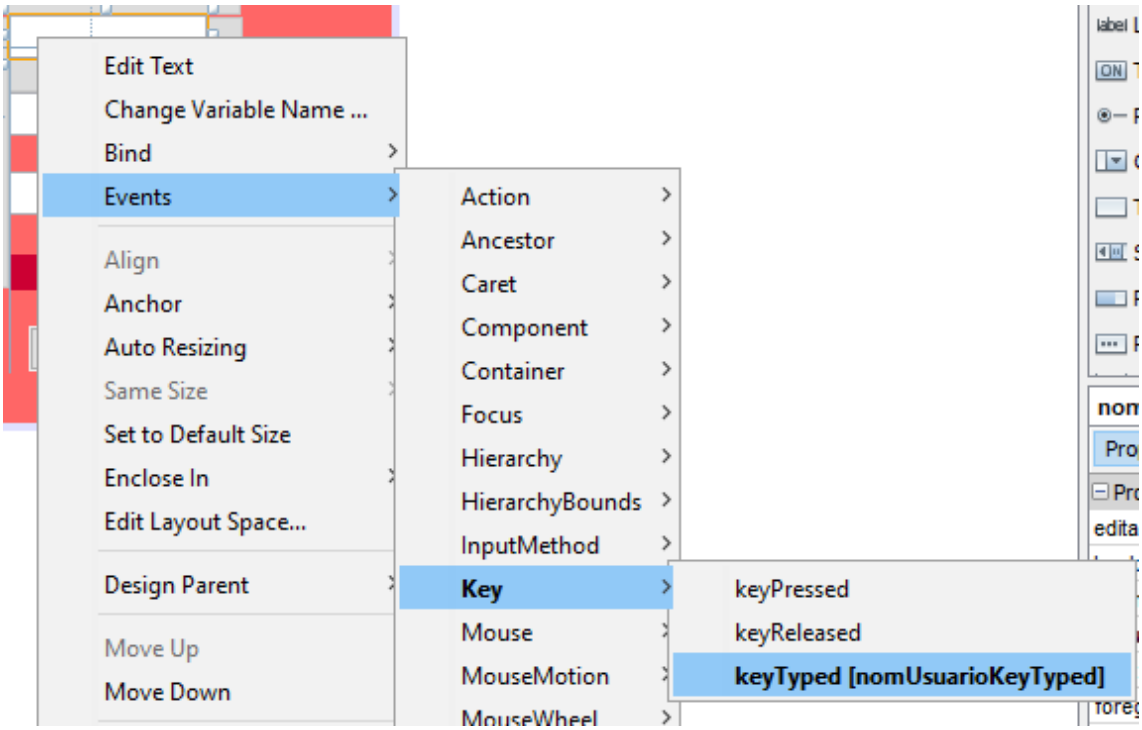
Veamos los parámetros:

- java.awt.event.KeyEvent evt,int: evt es el evento el cual ingresa cada tecla al pulsarla, con él controlamos que está pulsando el usuario.
- int cajaText: es una variable numérica que indicará cuantos caracteres lleva ingresando el usuario.
- int limite: es el límite de caracteres que puede ingresar el usuario

Explicación: if (cajaText==limite){ es un if que compara la cantidad de caracteres que el usuario a tecleado con el límite establecido, si alcanza dicho límite evt.consume(); termina con la acción impidiendo teclear un nuevo carácter y Toolkit.getDefaultToolkit().beep(); simplemente emite un pitido para alertar al usuario que alcanzó el límite de caracteres.

Ahora probaremos nuestro método

Para ello utilizaremos el **eveto keytiped** que hace llamado a la tecla que se presiona en el teclado el cual nos brindará la posibilidad de llevar un control sobre qué ingresa el usuario. Este proceso se repetirá 3 veces, empezaremos con **nomUsuario**; haremos clic derecho/**evets/key/key tiped.**



Ahora inserta el siguiente código

```
private void nomUsuarioKeyTyped(java.awt.event.KeyEvent evt) {

limiteCharter(evt,nomUsuario.getText().length(),8);
```

Veamos sus elementos:

- limiteCharter es el método que creamos para limitar los caracteres. A continuación se ingresa los parámetros.
- Evt es el evento de teclear una tecla en el teclado, la crea el **keytiped** y se la pasamos a nuestro método.
- nomUsuario.getText().length(), el método **getText().length()** nos devuelve el número de caracteres de un texto de un **textfield**, en este caso de **nomUsuario**.
- 8 es el límite que queremos establecer.

Así limitamos la cantidad los caracteres mediante un método, solo falta repetir este proceso para el ID y el password, los códigos serán los siguientes repitiendo el mismo proceso:

```
private void idUsuarioKeyTyped(java.awt.event.KeyEvent evt) {

limiteCharter(evt,idUsuario.getText().length(),4);

    }
```

```
  private void claveKeyTyped(java.awt.event.KeyEvent evt) {

limiteCharter(evt,clave.getText().length(),8);

   }
```

Ahora vamos a limitar los tipos de caracteres, para ello utilizaremos el *código ascii*:

El acrónimo ASCII significa American Standard Code for Information Interchange o Código estándar americano para el intercambio de información. Este es el nombre elegido para indicar el sistema de codificación de caracteres de siete bits utilizado inicialmente en los calculadores. Tiene la particularidad de asignar un número a cada símbolo empleado en los caracteres del sistema. Este código se descarga fácilmente de de internet y aunque puede tener ligeras variaciones según el idioma las diferencias son mínimas y sólo ocupa ciertos caracteres muy precisos que no vienen al caso para este ejercicio.

A continuación veremos el código ascii que suele utilizar nuestro teclado:

Caracteres ASCII de control			Caracteres ASCII imprimibles					
00	NULL	(carácter nulo)	32	espacio	64	@	96	`
01	SOH	(inicio encabezado)	33	!	65	A	97	a
02	STX	(inicio texto)	34	"	66	B	98	b
03	ETX	(fin de texto)	35	#	67	C	99	c
04	EOT	(fin transmisión)	36	$	68	D	100	d
05	ENQ	(consulta)	37	%	69	E	101	e
06	ACK	(reconocimiento)	38	&	70	F	102	f
07	BEL	(timbre)	39	'	71	G	103	g
08	BS	(retroceso)	40	(	72	H	104	h
09	HT	(tab horizontal)	41	)	73	I	105	i
10	LF	(nueva línea)	42	*	74	J	106	j
11	VT	(tab vertical)	43	+	75	K	107	k
12	FF	(nueva página)	44	,	76	L	108	l
13	CR	(retorno de carro)	45	-	77	M	109	m
14	SO	(desplaza afuera)	46	.	78	N	110	n
15	SI	(desplaza adentro)	47	/	79	O	111	o
16	DLE	(esc.vínculo datos)	48	0	80	P	112	p
17	DC1	(control disp. 1)	49	1	81	Q	113	q
18	DC2	(control disp. 2)	50	2	82	R	114	r
19	DC3	(control disp. 3)	51	3	83	S	115	s
20	DC4	(control disp. 4)	52	4	84	T	116	t
21	NAK	(conf. negativa)	53	5	85	U	117	u
22	SYN	(inactividad sínc)	54	6	86	V	118	v
23	ETB	(fin bloque trans)	55	7	87	W	119	w
24	CAN	(cancelar)	56	8	88	X	120	x
25	EM	(fin del medio)	57	9	89	Y	121	y
26	SUB	(sustitución)	58	:	90	Z	122	z
27	ESC	(escape)	59	;	91	[	123	{
28	FS	(sep. archivos)	60	<	92	\	124	|
29	GS	(sep. grupos)	61	=	93	]	125	}
30	RS	(sep. registros)	62	>	94	^	126	~
31	US	(sep. unidades)	63	?	95	_		
127	DEL	(suprimir)						

NOTA: Este no es todo el código, es solo una parte.

Para practicar la limitación de caracteres vamos a transformar todas las minúsculas en mayúsculas dentro del text fiel de **nomUsuario**. Si observamos el código ascii las letras

mayúsculas están ordenadas de tal manera que entre una minúscula y una mayúscula hay una diferencia exacta de posición de 32 espacios. EJ: 90- Z, 122-z, 85-U, 117-u, 70-F, 102-f. Por lo que llegamos a la conclusión de que si le restamos a la posición de cualquier letra minúscula 32 espacios tendremos su homónima minúscula.

Para aplicar esta idea tendremos que situarnos dentro del código en el evento de **keytiped** de **nomUsuario** que ya hemos creado, luego crearemos un if para saber si la tecla utilizada se encuentra entre el ascii de 96 y 123, esto es si comienza en la "a"=97 o termina en la "z"=122 con la ayuda del método **evt.getKeyChar** que nos devuelve dicho código ascii de la tecla empleada. Si se da el caso de entrar dentro del if entonces hay que proceder a restarle los 32 espacios, para ello se utiliza el método **evt.setKeyChar()** , el código de la operación completa es *evt.setKeyChar((char) (evt.getKeyChar()-32));*

El código final dentro del evento keytiped de nomUsuario es:

```
private void nomUsuarioKeyTyped(java.awt.event.KeyEvent evt) {
limiteCharter(evt,nomUsuario.getText().length(),8);
    if (evt.getKeyChar()>96 & evt.getKeyChar()<123){
      evt.setKeyChar((char) (evt.getKeyChar()-32));
    }
  }
```

Ahora para el id directamente vamos a limitar las posibilidad a solamente números. Usando la misma lógica que con las mayúsculas y minúsculas pero con números, vamos a verificar que el código ascii de la tecla empleada se encuentre dentro del rango 48 Y 57 solo que en este caso nos interesa los códigos que estén fuera de ese rango para limitarlos pues sólo queremos que se ingresen números.

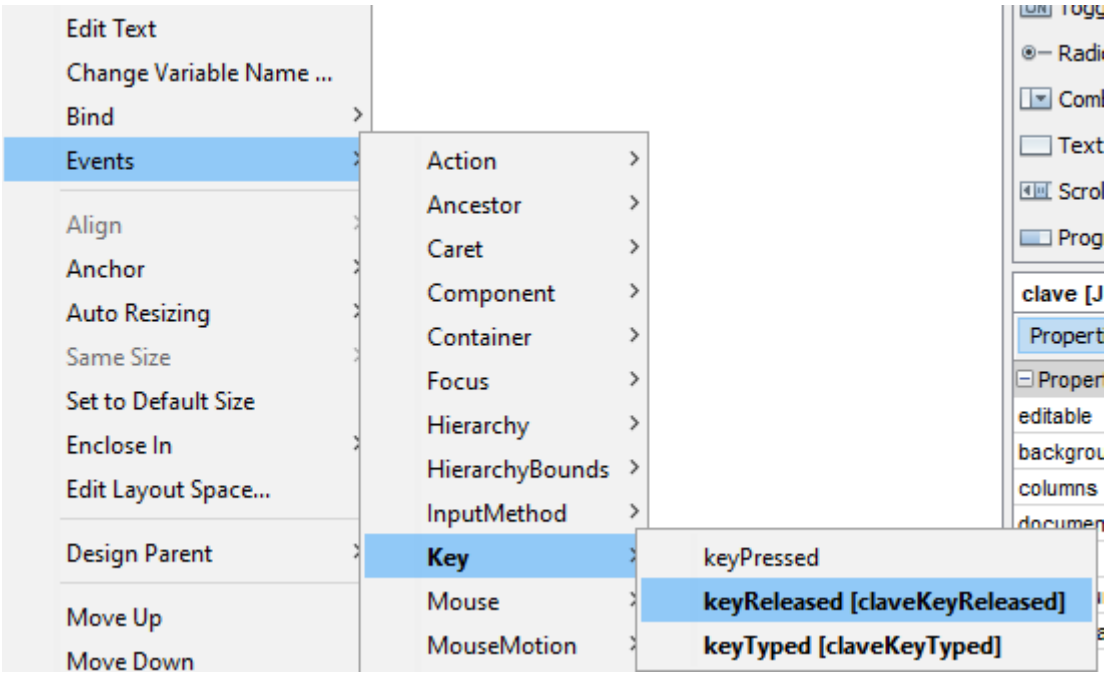
El código resultante es el siguiente:

```
private void idUsuarioKeyTyped(java.awt.event.KeyEvent evt) {
limiteCharter(evt,idUsuario.getText().length(),4);
if (evt.getKeyChar()>57 | evt.getKeyChar()<48){ // condición por fuera del rango de números
   evt.consume();
   Toolkit.getDefaultToolkit().beep();
```

```
}
   }
```

Con respecto al textfield de clave dejaremos que se ingrese mayúsculas,minúsculas o cualquier símbolo ya que esa es la esencia de un password, sólo lo limitaremos a 8 caracteres y le ingresaremos un pequeño código para que a la par que el usuario rellene la clave, automáticamente se autor rellene el label verClave que la "descubre".

Para que se realice de manera más eficiente utilizaremos el evento **keyReleased** que se activa cuando el usuario suelta la tecla que ha tecleado.



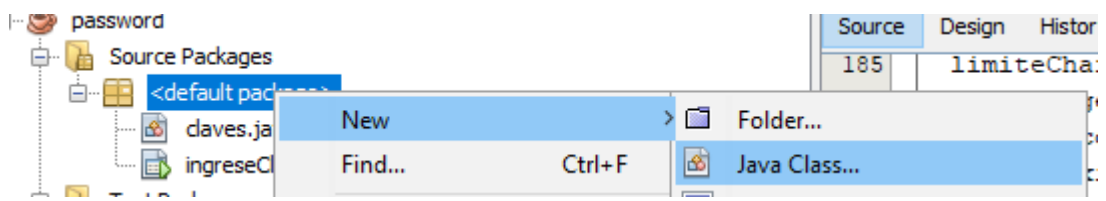

```
private void claveKeyReleased(java.awt.event.KeyEvent evt) {

verClave.setText(clave.getText());

   }
```

Hasta este punto hemos programado toda la interfaz gráfica, solo nos falta la comprobación de datos. Para ello vamos a crear una clase donde albergaremos los usurios, ids y contraseñas.

Recuerda que para crear una clase debes hacer clic secundario en el paquete donde se encuentra nuestro frame, new/ java class



Nombra a la clase como claves. Esta clase contiene 3 variables; usuarios, Id y password. Todas ellas son vectores, lo que significa que contienen más de un elemento en ellas. Ten en cuenta que un vector en java se declara la cantidad de elementos que posee pero se inicializa desde la posición 0, esto significa que si un vector tiene, por ejemplo, 3 elementos estos serán los elementos 0, 1 y 2. Para nuestro ejercicio cargaremos dos usuarios, por lo que declararemos nuestro vector con 2 elementos y la carga será los elementos 0 y 1.

Veamos el código:

```
public class claves {
    String [] usuarios= new String[2]; //declaración de variables vectorizadas
    int [] id= new int[2];
    String [] password= new String[2];

    public claves(){ //carga de valores
       usuarios[0]="ALDO";
    id[0]=77;
    password[0]="ALDO77";
    usuarios[1]="CARPINCH";
    id[1]=2221;
   password[1]="nordel";
}
```

Ahora vamos a crear un método que nos compruebe si los datos del usuarios son correctos, si así lo fueran nos devolverá una variable booleana true, caso contrario esta será false.

```
public boolean comprobar(String usuario, int numID, String contraseña){
boolean vOf =false;//de antemano considera la posible respuesta como falsa
for(int i=0;i<=1;i++){   //siclo que recorre todos los elementos de los vectores
if(usuarios[i].equals(usuario) & numID==id[i] & password[i].equals(contraseña)){
    vOf=true; //si se comprueba que los datos son correctos la respuesta será true
}
```

```
}
 return vOf; //retorna la respuesta
}
```

Nota: para comparar dos variables String en lugar de utilizar el típico == se utiliza el método .equals();

El código completo de la clase claves es la siguiente:

```
public class claves {
    String [] usuarios= new String[2];
    int [] id= new int[2];
    String [] password= new String[2];
    public claves(){
       usuarios[0]="ALDO";
    id[0]=77;
    password[0]="ALDO77";
    usuarios[1]="CARPINCH";
    id[1]=2221;
   password[1]="nordel";
}
 public boolean comprobar(String usuario, int numID, String contraseña){
boolean vOf =false;
for(int i=0;i<=1;i++){
if(usuarios[i].equals(usuario) & numID==id[i] & password[i].equals(contraseña)){
    vOf=true;
}
}
 return vOf;
}}
```

Finalmente vamos a programar el botón que ejecutará la comprobación que programamos en nuestra clase claves. Recuerda que para hacer el llamado a un método de otra clase tienes que crear un objeto al cual haces el llamado y le asignas sus métodos.

Devuelta en el Frame:

Vamos a redactar el siguiente código en el botón que creamos para ingresar los datos:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

      claves validar=new claves(); //creamos el objeto de la clase claves para hacer un llamado a su método

     if (validar.comprobar(nomUsuario.getText(),Integer.parseInt(idUsuario.getText()), clave.getText())){// se comprueba si los datos cargados son los correctos

       JOptionPane.showMessageDialog(this, "Usuario confirmado");//mensajes de correcto e incorrecto

     }

     else{

       JOptionPane.showMessageDialog(this, "Usuario no encontrado o error de clave");

     }

  }
```

El código del frame (sin el código de creación e iniciación de los elementos) es el siguiente:

```
import java.awt.Toolkit;

import javax.swing.JOptionPane;

public class ingreseClave extends javax.swing.JFrame {

   public ingreseClave() {

      initComponents();

      verClave.setVisible(false);

   }

   public void limiteCharter(java.awt.event.KeyEvent evt,int cajaText,int limite){

    if (cajaText==limite){
```

```
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    }
   }
private void nomUsuarioKeyTyped(java.awt.event.KeyEvent evt) {
limiteCharter(evt,nomUsuario.getText().length(),8);
      if (evt.getKeyChar()>96 & evt.getKeyChar()<123){
        evt.setKeyChar((char) (evt.getKeyChar()-32));
      }
   }

   private void idUsuarioKeyTyped(java.awt.event.KeyEvent evt) {
limiteCharter(evt,idUsuario.getText().length(),4);
if (evt.getKeyChar()>57 | evt.getKeyChar()<48){
   evt.consume();
   Toolkit.getDefaultToolkit().beep();
}
   }

   private void claveKeyTyped(java.awt.event.KeyEvent evt) {
limiteCharter(evt,clave.getText().length(),8);
   }

   private void checkClaveMouseClicked(java.awt.event.MouseEvent evt) {
if (checkClave.isSelected()){
   verClave.setVisible(true);
}  else{
   verClave.setVisible(false);
}
```

```
    }


    private void claveKeyReleased(java.awt.event.KeyEvent evt) {
verClave.setText(clave.getText());        // TODO add your handling code here:
    }


    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        claves validar=new claves();
       if (validar.comprobar(nomUsuario.getText(),Integer.parseInt(idUsuario.getText()),
clave.getText())){
         JOptionPane.showMessageDialog(this, "Usuario confirmado");
       }
       else{
          JOptionPane.showMessageDialog(this, "Usuario no encontrado o error de
clave");
       }
    }
```

Actividad: Añade en el ingreso de usuario la posibilidad de utilizar un nombre con la letra "Ñ" y que si se ingrese una "ñ" minúscula esta se transforme automáticamente en mayúscula. Para ello tendrás que buscar sus respectivos códigos ascii. Luego, añade un tercer usuario con su respectivo número id y clave en la clase claves. Para ello tendrás que modificar los respectivos vectores para que acepten un elemento más (pasan de 2 a 3) y no olvides añadir un ciclo más al for de comprobación (pasa de 1 a 2).