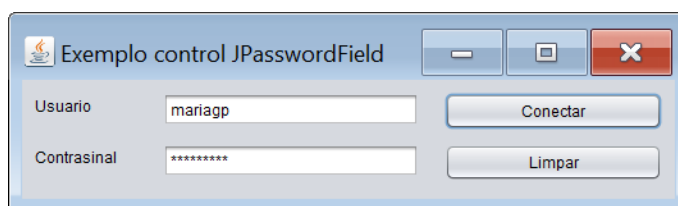


1. Compoñente Caixa de texto para contrasinais (JPasswordField)

El compoñente caixa de texto para contrasinais é un compoñente similar a unha caixa de texto, pero coa singularidade de que o texto escrito nela é substituído por unha máscara que impide que se poida visualizar o que estamos escribindo no compoñente. Este compoñente é empregado habitualmente para solicitar contrasinais ao usuario, de xeito que ninguén poida visualizar o que se está escribindo no compoñente. O compoñente caixa de texto para contrasinais defínese a través da clase `javax.swing.JPasswordField` e herda de `javax.swing.JTextField`. Graficamente o seu aspecto é o seguinte:



Na imaxe anterior podemos ver o emprego dunha caixa de texto para contrasinais.

Propiedades do control `JPasswordField` empregadas máis habitualmente e que pódense establecer a través do entorno de programación:

Propiedade	Función
Background	Cor de fondo do compoñente
Border	Borde do compoñente
Cursor	Mediante esta propiedade indícase a imaxe que amosará o punteiro do rato ao navegar sobre o compoñente
EchoChar	Mediante esta propiedade indicamos cal é o carácter empregado como máscara para a caixa de texto para contrasinais. Se esta propiedade vale 0 o comportamento do compoñente será similar a un <code>JTextField</code> .
Editable	Emprégase para indicar se podemos ou non escribir na caixa de texto para contrasinais
Enabled	Se esta propiedade está activada o compoñente será funcional. No caso de que esta propiedade estea desactivada o compoñente será visualizable, pero o usuario non poderá interaccionar con el.
Focusable	Se esta propiedade está activada o compoñente entrará na roda de reparto do foco de xeito que ao premer a tecla de cambio de foco (habitualmente o tabulador) nalgún momento tomará o foco da aplicación (cando sexa a súa quenda na roda de reparto do foco). Pola contra, se esta propiedade está desactivada o compoñente non entrará na roda de reparto do foco e soamente será posible acceder a el mediante o rato.
Font	Características da fonte do compoñente (tipo de fonte, tamaño, ...)
Foreground	Cor do texto do compoñente
HorizontalAlignment	Aliñación horizontal do texto do compoñente
SelectedTextColor	Color do texto seleccionado na caixa de texto para contrasinais
SelectionColor	Color de fondo do texto seleccionado na caixa de texto para contrasinais
Text	Emprégase para establecer o texto que aparece na caixa de texto para contrasinais

ToolTipText	Mediante esta propiedade establécese unha mensaxe (tooltip) que será amosado ao deixar o punteiro do rato sobre o compoñente. É habitual empregar esta mensaxe para indicar cal é a utilidade do compoñente
-------------	---

Eventos do control JPasswordField empregados máis habitualmente e que pódense establecer a través do entorno de programación:

Evento	Lanzamento
KeyPressed	Este evento é disparado cando pulsamos unha tecla sobre a caixa de texto para contrasinais. Se deixamos unha tecla pulsada, o evento é disparado tantas veces como caracteres escribanse na caixa de texto para contrasinais
KeyReleased	Este evento é disparado cando deixamos de pulsar unha tecla sobre a caixa de texto para contrasinais. Se deixamos unha tecla pulsada, o evento será disparado unicamente cando deixemos de pulsar a tecla.
KeyTyped	Este evento é disparado cando escribimos un carácter na caixa de texto para contrasinais.

Métodos do control JPasswordField empregados máis habitualmente:

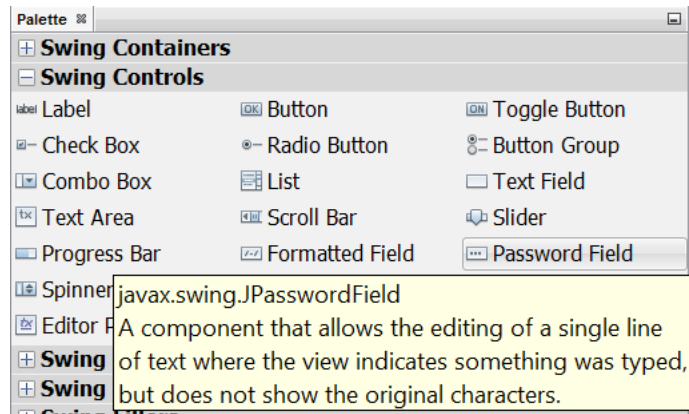
Método	Función
public char[] getPassword()	Emprégase para recuperar o texto dunha caixa de texto para contrasinais
public void setText(String text)	Emprégase para establecer o texto dunha caixa de texto para contrasinais

Xestión de caixas de texto para contrasinais empregando NetBeans

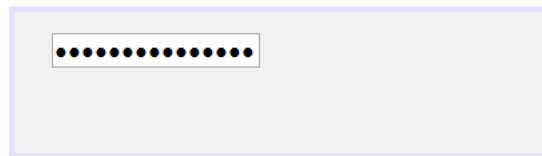
A continuación imos desenvolver o seguinte formulario:

O formulario consta de dúas caixas de texto. A primeira é unha caixa de texto normal para indicar o nome do usuario. A segunda é unha caixa de texto para contrasinais na cal o usuario introducirá a súa contrasinal. Cando pulsemos no botón Conectar, intentarase abrir unha conexión contra unha base de datos empregando a tupla usuario / contrasinal introducida no formulario (como non é parte desta actividade a xestión de bases de datos, simplemente amosaremos unha mensaxe por pantalla indicando o usuario e a contrasinal introducidas a través do formulario). Cando pulsemos no botón Limpar borraranse os contidos das caixas de texto do formulario. A aplicación debe informar ao usuario de calquera erro que poida acontecer ao longo do seu emprego.

Para engadir un compoñente de tipo JPasswordField no noso formulario primeiramente seleccionáremolo da paleta de compoñentes do entorno de programación:

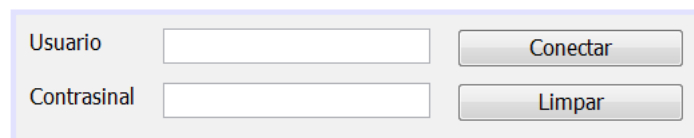


e arrastrámolo ata o formulario:

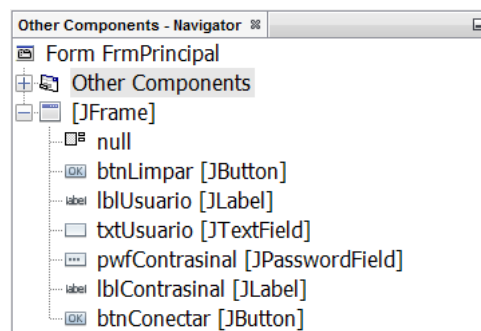


Unha vez situado dentro do formulario, e como sempre, adaptámolo para que teña o aspecto visual que desexamos que teña.

No caso que estamos desenvolvendo necesitamos engadir dúas etiquetas, unha caixa de texto, dous botóns e unha caixa de texto para contrasinais. Unha vez engadidos e colocados no seu lugar correspondente os compoñentes, este será o resultado do deseño:



O noso formulario quedará configurado do seguinte xeito:



Unha vez que temos colocados os compoñentes que precisamos, hai que configurar o seu aspecto. Neste caso non necesitamos modificar as propiedades de ningún compoñente xa que as predefinidas axústanse perfectamente aos requirimentos do programa.

O comportamento visual da aplicación xa está resolto. Agora imos desenvolver a súa funcionalidade.

A acción desenvólvese no momento no que fagamos clic sobre o botón Conectar. O código asociado ao clic sobre este botón é o seguinte:

```
private void btnConectarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String usuario=txtUsuario.getText();  
    if(usuario.trim().compareTo("")==0)  
    {  
        JOptionPane.showMessageDialog(this, "Debe indicar o nome de usuario");  
        return;  
    }  
  
    String contrasinal=new String(pwfContrasinal.getPassword());  
    System.out.println("contrasinal"+contrasinal.length());  
    if(contrasinal.trim().compareTo("")==0)  
    {  
        JOptionPane.showMessageDialog(this, "Debe indicar o contrasinal");  
        return;  
    }  
    JOptionPane.showMessageDialog(this, "A conexión contra a base de datos vaise realizar empregando  
o usuario \"\""+usuario+"\" e o contrasinal \"\""+contrasinal+"\"");  
}
```

Centrándonos na parte referente á xestión da caixa de texto de contrasinais, temos a seguinte liña:

```
String contrasinal=new String(pwfContrasinal.getPassword());
```

Como pódese observar, para recoller o contido da caixa de texto de contrasinais emprégase o seu método getPassword (poderíase empregar o seu método getText, pero é un método deprecated, polo tanto non é recomendable empregalo). Este método devolve un arrai de caracteres (char []), así que o máis habitual é ter que converter ese arrai a un String para poder manipulalo mediante as utilidades de cadea provistas pola clase String.

Respecto ao botón Limpar e centrándonos na xestión que se fai nel da caixa de texto de contrasinais, no método que xestiona a súa pulsación temos a seguinte liña:

```
pwfContrasinal.setText("");
```

Como podemos ver, a asignación de valores para unha caixa de texto de contrasinais é similar á asignación de valores para unha caixa de texto de tipo JTextField.