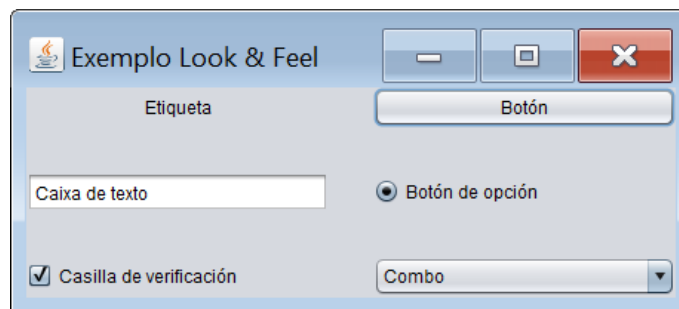


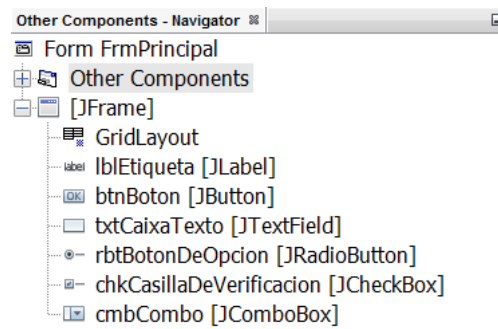
1.1.1 Máscaras

As máscaras, habitualmente chamadas look & feel serven para cambiar o aspecto visual xeral das nosas aplicacións gráficas. Un look & feel está composto por unha serie de clases que definen o aspecto visual de cada un dos compoñentes que forman parte do noso entorno gráfico. Ao establecer sobre a nosa aplicación de usuario un look & feel estarémoslle indicando ao compilador que cando debuxe os compoñentes da aplicación empregue as definicións gráficas establecidas nas clases do look & feel que estamos empregando.

Ata o momento, nas aplicacións desenvolvidas non tivemos que facer nada especial para establecer o aspecto visual dos compoñentes xa que NetBeans establece un look & feel por nos. Por exemplo, imos crear a seguinte aplicación:



Mediante o entorno de programación establecemos que queremos un formulario con unha serie de compoñentes xestionados a través dun GridLayout:



Non obstante, en ningún sitio dicímoslle ao compilador que look & feel ten que empregar para amosar os compoñentes. Isto é feito por NetBeans. Se observamos o código do método principal deste formulario (método main) atopámonos entre outras con estas liñas de código:

```
for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {  
    if ("Nimbus".equals(info.getName())) {  
        javax.swing.UIManager.setLookAndFeel(info.getClassName());  
        break;  
    }  
}
```

O método `javax.swing.UIManager.getInstalledLookAndFeels` devólvenos un arrai con todos os look & feels que hai preinstalados no noso sistema. Despois recollese ese arrai cun `for` e no caso de que algún dos look & feels almacenados no arrai chámese Nimbus, mediante o método `javax.swing.UIManager.setLookAndFeel` é establecido como look & feel da nosa aplicación. Inicialmente hai cinco look & feels preinstalados. Son os seguintes:

- Metal, xestionado mediante a clase `javax.swing.plaf.metal.MetalLookAndFeel`.
- Nimbus, xestionado mediante a clase `javax.swing.plaf.nimbus.NimbusLookAndFeel`.

- Motif, xestionado mediante a clase `com.sun.java.swing.plaf.motif.MotifLookAndFeel`.
- Windows, xestionado mediante a clase `com.sun.java.swing.plaf.windows.WindowsLookAndFeel`.
- Windows Classic, xestionado mediante a clase `com.sun.java.swing.plaf.windows.WindowsClassicLookAndFeel`.

Podemos empregar calquera de eles e ao facelo o aspecto da nosa aplicación variará en función do look & feel empregado. Entra algúns destes look & feels preinstalados non notaremos apenas diferencias, pero noutros casos estas serán notables.

Imos ver como modificar o noso código para empregar un look & feel en concreto. Substituiremos o bloque de código anterior pola seguinte liña:

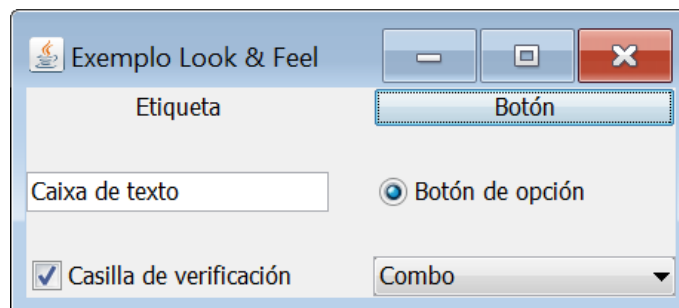
```
javax.swing.UIManager.setLookAndFeel("");
```

e dentro das comiñas indicaremos o nome da clase que xestiona o look & feel que queremos empregar. Imos ver algúns exemplos.

Se empregamos a seguinte liña:

```
javax.swing.UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
```

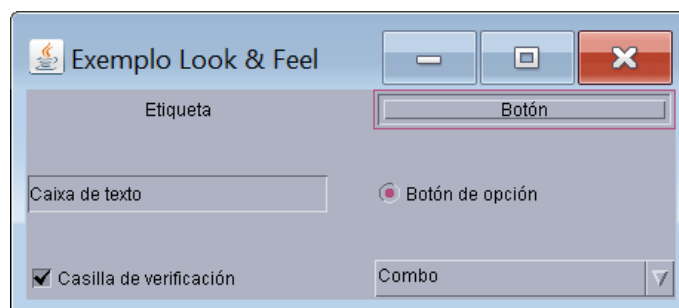
este será o resultado de executar a aplicación:



Se empregamos a seguinte liña:

```
javax.swing.UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");
```

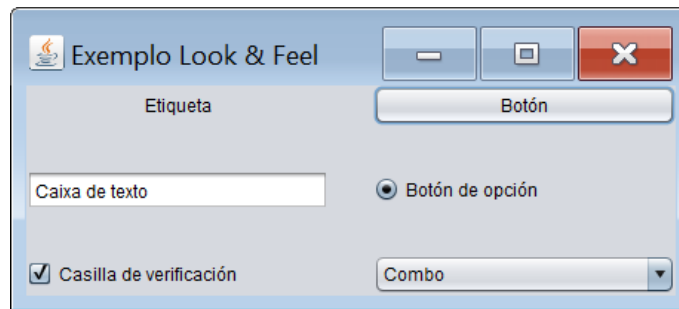
este será o resultado de executar a aplicación:



Se empregamos a seguinte liña:

```
javax.swing.UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");
```

este será o resultado de executar a aplicación:



En tódolos casos a aplicación é a mesma. O seu funcionamento é exactamente igual. O único que varía é o seu aspecto en función do look & feel seleccionado.

Afortunadamente non estamos limitados aos cinco look & feels citados con anterioridade. Se queremos empregar outro look & feel temos dúas opcións:

- Podemos crear o noso propio look & feel (pero isto non é parte desta actividade).
- Podemos empregar un look & feel xa desenvolvido.

Ímonos centrar na segunda opción. Na rede pódense atopar look & feels xa desenvolvidos e empregalos nas nosas aplicacións. Sempre deberase respectar a licencia do look & feel que empreguemos xa sexas este un look & feel libre ou un look & feel propietario.

A día de hoxe e desde hai bastantes anos, unha boa páxina para atopar look & feels para descargar, probar e empregar (insisto, sempre respectando as restricións impostas pola licencia do look & feel descargado) é o seguinte: <http://www.javootoo.com/>

Desde esta páxina poderemos acceder a multitude de look & feels libres e propietarios. Se prememos nesta páxina sobre a opción de menú Free Look And feel accederemos a un listado de look & feels libres (aínda que sempre haberá que comprobar as licencias para conformar que realmente son libres). Se por exemplo prememos sobre o look & feel Quaqua Look And Feel, amósasenos a seguinte páxina:

Home > Free Look And Feel > Quaqua Look And Feel >

Home

Free & Skinnable Look And Feel

Free Look And Feel

FH Look And Feel

InfoNode Look and Feel

JGoodies Looks

LipstikLF

Liquid Look And Feel

Napkin Look & Feel

NimROD Look & Feel

Office Look And Feels

PgsLookAndFeel

Quaqua Look And Feel

Squareness Look And Feel

TinyLAF

Tonic Look And Feel

Commercial Look And Feel

Discontinued Look And Feel

Icon Collections

Contact

Swing @ java.sun.com

Quaqua Look And Feel

[previous](#) | [next](#)

The Quaqua Look and Feel (Quaqua) is a user interface library for Java applications which wish to closely adhere to the Apple Human Interface Guidelines for Mac OS X.

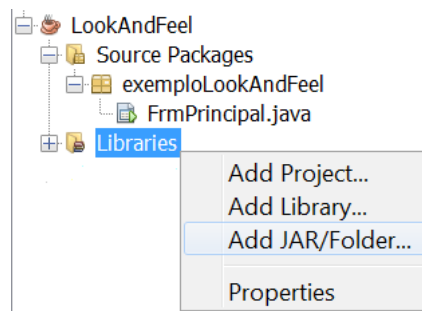
Screenshots

Facts

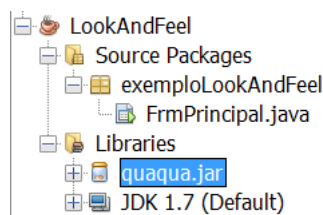
Developer Website	http://www.randelshofer.ch/quaqua/download.html
License	Free for commercial and non-commercial use.
Supported JDKs	1.3+, 5.0, 6.0
Webstart Demo	Launch

Nesta páxina temos información sobre o look & feel Quaqua (pantallazos, páxina de descarga, información da licencia, JDKs soportados e incluso unha demo webstart que nos amosa o funcionamento deste look & feel).

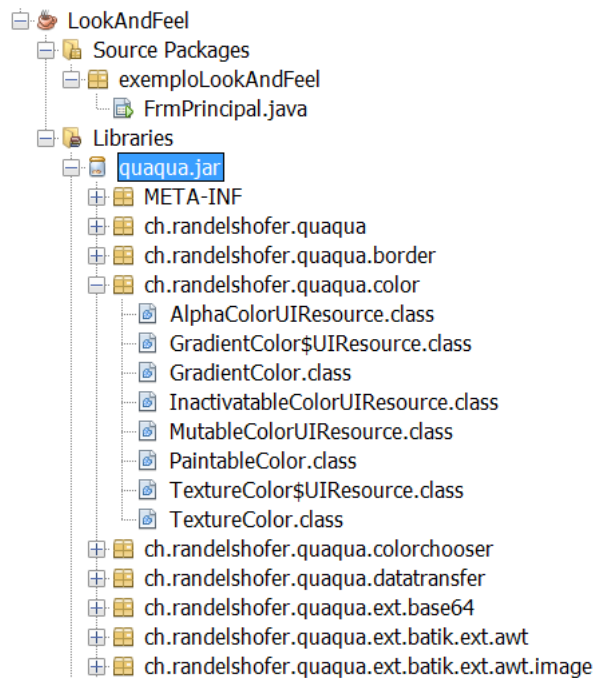
Imos descargar este look & feel (a páxina do desenvolvedor do Quaqua Look And Feel a día de hoxe é <http://www.randelshofer.ch/quaqua/index.html>). Unha vez descargado, descomprimímolos nun directorio calquera e revisamos a documentación adxunta (neste caso hai documentación, pero non sempre é así. En moitas ocasións a documentación adxunta é escasa, incompleta ou directamente inexistente polo cal hai que facer voar á imaxinación). Neste caso a documentación é moi clara e indícanos varios xeitos de empregar este look & feel. Un deles é o exposto anteriormente, así que o faremos dese xeito. O primeiro que nos di a documentación é que necesitaremos empregar as clases que veñen empaquetadas nun arquivo chamado quaqua.jar. Para facer isto desde NetBeans, o máis cómodo é engadir este jar ás librerías que emprega o noso proxecto. Para elo no noso proxecto prememos co botón dereito do rato sobre a carpeta Libraries e seleccionamos a opción Add JAR/Folder:



A continuación navegamos pola estrutura de carpetas da nosa máquina e seleccionamos o arquivo quaqua.jar para engadilo ás librerías do noso proxecto. Unha vez feito veremos que dentro da carpeta Libraries teremos o noso arquivo de clases quaqua.jar:



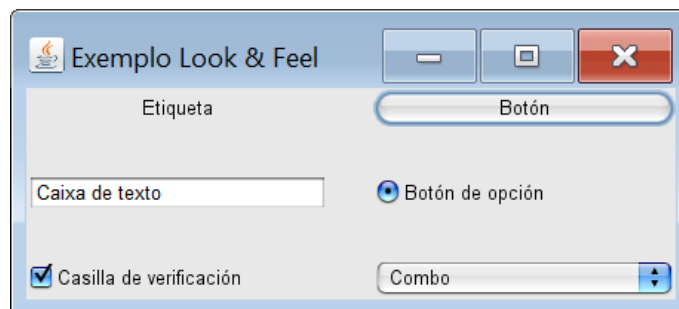
Se despregamos o contido do arquivo quaqua.jar podemos ver os seus contidos (paquetes, clases e recursos que compoñen o look & feel Quaqua):



Cando un look & feel non está ben documentado, ou simplemente non está documentado, o contido do jar dese look & feel case sempre pode darnos pistas sobre o seu funcionamento. Afortunadamente neste caso a documentación é moi completa e indica que unha vez engadido o arquivo quaqua.jar ás librerías do noso proxecto, para establecer Quaqua como o look & feel da aplicación deberemos empregar a seguinte liña de código:

```
javax.swing.UIManager.setLookAndFeel("ch.randelshofer.quaqua.QuaquaLookAndFeel");
```

Como pódese observar é o método empregado anteriormente, pero variando o nome da clase que xestiona o look & feel que queremos empregar. O resultado de executar a aplicación anterior con este novo look & feel é o seguinte:



Ao igual que antes, a aplicación sigue sendo a mesma. O seu funcionamento é exactamente igual. O único que varía é o seu aspecto xa que agora emprega o look & feel Quaqua.