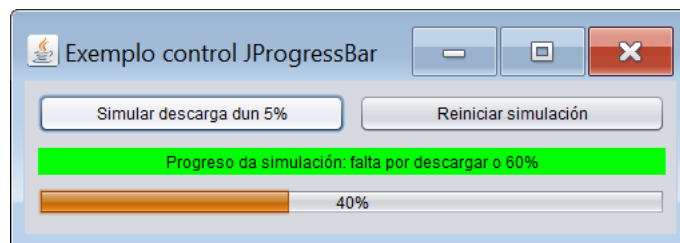


# 1. Compoñente Barra de progreso (JProgressBar)

A barra de progreso é un compoñente gráfico que como o seu nome indica emprégase para visualizar o progreso dalgunha tarefa que estamos realizando. Para visualizar o progreso da acción que estamos realizando, a barra de progreso debuxará unha barra cuxo tamaño vai ir aumentando a medida que se vai completando a tarefa á que fai referencia. Cando a tarefa sexa completada a barra estará debuxada completamente. Adicionalmente pode amosar o porcentaxe de barra debuxado (o cal corresponde ao porcentaxe de tarefa realizado). O compoñente barra de progreso defínese a través da clase `javax.swing.JProgressBar`. Gráficamente o seu aspecto é o seguinte:



Na imaxe anterior podemos ver o emprego dunha barra de progreso a cal amosa o porcentaxe realizado dunha tarefa.

Propiedades do control `JProgressBar` 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
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
Indeterminate	As veces non coñecemos cal vai ser a duración do proceso que estamos representando mediante a nosa barra de progreso. Neste caso é habitual amosar unha barra de progreso que simplemente se move (indicando ao usuario que o proceso se está levando a cabo). No caso de que queramos que a nosa barra de progreso funcione dese xeito debemos de activar esta propiedade. Se empregamos unha barra de progreso con esta propiedade activada é recomendable que no momento no que se dispoña da información sobre o que resta para terminar a tarefa, a propiedade sexa desactivada para amosar unha barra de progreso que informe realmente ao usuario do estado da súa tarefa.
Maximum	Valor máximo da barra de progreso. Cando a propiedade value da barra de progreso alcance este valor, a barra de progreso estará completa.

Minimum	Valor mínimo da barra de progreso. Cando a propiedade value da barra de progreso alcance este valor, a barra de progreso estará baleira.
Orientation	Mediante esta propiedade indicamos como vaise encher a barra de progreso para indicar o progreso da tarefa. Se vale 0 será horizontalmente de esquerda a dereita. Se vale 1 será verticalmente de abaixo a arriba.
String	Cadea de texto que se amosa dentro da barra de progreso.
StringPainted	Mediante esta propiedade indícase se queremos que se amose unha cadea de texto dentro da barra de progreso.
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
Value	Valor inicial da barra de progreso. Ten que ser un valor dentro do rango posible do compoñente.

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

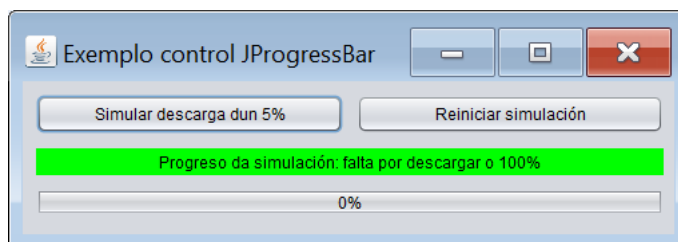
Evento	Lanzamento
StateChanged	Este evento é disparado cando cambiamos o valor da propiedade value da barra de progreso.

Métodos do control JProgressBar empregados máis habitualmente:

Método	Función
public int getValue()	Devolve o valor da barra de progreso.
public void setValue(int n)	Establece o valor da barra de progreso. Debe ser un elemento entre os valores posibles que pode tomar o compoñente.

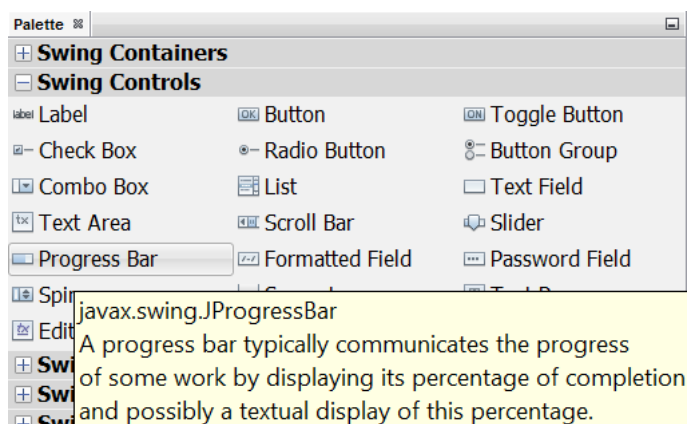
## Xestión de barras de progreso empregando NetBeans

A continuación imos desenvolver o seguinte formulario:

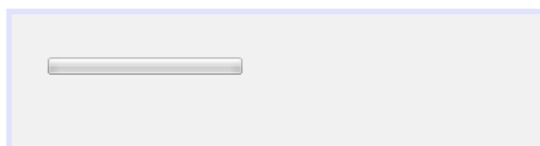


O formulario consta dunha barra de progreso. Nesta aplicación imos simular a descarga dun arquivo. O ideal sería facer unha descarga real, pero non é parte da materia o tratamento de conexións de datos, polo tanto imos simplificar o problema facendo unha simulación. O funcionamento da aplicación é o seguinte: cada vez que pulsemos o botón Simular descarga dun 5% equivale a descargar un 5% del arquivo e polo tanto debe reflectirse un avance de 5 puntos porcentuais na barra de progreso, a cal indica o avance da descarga. Ademais, temos unha etiqueta na cal amósase en todo momento canto resta por descargar. Cando a descarga sexa completada, esta etiqueta cambiará de cor e informará sobre a finalización da descarga. Ademais temos o botón Reiniciar simulación que é empregado para volver a aplicación a o seu estado inicial. Calquera erro que poida acontecer ao longo do emprego do programa debe ser reportado ao usuario.

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

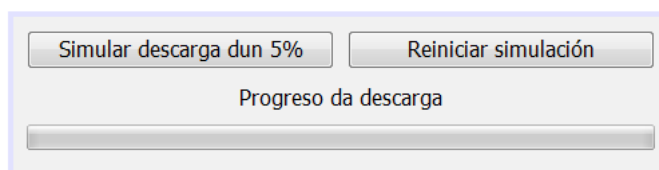


e arrastrámolo ata o formulario:

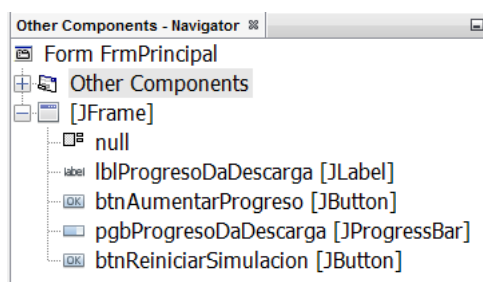


Unha vez situado o compoñente dentro do formulario, adaptámolo para que teña o aspecto visual que desexamos que teña.

No caso que estamos desenvolvendo necesitamos engadir unha etiqueta, dous botóns e unha barra de progreso. Unha vez engadidos e colocados no seu lugar correspondente, este será o resultado do deseño:



O noso formulario quedará configurado do seguinte xeito:



Unha vez que temos colocados os compoñentes que necesitamos hai que configurar o seu aspecto. Neste caso faremos as seguintes modificacións:

- Para a etiqueta `lblProgresoDaDescarga`, o texto estará centrado respecto á etiqueta. Os cambios de texto e cor serán feitos vía código.
- Para a barra de progreso, unicamente activaremos a súa propiedade `StringPainted` para indicar que queremos que se visualice no seu interior unha cadea de texto que amose a porcentaxe de avance da tarefa.

O comportamento visual da aplicación está practicamente resolto (unicamente faltaríanos xestionar o cambio de texto e cor da etiqueta en función do avance da simulación). Agora imos desenvolver a súa funcionalidade.

Os requirimentos do programa pídenos que cada vez que pulsemos o botón `btnAumentarProgreso`, modifiquemos a barra de progreso para amosar un avance de cinco unidades porcentuais. Vexamos a continuación o código fonte asociado ao clic sobre o botón `btnAumentarProgreso`:

```
private void btnAumentarProgresoActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    progreso+=5;  
    if(progreso>100)  
    {  
        JOptionPane.showMessageDialog(this, "A descarga xa foi completada. Reinicie a simulación");  
        return;  
    }  
    pgbProgresoDaDescarga.setValue(progreso);  
}
```

Para simular o avance da nosa descarga temos declarada unha variable `int` a nivel de clase chamada `progreso`. Inicialmente vale cero. Nesta variable almacenaremos o avance da simulación (terá un valor entre cero e cen). Cada vez que pulsamos o botón incrementámola en cinco unidades, reflectíndose nela o avance da simulación da descarga. Para amosar visualmente este avance no compoñente gráfico barra de progreso empregamos o seguinte código:

```
pgbProgresoDaDescarga.setValue(progreso);
```

Como pódese observar, para establecer o valor dunha barra de progreso hai que empregar o seu método `setValue`, pasándolle o novo valor que debe tomar (este valor debe ser un dos xestionables polo seu rango, o cal foi establecido coas propiedades `minimum` e `maximum`). Ao establecer un novo valor sobre a barra de progreso, este reflíctese automaticamente na súa visualización.

Os requirimentos do programa tamén pídenos que na etiqueta de texto `lblProgresoDaDescarga` sexa amosado en todo momento o estado da descarga mediante un código de cores. Cando realizamos algún cambio nos valores dunha barra de progreso dispárase o seu evento `stateChanged`. Polo tanto, a fin de cumprir os requirimentos do noso programa debemos implementar o evento `stateChanged` da barra de progreso `pgbProgresoDaDescarga`. Vexamos a continuación a súa implementación:

```
private void pgbProgresoDaDescargaStateChanged(javax.swing.event.ChangeEvent evt) {  
    // TODO add your handling code here:  
    if(pgbProgresoDaDescarga.getValue()<100)  
    {  
        lblProgresoDaDescarga.setText("Progreso da simulación: falta por descargar o "+(100-  
progreso)+"%");  
    }  
    else  
    {  
        lblProgresoDaDescarga.setBackground(Color.red);  
        lblProgresoDaDescarga.setText("Progreso da simulación: descarga completa");  
    }  
}
```

Como pódese observar no bloque de código exposto, para recuperar o valor da barra de progreso empregamos o seu método `getValue`. En función do valor recuperado xeramos a mensaxe apropiada para a etiqueta e asignámoslle a cor correspondente.

O botón `btnReiniciarSimulacion`, como dixemos anteriormente, encárgase de reiniciar a aplicación. No tocante á barra de progreso, ao facer clic sobre este botón execútase o seguinte código:

```
pgbProgresoDaDescarga.setValue(0);
```

Como xa vimos anteriormente, estamos dando un novo valor para a barra de progreso, o cal reflectirase automaticamente na súa visualización.

Na seguinte imaxe amósase a execución da aplicación:

