

1. A3. Localización, aliñamento e aspecto dos compoñentes

1.1 Introducción

Na actividade que nos ocupa aprenderanse os seguintes conceptos e manexo de destrezas:

- Xestionar os layout managers básicos e avanzados proporcionados pola linguaxe de programación java para localizar e aliñar axeitadamente os compoñentes das nosas aplicacións gráficas de usuario.
- Empregar máscaras para modificar o aspecto dos compoñentes das nosas aplicacións gráficas de usuario.

1.2 Actividade

1.2.1 Layout managers

Ao desenvolver interfaces gráficas de usuario atopámonos sempre co problema da xestión da distribución dos compoñentes gráficos. Este problema consiste no seguinte: nas nosas aplicacións gráficas de usuario cada contedor aloxa un número determinado de compoñentes gráficos, cada un dos cales terá unha posición e un tamaño dentro do contedor. O problema ocorre no momento no que modificamos o tamaño do contedor, é dicir, cando redimensionámolo. ¿Que ocorre coas posicións e tamaños dos compoñentes gráficos respecto ao novo tamaño do contedor?. É evidente que ao redimensionar o contedor, se non variamos as posicións e os tamaños dos seus compoñentes gráficos, estes quedarán descadrados respecto ao contedor, dando como resultado unha aplicación visualmente ineficiente. Polo tanto, cada vez que redimensionemos os nosos contedores, deberemos recalcular as posicións e os tamaños dos seus compoñentes gráficos. Evidentemente esta é unha ardua tarefa de programación, e é neste momento cando entran en xogo os layout managers para facilitarnos esta tarefa. Un layout manager, tamén coñecido como xestor de distribución, é unha clase java predefinida que se aplica sobre un contedor e que se encarga de posicionar e redimensionar os seus compoñentes gráficos en función do tamaño que poida tomar o contedor, descargando ao programador desta tarefa.

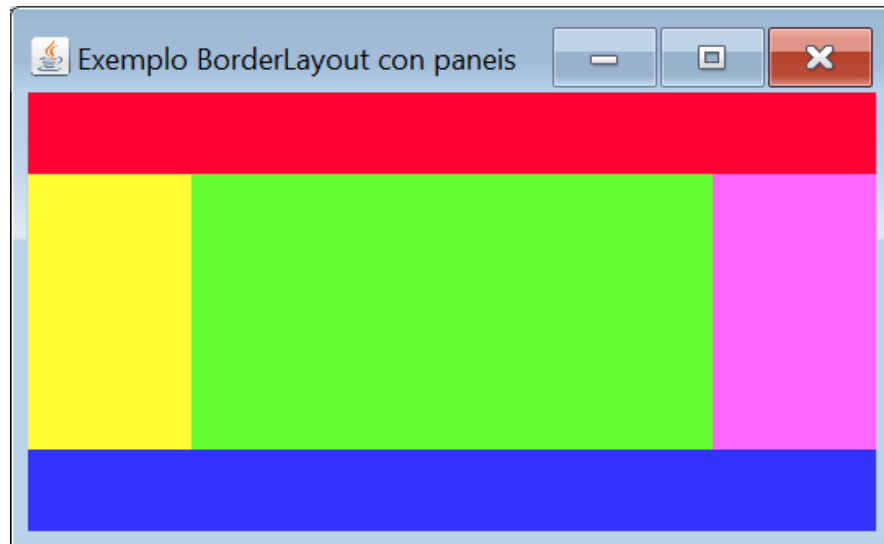
Existen diversos layout managers predefinidos en java, aínda que tamén é posible construír o noso propio layout manager. Cada layout manager xestiona os compoñentes gráficos do contedor sobre o que se aplica dun xeito determinado, de maneira que en función do layout manager aplicado sobre un contedor, o aspecto visual da distribución dos seus compoñentes gráficos variará. É tarefa do programador determinar que layout manager o grupo de layout managers empregar sobre cada contedor en función dos requisitos gráficos da aplicación que estea desenvolvendo. Os layout managers predefinidos empregados con maior asiduidade e que estudaremos a continuación dun xeito detallado son os seguintes:

- BorderLayout
- FlowLayout
- BoxLayout
- CardLayout
- GridLayout

- GroupLayout
- NullLayout
- AbsoluteLayout
- GridBagLayout

1.2.1.1 Layout manager BorderLayout

O layout manager BorderLayout defínese a través da clase `java.awt.BorderLayout`. O seu aspecto visual é o seguinte:



Na imaxe anterior amósase un formulario sobre o que foi establecido un BorderLayout. Sobre o layout manager BorderLayout podemos establecer ata cinco zonas (aínda que non é obrigatorio que existan todas). Neste caso en cada unha das zonas temos colocado un panel e cada un deles está coloreado de distinto xeito co fin de que poidamos visualizar perfectamente ata onde chega cada zona.

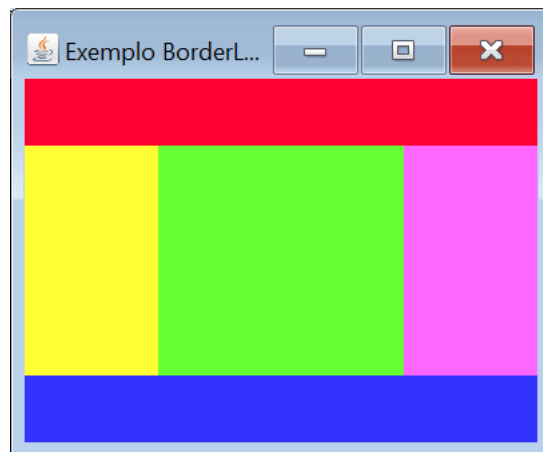
- O panel vermello está na zona North
- O panel azul está na zona South
- O panel amarelo está na zona West
- O panel rosa está na zona East
- O panel verde está na zona Center

Aínda que neste caso temos colocado un panel en cada zona, tamén sería posible colocar un ou máis controis sobre cada unha das zonas. Non obstante, o emprego de paneis facilita a comprensión do modelo, permite o uso de novos layout managers sobre os diferentes paneis e ademais posibilita a reutilización dos paneis xerados.

¿Cal é a vantaxe de empregar o layout manager BorderLayout?. Ata agora, o único que temos feito é dividir o noso formulario en varias zonas sobre as que colocar controis. Aparentemente isto non nos proporciona ningunha vantaxe. Pero, ¿que acontece se redimensionamos o noso formulario?. Se ampliamos o formulario, este é o resultado:



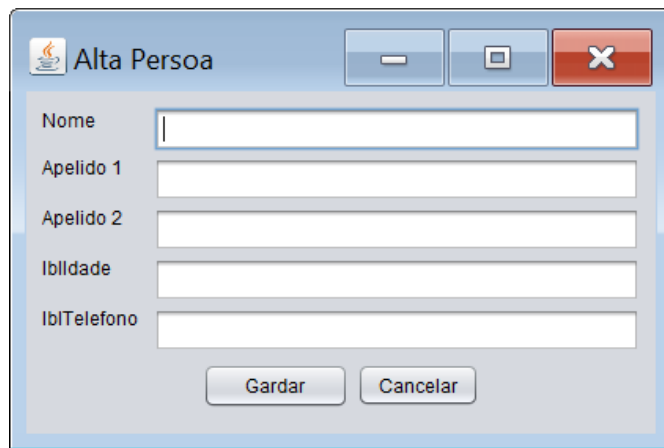
Se pola contra reducimos o formulario, o resultado é estoutro:



Como ben pódese observar, independentemente do tamaño que lle asignemos ao noso formulario, este sempre cumpre co seguinte:

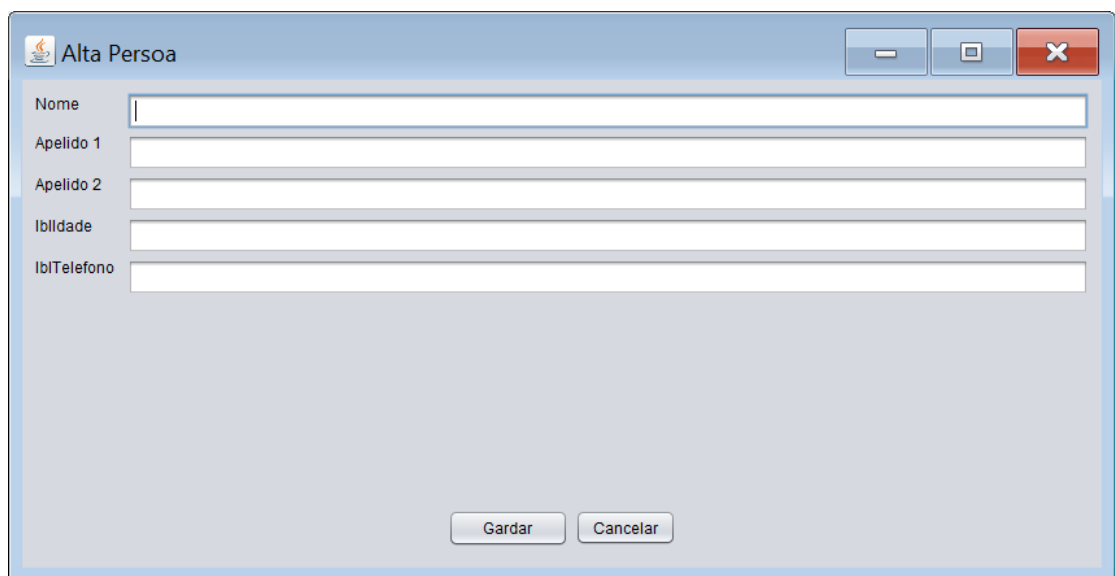
- Os paneis vermello e azul manteñen o seu tamaño vertical, mentres que varían o seu tamaño horizontal.
- Os paneis amarelo e rosa manteñen o seu tamaño horizontal, mentres que varían o seu tamaño vertical.
- O panel verde, o central, gaña espazo horizontal e vertical ao aumentar o tamaño da xanela e pérdeo ao reducir o tamaño da mesma.

Este tipo de layout manager é amplamente empregado para deseñar aplicacións con zonas cuxos tamaños non deben variar (zonas norte, sur, leste e/ou oeste) e nas cales adoitase colocar controis coas accións básicas a desenvolver e con unha zona central na cal desenvólvese realmente a funcionalidade da aplicación. Esta zona central é a que gañará ou perderá espazo en función do tamaño que tome o formulario. Un exemplo desta funcionalidade pódémolo ver no seguinte formulario:



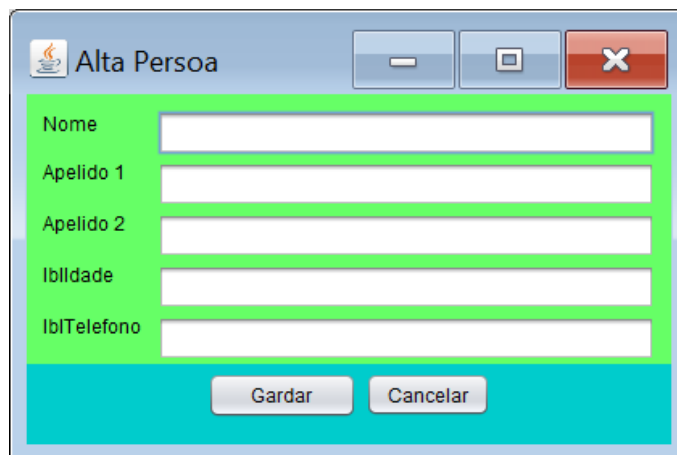
A small Java Swing window titled "Alta Persoa" with a standard Mac OS X title bar (red, yellow, green buttons). The window contains five text input fields labeled "Nome", "Apelido 1", "Apelido 2", "Ibldade", and "IbTelefono". At the bottom are two buttons: "Gardar" and "Cancelar".

Se modificamos o tamaño deste formulario podemos ver que sucede o seguinte:



The same "Alta Persoa" window, but resized to a larger size. The layout of the form fields and buttons remains the same, but the overall dimensions are increased, showing how the components are distributed within the larger frame.

Podemos observar que o formulario a efectos visuais está realmente dividido en dúas partes: a botoeira (zona sur) e a zona das caixas de texto (zona central). Co fin de visualizar dun xeito máis intuitivo as diferentes zonas configuradas no layout manager BorderLayout imos colorear os paneis, de xeito que o resultado sería o seguinte:



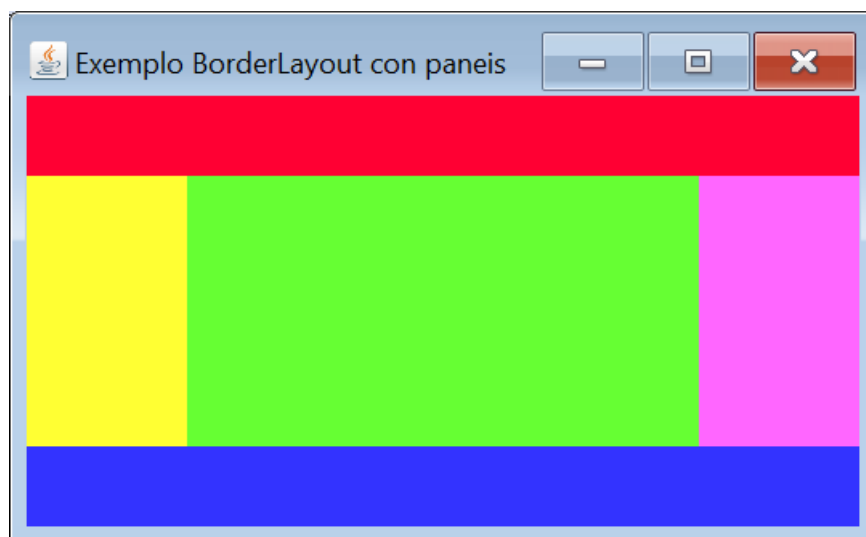
The "Alta Persoa" window with its components highlighted by background colors to represent BorderLayout regions. The area containing the text input fields is highlighted in green, representing the center (Zona Central). The area containing the "Gardar" and "Cancelar" buttons is highlighted in blue, representing the south (Zona Sur). The title bar and window controls remain in their original colors.

No noso layout manager BorderLayout temos definida unha zona sur (panel azul) e unha zona central (panel verde). Se modificamos o tamaño do formulario este é o resultado:

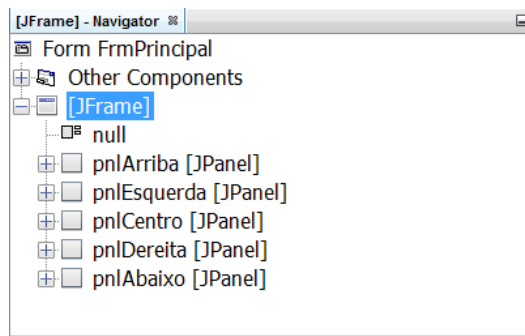
Como pódese observar, ao modificar o tamaño do formulario, a zona sur amplía o seu tamaño horizontal pero mantén o seu tamaño vertical. Non obstante, a zona central amplía o seu tamaño horizontal e vertical co fin de gañar o espazo xerado coa ampliación do formulario. O resultado final é que independentemente do tamaño do formulario, a botoeira sempre estará na parte inferior (sendo o seu tamaño vertical sempre o mesmo), mentres que a zona de toma de datos do usuario axustarase dinamicamente ao tamaño do formulario.

Configuración dun BorderLayout empregando NetBeans

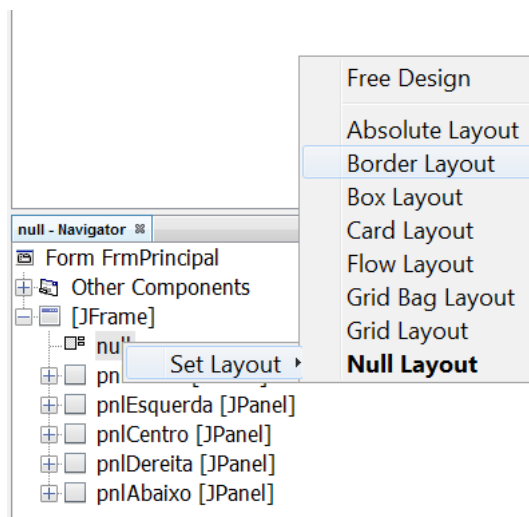
A continuación imos desenvolver o seguinte formulario:



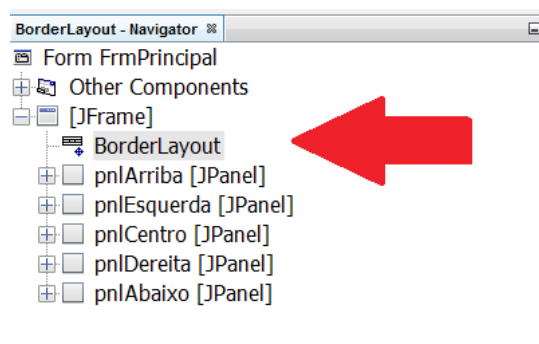
Este formulario emprega un layout manager de tipo BorderLayout o cal ten establecidas cinco zonas. Para empezar creamos un proxecto e engadimos nel un formulario. Dentro do noso formulario engadimos cinco paneis (pnlArriba, pnlEsquerda, pnlCentro, pnlDereita e pnlAbaixo). A pnlArriba dámoslle como cor de fondo vermello (propiedade background do panel), a pnlEsquerda amarelo, a pnlCentro verde, a pnlDereita rosa e a pnlAbaixo azul:



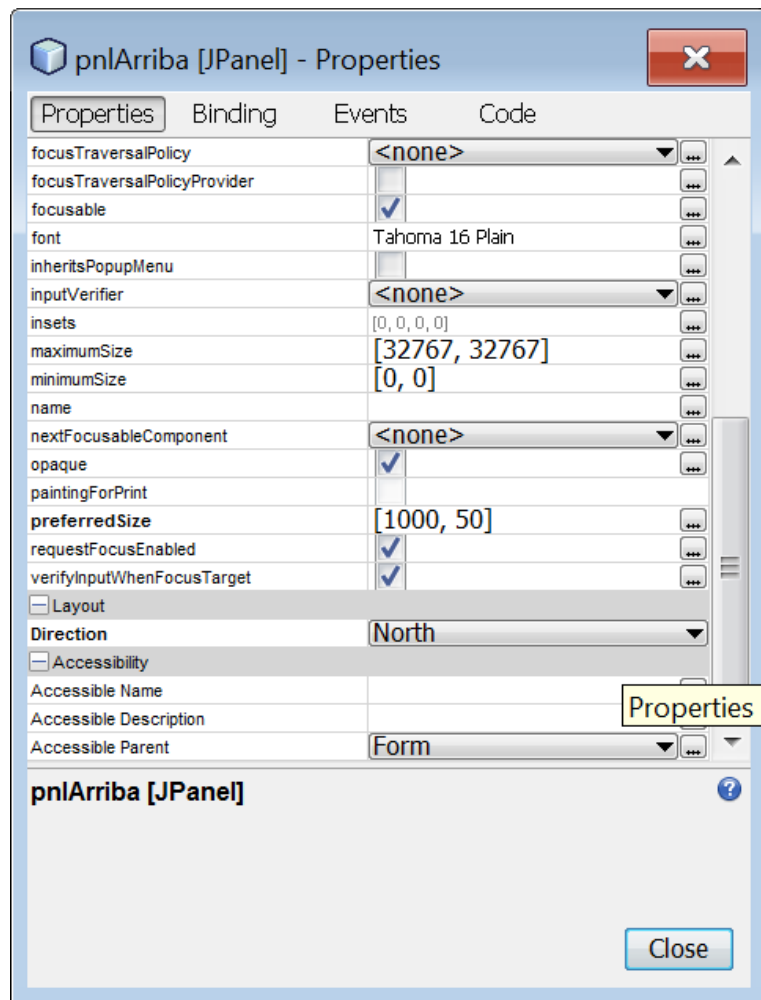
O seguinte que debemos facer é indicar que queremos establecer un layout manager de tipo BorderLayout sobre o noso formulario. Para isto prememos sobre o JFrame co botón dereito, eliximos a opción Set Layout e seleccionamos Border Layout:



A partir deste momento teremos establecido un layout manager de tipo BorderLayout para o noso formulario:



O seguinte paso é configurar en que zona do BorderLayout queremos colocar cada panel. Para situar o panel pnlArriba na zona norte do BorderLayout accedemos ás propiedades do panel pnlArriba e á propiedade Direction dámoslle o valor North:



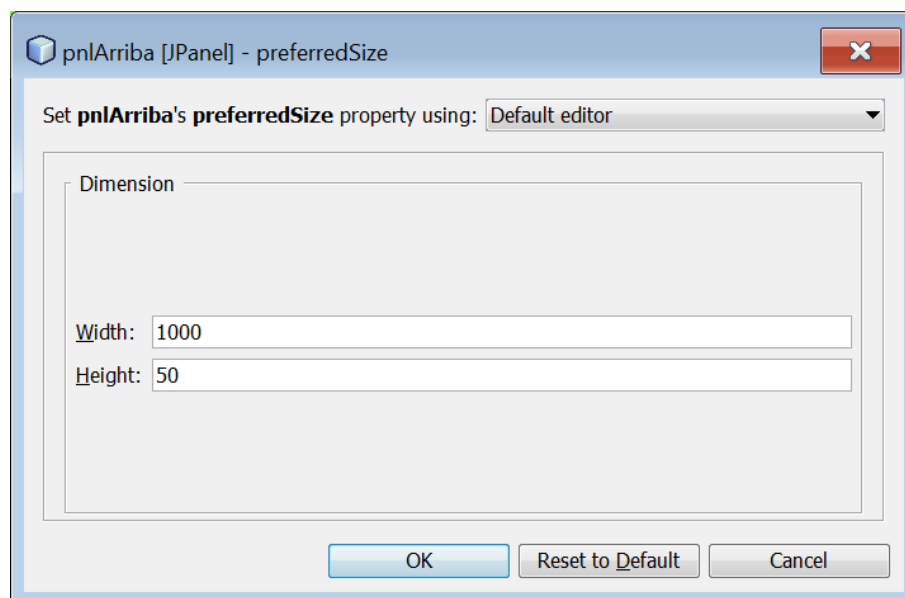
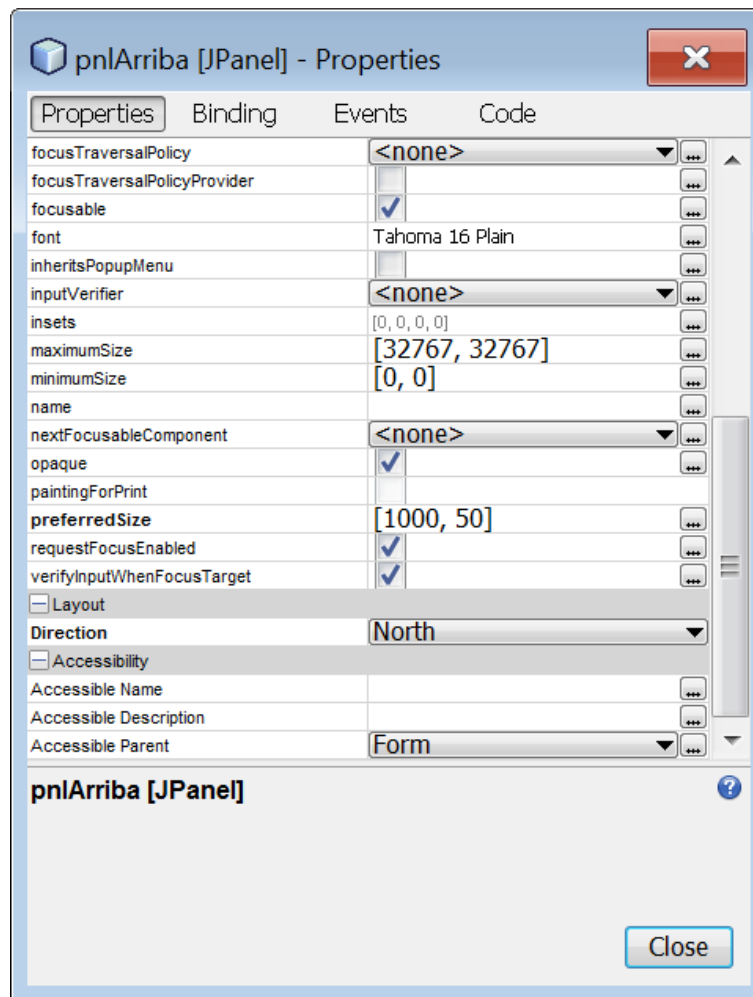
Facemos o mesmo para o resto dos paneis, pero variando o valor da propiedade Direction:

- Ao panel pnlEsquerda dámoslle o valor West
- Ao panel pnlCentro dámoslle o valor Center.
- Ao panel pnlDereita dámoslle o valor East.
- Ao panel pnlAbaixo dámoslle o valor South.

Por último debemos establecer o tamaño que queremos darlle a cada un dos paneis. O primeiro que temos que facer é ver o que queremos que ocorra ao redimensionar o formulario. Neste caso queremos o seguinte:

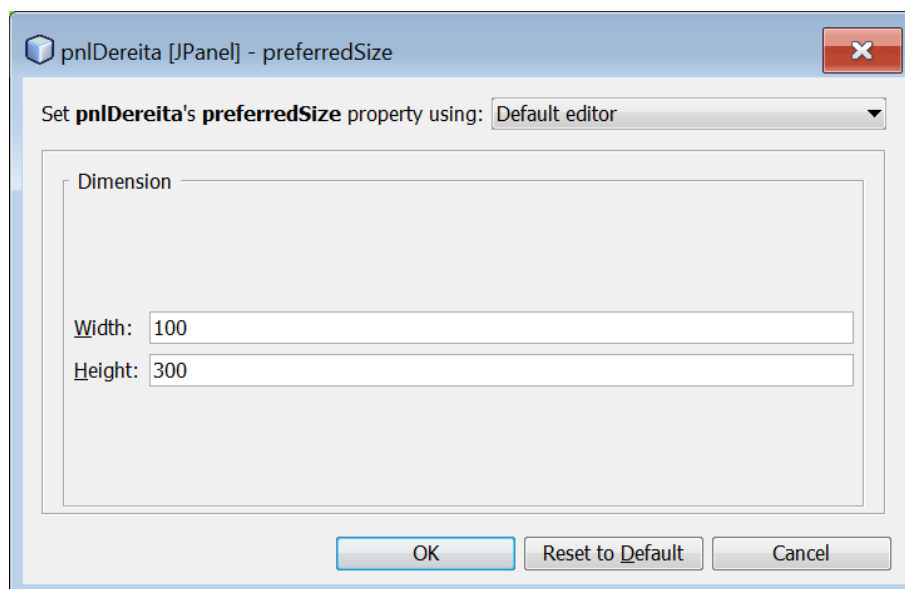
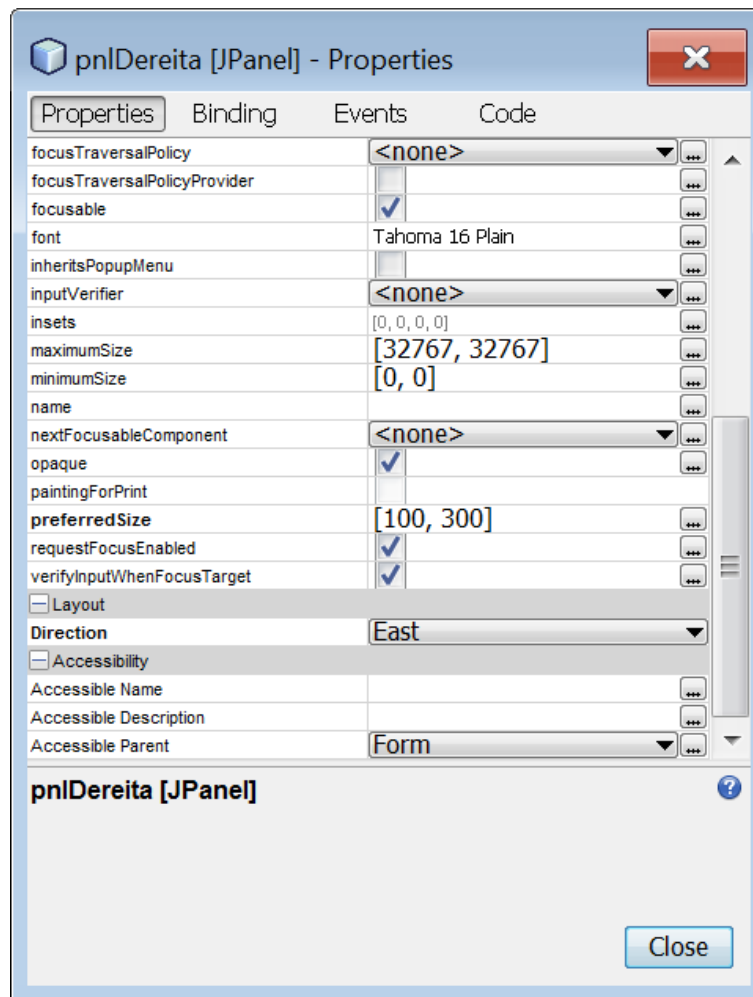
- Paneis pnlArriba e pnlAbaixo: queremos que a súa altura sexa sempre de 50 píxels
- Paneis pnlDereita e pnlEsquerda: queremos que o seu ancho sexa sempre de 100 píxels
- Panel pnlCentro: tomará o espazo sobrance ao redimensionar o formulario.

Para establecer que a altura do pnlArriba sexa sempre de 50 píxels facemos o seguinte: accedemos ás propiedades do panel pnlArriba e á propiedade preferredSize asignámoslle 50 no compoñente Height:



O mesmo faremos para o panel pnlAbaixo.

Para establecer que o ancho do panel pnlDereita sexa sempre de 100 píxels facemos o seguinte: accedemos ás propiedades do panel pnlDereita e á propiedade preferredSize asígnámoslle 100 no compoñente Width:



O mesmo faremos para o panel pnlEsquerda.

Respecto ao panel pnlCentro, non é preciso facer nada en especial xa que ao ter establecida a súa propiedade Direction como Center, ao redimensionar o formulario, o panel automaticamente tomará o espazo restante tanto ao ancho como ao alto.