

## EJERCICIO GUIADO. JAVA: LAYOUTS

### El problema de la distribución de elementos en las ventanas

Uno de los problemas que más quebraderos de cabeza da al programador es el diseño de las ventanas y la situación de los distintos componentes en ellas.

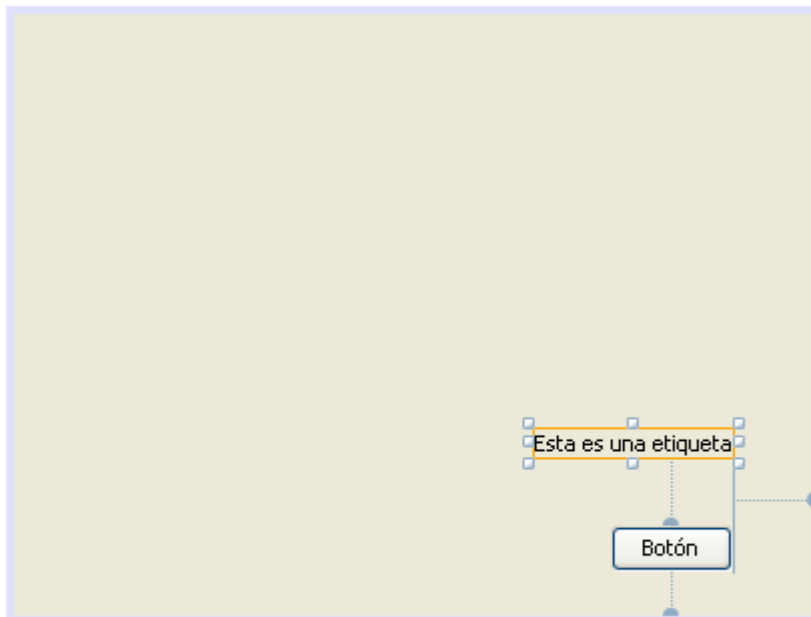
Para diseñar más cómodamente las ventanas, Java proporciona una serie de objetos denominados Layouts, que definen la forma que tendrán los elementos de situarse en las ventanas.

Así pues, un Layout define de qué forma se colocarán las etiquetas, botones, cuadros de textos y demás componentes en la ventana que diseñamos.

### Ejercicio guiado

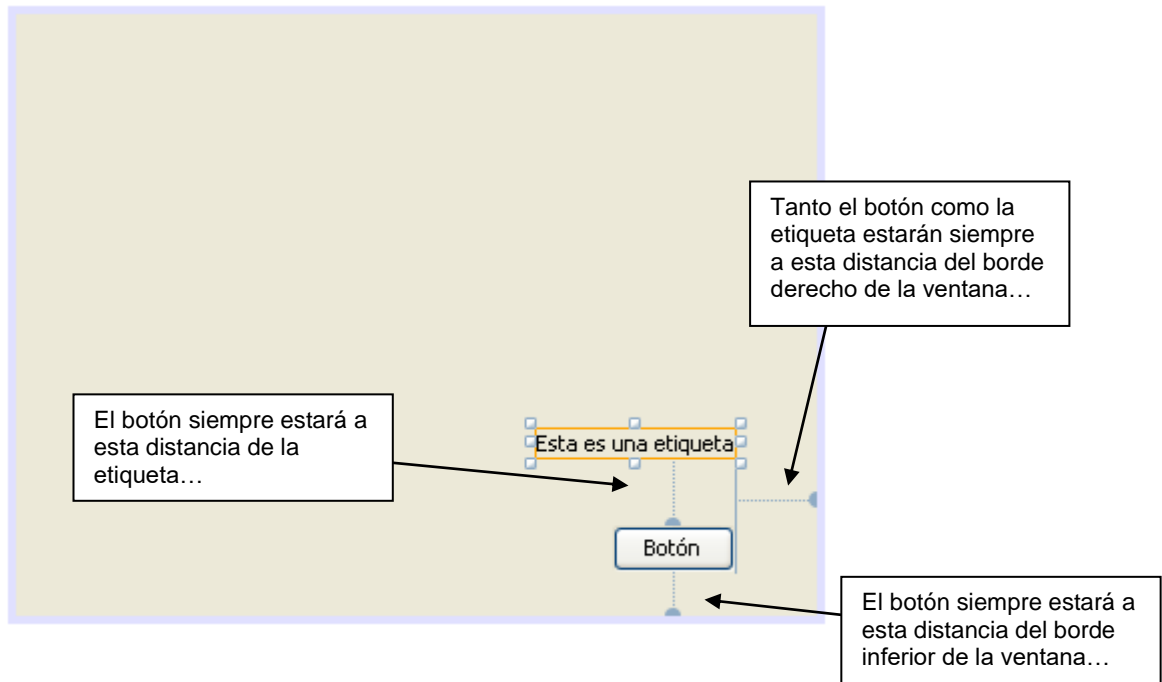
#### “Diseño Libre”

1. Crea un nuevo proyecto en java.
2. Añade una etiqueta y un botón. Muévelos a la posición que se indica en la imagen. Deben aparecer las líneas “guía” de color azul que se muestran:

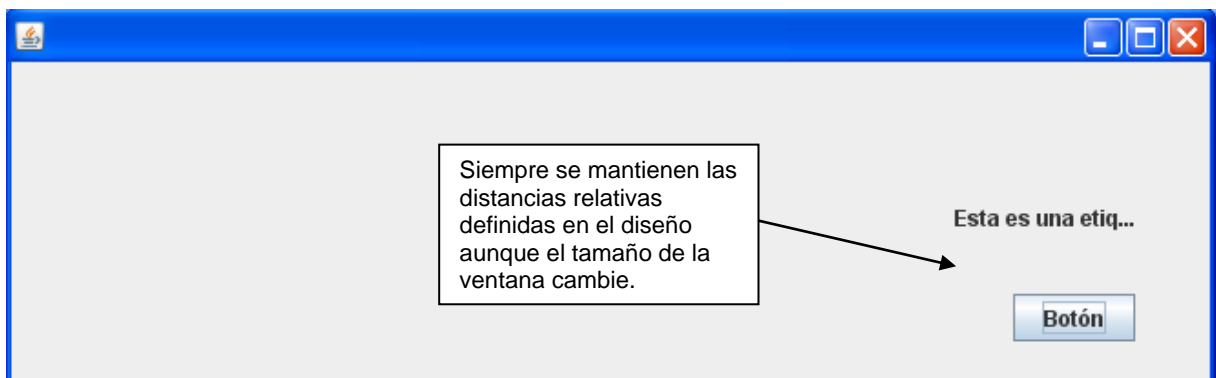


3. Las líneas azules que aparecen indican con qué otro elemento está relacionado un componente de la ventana. La situación de un elemento dependerá siempre de la situación del otro.

Dicho de otra forma, las líneas azules indican las distancias que siempre se respetarán. Observa la siguiente imagen:



4. Ejecuta el programa y prueba a ensanchar (o achicar) la ventana por el lado derecho y por el lado inferior. Debes observar como la etiqueta y el botón mantienen sus distancias relativas entre sí y con los bordes derecho e inferior de la ventana.



5. Este comportamiento de los elementos en la ventana viene dado por una opción del NetBeans llamada *Diseño Libre* (Free Design)

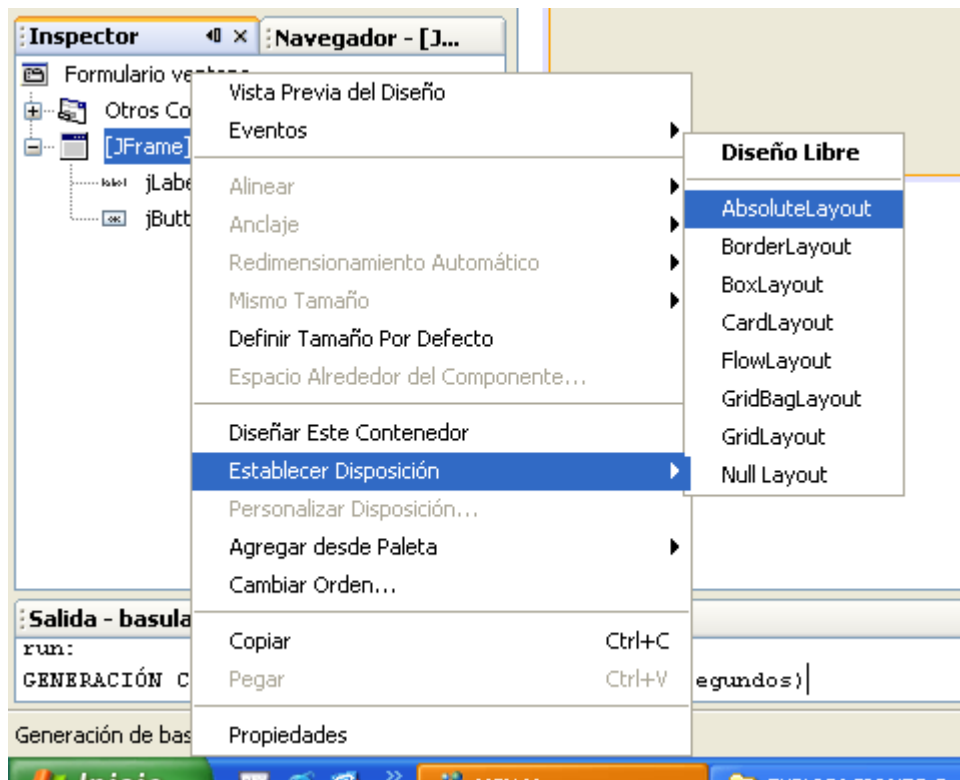
El Diseño Libre permite que los elementos de una ventana mantengan una distribución relativa da igual el tamaño que tenga la ventana. Dicho de otra forma, los elementos se redistribuyen al cambiar el tamaño de la ventana.

El problema del Diseño Libre es el poco control que se tiene sobre los elementos que se añaden a la ventana.

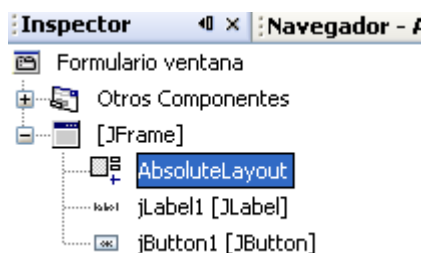
Se puede observar como a veces los elementos no se colocan en la posición que deseamos o como cambian de tamaño de forma inesperada. Todo esto es debido a la necesidad de dichos elementos de mantener unas distancias relativas con otros elementos de la ventana. Cuantos más elementos tengamos en una ventana, más difícil será el colocarlos usando el Diseño Libre.

## “AbsoluteLayout. Posiciones Absolutas”

6. El Diseño Libre es la opción que está activada por defecto cuando se crea un proyecto en NetBeans. Sin embargo, esta opción se puede cambiar por distintos “Layouts” o “Distribuciones”.
7. En el *Inspector* de tu proyecto pulsa el botón derecho del ratón sobre el objeto JFrame y activa la opción *Establecer Disposición – AbsoluteLayout*.



8. El *Inspector* tendrá la siguiente forma ahora:



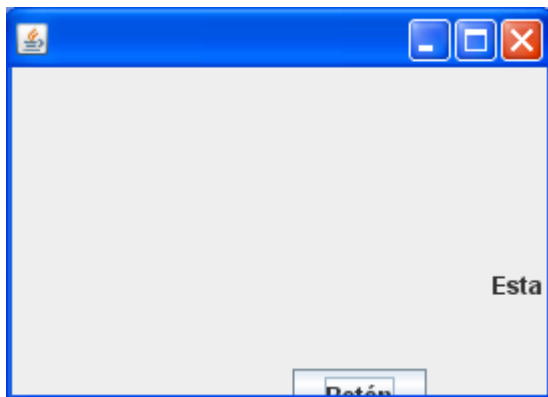
Como ves, aparece un objeto dentro del JFrame llamado AbsoluteLayout. Este objeto define otra forma de situar los elementos en la ventana. Concretamente, la distribución AbsoluteLayout permite al programador colocar cada elemento donde él quiera, sin restricciones, sin tener en cuenta distancias relativas.

9. Sitúa la etiqueta y el botón donde quieras. Observa que no aparece ninguna línea guía que defina distancias relativas:



10. La ventana de definir una distribución `AbsoluteLayout` es la facilidad para colocar cada elemento en la ventana (no tendrás los problemas del Diseño Libre). Sin embargo, la desventaja es que los elementos no mantienen una distribución relativa respecto al tamaño de la ventana.

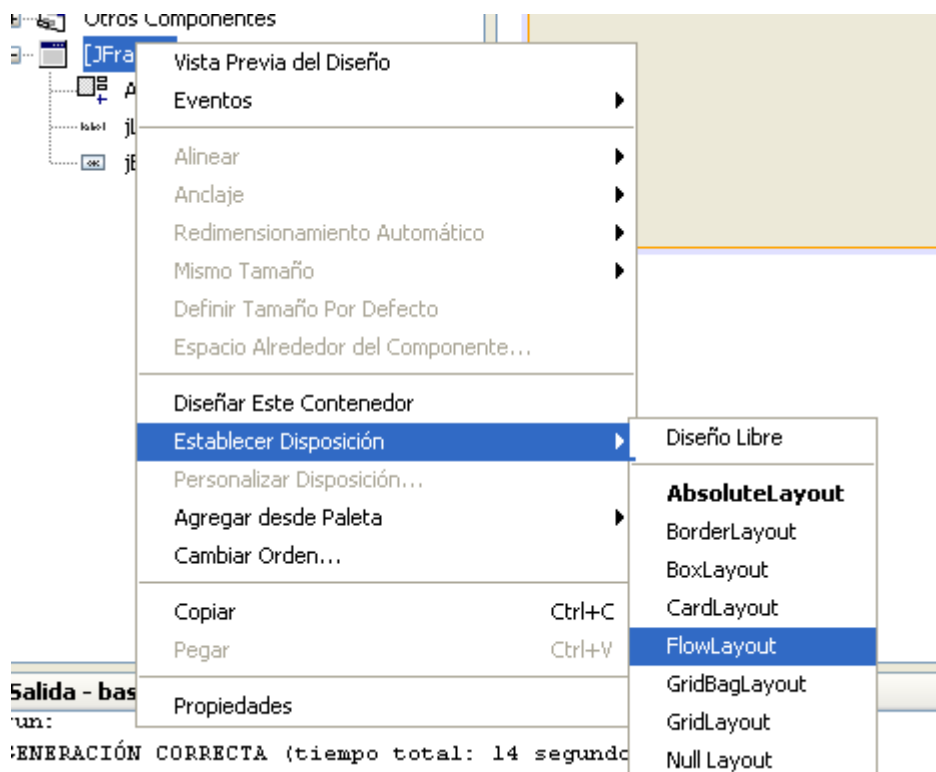
11. Ejecuta el programa y reduce su ancho. Observa lo que ocurre:



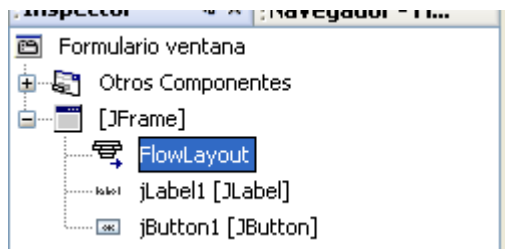
Verás que los elementos de la ventana son inamovibles aunque la ventana cambie de tamaño. En cambio, en el Diseño Libre los elementos intentaban siempre estar dentro de la ventana.

### “Distribución en línea. `FlowLayout`”

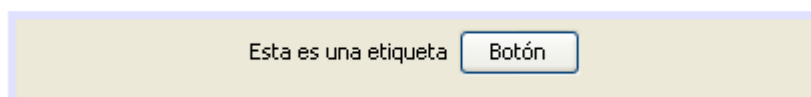
12. Practiquemos ahora con otro tipo de distribución. Accede al *Inspector* y pulsa el botón derecho del ratón sobre el objeto `JFrame`. Activa la opción *Establecer Disposición – FlowLayout*.



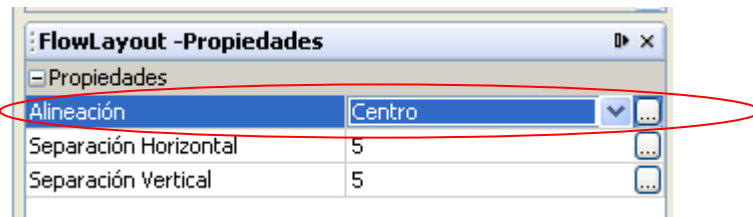
13. Observa como el layout “AbsoluteLayout” es sustituido por la distribución “FlowLayout”. Una elemento solo puede tener un tipo de distribución a la vez.



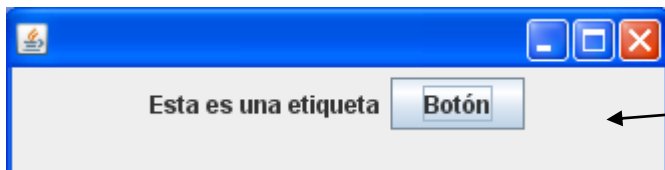
14. Observa la ventana. Los elementos se han colocado uno detrás de otro. Se han colocado “en línea”. Esto es lo que hace el “FlowLayout”. Fuerza a los distintos elementos a que se coloquen en fila.



15. Si seleccionas el FlowLayout en el *Inspector*, podrás acceder a sus propiedades (los layout son objetos como los demás) Una de las propiedades del FlowLayout se llama *alineación* y permite que los elementos estén alineados a la izquierda, derecha o centro. El FlowLayout tiene una alineación centro por defecto.



16. El FlowLayout no permite controlar la posición de los elementos en la ventana, pero sí procura que los elementos estén siempre visibles aunque la ventana se cambie de tamaño. Ejecuta el programa y observa el comportamiento del FlowLayout al agrandar o achicar la ventana:

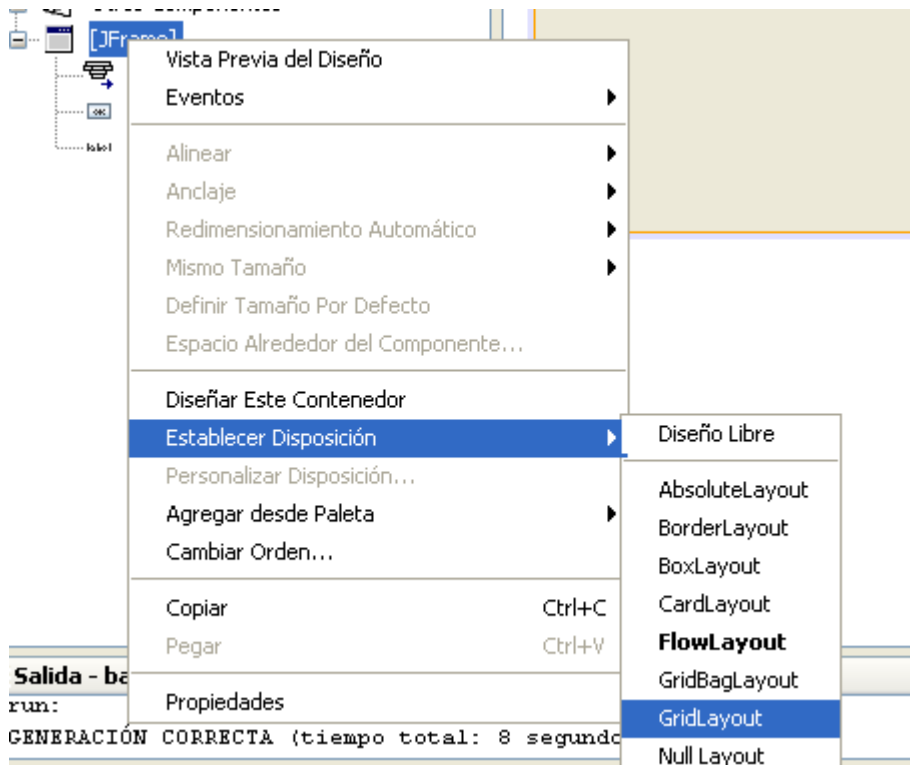


En el FlowLayout, los elementos intentan siempre estar dentro de la ventana, aunque esta se cambie de tamaño...

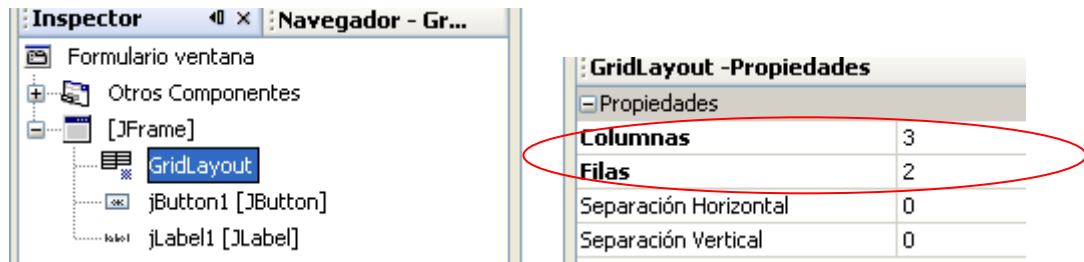
### “Distribución en rejilla. GridLayout”

17. Otra distribución que se puede usar es la distribución GridLayout. Esta distribución coloca a los elementos en filas y columnas, como si formaran parte de una tabla. Al añadir esta distribución es necesario indicar cuantas filas y columnas tendrá la rejilla.

18. Cambia el layout del JFrame por un GridLayout:



19. Marca el GridLayout y cambia sus propiedades Filas y Columnas. Asigna a la propiedad Filas un 2 y a la propiedad Columnas un 3.



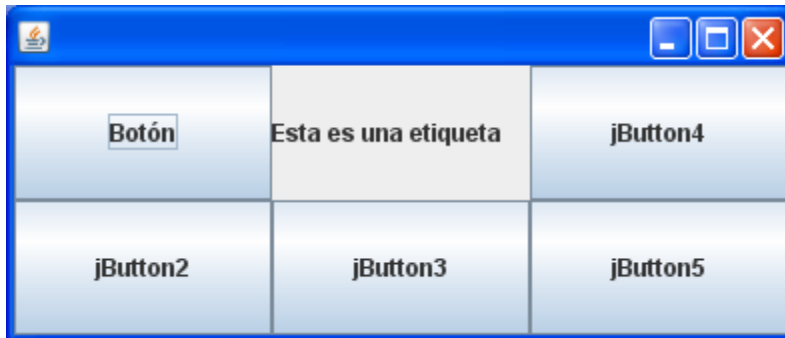
20. Al asignar 2 filas y 3 columnas al GridLayout, conseguiremos que los elementos de la ventana se distribuyan en una tabla como la siguiente:


Los distintos elementos se adaptarán al espacio que tienen asignado, y cambiarán de tamaño.

21. Ya que solo tienes dos elementos en la ventana (una etiqueta y un botón), añade otros cuatro elementos más (cuatro botones) para observar como se distribuyen en la cuadrícula.



22. En un GridLayout, los elementos estarán situados siempre en una casilla de la rejilla, ocupando todo su espacio. El programador no tiene mucho control sobre la disposición de los elementos.
23. Ejecuta el programa y agranda y achica la ventana. Observa como los elementos siempre mantienen su disposición en rejilla y siempre aparecen dentro de la ventana aunque el tamaño de esta varíe.



Con un GridLayout los elementos aparecen en filas y columnas.

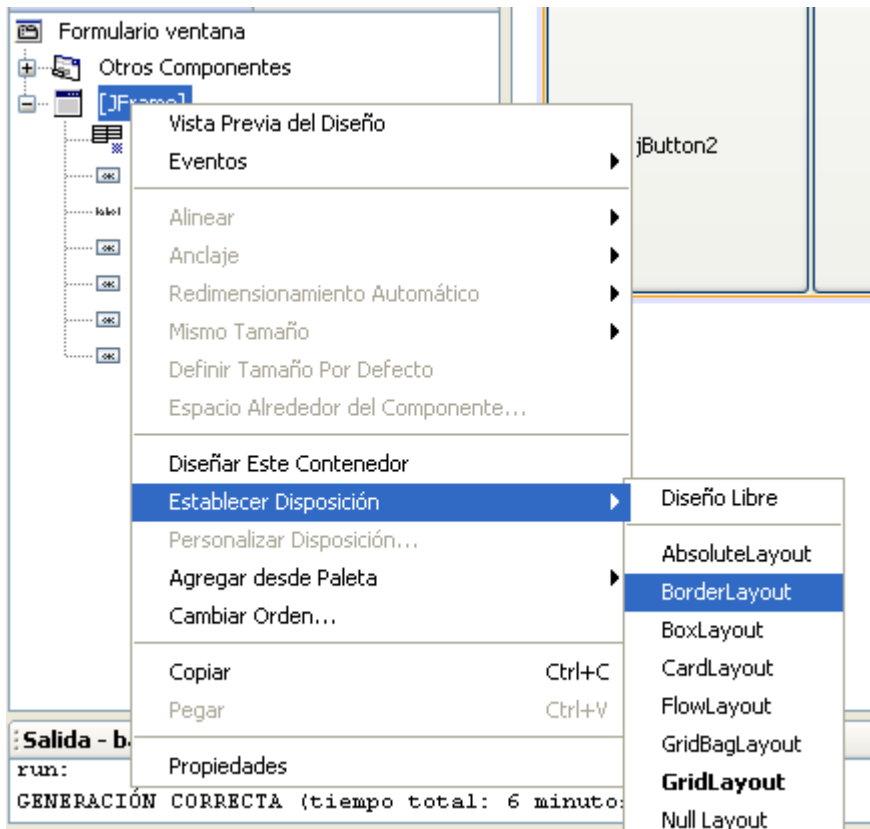
Siempre aparecen dentro de la ventana aunque el tamaño de esta cambie.

## “BorderLayout”

24. Otra de las distribuciones posibles es la llamada BorderLayout. Esta distribución coloca los elementos de la ventana en cinco zonas:

- Zona norte (parte superior de la ventana)
- Zona sur (parte inferior de la ventana)
- Zona este (parte derecha de la ventana)
- Zona oeste (parte izquierda de la ventana)
- Zona centro.

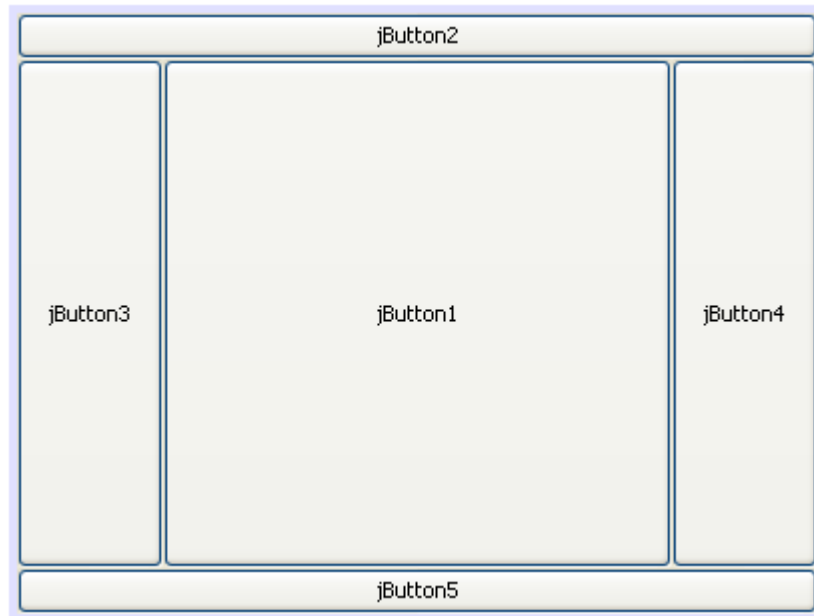
25. Haz clic con el derecho sobre el JFrame y asigna una distribución “BorderLayout”.





26. Para poder entender el funcionamiento del BorderLayout, se recomienda que el JFrame contenga únicamente 5 botones (elimine los elementos que tiene ahora y añada cinco botones)

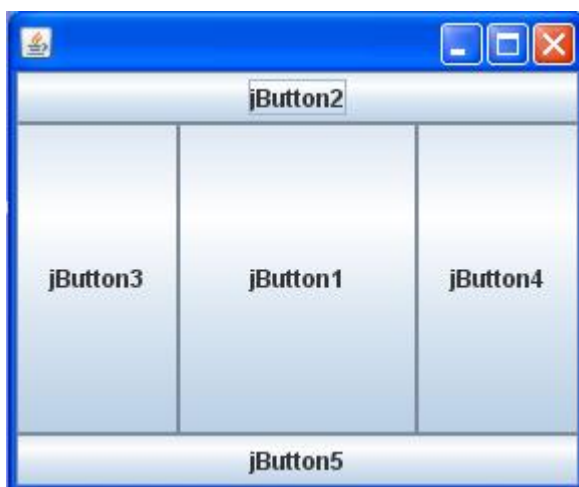
La ventana tendrá un aspecto parecido al siguiente:



27. Como se puede observar, cada botón se ha colocado en una zona, y su tamaño ha variado hasta ocupar la zona entera. Tenemos un botón en el norte, otro al sur, uno al este, otro al oeste y uno en el centro.

El programador no tiene mucho control sobre la disposición de los elementos en la ventana al usar esta distribución.

28. Ejecuta el programa y observa como los elementos siempre se mantienen dentro de la ventana aunque esta cambie de tamaño.



Con un GridLayout los elementos aparecen zonas.

Siempre aparecen dentro de la ventana aunque el tamaño de esta cambie.

## **CONCLUSIÓN**

El diseño de la ventana viene definido por los Layouts o distribuciones.

**Diseño Libre** – Esta distribución viene activada por defecto en el NetBeans, y define una distribución de componentes en la que se respetan las distancias entre ellos cuando la ventana cambia de tamaño.

**AbsoluteLayout** – En esta distribución el programador puede colocar cada elemento en la posición que desee de la ventana. Los distintos elementos mantienen su posición aunque la ventana cambie de tamaño, lo que puede hacer que si la ventana se reduce de tamaño algunos elementos no se vean.

**FlowLayout** – En esta distribución los elementos se colocan uno detrás de otro. Los elementos intentarán estar dentro de la ventana aunque esta se reduzca de tamaño.

**GridLayout** – Esta distribución coloca a los elementos en filas y columnas. Los elementos siempre estarán dentro de la ventana aunque esta se reduzca de tamaño.

**BorderLayout** – Esta distribución coloca a los elementos en zonas. Los elementos siempre estarán dentro de la ventana aunque esta se reduzca de tamaño.