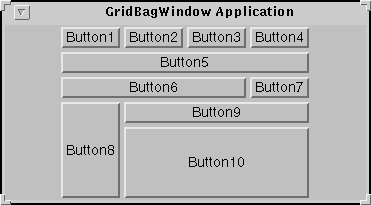
**Investigación:** GridbagLayout

**Unidad de Aprendizaje:** Programación Avanzada

**Integrantes:**

* Sales Escamilla Diana Angélica
* Guzmán Catorce Edgar
* Mares Tamara
* Santiago Pérez Mónica
* Hernández Pérez Alejandro

Es igual que la composición de GridLayout, con la diferencia que los Componentes no necesitan tener el mismo tamaño. Es quizá el layout más sofisticado y versátil de los que actualmente soporta AWT.

En la figura siguiente vemos la imagen generada por un panel que usa el GridBagLayout para soportar diez botones en su interior:

El GridBagLayout es uno de los Layout de java más potentes, pero también es el que más cuesta entender.

Vamos a construir una ventana en la que situamos determinados componentes (botones) en determinadas posiciones usando el GridBagLayout. Puedes ver los fuentes de ejemplo y applets funcionando pinchando en algunas de las imágenes que se ven durante la explicación.

**Concepto de layout**

A la hora de hacer "ventanas" en java, un componente es cualquiera de los botones, etiquetas, listas, menús, etc que podemos colocar en la ventana. Hay determinados componentes que no tienen un aspecto claro y que únicamente están destinados a contener a los demás y se les llama contenedores. Por ejemplo, si una ventana contiene dos botones, la ventana en si misma en otro componente java que recibe el nombre de contenedor. Otro ejemplo, si te fijas en la barra de herramientas de tu navegador (echa el ojo un poco para arriba), verás que está llena de botones-iconos. La barra de herramientas en sí misma es otro contenedor (aunque posiblemente no sea de java, sino del internet explorer o netscape).

**Hacer una rejilla adecuada**

El primer paso es dibujar la ventana que queremos (a los más espabilados les bastará con hacerse una idea mental). Es importante dibujarla más o menos estirada, para tener claro si los botones deben hacerse grandes o no. Por ejemplo, en la figura anterior, los botones deben mantener su tamaño aunque estiremos la ventana, sin embargo el area de texto debe crecer con la ventaa y el campo de texto sólo debe crecer horizontalmente.

Luego, sobre el dibujo, hay que trazar unas lineas horizontales y verticales, para tratar de delimitar una rejilla en la que iran colocados los botones. Para trazar esta rejilla y que luego el código no nos salga muy complicado conviene tener en cuenta lo siguiente:

* Debemos tratar de meter los componentes en las celdas. Cada componente debe ocupar una o más celdas. Dos componentes no pueden ocupar la misma celda.
* No es necesario que las celdas de la rejilla sean todas del mismo tamaño ni es necesario que un componente ocupe una celda completa.
* Para que no se lie el código, conviene que el componente ocupe toda la celda, o bien que esté centrado en la misma, o bien pegado al centro de uno de los bordes de la celda o a una esquina de la misma.

Aquí un ejemplo de las ventana creada por Layout:

package ejemplo1;

import javax.swing.\*;

import java.awt.\*;

/\*\*

\* Ejemplo de uso del GridBagLayout. Los componentes salen apelotonados en el

\* centro de la ventana.

\*/

public class Ventana extends JFrame

{

/\*\*

\* Constructor por defecto.

\*/

public Ventana()

{

super ("Ejemplo 1 GridBagLayout");

this.getContentPane().setLayout(new GridBagLayout());

GridBagConstraints constraints = new GridBagConstraints();

// JTextArea en la fila 0, columna 0 y que ocupa dos filas y dos

// columnas de ancho.

JTextArea cajaTexto = new JTextArea("Area texto");

constraints.gridx = 0;

constraints.gridy = 0;

constraints.gridwidth = 2;

constraints.gridheight = 2;

this.getContentPane().add (cajaTexto, constraints);

// JButton en la fila 0, columna 2 y que ocupa 1 celda de ancho por

// una de alto.

JButton boton1 = new JButton ("Boton 1");

constraints.gridx = 2;

constraints.gridy = 0;

constraints.gridwidth = 1;

constraints.gridheight = 1;

this.getContentPane().add (boton1, constraints);

// JButton en la fila 1, columna 2 y que ocupa 1x1 celdas.

JButton boton2 = new JButton ("Boton 2");

constraints.gridx = 2;

constraints.gridy = 1;

constraints.gridwidth = 1;

constraints.gridheight = 1;

this.getContentPane().add (boton2, constraints);

// JButton en la fila 2, columna 0 y que ocupa 1x1 celdas.

JButton boton3 = new JButton ("Boton 3");

constraints.gridx = 0;

constraints.gridy = 2;

constraints.gridwidth = 1;

constraints.gridheight = 1;

this.getContentPane().add (boton3, constraints);

// JButton en la fila 2, columna 2 y que ocupa 1x1 celdas.

JButton boton4 = new JButton ("Boton 4");

constraints.gridx = 2;

constraints.gridy = 2;

constraints.gridwidth = 1;

constraints.gridheight = 1;

this.getContentPane().add (boton4, constraints);

// JTextField en la fila 2, columna 1 y que ocupa 1x1 celdas.

JTextField campoTexto = new JTextField ("Campo texto");

constraints.gridx = 1;

constraints.gridy = 2;

constraints.gridwidth = 1;

constraints.gridheight = 1;

this.getContentPane().add (campoTexto, constraints);

}

}