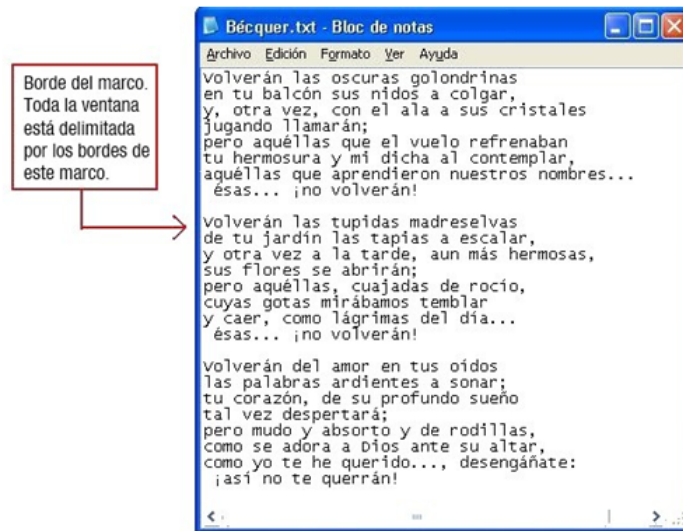


## 11.C. Generando programas en entorno gráfico.

### 1. Generación de programas en entorno gráfico.

#### 1.1. Contenedores.

Por los ejemplos vistos hasta ahora, ya te habrás dado cuenta de la necesidad de que cada aplicación "contenga" de alguna forma esos componentes. ¿Qué componentes se usan para contener a los demás?



En Swing esa función la desempeñan un grupo de componentes llamados **contenedores** Swing.

Existen dos tipos de elementos contenedores:

- **Contenedores de alto nivel** o "peso pesado".
  - Marcos: `JFrame` y `JDialog` para aplicaciones
  - `JApplet`, para applets.
- **Contenedores de bajo nivel** o "peso ligero". Son los paneles: `JRootPane` y `JPanel`.

Cualquier aplicación, con interfaz gráfico de usuario típica, comienza con la apertura de una ventana principal, que suele contener la barra de título, los botones de minimizar, maximizar/restaurar y cerrar, y unos bordes que delimitan su tamaño.

Esa ventana constituye un marco dentro del cual se van colocando el resto de componentes que necesita el programador: menú, barras de herramientas, barra de estado, botones, casillas de verificación, cuadros de texto, etc.

Esa ventana principal o marco sería el contenedor de alto nivel de la aplicación.

Toda aplicación de interfaz gráfica de usuario Java tiene, al menos, un contenedor de alto nivel.

Los contenedores de alto nivel extienden directamente a una clase similar de AWT, es decir, `JFrame` extiende de `Frame`. Es decir, realmente necesitan crear una ventana del sistema operativo independiente para cada uno de ellos.

Los demás componentes de la aplicación no tienen su propia ventana del sistema operativo, sino que se dibujan en su objeto contenedor.

En los ejemplos anteriores del tema, hemos visto que podemos añadir un `JFrame` desde el diseñador de NetBeans o el WindowBuilder de Eclipse, o bien escribiéndolo directamente por código. De igual forma para los componentes que añadamos sobre él.

#### Para saber más

Te recomendamos que mires la siguiente web para ver información sobre `JFrame` y `JDialog`.

[JFrame y JDialog](#)

### Autoevaluación

Cualquier componente de gráfico de una aplicación Java necesita tener su propia ventana del sistema operativo.

- ☐ Verdadero.
- ☐ Falso.