

**UNIVERSIDAD CATOLICA DE COLOMBIA  
FACULTAD DE INGENIERIA DE SISTEMAS**

**CURSO:** JAVA BASICO  
**PROFESOR:** EMERSON CASTAÑEDA SANABRIA

**TEMA:** AWT

**OBJETIVOS:**

- Aprender la forma de organización del paquete java.awt.
- Presentar las clases del paquete java.awt.
- Comprender los mecanismos para realizar gestión de eventos.
- Manejar los elementos para la organización de ventanas.

**CONTENIDO:**

1. Que es AWT?
2. Contenedores y Componentes
3. Organización de Ventanas (Layouts)
4. Gestión de eventos

**DESARROLLO:**

1. Que es AWT?

El AWT (Abstract Windowing Toolkit) proporciona un gran número de clases que soportan el desarrollo de programas de ventanas. Estas clases se emplean para crear y organizar ventanas, implementar componentes de GUI, gestionar eventos, dibujar texto y gráficos, procesar imágenes y obtener acceso a la implementación nativa de ventanas.

2. Contenedores y Componentes

La clase Component

La clase Component es la clase superior del conjunto de clases del AWT que implementa los controles de la interfaz grafica de usuario. Estos componentes incluyen ventanas, cuadros de dialogo, botones, etiquetas campos de texto y otros componentes habituales de una GUI.

Component proporciona un conjunto de métodos que son empleados por todas sus subclases. Dichos métodos incluyen métodos para gestionar eventos y trabajar con imágenes, fuentes y colores. Esta clase implementa más de 70 métodos. Es aconsejable consultar la documentación correspondiente a la clase Component para familiarizarse con los tipos de métodos disponibles.

La clase Container

La clase Container es una subclase de Component que se utiliza para definir componentes que pueden contener otros componentes. Proporciona métodos para agregar, recuperar, visualizar, contar y eliminar los componentes que contiene. Incorpora el método deliverEvent() para dirigir eventos hacia sus componentes. Asimismo la clase Container proporciona métodos para trabajar con formatos (Layouts).

La clase Container posee dos subclases principales: Window y Panel. Window ofrece una clase superior común para las ventanas principales de aplicaciones (objetos Frame y Dialog). La clase Panel es un contenedor genérico que puede visualizarse dentro de una ventana. Una de sus subclases es la clase Applet, que es la clase básica para todas las miniaplicaciones java.

### La clase Window

La clase Window proporciona el encapsulamiento necesario para un objeto ventana genérico. Sus subclases son Frame y Dialog, que conoce la capacidad necesaria para soportar las ventanas principales y cuadros de dialogo de una aplicación.

La clase Window contiene un único constructor que crea una ventana cuyo padre es la ventana Frame. La ventana Frame es necesaria por que únicamente los objetos de la clase Frame cuentan con la capacidad funcional precisa para soportar la implementación de una ventana de aplicación independiente. La clase Window implementa métodos importantes que son utilizados por sus subclases. El método pack() se usa para organizar los componentes de una ventana de acuerdo con el estilo de formato de la misma(layout). El método show() se emplea para presentar una ventana. Por defecto las ventanas están ocultas y se muestran solamente al llamar su método show(). Los métodos toFront() y toBack() se utilizan para ubicar las ventanas con relación a otras ventanas. El método dispose() se emplea para liberar los recursos asociados a una ventana y para borrar el objeto Window.

En el momento de crearse una ventana no posee borde ni tampoco barra de menús. En este estado, puede emplearse para implementar una ventana emergente. El formato por defecto para un objeto Window es BorderLayout.

### La clase Frame

La clase Frame se utiliza para proporcionar la ventana principal de una aplicación. Es una subclase de Window que soporta la capacidad de especificar el icono, cursor, barra de menús y titulo. Dado que implementa la interfaz MenuContainer, es capaz de trabajar con objetos MenuBar.

La clase Frame define 14 constantes que emplean para especificar los diversos tipos de cursor que pueden utilizarse dentro del marco. Frame proporciona dos constructores, un constructor predeterminado que no recibe parámetros y crea un objeto Frame sin titulo, y un constructor que acepta un argumento String que se usa como titulo de la ventana.

### La clase Dialog

Dialog es una subclase de la clase Window que se emplea para implementar ventanas de cuadros de dialogo. Un cuadro de dialogo es una ventana que acepta entradas por parte del usuario. La clase Dialog permite construir cuadro de dialogo de tipo modal y no modales, es decir que precisan cerrarse o no antes de devolver el control a las otras ventanas del programa.

La clase Dialog proporciona dos constructores, ambos requieren que se especifique el objeto Window que contiene el cuadro de dialogo, así como el indicador de modal. El segundo constructor permite especificar adicionalmente el titulo del cuadro de dialogo.

### La clase Panel

Panel es una subclase de la clase Container que se emplea para organizar componentes de la GUI dentro de otros objetos contenedores. Posee un único constructor que no recibe ningún parámetro.

### La clase Label

La clase Label se utiliza para representar etiquetas de texto de solo lectura dentro de una ventana u otro contenedor de la GUI. Posee tres constructores. El primero no emplea parámetros y se usa para construir una etiqueta vacía. El segundo emplea como parámetro un String que se usa para especificar el texto que contiene la etiqueta. El tercer constructor posee un parámetro de alineación además del texto. Este parámetro especifica como debe alinearse la etiqueta.

La clase Label proporciona métodos para obtener y definir la etiqueta presentada y su valor de alineación.

#### La clase Button

La clase Button implementa un botón que puede pulsarse con el ratón. El botón es capaz de mostrar una etiqueta de texto incorpora dos constructores. El primero de ellos no utiliza ningún parámetro y crea un botón con la etiqueta vacía. El segundo constructor acepta un objeto String que se muestra como la etiqueta del botón.

#### La clase Checkbox

Se utiliza para implementar controles de cuadro de verificación y de control radial de la GUI. El cuadro de verificación o botón radial está asociado a una etiqueta. Si en un objeto Checkbox no está asociado a un objeto CheckboxGroup se implementa como cuadro de verificación normal. Si lo está, se implementa como botón radial.

La clase Checkbox proporciona tres constructores. El primero no utiliza ningún parámetro e implementa un cuadro de verificación vacío. El segundo constructor emplea un parámetro String que se usa como título del cuadro de verificación. El tercero permite especificar un objeto CheckboxGroup y el estado inicial del botón radial, además de su etiqueta.

#### La clase CheckboxGroup

La clase CheckboxGroup se utiliza con la clase Checkbox a fin de implementar botones radiales. Todos los objetos Checkbox asociados a un objeto CheckboxGroup son tratados como un conjunto individual de botones radiales. Proporciona un único constructor sin parámetros, además de métodos para obtener y definir el objeto Checkbox.

#### La clase Choice

Se utiliza para implementar listas desplegables que pueden situarse en una ventana. Dichas listas se conocen con el nombre de menús de opciones o menús emergentes de opciones, que permiten al usuario seleccionar uno de los valores que incluyen. La clase Choice incorpora un único constructor sin parámetros. Asimismo, proporciona métodos de acceso que se emplean para agregar elementos a la lista, contar el número de elementos que contiene ésta, seleccionar un elemento y determinar qué elemento de la lista está seleccionado.

#### La clase List

La clase List implementa controles de la GUI de listas de selección individual y múltiple. Las listas que incorpora la clase List están más perfeccionadas que las proporcionadas por la clase Choice: la clase List ofrece la posibilidad de determinar el tamaño de la ventana desplazable en la que se presentan los elementos de la lista y seleccionar múltiples elementos de ésta. List posee dos constructores. El primero no utiliza parámetros y construye un objeto genérico List. El segundo permite especificar el número de filas de la ventana visible y si se permite o no efectuar múltiples selecciones.

Proporciona diversos métodos de acceso que se emplean para agregar, borrar y reemplazar elementos de la lista, contar el número de elementos que contiene, determinar qué elementos están seleccionados y seleccionar elementos de la lista.

#### La clase TextComponent

TextComponent es la clase superior de todas las clases basadas en texto. Proporciona un conjunto de métodos que emplea sus subclases TextField y TextArea. No proporciona ningún constructor y no es posible crear instancias de la misma. Facilita métodos para obtener y definir el texto que se muestra en el objeto, establece el estado de edición o de solo lectura, o selecciona el texto que contiene el objeto.

### La clase TextField

Implementa un campo de entrada de texto de una sola línea. Proporciona cuatro constructores que se emplean para especificar el ancho del cuadro de texto o el texto predeterminado que debe aparecer en el campo. Incorpora diversos métodos para acceder al tamaño del campo y para especificar si los caracteres introducidos por el usuario deben o no presentarse. El método `setEchoCharacter()` se utiliza para especificar un carácter que debe presentarse en lugar del texto introducido por el usuario. Este método se utiliza para implementar campos de contraseña.

### La clase TextArea

Implementa objetos de entrada de texto desplazables que ocupan múltiples líneas y columnas. Proporciona cuatro constructores que permiten especificar el número de filas y columnas y el texto que debe aparecer por defecto. Asimismo, posee diversos métodos que retornan las dimensiones del área de texto e insertan, anexan y remplazan el texto contenido en el área de texto.

### La clase Scrollbar

Se utiliza para implementar barras de desplazamiento horizontal y vertical. Ofrece tres constructores que permiten especificar la orientación de la barra, así como los parámetros que controlan su funcionamiento. Incorpora diversos métodos que permitan leer y definir los parámetros y el valor actual de la barra de desplazamiento.

## 3. Organización de Ventanas (Layouts)

El método según el cual se organizan los componentes dentro de un objeto `Container` depende de un objeto asociado que implemente la interfaz `LayoutManager`. El formato para un objeto `container` se especifica usando el método `setLayout()` de la clase `container`, que asocia al contenedor un objeto que implemente la interfaz `LayoutManager` como parámetro.

### La interfaz LayoutManager

`LayoutManager` proporciona un conjunto de métodos que implementan clases que controlan el formato de un contenedor. Estos métodos incluyen los métodos que agregan o suprimen los componentes de formato, especifican el tamaño del contenedor y determinan sus componentes.

### La clase BorderLayout

La clase `BorderLayout` se emplea para definir los componentes de la GUI que contienen un objeto `container`. Especifica la posición de los componentes según los bordes norte, sur, este y oeste del contenedor, y también según el centro de éste. La sección central aprovecha cualquier espacio no utilizado por los componentes de los bordes. Es el formato predeterminado de las clases `Window`, `Frame` y `Dialog`. Asimismo, ofrece la posibilidad de especificar el espacio horizontal y vertical entre los componentes situados en el contenedor.

### La clase CardLayout

Se utiliza para situar los componentes dentro de un objeto `container` en la forma de una baraja de cartas en la que solamente es visible una de ellas. Esta clase proporciona los métodos que se emplean para especificar los componentes inicial, último, siguiente y anterior.

### La clase FlowLayout

La clase `FlowLayout` se usa para situar los componentes dentro de un objeto `container` en una disposición de izquierda a derecha y de arriba hacia abajo. Es el formato predeterminado de la

clase Panel. Permita alinear los componentes de acuerdo a los valores especificados por las constantes. LEFT, CENTER y RIGHT.

#### La clase GridLayout

Se utiliza para disponer los objetos dentro de un objeto container en forma de cuadrícula donde todos los objetos tiene el mismo tamaño. En su constructor se especifica el numero de filas y el numero de columnas de la cuadrícula.

#### La clase GridBagLayout

Dispone los componentes dentro de un objeto container de la forma similar a una cuadrícula, donde algunos objetos pueden ocupar mas de una fila o columna. La clase GridBagConstraints se emplea para especificar los parámetros para la ubicación de un componente situado dentro de un contenedor con formato objeto GridBagLayout. La clase Insets se utiliza para especificar los márgenes asociados a un objeto dispuesto por el formato GridBagLayout.

#### 4. Gestión de eventos

El usuario se comunica con programas basados en ventanas con acciones como la pulsación de un botón del ratón o de una tecla. Estas acciones dan como resultado la generación de objetos Event. El proceso de respuesta ante un evento se conoce como gestión de eventos. Se dice que los programas basados en ventanas son dependientes de eventos porque funcionan realizando acciones como respuesta a determinados eventos.

La clase Event encapsula todo el procesamiento de eventos de Windows y es, por tanto, una clase muy importante. Como la interfaz de usuario de Windows es dependiente de Eventos, todos los programas basados en ventanas deben ser capaces de manejar eventos.

La clase Event define una lista completa de eventos gestionables por programas basados en ventanas que utilizan constantes de clases. Dichas constantes se emplean para identificar los eventos que se transfieren a los métodos de gestión de eventos.

Event proporciona cuatro constructores para construir eventos, aunque lo mas probable es que no se utilicen dado que el sistema en tiempo de ejecución de java los genera internamente como respuesta a las acciones de la interfaz de usuario. Asimismo, la clase Event proporciona métodos para averiguar si se pulsan las teclas <Ctr><Shif><Alt> durante la generación de un evento.