



Interfaz Gráfica de Usuario (GUI)

Unidad: 2
Laboratorio de Programación

Universidad Nacional de la Patagonia Austral
Unidad Académica Río Gallegos



Indice

- Generalidades
- Concepto GUI y AWT
- Estructura
- Jerarquía AWT
- Contenedores: Panel, Applet, ScrollPane, Window, Frame, Dialog
- Componentes: Label, Button, CheckBox y CheckBoxGroup, Choice, List, Scrollbar, TextField, TextArea



Generalidades

- La influencia reciente de las tecnologías de información en el comportamiento de las personas es cada vez más notoria, incluso ha traspasado fronteras internacionales e interculturales.
- Por ejemplo, en las aplicaciones de Internet (compras electrónicas, gestión de mensajes, etc.), se orienta buena parte de los esfuerzos de diseño e implementación a satisfacer las demandas potenciales de los usuarios en:
 - Usabilidad,
 - Comprensión,
 - Manipulación de las utilidades de las aplicaciones.
- Por estas razones es necesario realizar un buen diseño de interfaz con el fin de que los usuarios puedan cumplir su cometido.



Generalidades...

- Por muy buenos que hayan sido el análisis, el diseño y la implantación de cualquier aplicación, de nada servirá si los servicios que proporciona a los usuarios no se muestran de una forma **clara, sencilla y amigable**.
- La mayoría de los usuarios juzgan las aplicaciones por la primera impresión que les producen, siendo un factor importante la manera de interactuar con el programa.



Qué es una GUI?

- Una GUI es cualquier cosa que aparezca en una pantalla que le permita a un usuario hacer lo que tiene que hacer de forma gráfica.



AWT: Que es?

- AWT: Abstract Window Toolkit
- Librería que define un conjunto de clases e interfaces Java que permiten la construcción de interfaces gráficas de usuario.
- Es la librería básica y se ha construido una posterior que es la Swing.
- 2 elementos básicos
 - Contenedores (`java.awt.Container`)
 - Componentes (`java.awt.Component`)



AWT: Estructura

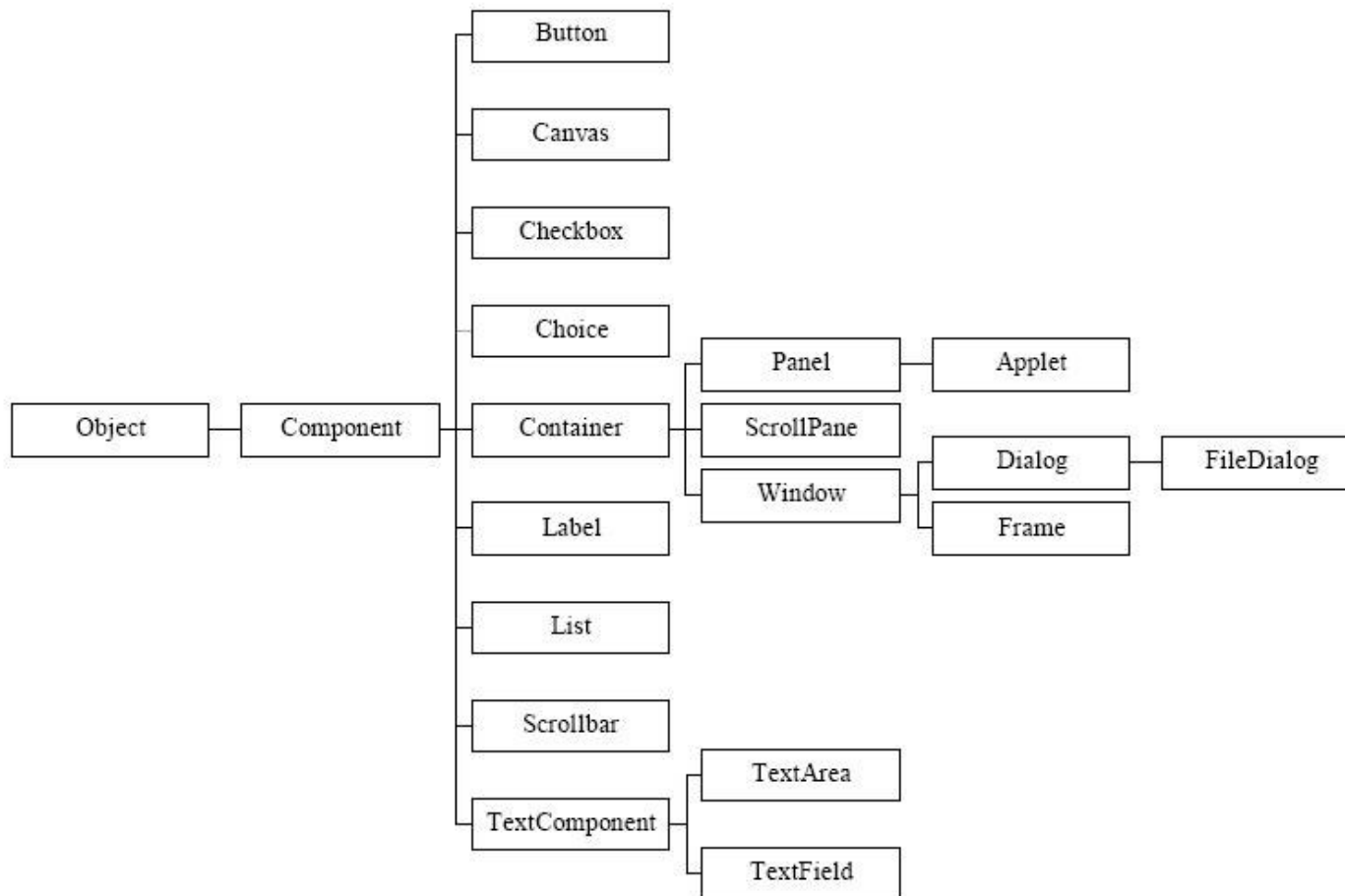
- La librería AWT esta compuesta por:
 - Los Componentes (`java.awt.Component`), como los Buttons, Labels,..
 - Los Contenedores (`java,awt.containers`), contienen componentes.
 - Los gestores de posición (`java.awt.LayoutManager`), que posiciona los componentes dentro de los contenedores.
 - Los eventos (`java.awt.AWTEvent`), que nos indican las acciones del usuario.



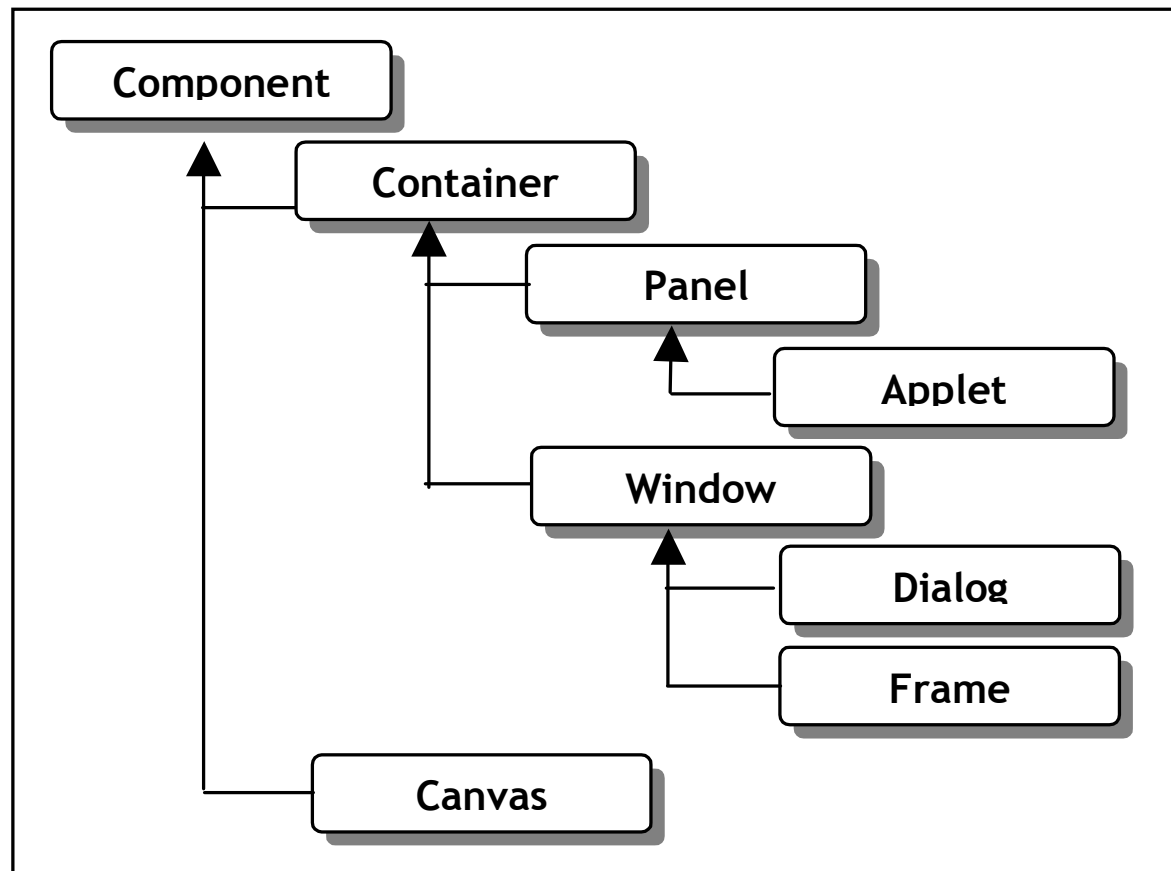
Contenedores & Componentes

- Contenedor
 - Elemento gráfico que puede contener dentro a componentes
 - Lo son todas las clases que derivan de `java.awt.Container`
 - Ejemplos:
 - Ventana (`java.awt.Frame`)
 - Dialogo (`java.awt.Dialog`)
 - Panel (`java.awt.Panel`)
- Componentes
 - Cualquier elemento gráfico que puede aparecer en una GUI
 - Lo son todas las clases que derivan de `java.awt.Component`
 - Ejemplos:
 - Botones (`java.awt.Button`)
 - Campos de texto (`java.awt.TextField`)
 - Etiquetas (`java.awt.Label`)
 - Lienzos de Dibujo (`java.awt.Canvas`)
- Un contenedor es también un componente → 1 contenedor puede contener dentro a otros contenedores

Jerarquía AWT



Contenedores





Contenedores: Tipos

- Window
 - Ventana tradicional de Windows, sin bordes y sin barra de menús
- Frame
 - A diferencia de Windows, acepta bordes y barra de menús, es la mas utilizada.
- Dialog
 - Ventana que sirve para lanzar preguntas o advertencias al usuario.
 - Depende de un Frame principal y no admite elementos como menús, barras de herramientas, ...
- Panel
 - Permite agrupar y organizar los elementos de una ventana
 - Es insertado dentro de un frame
- Applet
 - Es una subclase de Panel que se utiliza para ejecutarse en un navegador
- ScrollPane
 - Es como una ventana de tamaño limitado en la que se puede mostrar un componente de mayor tamaño con dos Scrollbars (barras de desplazamiento), una horizontal y otra vertical.
- ¿Cómo se crean los contenedores?

```
Frame f = new Frame();  
Panel p = new Panel();
```



Contenedores: Panel

- Un panel es la forma más simple de un contenedor
- Un panel se gestiona mediante la clase Panel, la cual extiende o deriva de la clase Container.

```
java.lang.Object  
java.awt.Component  
java.awt.Container  
java.awt.Panel
```

- El organizador o la disposición por defecto de un panel es de tipo FlowLayout (disposición de flujo)
- Esta disposición puede ser modificada mediante el método setLayout
- La clase Panel posee dos constructores:
 - `public Panel () // Crea un panel con una disposicion FlowLayout`
 - `public Panel (LayoutManager layout) // Crea un panel con la disposicion especificada`
- Como métodos, la clase Panel utiliza los heredados por su clase padre Container o los de la clase Component.



Contenedores: ScrollPane

- Panel de desplazamiento consiste en tener una superficie que visualiza una parte de su contenido
- El campo de visualización del panel es menor que el contenido
- Para poder acceder a las partes ocultas se utilizan las barras de desplazamiento
- Esta clase posee dos constructores:
 - `public ScrollPane ()`
 - `public ScrollPane (int scrollbarDisplayPolicy)`
- El segundo constructor permite indicar qué política debe seguir el panel de desplazamiento para visualizar las barras de desplazamiento. Esta política se rige por las siguientes constantes definidas en la misma clase:
 - `ScrollPane.SCROLLBARS_AS_NEEDED` // cuando sean necesarias
 - `ScrollPane.SCROLLBARS_ALWAYS` // Se muestran siempre
 - `ScrollPane.SCROLLBARS_NONE` // No se muestran nunca



Contenedores: Window

- Las ventanas son un tipo de contenedor de alto nivel que no puede estar contenido en otro contenedor.
- La clase base para todos los tipos de ventanas es la clase Window.
- No posee borde ni menús.
- A partir de esta clase se pueden crear clases de ventana más especializadas, como pueden ser Frame o Dialog
- La clase Window posee dos constructores:
 - `public Window (Frame owner)`
 - `public Window (Window owner)`



Contenedores: Frames

- Un marco es un tipo de ventana que posee un borde y un título, representando el tipo normal o estándar de ventana
- La clase Frame posee dos constructores:
 - `public Frame ()`
 - `public Frame (String title)`
- Para gestionar los títulos del marco de ventana existen dos métodos:
 - `public void setTitle (String title)`
 - `public String getTitle ()`
- Los marcos de ventana pueden ser redimensionados por el usuario
 - `public void setResizable (boolean resizable)`
- Los marcos de ventana pueden tener un icono o imagen que los representa cuando el marco se minimiza
 - `public void setIconImage (Image image)`
- Los marcos de ventana pueden tener barras de menú.
 - `public void setMenuBar (MenuBar mb)`



Contenedores: Dialog

- En esta ventana se pueden alojar componentes, conformando un formulario de datos que interactúe con el usuario
- Un diálogo no puede ser minimizado ni maximizado, y por tanto, no poseerá dichos botones en la barra de título
- Un cuadro de diálogo está vinculado a otra ventana
- Diferencia entre frame y diálogo: los frames pueden contener barras de menús
- La clase Dialog posee diversos constructores, dependiendo de cuál sea la ventana padre:
 - `public Dialog (Frame owner)`
 - `public Dialog (Frame owner, boolean modal)`
 - `public Dialog (Frame owner, String title)`
 - `public Dialog (Frame owner, String title, boolean modal)`
 - `public Dialog (Dialog owner)`
 - `public Dialog (Dialog owner, String title)`
 - `public Dialog (Dialog owner, String title, boolean modal)`
- Un diálogo puede ser modal cuando acapara toda la atención del usuario

Contenedores: ejemplo

```
import java.awt.*;

/* Esta clase hereda de la clase Frame de Java, es la
 * clase que va a representar un Frame. */

public class MiFrame extends Frame {
    public MiFrame(){
        setSize(300, 200);
        setTitle("Hola Mundo");
    }

    public static void main(String[] args) {
        // Creamos una instancia de esta clase
        MiFrame f = new MiFrame();
        // Mostramos el frame
        f.setVisible(true);
    }
}
```



Contenedores...

- ¿Cómo se añaden componentes a un contenedor?
 - Mediante el método add
 - <contenedor>.add(<componente>);

```
import java.awt.*;

public class MiFrame2 extends Frame {
    Panel panel;
    public MiFrame2(){
        setSize(300, 200);
        setTitle("Hola Mundo");
        // creación del objeto panel
        panel = new Panel();
        // color de fondo del panel (usando la clase Color)
        panel.setBackground(Color.red);
        // agregado del objeto panel al marco
        this.add(panel);
    }
    public static void main(String[] args) {
        MiFrame2 f = new MiFrame2();
        f.setVisible(true);
    }
}
```





Ejemplo Contenedores

```
import java.awt.*;

class EjemploMarco1 extends Frame {
    Panel panel;
    Dialog dialogo;

    EjemploMarco1() {
        this.setTitle("Primer ejemplo de Marco...");
        this.setBounds(200, 100, 600, 350);
        this.setBackground(Color.gray);
        panel = new Panel();
        panel.setBackground(Color.red);
        this.add(panel);
        dialogo = new Dialog(this, "Ejemplo de diálogo...", false);
        dialogo.setBounds(250, 200, 300, 100);
        dialogo.setVisible(true);
    }
}

GUI
```



Ejemplo Contenedores...

```
...  
class EjemploContenedores1 {  
public static void main(String[] args) {  
    EjemploMarco1 marco = new EjemploMarco1();  
    marco.setVisible(true);  
}  
}
```



Contenedores...

- ¿Cómo se añaden componentes a un contenedor?
 - Inserción de dos botones y dos campos de texto en una ventana

```
Frame ventana = new Frame();  
ventana.add(new Button("Aceptar"));  
ventana.add(new Button("Cancelar"));  
ventana.add(new TextField(20));  
ventana.add(new TextField(30));
```

- Inserción de una etiqueta, una lista y un botón en un panel

```
Panel panel = new Panel();  
panel.add(new Label("Soy una etiqueta"));  
panel.add(new List());  
panel.add(new Button("OK"));
```

- Inserción de un panel dentro de una ventana
- ```
ventana.add(panel);
```



# Contenedores

---

- Algunos métodos de los contenedores
  - Asignar título y tamaño a una ventana:

```
ventana.setTitle("Titulo de la ventana");
ventana.setSize(300,200);
```

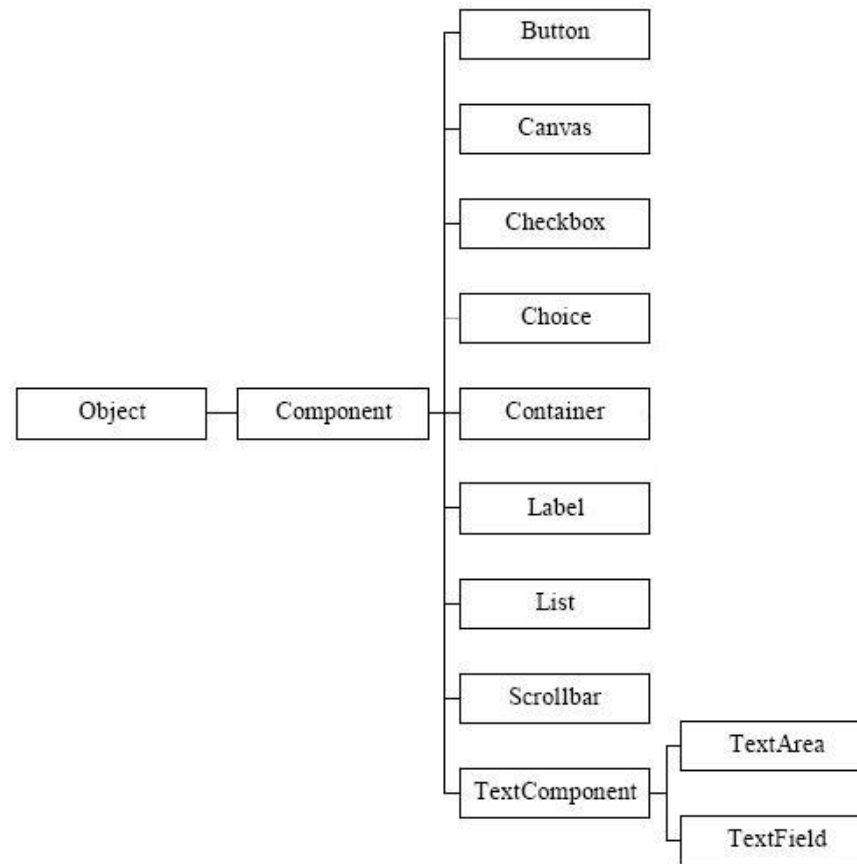
- Asignar color de fondo a un panel:

```
panel.setBackground(Color.blue)
```

- Más métodos → API



# Componentes





# Componentes

¿Como se crean?

- Como cualquier otro objeto en Java → Constructores

```
Button boton = new Button();
Label etiqueta = new Label();
TextField campoTexto = new TextField();
```

- Constructores

- Cada tipo de componente tiene distintos tipos de constructores

```
TextField c = new TextField();
TextField c = new TextField("Hola a todos");
TextField c = new TextField(20);
TextField c = new TextField("Hola", 20);
```

```
Label etiq = new Label();
Label etiq = new Label("Apellido: ");
Label etiq = new Label("Edad: ", Label.CENTER);
```

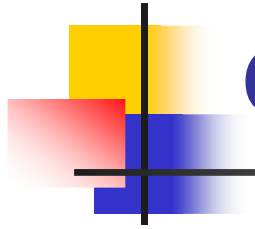
- Métodos más comunes

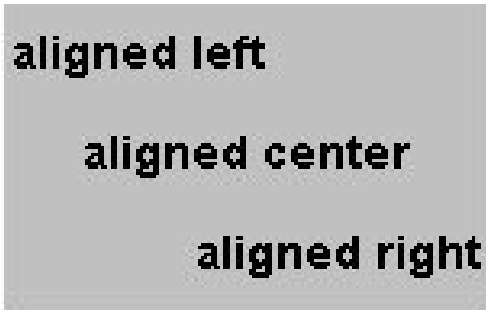
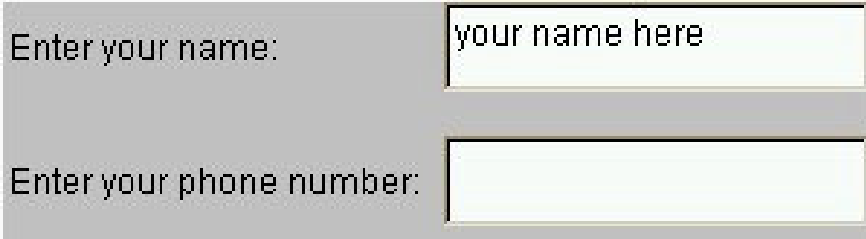
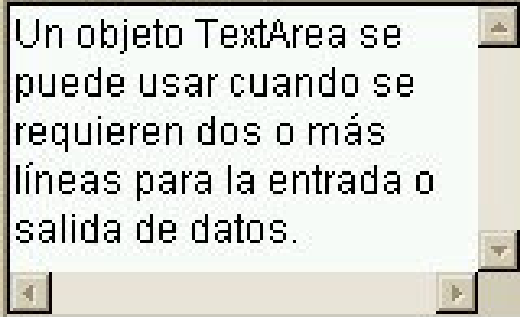

- Los componentes, como todos los objetos proporcionan métodos para manipular sus propiedades

```
boton.setLabel("Aceptar");
etiq.setText();
```

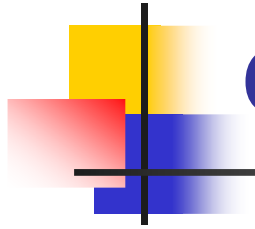
- Más información → API





# Widgets o componentes elementales



| <b>Label</b>                                                                        | <b>TextField</b>                                                                      |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|    |    |
| <b>TextArea</b>                                                                     | <b>Button</b>                                                                         |
|  |  |

# Widgets o componentes elementales...



| <b>CheckBox</b>                                                                      | <b>CheckboxGroup</b>                                                                  |
|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|    |    |
| <b>Choice</b>                                                                        | <b>List</b>                                                                           |
|  |  |



# Labels (Etiquetas)

---

- Son usadas para mostrar información
- Constructores
  - Label ( )
  - Label (String s)
  - Label (String s, int alignment)
- Métodos
  - String getText( )
  - void setText (String s)
  - void setAlignment (int alignment)
    - Label.LEFT, Label.CENTER, Label.RIGHT



# Buttons (Botones)

---

- Constructores:
  - Button()
  - Button(String label)
- Métodos:
  - void setLabel(String label): Setea la etiqueta del botón con el nombre especificado.
  - String getLabel(): Obtiene la etiqueta del botón.
  - void addActionListener(ActionListener l): Agrega un listener para detectar los eventos que se producen con un botón.



# Choice

---

- Este componente puede presentar al usuario una serie de alternativas de las cuales solo una selección puede hacerse.
- **Constructor:**
  - Choice(): Crea una nueva lista de selección.
- **Métodos:**
  - void add (String item): Agrega un item a la lista.
  - String getItem(int index): Retorna el string correspondiente al índice.
  - void addItemListener(ItemListener l): Agrega un listener para detectar los eventos que se producen componente Choice.



# Algunos métodos de Choice

---

- `int getItemCount()`: Retorna el número de items en el Choice.
- `String getItem(int index)`: Obtiene la representación `String` del correspondiente índice (Los índices comienzan en cero).
- `int getSelectedIndex()`: Retorna el índice del item seleccionado actualmente.
- `String getSelectedItem()`: Representación `String` del item seleccionado.



# List

---

- Permiten escoger entre una o más opciones.
- **Constructores:**
  - List(): Crea una nueva lista
  - List(int rows): Crea lista con un número específico de líneas.
  - List(int rows, boolean multipleMode): Crea una nueva lista con cierto numero de filas y especifica si permite selección múltiple.
- **Métodos:**
  - void add(String item): agrega un item al final de la lista.
  - void add(String item, int index): agrega un item en cierto índice.



# Algunos métodos de List

---

- `int getItemCount()` : Retorna el número de items en el List.
- `void removeAll()` : Remueve todos los items del List.
- `int getSelectedIndex()` : Retorna el item seleccionado (-1 si ningún item es seleccionado).
- `int[] getSelectedIndexes()` : Arreglo de items seleccionados. Si no hay items seleccionados, arreglo de longitud cero.
- `String getSelectedItem()` : Item seleccionado o null.
- `String[] getSelectedItems()` : Retorna arreglo de items seleccionados.



# TextField

---

- **Cuadro de texto de una sola línea.**
- **Constructores:**
  - TextField()
  - TextField(int columns): Construye un campo de texto con un número específico de columnas.
  - TextField(String s): Construye un campo de texto que displaya s.
- **Métodos:**
  - void setEchoChar(char c): Setea el caracter a ser displayado a medida que el usuario tipea.
  - void setEditable(boolean b) //true==editable Especifica si un TextField es editable.



# TextArea

---

- **Constructores:**

- TextArea()
- TextArea(String text): Construye una nueva área de texto con un texto específico
- TextArea(int rows, int columns): Construye un area de texto con un número específico de filas y columnas.

- **Métodos:**

- void append(String str): agrega un texto dado a un text area.
- void setColumns(int columns): Setea el número de columnas para esta área de texto.
- void setRows(int rows): Setea el número de filas para el área de texto.



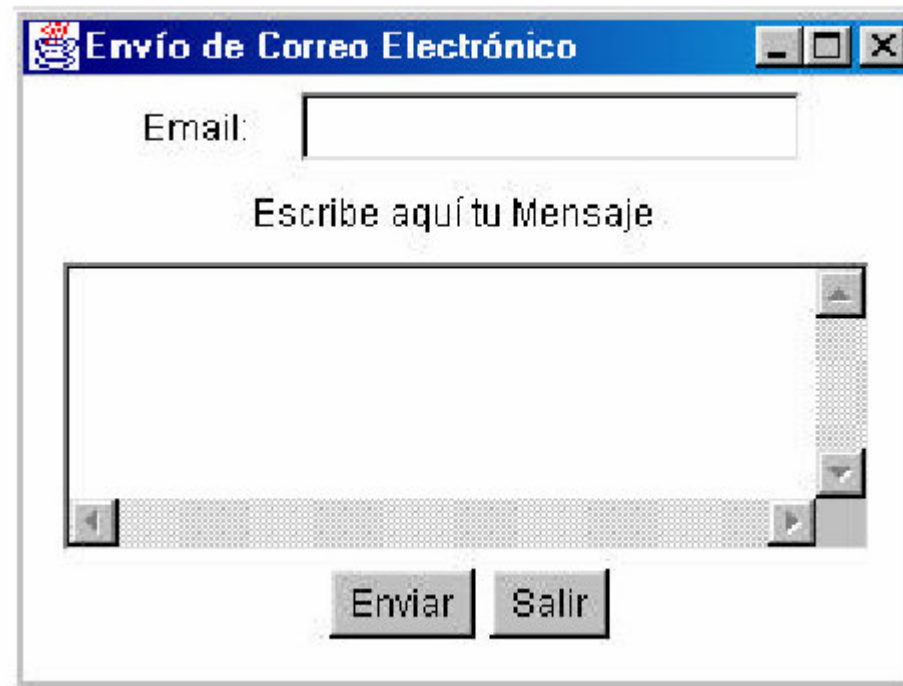
# Resumen

---

- No se trabaja con las ventanas por defecto que trae Java (clase `java.awt.Frame`)
- Se crean ventanas propias y se configuran a nuestro gusto
- Para ello:
  - Crear una clase derivada de `Frame`
  - Se le añaden todos los elementos (componentes) que se quieran
  - Ejercicio: ventana que posee un `Panel`, dos etiquetas, un campo de texto, un área de texto y dos botones

# Ejercicio:

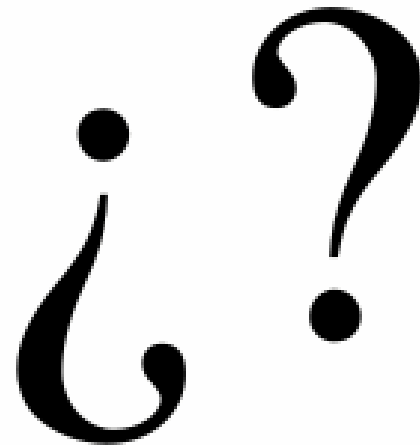
- Ventana generada:





# Consultas...

---





# Próxima clase

---

- Repaso
- Administradores de diseño (Layouts Managers): ventajas, tipos, como definirlos
- Eventos: gestión, tipos
- Ejemplos y ejercitación