

# Práctica III - Introducción a Swing

Swing es un potente conjunto de controles que nos permite crear interfaces ricas fácilmente. Al momento de crear programas en algún momento requeriremos diversos datos del usuario, como su nombre, su edad, si es hombre o mujer, etc. Para crear formularios debemos explorar los controles y analizar las propiedades de cada control para determinar si nos será útil para recabar información en nuestro sistema.

Si no estás conforme con los pocos componentes de *swing* no te angusties y visita

<http://www.tutego.de/java/additional-java-swing-components.htm>

<https://dairdev.wordpress.com/2011/12/05/actualizado-instalar-swingx-en-netbeans-componentes-swingx-de-swinglabs/>

## Etiquetas

---

`JLabel` nos permite mostrar un texto en una línea e incluso colocar contenido html para mayor presentación. Algunas propiedades interesantes son:

`text` - Ajusta el texto, prueba `<html>Hola<br>mundo</html>` que inserta código html

`toolTipText` - Muestra un texto de ayuda cuando se posiciona el cursor sobre él

`labelFor` - Cede el foco a otro elemento objetivo al ser pulsado

`icon` - Muestra una imagen dentro del label

## Botones

---

`JButton` nos permite mostrar un botón pulsable. Algunas propiedades interesantes son:

`cursor` - Cambia el cursor del *mouse* cuando este pasa por encima

`enabled` - Deshabilita el botón para que no se pueda pulsar

## Agrupador de botones

---

`ButtonGroup` permite definir un grupo de botones que compartirán funcionalidad dentro del grupo, por ejemplo, en el caso de utilizar controles `JRadioButton` o `JToggleButton` estos funcionarán de forma

exclusiva (excluyente). Observa que este control no tiene representación gráfica y sólo sirve para ajustar las propiedades de otros controles.

## Botones alternadores

---

`JToggleButton` permite crear un botón que preserva su estado de pulsado. Algunas propiedades interesantes son:

`buttonGroup` - Define el grupo al que pertenece y excluye a otros controles del grupo para tomar el mando (intenta crear un grupo y dos `JToggleButton` )

`selected` - Indica si el botón está pulsado

`rolloverIcon/rolloverSelectedIcon` - Ajusta íconos cuando el botón está en estado seleccionado y deseleccionado

## Marcadores lógicos

---

`JCheckBox` permite mostrar una caja que preserva un estado de marcado o desmarcado, aunque soporta `ButtonGroup` no se recomienda asignar grupo ya que funcionaría como un `JRadioButton` . Algunas propiedades interesantes son:

`selected` - Indica si la caja está seleccionada ( `true` )

## Botones excluyentes

---

`JRadioButton` muestra unos botones en forma de círculo que permiten mantener el estado de seleccionado, junto a `ButtonGroup` funciona para seleccionar uno a la vez.

## Cajas de selección

---

`JComboBox` es una caja que muestra diversas opciones para seleccionar. Este control es especial ya que requiere un modelo para funcionar y se verá en una práctica independiente.

## Listas

---

`JList` es similar a `JComboBox` con la diferencia que los elementos son mostrados en una región definida, también posee un modelo y se verá en una práctica independiente.

# Cajas de texto

---

`TextField` funciona como `Label` con la diferencia que en esta el usuario puede introducir texto. `PasswordField` funciona de forma similar sólo que esconde los caracteres. Algunas propiedades interesantes son:

`editable` - Determina si se puede modificar el texto de la caja

`TextArea` funciona para introducir texto, sin embargo, hablaremos más a fondo de este control en otra práctica.

## Problemas

---

- Crea un formulario que solicite los datos personales del usuario: nombre, edad, lugar de nacimiento, dirección, género y si es soltero o casado.
- Agrega al formulario los gustos musicales: rock, pop, regueton, cumbia, salsa, ninguna.
- Agrega una caja de selección con 5 colores para que elija su favorito.
- Agrega un botón que diga enviar y otro que diga limpiar.
- Agrega una etiqueta que diga comentarios seguido de una caja de texto en varias líneas
- Agrega un botón que diga enviar comentarios.

**Nota:** Comprime tu proyecto en `ZIP` (no `RAR`, no `7z`, no otros) y guardalo en una memoria USB o envíalo a tu correo. Las prácticas deberás entregarlas a más tardar antes de la siguiente clase de viernes (una semana máximo) al correo `badillo.soft@hotmail.com` con el asunto `Práctica I - <TU PRIMER NOMBRE Y PRIMER APELLIDO>`.

**Nota:** En el mismo archivo comprimido `ZIP` deberás adjuntar un archivo `<TU PRIMER NOMBRE Y PRIMER APELLIDO>.txt` que contenga las respuestas que se piden como texto (como la lista de las 5 propiedades con su descripción). En caso de no solicitar respuestas escritas deberás crear igualmente el archivo pero dejarlo vacío. Si se requieren capturas de pantalla, deberás adjuntarlas en una carpeta llamada `capturas`.

**Nota:** En caso de no seguir estas instrucciones la práctica no será recibida hasta el final del curso con un puntaje del `50%` solamente. La suma total de cada práctica es de `100 puntos` distribuidos por número de incisos sin importar si uno es más difícil que otro.

**Diplomado de Java - Alan Badillo Salas (badillo.soft@hotmail.com)**

**Instituto Politécnico Nacional - Centro de Investigación en Computo**