

## Enunciado

Con el fin de acelerar la creación de interfaces y que estas tengan un diseño semejante, se desea crear una serie de componentes que deberán ser empaquetados en un solo archivo.

Todos los componentes deben permitir introspección y persistencia.

Las etiquetas, cuadros de texto, y demás componentes tendrán un tipo de letra, tamaño, color de texto, color de fondo, etc., por lo que, en primer lugar, el alumno debe indicar la fuente de texto que van a tener los componentes que va crear.

El proyecto consta de dos apartados:

- Apartado 1. Crear componentes y empaquetarlos.
- Apartado 2. Probar componentes sobre la aplicación “Venta de productos”.

### Apartado 1. Creación de componentes

1. Crear los componentes más comunes de una interfaz de forma que todos tengan el mismo formato (fuente, tamaño, color de fondo, color del texto, etc.). Los componentes a crear son: JLabel, JTextField, JComboBox, JList, JTextArea, JRadioButton, JCheckbox y Botones.

Cada uno de estos componentes debe tener unas dimensiones iniciales cuando se sitúen sobre el contenedor.

2. Crear un componente del tipo JList compuesto por un JLabel y el JList que contenga todos los países del mundo. Este componente deberá devolver el código del país y mediante una **propiedad ligada** notificará que se ha realizado una elección. Se proporciona un archivo con el código de una matriz con los países y sus respectivos códigos (Matriz\_Simple\_Paises.txt). Este componente se emplea en la gestión de fabricantes.

### 3. Etiqueta y cuadro de texto.

- a. Crear un componente formado por una etiqueta y un cuadro de texto. Este componente tendrán las propiedades necesarias para que el usuario pueda establecer/acceder al texto de la etiqueta y del cuadro de texto (setter y getter tanto para la etiqueta como para el componente).
- b. Este componente tendrá una propiedad de tipo booleano para indicar si es obligatorio cumplimentar ese componente, de forma que la apariencia de la etiqueta será diferente a la de otro componente que no sea obligatorio cumplimentar. Por ejemplo, si se trata de un campo obligatorio, entonces puede aparecer el texto de la etiqueta en negrita y añadirse el carácter asterisco.
- c. Asociado a este componente habrá dos eventos de forma que, cuando el **cuadro de texto pierda el foco o se pulse la tecla “enter”**, entonces se generará un EventObject que contendrá:
  - Un objeto, ya sea el propio componente o una tabla.
  - Un booleano que indicará si el cuadro de texto está vacío.
  - Un booleano que indicará si el valor introducido por el usuario se encuentra en la lista. Este componente debe permitir, por tanto, que se le pase una lista (ArrayList o Array) de String. Esta comprobación la hará si se le ha pasado una lista, si no, simplemente devolverá false.

Se debe implementar el componente EventSource, el EventObject y la interfaz del *listener*. El EventListener se implementará cuando se utilice el componente en el programa principal.
- d. Este componente tendrá una propiedad ligada. La propiedad ligada es la que contiene el texto introducido por el usuario. La propiedad ligada se desencadena cuando el cuadro de texto pierde el foco.
- e. Este componente debe tener la función de realizar dos comprobaciones:
  - Comprobar que el cuadro de texto no está vacío en caso de que este campo sea obligatorio.
  - Comprobar si el valor introducido en el cuadro de texto se encuentra en un ArrayList o Array que se le pase como parámetro.

En caso de que se haya producido un error, la apariencia del componente debe cambiar. Por ejemplo, la etiqueta puede aparecer en color rojo (esto lo dejo a la elección del alumno). También deberá implementarse algún mecanismo para mostrar un mensaje de aviso u error.

#### 4. Componente etiqueta y combobox.

- a. Crear un componente formado por una etiqueta y un combobox. Este componente tendrán las propiedades necesarias para que el usuario pueda establecer/acceder al texto de la etiqueta y del combobox (setter y getter tanto para la etiqueta como para el componente).  
Este componente recibirá obligatoriamente un ArrayList o Array con los valores y, opcionalmente, podrá recibir una segunda lista con los “códigos” de cada opción.  
Este componente deberá tener propiedades para almacenar el índice seleccionado, el texto seleccionado del combobox y el código (si no se ha pasado la lista de código, entonces el valor será null). Estas propiedades tendrán los correspondientes getter y setter.
- b. Este componente tendrá una propiedad de tipo booleano para indicar si es obligatorio cumplimentar ese componente, de forma que la apariencia de la etiqueta será diferente a la de otro componente que no sea obligatorio cumplimentar. Por ejemplo, si se trata de un campo obligatorio, entonces puede aparecer el texto de la etiqueta en negrita y añadirse el carácter asterisco.
- c. Asociado a este componente habrá un evento de forma que, cuando se seleccione algo del combobox, entonces se generará un EventObject que contendrá:
  - Un objeto, ya sea el propio componente o una tabla.
  - Un Integer que indicará el índice seleccionado.
  - El texto de la opción elegida.
  - Si se ha introducido la lista de códigos, entonces también el código, si no el valor será null.

Se debe implementar el componente EventSource, el EventObject y la interfaz del *listener*. El EventListener se implementará cuando se utilice el componente en el programa principal.

- d. Este componente tendrá una propiedad ligada. La propiedad ligada es la que contiene el índice de la opción elegida. La propiedad ligada se desencadena cuando se elige una opción.
- e. Este componente debe tener la función de comprobar:
  - a. En caso de ser un valor obligatorio, deberá comprobar si se ha seleccionado un valor y, además, que este valor no sea el del índice 0 (suele utilizarse para poner algo así como “Elija una opción”).

En caso de que se haya producido un error, la apariencia del componente debe cambiar. Por ejemplo, la etiqueta puede aparecer en color rojo (esto lo dejo a la elección del alumno). También deberá implementarse algún mecanismo para mostrar un mensaje de aviso u error.

## Apartado 2. Probar componentes

El objetivo de este apartado es probar la utilización de los componentes creados e implementar los EventObject indicados en el apartado anterior.

Una vez empaquetados los componentes del apartado 1, estos deben emplearse para implementar las secciones fabricantes y empleados. Si el alumno ya tiene hecha estas secciones, entonces deberá duplicar el proyecto y trabajar sobre la copia.

### Gestión de empleados

Para probar los componentes solo es necesario permitir buscar, añadir empleados. Las vistas son semejantes en lo expuesto en el proyecto de la unidad 1.

#### Buscar empleados

Código	Nombre y apellidos	Ciudad	Cargo		
				Modificar	Eliminar
				Modificar	Eliminar
				Modificar	Eliminar

Apellido:  País:  Ciudad:

Cargo:  ☐

En esta vista solamente se va a implementar los componentes para filtrar por Apellido y Cargo.

Se debe utilizar los componentes de los ejercicios 4 y 5. Se creará un EventListener para cada uno de ellos de forma que cuando pierdan el foco se desencadene una búsqueda por el dato introducido o seleccionado. Obviamente se debe haber introducido un apellido o seleccionado un cargo.

#### Alta de un empleado

A la hora de dar de alta un empleado, es obligatorio introducir el nombre, los apellidos y el seleccionar el cargo. Se deben utilizar los componentes creados en los apartados 4 y 5 para que, en caso de que reciban el

Formulario de alta de empleado

Nombre:

Apellidos:

País:  Ciudad:  Cargo:

☐ ☐ ☐

foco y no se introduzca los datos o seleccione la opción de los campos obligatorios, cuando pierda el foco se muestre un mensaje de error. El mensaje debe ser un cuadro desplegable informativo.

Aquí se debe crear el ObjectListener para cada componente.

Cuando se pulse el botón de aceptar, se debe poder recuperar los datos introducidos por el usuario para dar de alta un nuevo empleado.

## Gestión de fabricantes

Para probar los componentes solo es necesario permitir buscar, añadir fabricantes. Las vistas son semejantes en lo expuesto en el proyecto de la unidad 1.

### Buscar fabricantes

MENÚ

FABRICANTES - BÚSQUEDA

CLAVE	CÓDIGO	NOMBRE		
			MODIFICAR	ELIMINAR
			MODIFICAR	ELIMINAR
			MODIFICAR	ELIMINAR

CÓDIGO

NOMBRE

PAÍS

☐ ACTIVO  
☐ NO ACTIVO

En esta vista solo es necesario implementar la búsqueda por código y por país.

Se debe utilizar los componentes de los ejercicios 4 y 6. Se creará un EventListener para cada uno de ellos de forma que pierdan el foco cuando se desencadene una búsqueda por el dato introducido o seleccionado. Obviamente se debe haber introducido un código o seleccionado un país.