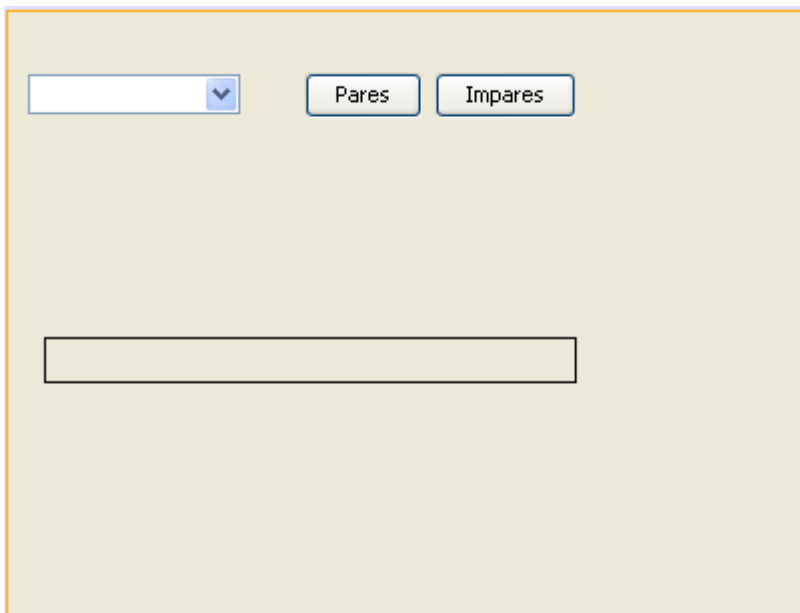


## EJERCICIO GUIADO. JAVA: MODELOS DE CUADRO DE LISTA

1. Realiza un nuevo proyecto.
2. En la ventana principal debes añadir lo siguiente:
  - a. Un combo llamado cboNumeros.
  - b. Un botón "Pares" llamado btnPares.
  - c. Un botón "Impares" llamado btnImpares.
  - d. Una etiqueta con borde llamada etiResultado.
3. Elimina todos los elementos que contenga el combo. Recuerda, debes usar la propiedad "model" del combo para cambiar sus elementos.
4. Después de haber hecho todo esto, tu ventana debe quedar más o menos así:



5. En el evento *actionPerformed* del botón Pares, programa lo siguiente:

```
int i;

DefaultComboBoxModel modelo = new DefaultComboBoxModel();

for (i=0;i<10;i+=2) {
    modelo.addElement("Nº "+i);
}

cboNumeros.setModel(modelo);
```

6. Observa lo que hace este código:
  - a. Crea un objeto "modelo" para el combo.

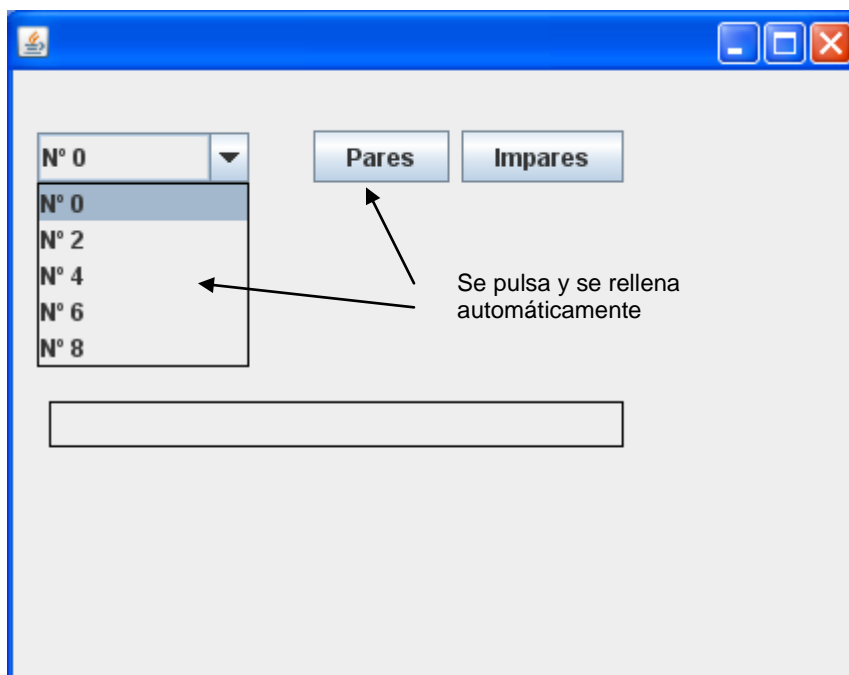
Al igual que pasa con los cuadros de lista, los combos tienen un objeto "modelo" que es el que realmente contiene los datos. En el caso de los combos, para crear un objeto "modelo" se usará esta instrucción:

```
DefaultComboBoxModel modelo = new DefaultComboBoxModel();
```

- b. A continuación, se usa el objeto “modelo” creado y se rellena de datos. Concretamente, se rellena con los números pares comprendidos entre 0 y 10.
- c. Observa el uso de la propiedad `addElement` para añadir un elemento al modelo del combo.
- d. Se ha usado un bucle `for` para hacer la introducción de datos en el modelo más fácil.
- e. Finalmente, se asocia el modelo al combo a través de la siguiente línea, con lo que el combo aparece relleno con los elementos del modelo:

```
cboNumeros.setModel(modelo);
```

7. Ejecuta el programa y observa el funcionamiento del botón Pares.



8. El botón Impares es similar. Programa su *actionPerformed* como sigue:

```
int i;
DefaultComboBoxModel modelo = new DefaultComboBoxModel();

for (i=1;i<10;i+=2) {
    modelo.addElement("Nº "+i);
}

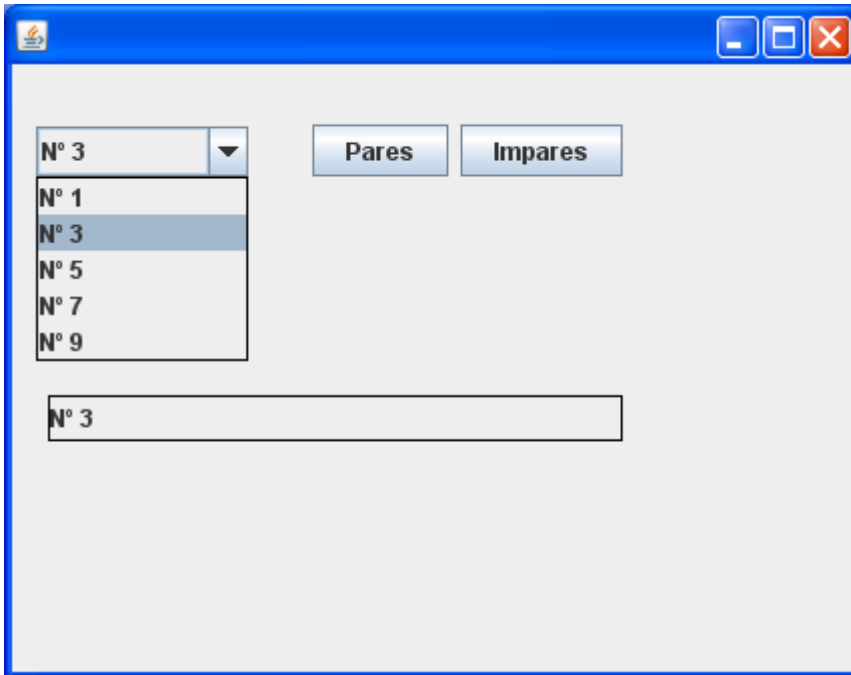
cboNumeros.setModel(modelo);
```

9. La única diferencia de este código es el `for`, que está diseñado para que se introduzcan los números impares comprendidos entre 0 y 10 dentro del modelo.
10. Finalmente se programará el *actionPerformed* del combo para que al seleccionar un elemento este aparezca en la etiqueta. Esto se hace con una simple instrucción:

```
etiResultado.setText(cboNumeros.getSelectedItem().toString());
```

Recuerda el uso de `getSelectedItem()` para recoger el elemento seleccionado, y el uso de `toString()` para convertirlo a texto.

11. Prueba el programa. Prueba los botones Pares e Impares y prueba el combo.



12. Sería interesante añadir un botón “Vaciar” llamado `btnVaciar` que vaciara el contenido del combo. Esto se haría simplemente creando un modelo vacío y asignarlo al combo. Se anima al alumno a que realice esta mejora.

## CONCLUSIÓN

Un combo, al igual que los cuadros de lista, es un objeto que contiene a su vez otro objeto denominado “modelo”.

El objeto “modelo” es el que realmente contiene los datos del combo.

Combo → Modelo → Datos

Se puede crear un “modelo” y luego introducir datos en él. Luego se puede asociar ese “modelo” al combo. De esta manera se puede cambiar el contenido del combo en cualquier momento.