



# Aplicaciones RIA con ZK

- Sesión 4: Componentes (1)



# El lenguaje zscript

- El lenguaje zscript
- Expresiones EL
- Atributos ZK
- Espacio de identificadores
- Procesamiento y carga de una página ZUL

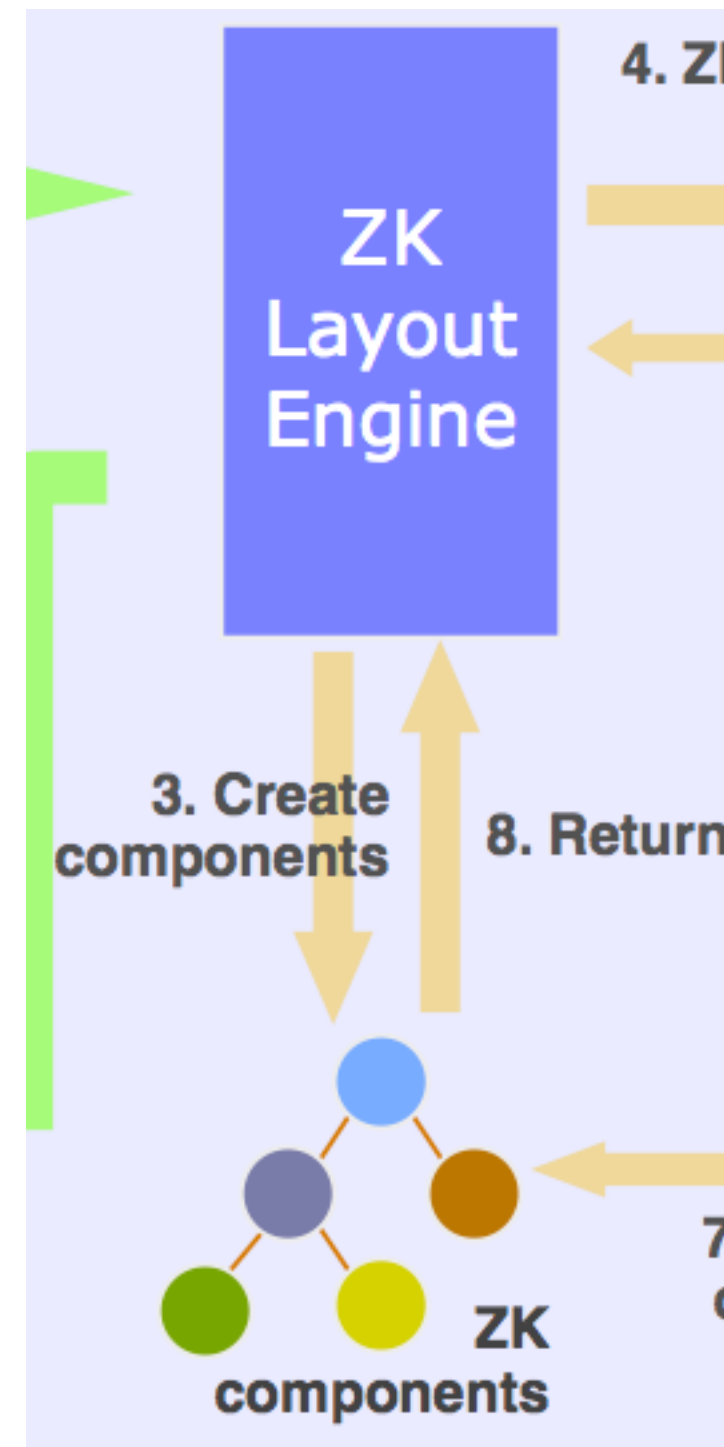


# Componentes ZUL y componentes Java

- Los componentes ZUL se definen por etiquetas: `<button>`, `<label>`, ...
- Al procesar la página ZUL se convierten en objetos Java de las clases `Button`, `Label`, ...
- Las clases están definidas en el paquete `org.zkoss.zul`
- Cada atributo en ZUML se corresponde con una pareja de métodos `get/set` en la clase Java

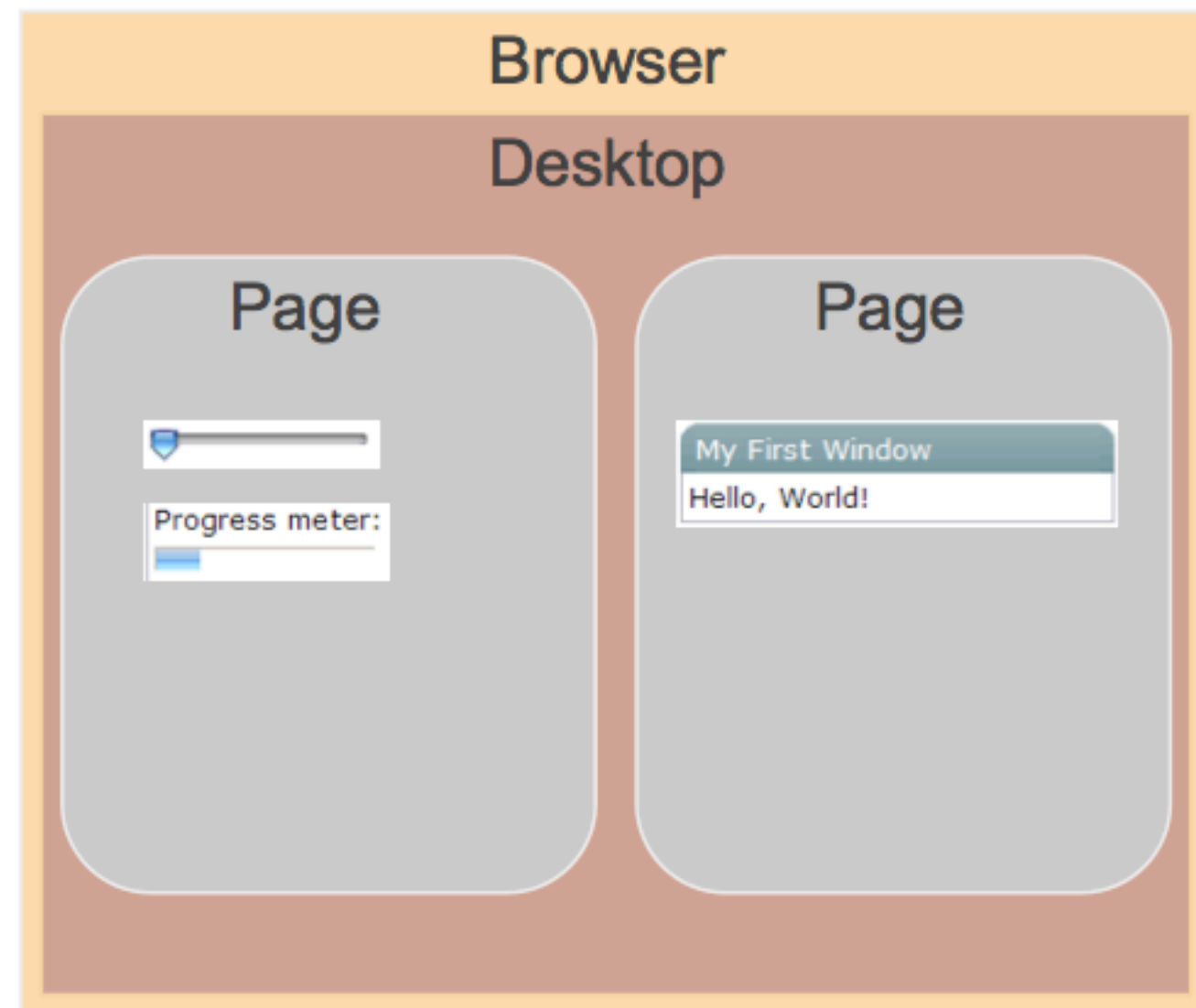
# Árboles de componentes

- Al procesar la página ZUL el ZK loader construye un árbol de componentes (objetos Java)



# Desktop y page

- Una interfaz de un usuario define un escritorio Desktop
- Un escritorio puede usar más de una página, que se crean cuando el usuario accede a distintas páginas zul
- El escritorio se guarda en una variable (`desktop`) que puede usarse para comunicar distintas páginas
- Dentro de cada página se definen componentes





# Añadir componentes a otros

- Se puede llamar al método `setPage` para hacer que el componente se convierta en un componente raíz de la página especificada
- Se puede llamar a los métodos `setParent`, `insertBefore` o `appendChild` para hacer que el componente sea hijo de otro
- Los métodos están definidos en la interfaz `org.zkoss.zk.ui.Component`



# Eliminar componentes

- Es posible desconectar un componente raíz de una página llamando a `setPage` con `null`
- Un hijo es desconectado de su padre llamando a `setParent` con `null`



# Ejemplo

```
<zk>
<zscript>
    public void nuevaEtiqueta() {
        Label label = new Label();
        label.setValue("Nueva etiqueta ");
        label.setParent(win_2);
    }
</zscript>
<hbox>
    <window id="win_1" title="Ventana 1" border="normal">
        <button label="añade" onClick="nuevaEtiqueta()" />
    </window>
    <window id="win_2" title="Ventana 2" border="normal">
    </window>
</hbox>
</zk>
```





# Clasificación de componentes en ZK

- Elementos simples
- Contenedores y diálogos
- Gestores de disposición
- Presentación de datos
- Menús
- Especializados



# Componentes simples

- Básicos: button, label
- Imágenes: image, imagemap
- Input: textbox, intbox, doublebox, datebox, timebox, spinner, slider
- Selección: checkbox, radio, combobox, bandbox
- Elementos XHTML
- Varios: progressmeter, timer



# Contenedores y diálogos

- Contenedores: window, panel, tabbox, tabs, tabpanel, tabpanels
- Diálogos: MessageBox, fileupload, Filedownload



# Disposición

- Disposición por defecto: space
- Modelo de caja: box, vbox, hbox, splitter
- Border layout: borderlayout, north, south, center, east, west
- Tablas de paneles: columnlayout, columnchildren, tablelayout, tablechildren
- Agrupaciones: groupbox, toolbar, toolbarbutton



# Datos y elementos múltiples

- Listas: listbox, listhead, listheader, listitem, listcell
- Rejillas: grid, columns, column, rows, row, detail
- Árboles: tree, treechildren, treeitem, treerow, treecell, treecols, treecol, treefooter, treefoot



# Menús

- Menús: menubar, menupopup, menu, menuitem, menuseparator
- Menús contextuales: popup



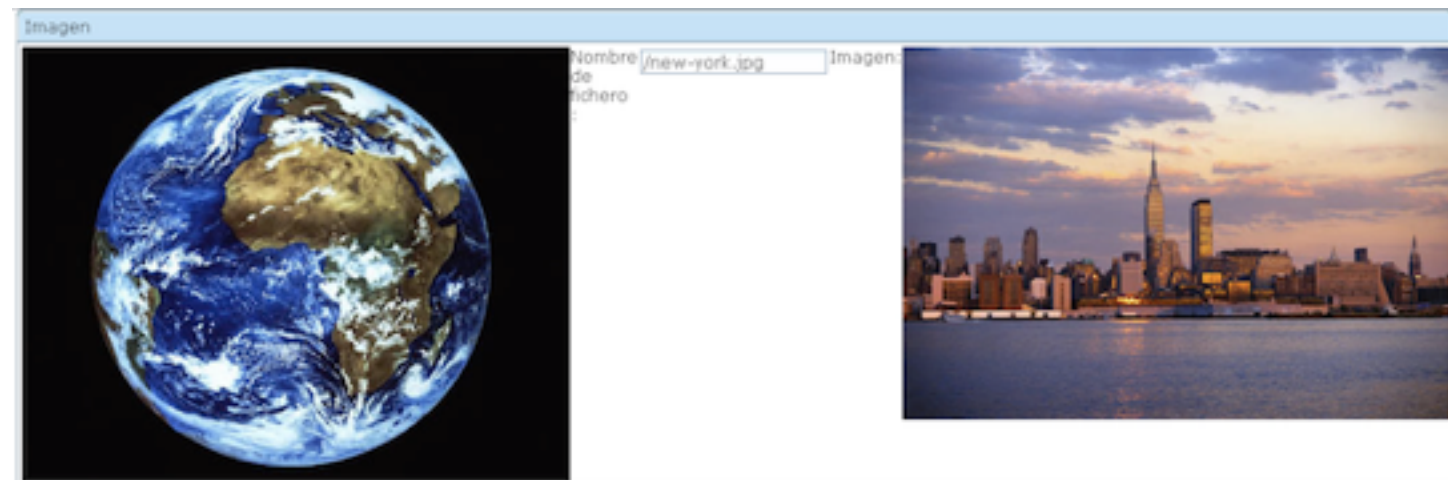
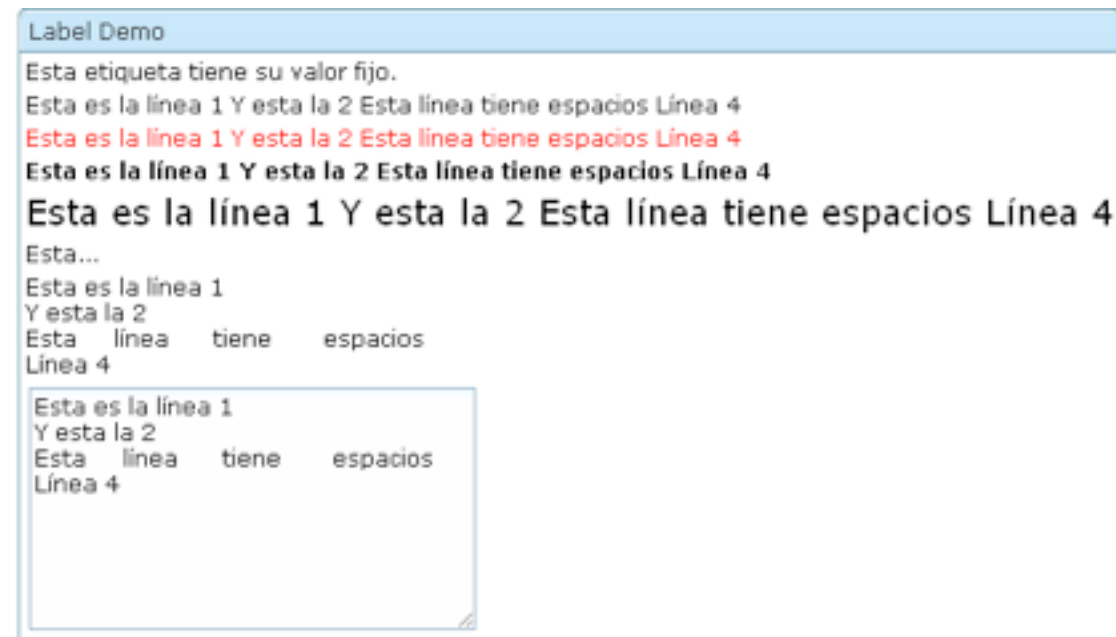
# Especializados

- Captcha: captcha
- Gráficos: chart
- Mapas: gmaps
- Editor: fckeditor
- Línea temporal: timeline
- Plots de datos: timeplot



# Probamos el código de los ejemplos

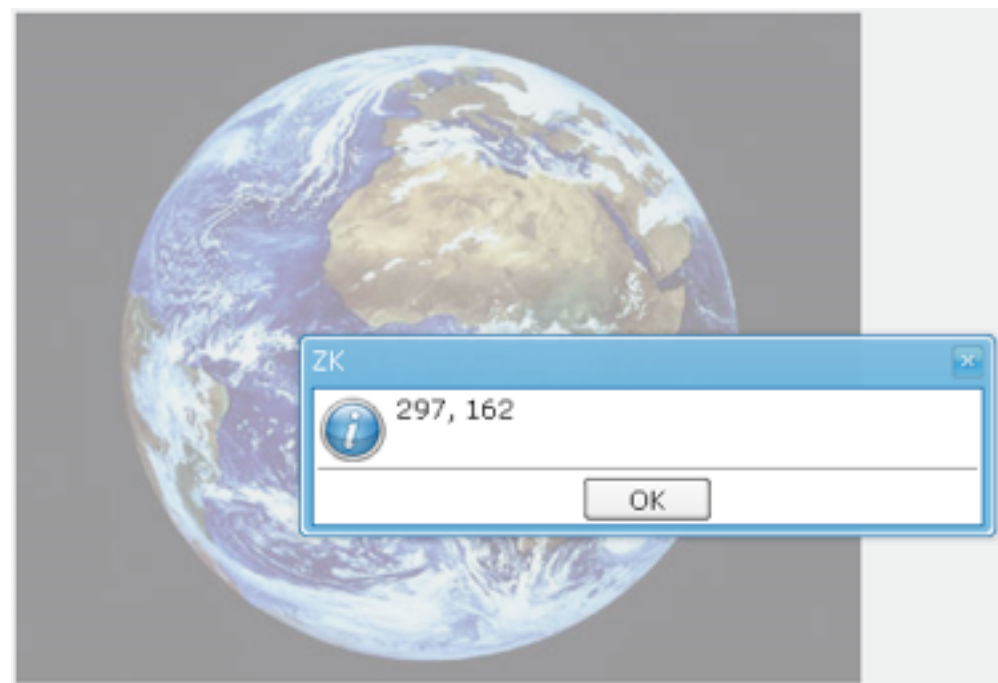
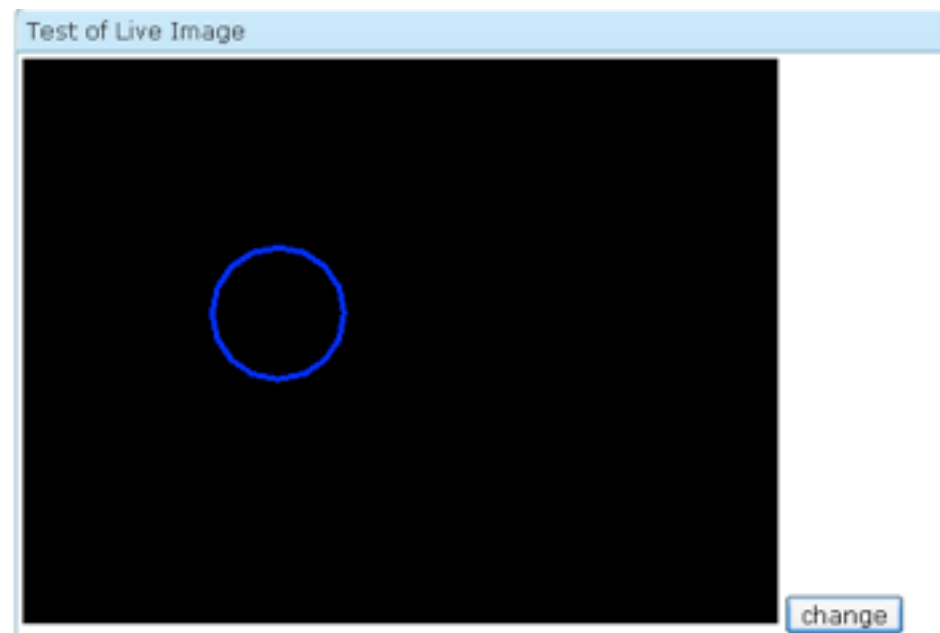
- Label
- Button
- Image





## Ejemplos (2)

- Imagen generada
- Imagemap





# Entrada

- Distintos componentes según el tipo de datos
- Es posible definir restricciones

```
constraint="/.+@.+\. [a-z]+/ :  
    Por favor introduce una dirección e-mail correcta" />  
  
<decimalbox format="#,##0.##" />  
  
<intbox constraint="no negative,no zero" />  
  
<datebox constraint="no empty, no future: now or never" />  
  
<datebox constraint="between 20071225 and 20071203" />  
  
<datebox constraint="after 20071225" />
```



# Ejemplo de entrada

The screenshot shows a web form titled "Textbox" with the following elements:


- A text input field containing "Texto por defecto".
- A password input field containing "secret" (represented by dots).
- A text input field with a red border, indicating a validation error.
- A validation message box with a red background and a close button (X) that says "Por favor introduce una dirección e-mail correcta".
- A multi-line text area containing "texto linea1..." and "texto linea2...".

```
<textbox value="Texto por defecto" />
<textbox value="secret" type="password" />
<textbox cols="40"
  constraint="/.+@.+\. [a-z]+/:"
  Por favor introduce una dirección e-mail correcta" />
<textbox rows="5" cols="40">
  <attribute name="value">
texto linea1...
texto linea2...</attribute>
</textbox>
```



# Ejercicio

- Nombre (Texto): Sólo se permiten letras y espacios
- Contraseña (Texto)
- Re-escribir contraseña (Texto)
- Edad (Entero)
- Fecha de nacimiento (Fecha)
- Email (Texto): Sólo se permiten direcciones de correo que terminen en .es

Registro	
Dato	Contenido
Nombre	<input type="text"/>
Contraseña	<input type="password"/>
Re-escribir contraseña	<input type="password"/>
Edad	<input type="text"/>
Fecha de nacimiento:	<input type="text"/> 
E-mail:	<input type="text"/>

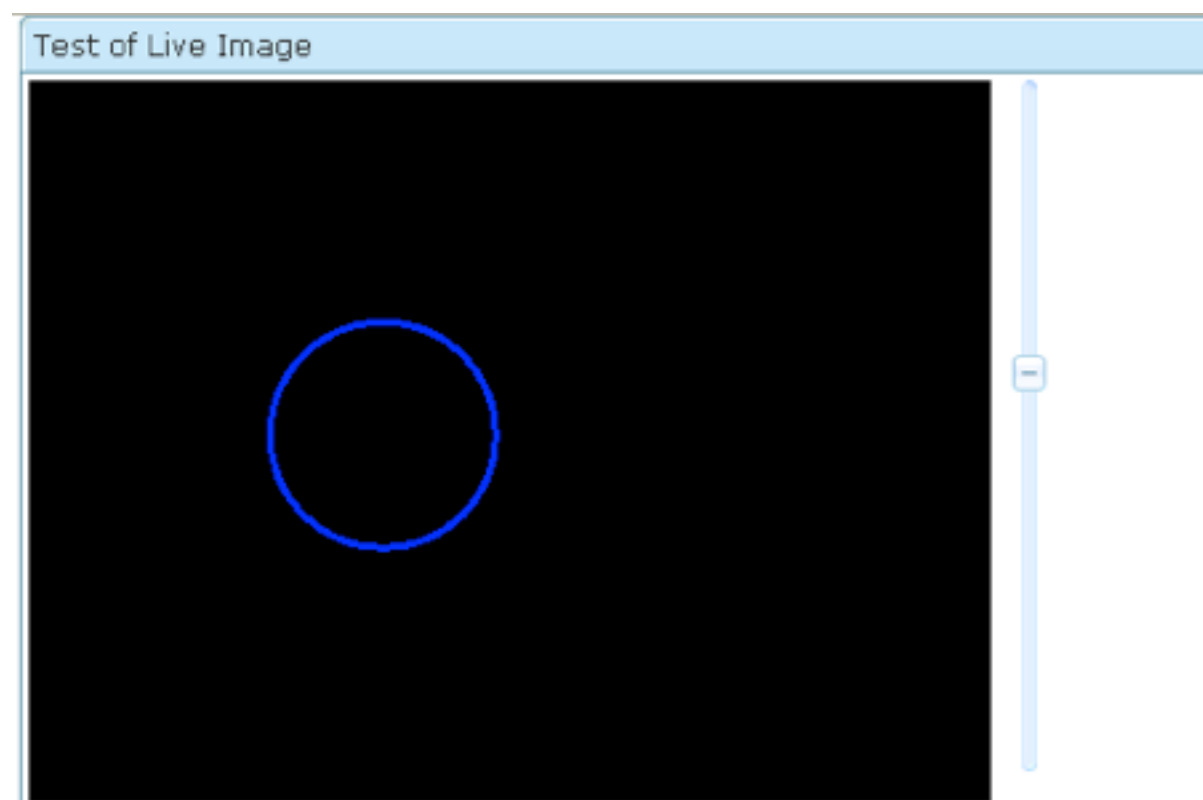
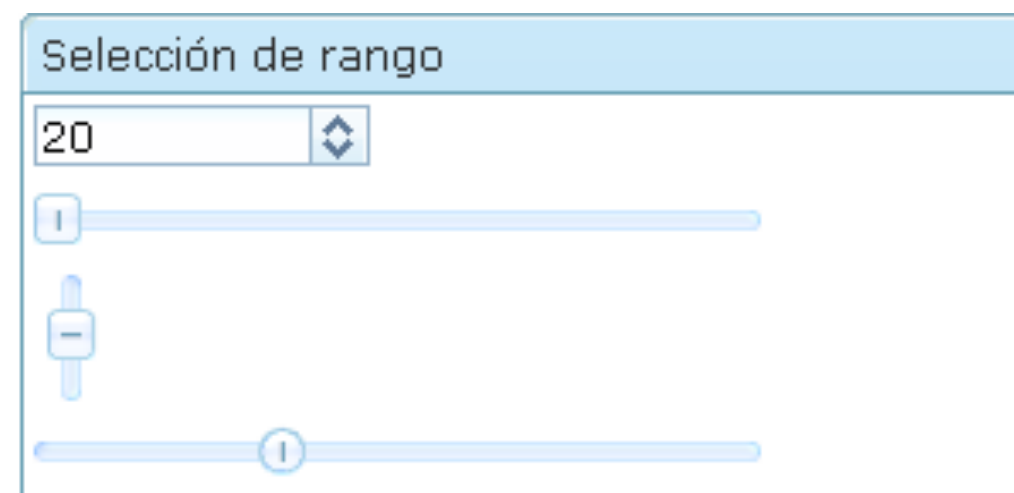


# Plantilla

```
<grid fixedLayout="true">
  <columns>
    <column label="Dato" width="150px" />
    <column label="Contenido" />
  </columns>
  <rows>
    <row>
      Nombre
      <textbox />
    </row>
    ...
  </rows>
</grid>
```

# Rango

- Probar el código de los apuntes
- Combinarlo con el ejemplo de imagen dinámica





# Selección

- Radiogrupop y Radio
- Checkbox

Selección componentes

Elige una fruta

Manzana

Naranja

Pera

Elige varias frutas

Manzana

Naranja

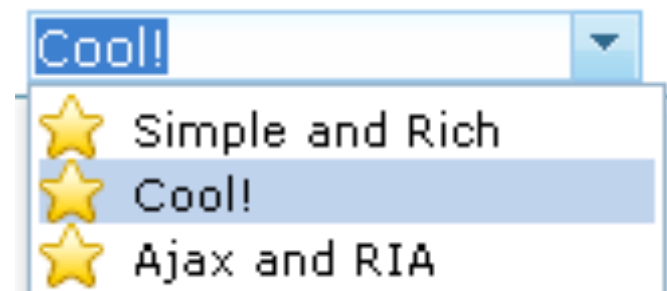
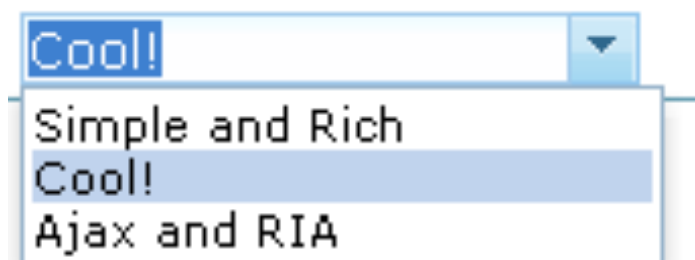
Pera

Has seleccionado: Naranja Pera



# Combobox

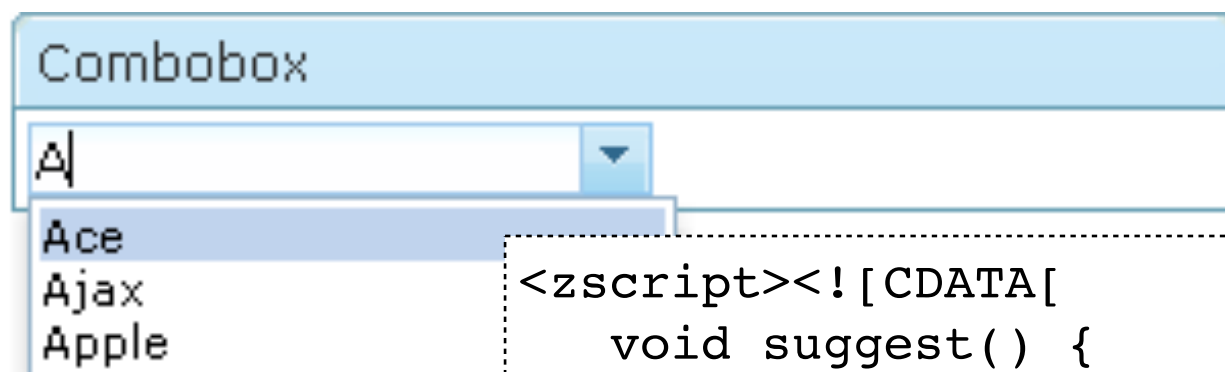
- Combobox y comboitem





# Autocompletado

- Es posible implementar autocompletado en un combobox cambiando dinámicamente sus elementos con el evento onChanging



```
<zscript><![CDATA[
    void suggest() {
        String[] palabras = { "Ace", "Ajax", "Apple", "Best",
            "Blog" };
        combo.getItems().clear();
        for (String str : palabras)
            if (str.startsWith(event.value))
                combo.appendItem(str);
    }
}]>
</zscript>
<combobox id="combo" autodrop="true" onChanging="suggest()" />
```