# Trabajo sobre PrimeFaces

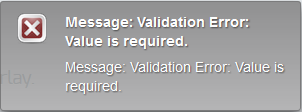
## Componentes

### Messages

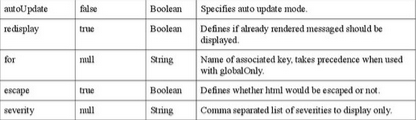
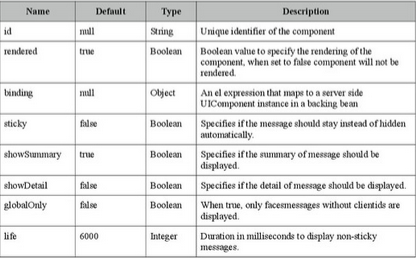
Este framework nos permite la utilización de distintos tipos de mensajes que nos facilitarán mensajes que nos solucionarán la mayor parte de los problemas de una manera simple.

#### Growl

Este tipo de mensajes está basado en el widget de notificaciones de Mac y se utiliza para mostrar ‘FacesMessages’.



##### Atributos



##### Ejemplo de código

<h:form>

<p:growl id=**"growl"** showDetail=**"true"** sticky=**"true"** />

<p:panel header=**"Growl"**>

<h:panelGrid columns=**"2"** cellpadding=**"5"**>

<p:outputLabel for=**"msg"** value=**"Message:"** />

<p:inputText id=**"msg"** value=**"#{growlView.message}"** required=**"true"** />

</h:panelGrid>

<p:commandButton value=**"Save"** actionListener=**"#{growlView.saveMessage}"** update=**"growl"** />

</p:panel>

</h:form>

#### Messages

Estos mensajes son extensiones de los componentes básicos de JSF para mensajes.

Nos sirven para comunicarnos de forma directa con el usuario.



##### Atributos

##### 

##### Ejemplo de código

<p:tooltip />

<h:form>

<p:messages id=**"messages"** showDetail=**"true"** autoUpdate=**"true"** closable=**"true"** />

<p:commandButton value=**"Info"** actionListener=**"#{messagesView.info}"** />

<p:commandButton value=**"Warn"** actionListener=**"#{messagesView.warn}"** />

<p:commandButton value=**"Error"** actionListener=**"#{messagesView.error}"** />

<p:commandButton value=**"Fatal"** actionListener=**"#{messagesView.fatal}"** />

</h:form>

### Multimedia

Estos componentes permiten una mayor interacción gracias a la representación de contenido multimedia en el navegador. Este se presenta de una manera sencilla al usuario y también al programador mediante los siguientes elementos.

#### Barcode

Este componente nos permite mostrar desde códigos de barras hasta códigos QR de una manera sencilla.

##### Atributos

##### 

##### Ejemplo de código

(Código QR)

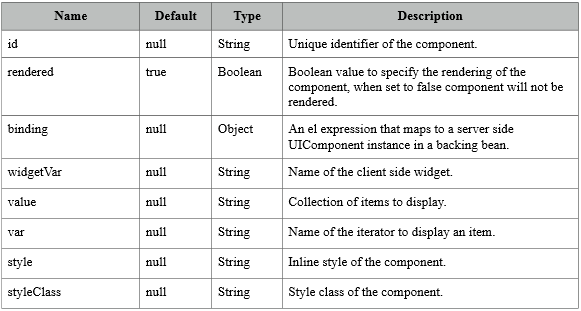
<p:barcode value=**"0123456789"** type=**"qr"**/>

#### ContentFlow



Este component consta de una galería horizontal donde se pueden mostrar los componentes con una animación para el paso entre ellas.

##### Atributos



##### Ejemplo de código

<p:contentFlow value=**"#{imagesView.images}"** var=**"image"**>

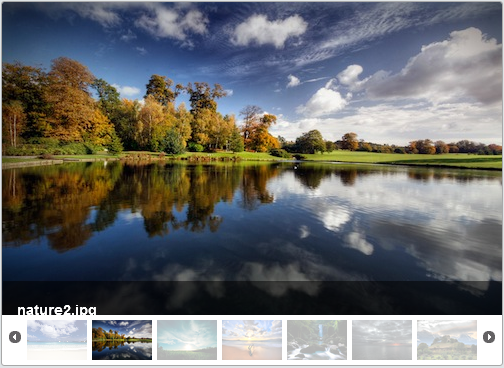
<p:graphicImage name=**"demo/images/nature/#{image}"** styleClass=**"content"** />

<div class=**"caption"**>**#{image}**</div>

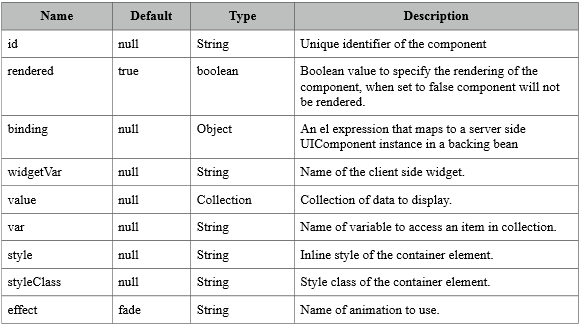
</p:contentFlow>

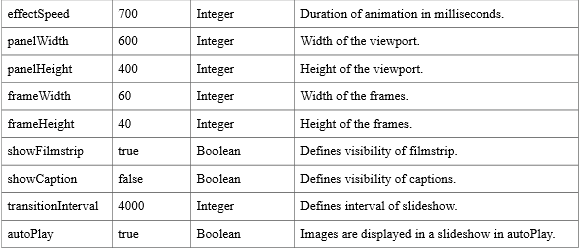
#### Galleria

Este componente es parecido al anterior, pero nos permite mostrar la información como una galería de imágenes.



##### Atributos



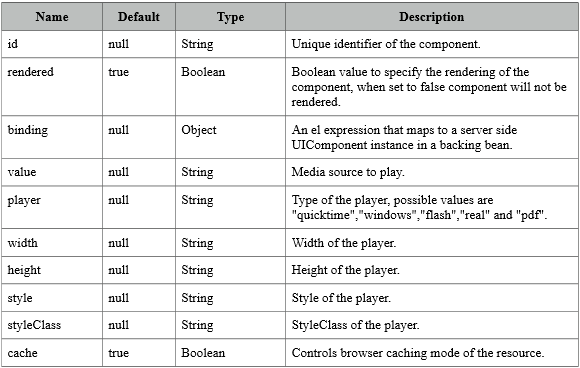


#### Media

Este componente es el reproductor genérico de contenido multimedia empotrable en una página JSF. Soporta varios formatos como Flash, quicktime, Windows media, realplayer y pdf.



##### Atributos



##### Ejemplo de código

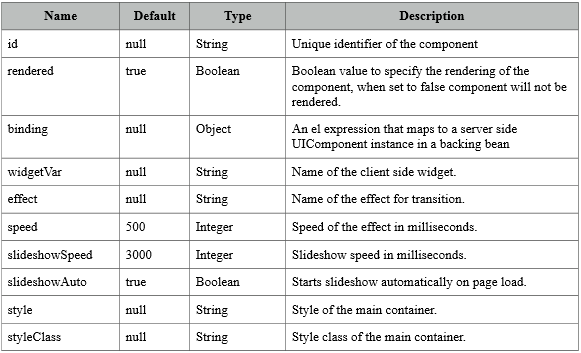
(FlashPlayer)

<p:media value=**"http://www.youtube.com/v/KZnUr8lcqjo"** width=**"420"** height=**"315"** player=**"flash"**/>

#### ImageSwitch

Es otro estilo de galería de imágenes que permite múltiples efectos para el cambio entre imagen.

##### Atributos



##### Ejemplo de código

<p:imageSwitch effect=**"fade"** id=**"fadeEffect"**>

<ui:repeat value=**"#{imagesView.images}"** var=**"image"** id=**"fadeEffectImages"**>

<p:graphicImage name=**"/demo/images/nature/#{image}"** id=**"image"** />

</ui:repeat>

</p:imageSwitch>

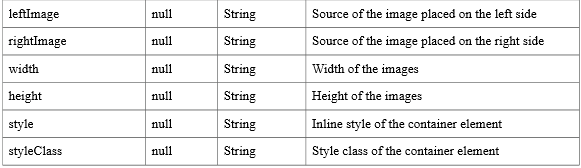
#### ImageCompare

El comparador de imágenes proporciona una interfaz gráfica que permite mostrar similares imágenes para poder ver más claramente similitudes o diferencias.



##### Atributos

### 



##### Ejemplo de código

<p:imageCompare leftImage=**"/resources/demo/images/compare/lara-ps3.png"**

rightImage=**"/resources/demo/images/compare/lara-ps4.png"**

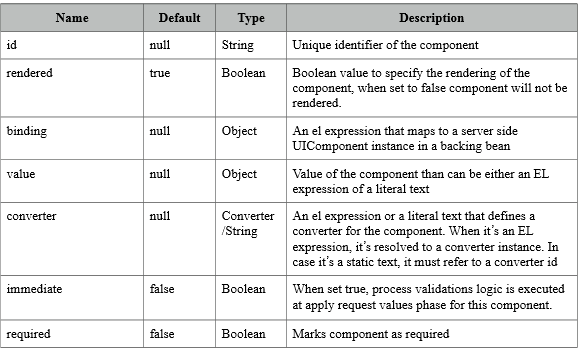
width=**"450"** height=**"435"**/>

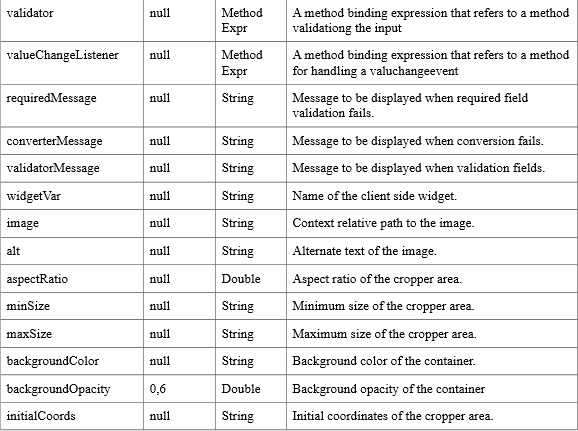
#### Cropper

Este es un component que nos permite extraer una parte concreta de una imagen para crear una nueva.



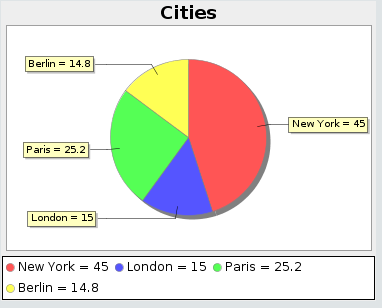
##### Atributos



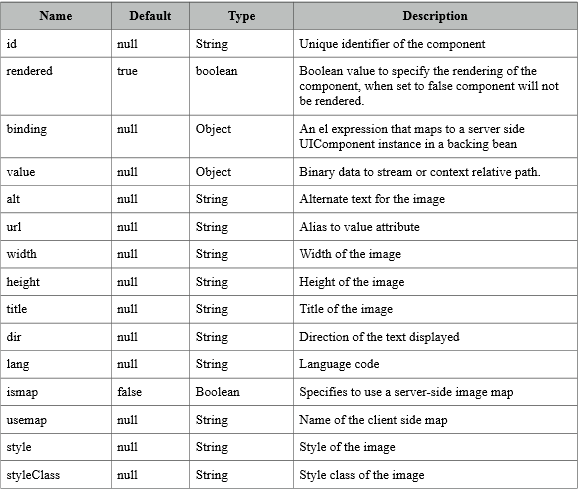


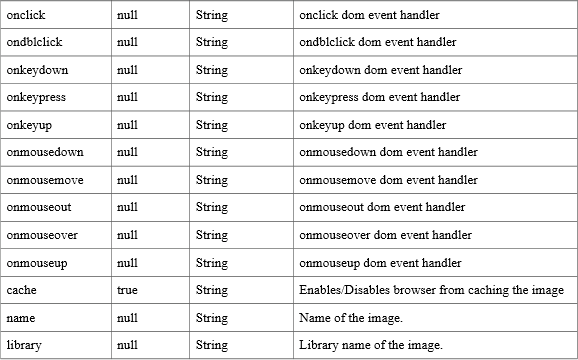
#### GraphicImage

Este es un componente que nos permite presentar imágenes que se crean en tiempo de ejecución o imágenes almacenadas en bases de datos.



##### Atributos



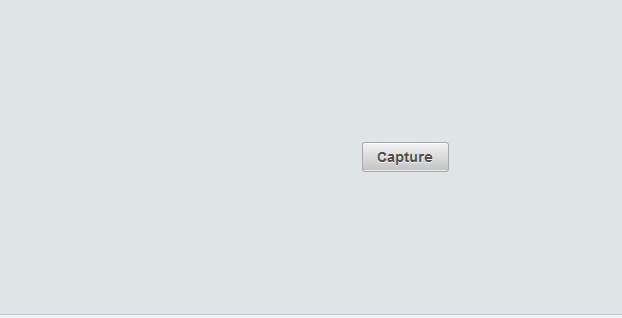


##### Ejemplo de código

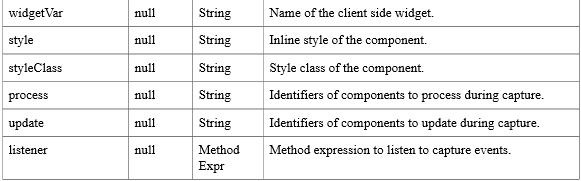
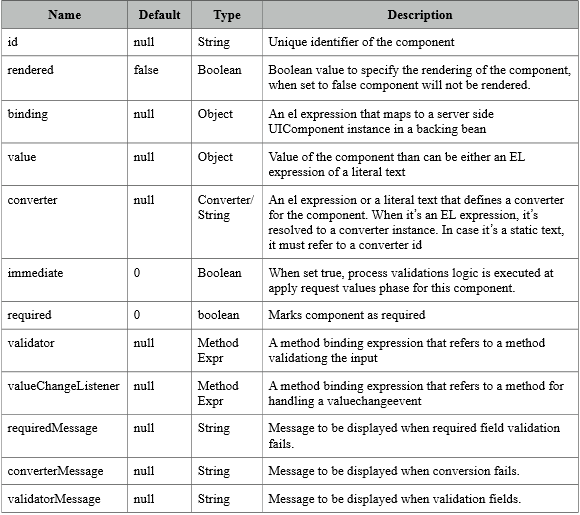
<p:graphicImage value=**"#{graphicImageView.chart}"** />

#### PhotoCam

Es un componente de entrada que permite tomar fotos desde la webcam y enviarlos.



##### Atributos



##### Ejemplo de Código

<h:form>

<h:panelGrid columns=**"3"** cellpadding=**"5"**>

<p:photoCam widgetVar=**"pc"** listener=**"#{photoCamView.oncapture}"** update=**"photo"** />

<p:commandButton type=**"button"** value=**"Capture"** onclick=**"PF('pc').capture()"**/>

<p:outputPanel id=**"photo"**>

<p:graphicImage name=**"demo/images/photocam/#{photoCamView.filename}.jpeg"** rendered=**"#{not empty photoCamView.filename}"**/>

</p:outputPanel>

</h:panelGrid>

</h:form>

### File

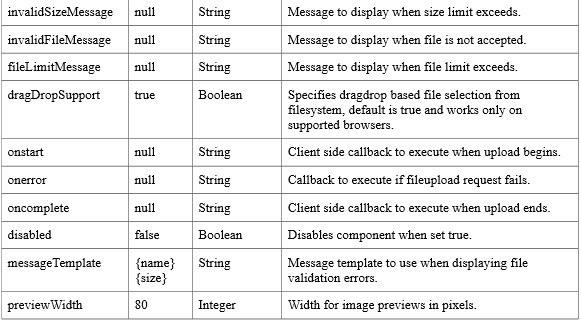
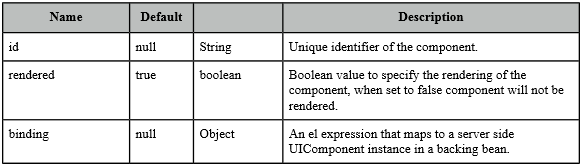
Estos componentes permiten la interacción mediante datos entre el usuario y la aplicación permitiendo tanto la subida como la descarga de datos.

#### FileUpload

Este es un simple componente que utiliza el método nativo del navegador para subir los archivos.



##### Atributos



##### Ejemplo de Código

<h:form enctype=**"multipart/form-data"**>

<p:growl id=**"messages"** showDetail=**"true"** />

<p:fileUpload value=**"#{fileUploadView.file}"** mode=**"simple"** skinSimple=**"true"**/>

<p:commandButton value=**"Submit"** ajax=**"false"** actionListener=**"#{fileUploadView.upload}"** disabled=**"true"** />

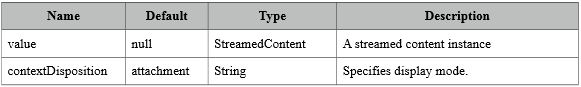
</h:form>

#### Download

Este componente se utiliza para que los contenidos subidos puedan ser descargados por el usuario.



##### Atributos



##### Ejemplo de código

<p:dialog modal=**"true"** widgetVar=**"statusDialog"** header=**"Status"** draggable=**"false"** closable=**"false"** resizable=**"false"**>

<p:graphicImage name=**"/demo/images/ajaxloadingbar.gif"** />

</p:dialog>

<h:form>

<p:commandButton value=**"Download"** ajax=**"false"** onclick=**"PrimeFaces.monitorDownload(start, stop);"** icon=**"ui-icon-arrowthick-1-s"**>

<p:fileDownload value=**"#{fileDownloadView.file}"** />

</p:commandButton>

</h:form>

<script type=**"text/javascript"**>

function start() {

PF('statusDialog').show();

}

function stop() {

PF('statusDialog').hide();

}

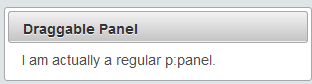
</script>

### DragDrop

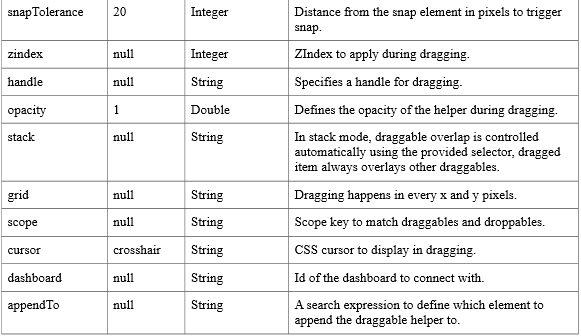
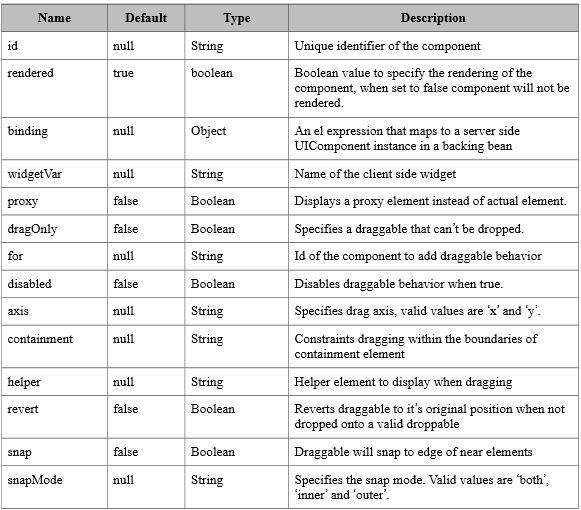
Estos elementos se basan en la característica de ser arrastrables por el usuario, permitiendo una mayor interacción entre la aplicación y el usuario. También permiten la creación de unos elementos propios.

#### Draggable

Estos son componentes con la capacidad de ser arrastrados por el usuario.



##### Atributos



##### Ejemplo de código

<p:panel id=**"pnl"** header=**"Draggable Panel"**>

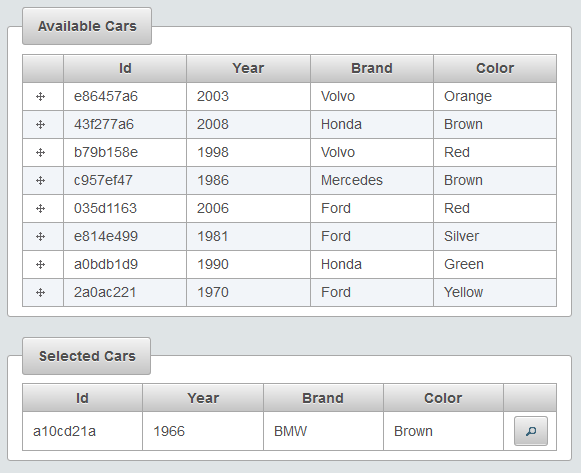
<h:outputText value=**"I am actually a regular p:panel."** />

</p:panel>

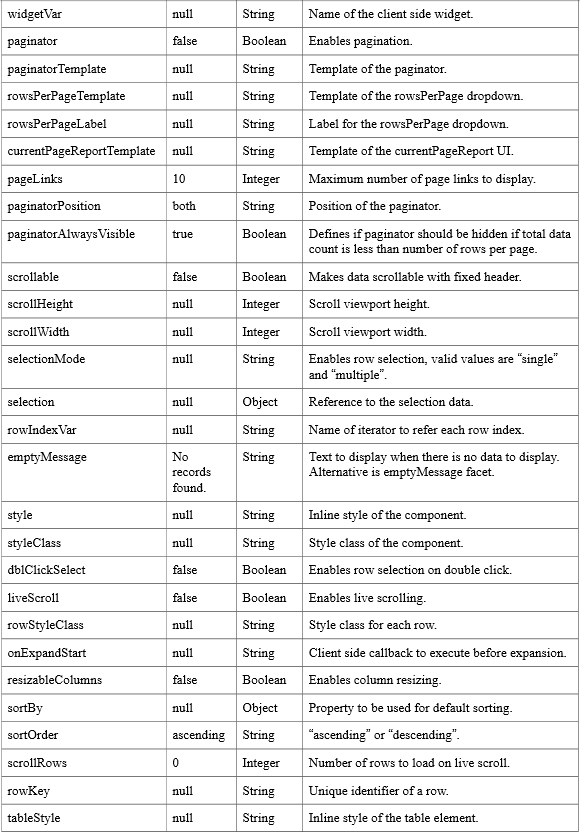
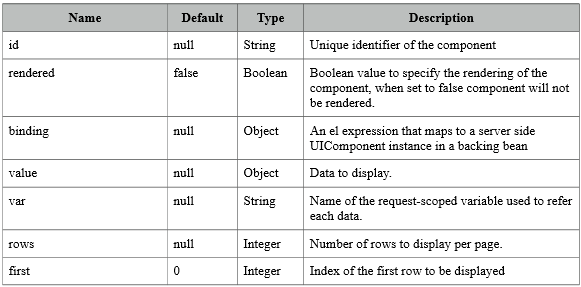
<p:draggable for=**"pnl"** />

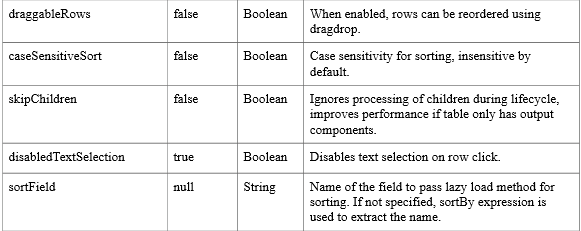
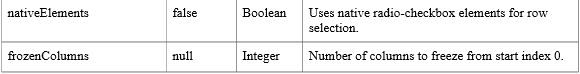
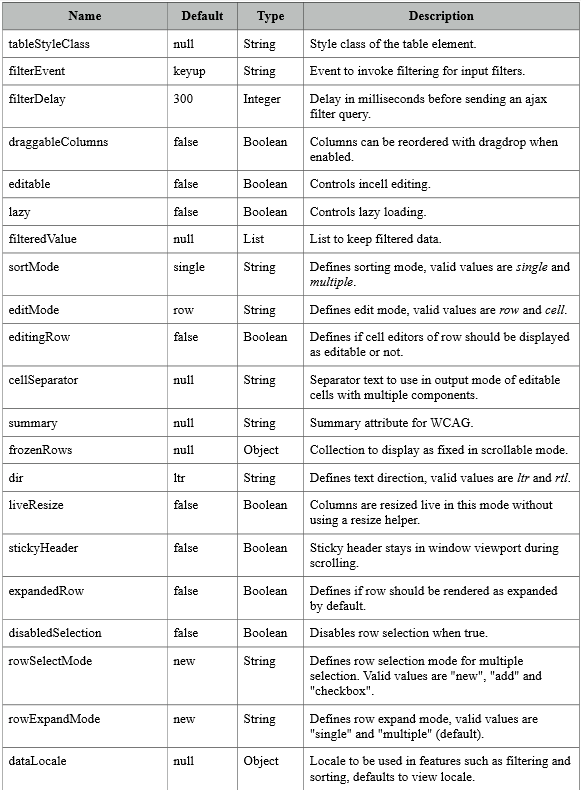
#### DataTable

Los elementos arrastrables amplian la utilidad de las tablas, por ejemplo cambiando de sitio ciertos elementos de una table.



##### Atributos





##### Ejemplo de código

<p:dataTable id=**"availableCars"** var=**"car"** value=**"#{dndCarsView.cars}"**>

<p:column style=**"width:20px"**>

<h:outputText id=**"dragIcon"** styleClass=**"ui-icon ui-icon-arrow-4"** />

<p:draggable for=**"dragIcon"** revert=**"true"** helper=**"clone"**/>

</p:column>

<p:column headerText=**"Id"**>

<h:outputText value=**"#{car.id}"** />

</p:column>

<p:column headerText=**"Year"**>

<h:outputText value=**"#{car.year}"** />

</p:column>

<p:column headerText=**"Brand"**>

<h:outputText value=**"#{car.brand}"** />

</p:column>

<p:column headerText=**"Color"**>

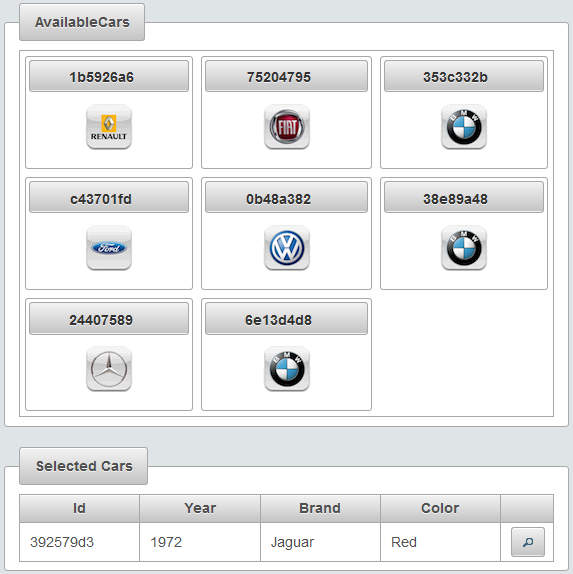
<h:outputText value=**"#{car.color}"** />

</p:column>

</p:dataTable>

#### DataGrid

Este mismo concepto puede aplicarse a otros componentes y de ello surgen los **DataGrid**, que son rejillas donde pueden arrastrarse otro tipo de objetos para, por ejemplo, ampliar su información.



##### Ejemplo de código

En este caso la sintaxis es igual que la anterior pero incluyendo el nuevo elemento

<p:dataGrid id=**"availableCars"** var=**"car"** value=**"#{dndCarsView.cars}"** columns=**"3"**>

<p:panel id=**"pnl"** header=**"#{car.id}"** style=**"text-align:center"**>

<h:panelGrid columns=**"1"** style=**"width:100%"**>

<p:graphicImage name=**"/demo/images/car/#{car.brand}.gif"** />

</h:panelGrid>

</p:panel>

<p:draggable for=**"pnl"** revert=**"true"** handle=**".ui-panel-titlebar"** stack=**".ui-panel"**/>

</p:dataGrid>