PrimeFaces

Trabajo SDI

Contenido

[Ajax Core 3](#_Toc419059099)

[Listener 3](#_Toc419059100)

[Process 3](#_Toc419059101)

[Dropdown 3](#_Toc419059102)

[Selector 3](#_Toc419059103)

[Poll 3](#_Toc419059104)

[Status 3](#_Toc419059105)

[Events 4](#_Toc419059106)

[Validation 4](#_Toc419059107)

[Partial Submit 4](#_Toc419059108)

[Fragment 4](#_Toc419059109)

[Input 4](#_Toc419059110)

[Autocompletar 4](#_Toc419059111)

[BooleanButton 4](#_Toc419059112)

[Calendar 5](#_Toc419059113)

[SelectOneMenu 5](#_Toc419059114)

[InputText 5](#_Toc419059115)

[SelectOneListbox 5](#_Toc419059116)

[SelecRadio 5](#_Toc419059117)

[Password 5](#_Toc419059118)

[Rating 6](#_Toc419059119)

[InputTextArea 6](#_Toc419059120)

[Editor 6](#_Toc419059121)

[InputMask 6](#_Toc419059122)

[Keyboard 6](#_Toc419059123)

[Button 7](#_Toc419059124)

[Data 7](#_Toc419059125)

[Carousel 7](#_Toc419059126)

[DataList 7](#_Toc419059127)

[DataTable 8](#_Toc419059128)

[PickList 8](#_Toc419059129)

[GMap 9](#_Toc419059130)

[Paneles 9](#_Toc419059131)

[Wizard 9](#_Toc419059132)

[TabView 10](#_Toc419059133)

[Fieldset 11](#_Toc419059134)

[OutputPanel 12](#_Toc419059135)

[Overlay 13](#_Toc419059136)

[LightBox 13](#_Toc419059137)

[Confirm Dialog 14](#_Toc419059138)

[Tooltip 14](#_Toc419059139)

# Ajax Core

Mediante Ajax podemos realizar cambios sobre nuestra página Web sin necesidad de recargarla. Algunos de los componentes sobre los que podemos utilizar Ajax con PrimeFaces son los siguientes:

## Listener

Permite que un método de java sea invocado con Ajax usando la opción “listener”.

## Process

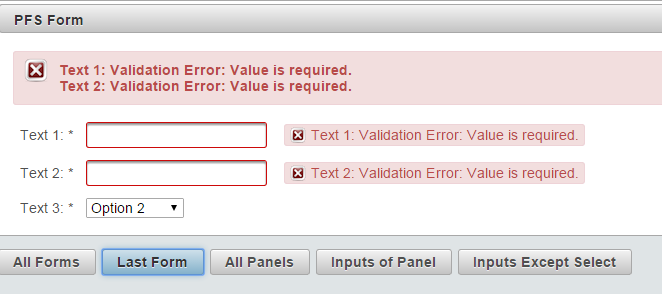
Permite un procesamiento parcial de alguna de los componentes de una Web ignorando otros.

## Dropdown

Con este componente podemos crear una dependencia entre varios desplegables.

## Selector

Este componente permite usar la API jQuery Selector para hacer referencias a componentes.



## Poll

Este componente realiza llamadas a métodos periódicamente.

## Status

Mediante AjaxStatus se proporciona información acerca de la solicitud Ajax que se está realizando en ese momento.

## Events

Permite realizar eventos compatibles en los componentes.

## Validation

Este componente permite realizar una comprobación en el servidor y actualizarse con el resultado.



## Partial Submit

Permite reducir el tráfico de la red al cargar una página. En páginas grandes con muchos componentes de entrada, carga en primer lugar los más rápidos.

## Fragment

Permite fragmentar componentes y cargar la parte que se desee.

# Input

Podemos personalizar los inputs de múltiples formas con PrimeFaces:

## Autocompletar

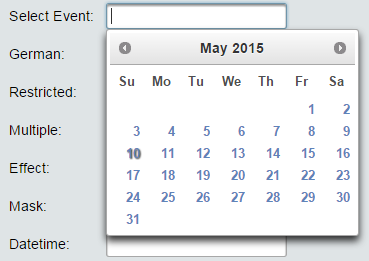
Permite mostrar sugerencias para completar una palabra que se está escribiendo en una página Web.

## BooleanButton

Permite crear un botón con un valor boolean que cambie al pulsarlo.

## Calendar

Es un componente que permite visualizar un calendario y seleccionar una fecha en él para introducirla en un input.



## SelectOneMenu

Muestra un menú con diferentes opciones para seleccionar, permite elegir el tipo de menú que deseemos: Basic, Grouping, Editable, Advanced…

## InputText

Campo de texto que se puede personalizar el estilo.

## SelectOneListbox

Componente que muestra una lista de opciones que se pueden seleccionar.

## SelecRadio

Permite elegir un elemento de una colección.

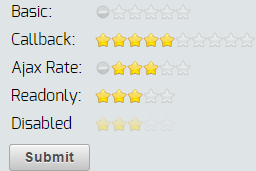
## Password

Es un campo para introducir contraseñas al que se le puede añadir un indicador de seguridad, para saber cómo de segura es la contraseña que hemos introducido.



## Rating

Es un componente para introducir valoraciones mediante estrellas.



## InputTextArea

Permite crear componentes textarea en los que se pueden añadir opciones como auto completado, cambiar el tamaño, contadores, temas…

## Editor

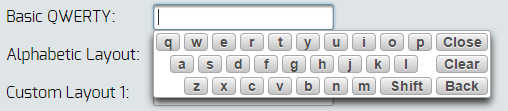
Podemos introducir en nuestra Web un editor de texto con múltiples opciones para editar el texto, con barra de herramientas y posibilidad de configurar los controles.

## InputMask

Permite mostrar el formato en el que debemos introducir el texto.

## Keyboard

Permite introducir un teclado en la pantalla al seleccionar un elemento de introducción de texto.



# Button

Este componente es una extensión del componente de JSF “h:button”. Con esta opción podemos modificar múltiples características del botón que de la otra manera no podíamos, ponerle un icono, cambiar el skin…

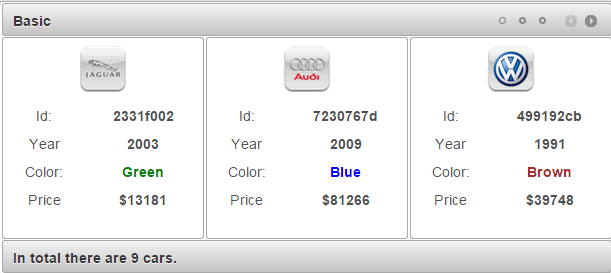


# Data

Con primefaces tenemos múltiples opciones de mostrar datos en una página Web:

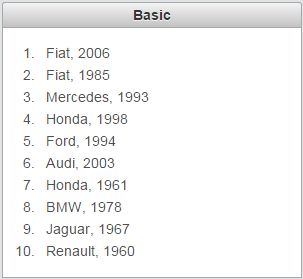
## Carousel

Con este componente podemos mostrar datos usando un slide al que le podemos dar la forma que queramos (básico, tabla, avanzado).



## DataList

Con este componente podemos mostrar los datos en un layout con forma de lista.



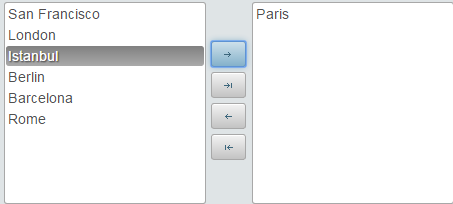
## DataTable

Este componente permite mostrar los datos en una tabla tabulada.



## PickList

Este componente permite seleccionar datos de una lista inicial y añadirlos a otra.



## GMap

Con este componente podemos mostrar en nuestra Web un mapa con una posición.

# Paneles

Los componentes de paneles son útiles para agrupar la información que mostremos en nuestra web, así como los formularios, logrando no saturar con demasiado contenido al visitante.

## Wizard



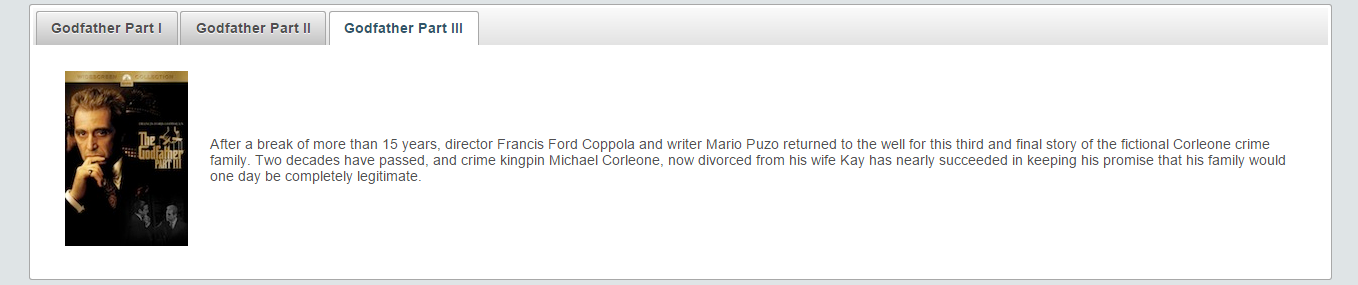
Con este componente podemos agrupar el contenido de largos formularios por categorías. De esta forma nuestro usuario no se agobiará si tiene que rellenar muchos campos. Una de sus ventajas es la validación por partes. Es decir, en la imagen mostrada por ejemplo, no te dejará acceder a rellenar “Address” sin antes haber validado correctamente los campos de “Personal”.

El uso de esta etiqueta es bastante sencillo. Siguiendo el ejemplo mostrado en la imagen, su correspondiente .xhtml sería el siguiente:



Como podemos observar, tan solo sería necesario especificar las agrupaciones de los campos del formulario, así como sus condiciones de validación, encargándose PrimeFaces de realizar el resto del trabajo.

## TabView



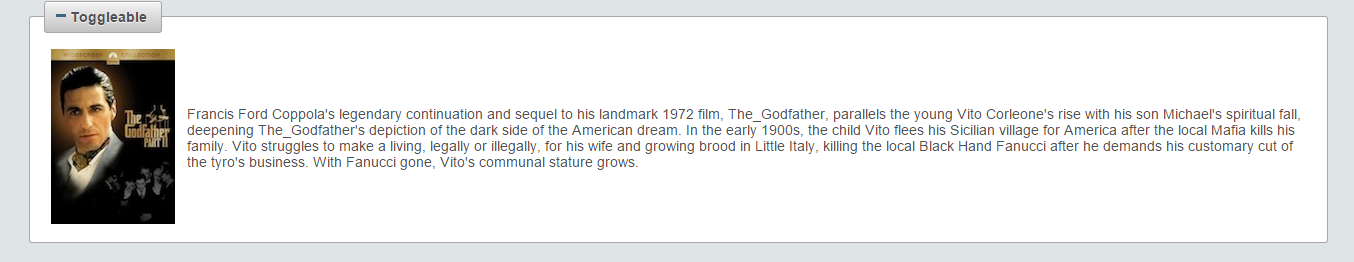
Esta etiqueta agrupa la información que antes mostraríamos como una lista en pestañas. Gracias a ella logramos reducir el contenido mostrado de cada vez al visitante, y dependiendo de los parámetros escogidos, incluso el ancho de banda consumido por la página al cargar las vistas sólo cuando el usuario pinche en ellas.

Su funcionamiento al igual que el resto de componentes de este framework es muy sencillo:



Simplemente, instanciamos el conjunto de pestañas con *<p:tabView>,* y definimos cada una con *<p:tab>*, pudiendo añadir el contenido que queramos dentro de ellas.

## Fieldset



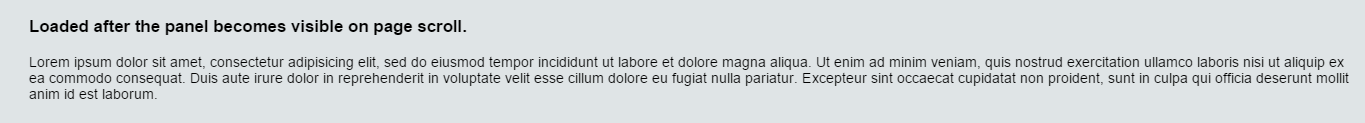
Con <p:fieldset> logramos resaltar cierto contenido, del mismo modo que podemos darla opción al usuario de ocultarlo si este lo considerase necesario. Un buen ejemplo aparte del aquí mostrado, podría ser mostrar código al usuario convenientemente formateado dentro de un tutorial.

El funcionamiento de esta etiqueta no puede ser más simple:



El componente se instancia con <p:fieldset>, y con los atributos toggleable, y toggleSpeed definimos respectivamente si el usuario puede ocultar el contenido de este, y la velocidad de la animación respectivamente.

## OutputPanel



Este componente permite agregar contenido después la descarga de la página. Gracias a él podemos reducir el ancho de banda consumido por el servidor, además de incrementar la velocidad de carga de la web. Es muy interesante, ya que los visitantes no siempre cuentan con una buena conexión, y pueden evitar la carga de partes de la web que en algunas circunstancias no les son relevantes.

Al igual que las anteriores su uso no tiene más complicación:

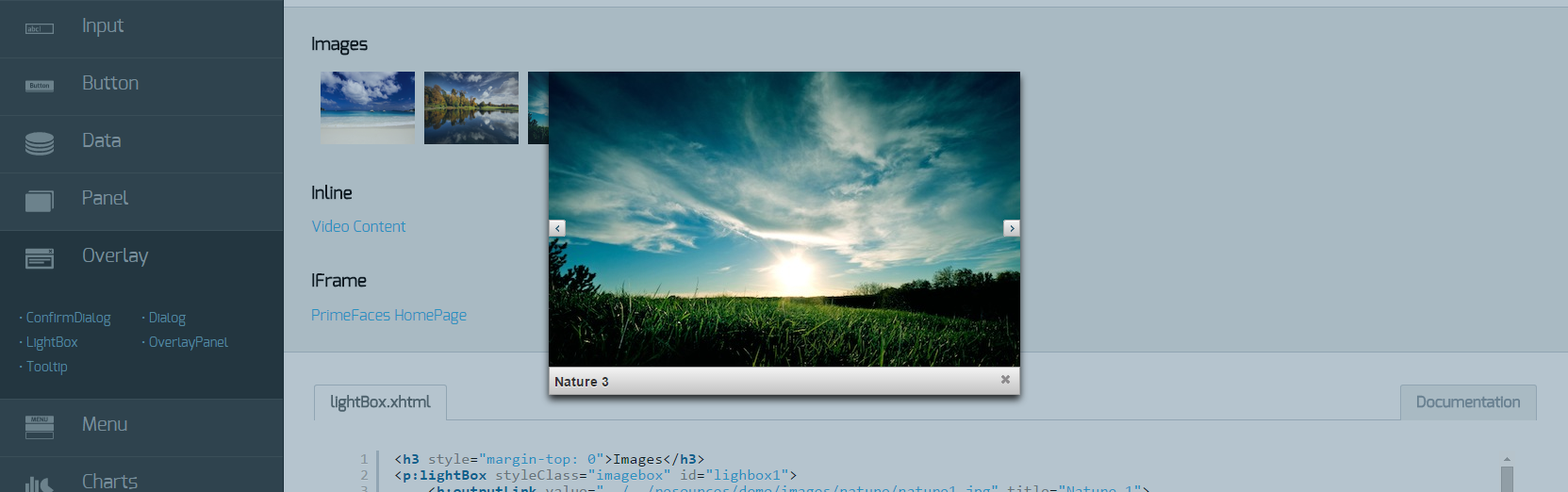


Se instancia con <p:outputPanel>, y con los atributos deferred, y deferredMode, establecemos que se cargue más tarde que el resto de la página, y cuando exactamente ocurrirá esto respectivamente.

# Overlay

En esta categoría nos encontramos componentes con los que generar contenido extra sin romper la estructura de la página, y que pueden ser útiles para hacer la web más amigable de cara al usuario, así como para mejorar su rendimiento y aspecto.

## LightBox



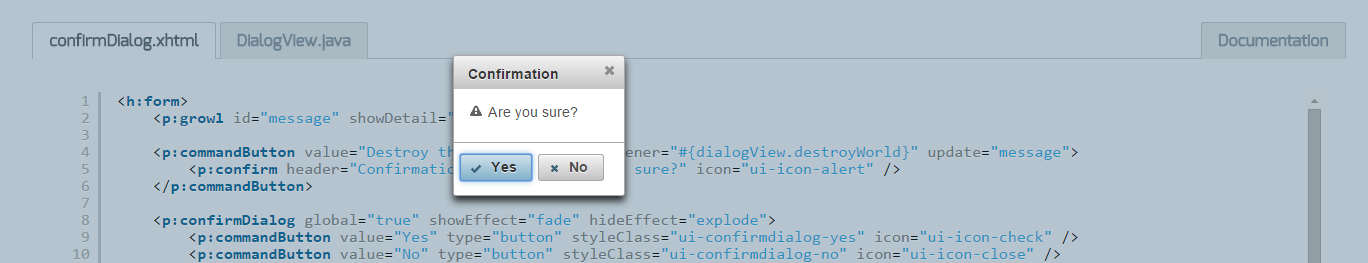
Este elemento nos permite integrar contenido multimedia en un modal. Es una buena forma de mostrar contenido que ocupará un gran espacio en la pantalla, y que concentrará nuestra atención totalmente mientras es visualizado, como por ejemplo videos o imágenes en alta resolución.

En el siguiente ejemplo vemos su funcionamiento con una galería de imágenes:



Para usarlo, como vemos sólo debemos instanciar el elemento con <p:lightBox>, y a continuación añadir los <h:outputLink> con el contenido a mostrar dentro de ellos.

## Confirm Dialog



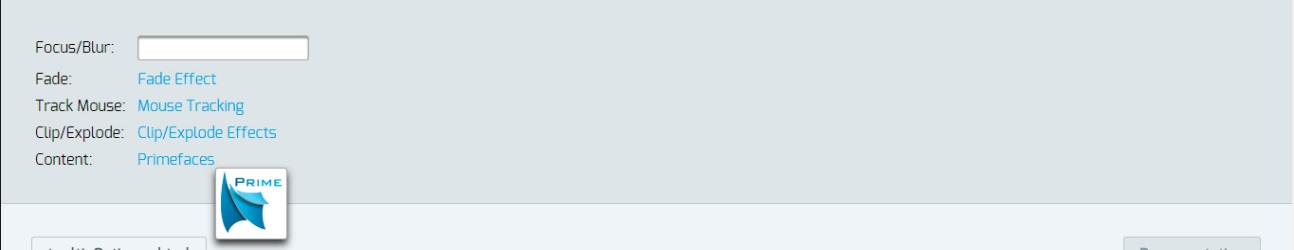
Con este componente, mostraremos al usuario un panel de confirmación al realizar alguna acción “peligrosa”, como podría ser borrar su usuario de la web. Con él, evitaremos clicks accidentales en botones, así como dar una oportunidad a que el usuario recapacite al realizar acciones que no tengan vuelta atrás.

En este ejemplo se muestra un ejemplo de botón que activaría este diálogo:



Si nos fijamos en el código necesario para agregarlo a nuestra web, necesitaremos un commandButton que al pulsarlo activará el diálogo de confirmación. Dentro este agregaremos el mensaje a mostrar al usuario con <p:confirm>, y finalmente un <p:confirmDialog> junto a este botón en el que agregaremos las distintas opciones a ofrecer al usuario, así como efectos o animaciones que queramos que este tenga. Este <p:confirmDialog> puede ser global y ser usado en toda la página, activado por distintos botones.

## Tooltip



Este componente agregará funcionalidad a los ya conocidos tooltips, permitiéndonos agregar imágenes, efectos, o que se muevan junto con el cursor del ratón.

Una muestra del código es la siguiente:



Agregarlo es muy fácil, y ayudará a hacer nuestra web más fácil de usar por el usuario. Con <p:tooltip> instanciamos el elemento, y con el atributo “for” le indicamos a que otro objeto pertenece.