Es evidente que uno de los standares que muchas aplicaciones JEE utilizan es JSF .Aunque es un standard tiene la peculiaridad de que no es muy sencillo de entender ya que el ciclo de vida de las páginas puede ser muy complejo .En estas situaciones nos encontramos que a veces hay que usar AJAX con lo todo se complica un poco mas . Vamos a intentar comentarlo y aclarar las dudas. Supongamos que disponemos del siguiente formulario .


Este formulario esta compuesto de una página JSF y un ManagedBean .Vamos a mostrar el código de ambos.

```
<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
 <html xmlns="http://www.w3.org/1999/xhtml"
 xmlns:h="http://java.sun.com/jsf/html"
 xmlns:f="http://java.sun.com/jsf/core"
 >
 <h:head>
 <title>Bienvenido</title>
 </h:head>
 <h:body>
```

```
<h:form>
<h3>Introduce tu nombre</h3>
<table>
<tr>
<td>Nombre:</td>
<td><h:inputText value="#{personaBean.nombre}"/></td>
</tr>
</table>
<p><h:commandButton value="Aceptar" action="aceptar"/></p>
<h:outputText value="#{personaBean.nombre}"/>
</h:form>
</h:body>
</html>
```

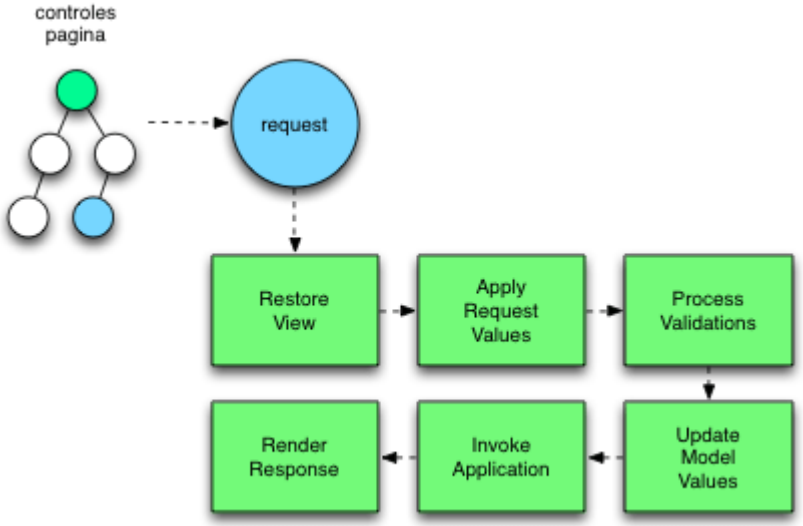
```
package com.arquitecturajava;

import javax.faces.bean.ManagedBean;

@ManagedBean
public class PersonaBean {

 private String nombre;

public String getNombre() {
 return nombre;
 }
```

```
public void setNombre(String nombre) {
 this.nombre = nombre;
 }
}
```

Si rellenamos el nombre y pulsamos el boton de Aceptar obtendremos el siguiente resultado.


## JSF y Ciclo de vida

¿Ahora bién como ha funcionado realmente la ejecución del boton? . Bueno lo que ha sucedido es que JSF ha enviado una petición al servidor con la pagina cargada con los controles iniciales y se ha ejecutado en el servidor todo el ciclo de vida de JSF .

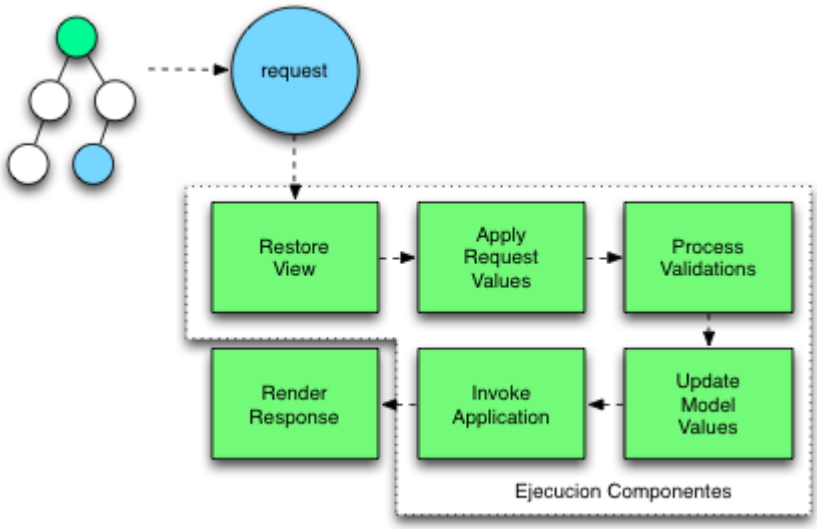



Sin entrar a detalle a explicar el ciclo podemos decir lo siguiente de las distintas fases del ciclo de vida:

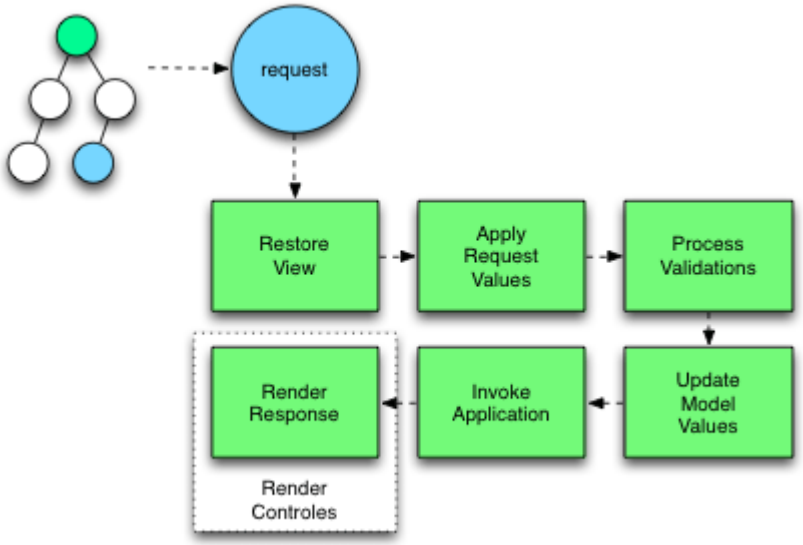
1. RestoreView : Restaura  los controles de la vista y los prepara para las modificaciones
2. Apply Request Values : Aplica a los controles los valores que nosotros hubieramos modificado en la página HTML
3. Process Validations : Aplica las validaciones pertinentes.
4. Update Model Values :Actualiza los datos del Modelo
5. Invoke Application : Invoca La lógica de negocio pertinente que quede
6. Render Response : Renderiza el resultado de todo el proceso y lo envia al cliente .

## JSF Ajax (Ejecución y Render)

Para entender como funciona AJAX en JSF hay que entender que aunque el ciclo de vida de JSF es muy complejo se puede dividir en dos grupos fundamentales . A estos grupos se les denomina Ejecución y Renderizado .Vamos a verlo un poco mas a detalle.

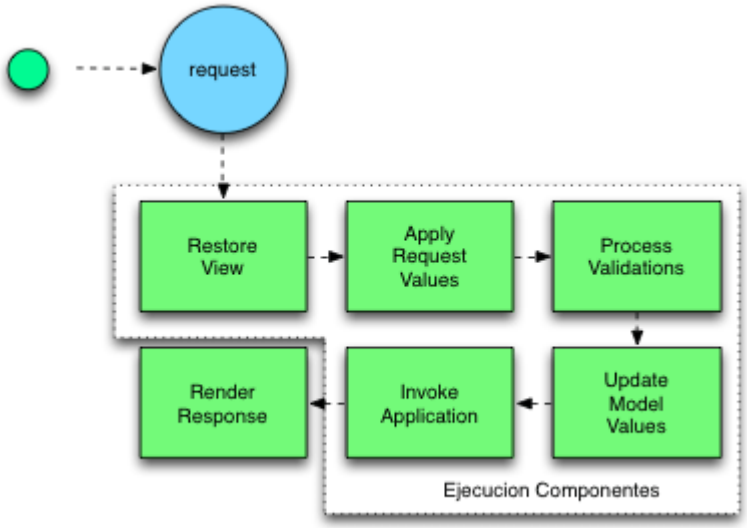



La parte de ejecución hace referencia como su nombre indica a todo el ciclo de vida que esta ligado a la propia ejecución de los componentes dejando de lado la ultima fase, la fase de renderizado. En cambio la segunda fase es opuesta a la primera solo se encarga del renderizado.

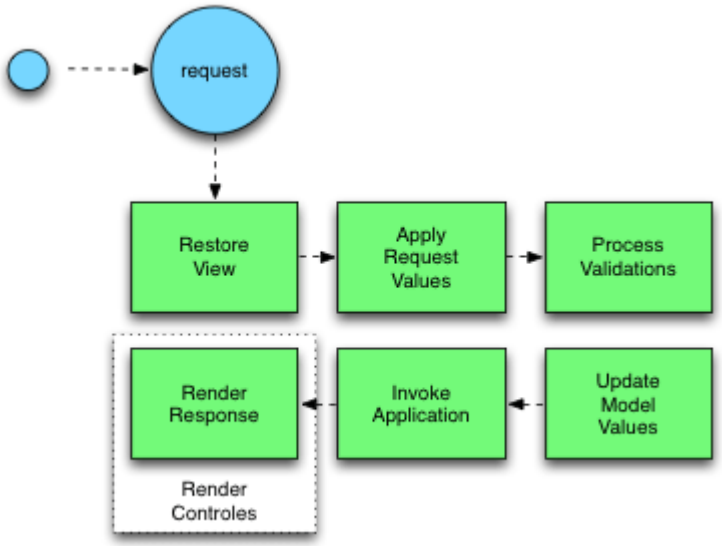



JSF y AJAX nos permiten aplicar estas fases a controles diferentes y obtener un resultado .

Por ejemplo podemos hacer que el control de h:inputText pase por todas las fases de ejecución.




Pero que no le haga falta pasar por la fase de render ya que esta renderizado correctamente .Ahora podemos hacer que el control h:outputText funcione de la forma contraria y no pase por ninguna fase salvo la ultima.



Vamos a verlo en código para terminar de aclarar dudas.

```
<?xml version="1.0" encoding="UTF-8"?></pre>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
 <html xmlns="http://www.w3.org/1999/xhtml"
 xmlns:h="http://java.sun.com/jsf/html"
 xmlns:f="http://java.sun.com/jsf/core" >
 <h:head>
 <title>Welcome</title>
 </h:head>
 <h:body>
<h:form>
<h3>Introduce tu nombre</h3>
<table>
<tr>
<td>Nombre:</td>
<td>
<h:inputText value="#{personaBean.nombre}">
 <f:ajax event="keyup" execute="@this" render="zona"/>
</h:inputText>
</td>
</tr>
</table>
<p><h:commandButton value="Aceptar" action="aceptar"/>
</p>
<h:outputText id="zona" value="#{personaBean.nombre}"/>
</h:form>
</h:body>
</html>
```

<pre>

Lo importante de todo este código son las siguientes lineas

```
<h:inputText value="#{personaBean.nombre}">
 <f:ajax event="keyup" execute="@this" render="zona"/>
 </h:inputText>
```

En ellas concretamente en <f:ajax> se especifica el evento keyUp  y  como el control actual (@this) que es el h:inputText  necesita de la fase de ejecución .Mientras por el otro lado el control zona (identificador del outputText) solo necesita la fase de render. De esta forma segun vamos pulsando teclas en el inputText via ajax se actualiza el outputText.