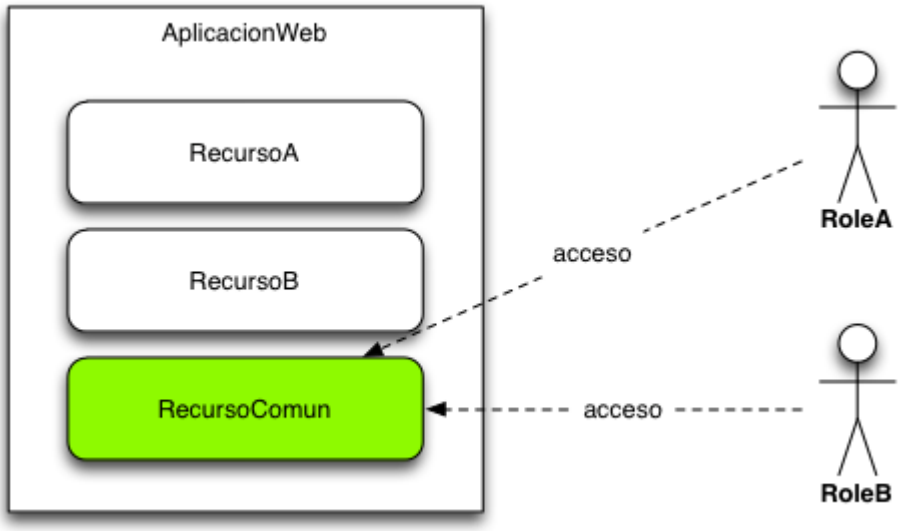
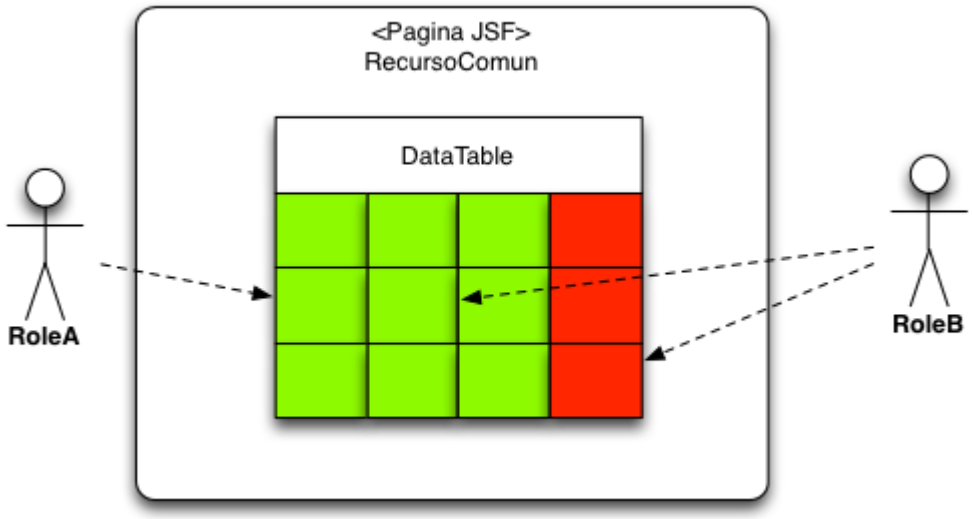
En muchas ocasiones usamos JSF como framework de capa de presentación y normalmente JAAS nos cubre nuestras necesidades de seguridad básicas. Ahora bien hay situaciones que tienen tratamientos especiales. Una de ellas es cuando queremos controlar la seguridad no a nivel de carpeta sino a nivel de página ya que disponemos de un recurso que dos roles comparten.



En este caso podemos encontrarnos con la situación de que cada uno de los roles tiene que ver una información distinta de la página . Por ejemplo en el caso de usar un DataTable un rol ve solo un subconjunto de columnas mientras que el otro rol ve todo.



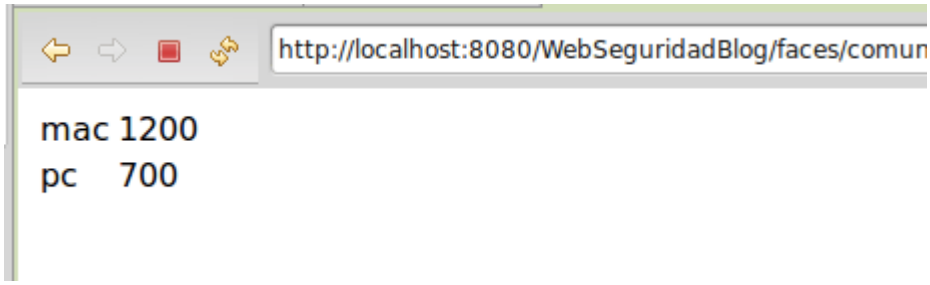
En esta casuística podemos proteger el recurso para que los dos roles tengan acceso via web.xml .

```
<security-constraint>
 <web-resource-collection>
 <web-resource-name>comun</web-resource-name>
 <url-pattern>
 *.xhtml
 </url-pattern>
 <http-method>GET</http-method>
 <http-method>POST</http-method>
 </web-resource-collection>
 <auth-constraint>
 <role-name>ROLEA</role-name>
 <role-name>ROLEB</role-name>
 </auth-constraint>
 </security-constraint>
```

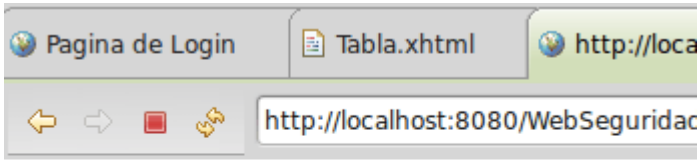
Ahora bien eso no impedirá a los dos roles acceder a la información de la tabla de la misma manera. Si nosotros deseamos limitar el acceso a esta información. Deberemos apoyarnos en los atributos rendered que vimos hace unos post y en las capacidades que tiene JSF 2 y Expression languaje para invocar métodos desde páginas JSF.

```
<h:dataTable value="#{beanFactura.listaFacturas}" var="factura">
 <h:column>
 #{factura.concepto}
 </h:column>
 <h:column rendered="#{request.isUserInRole('ROLEA')}">
 #{factura.importe}
 </h:column>
 </h:dataTable>
```

De esta forma el usuario con el ROLE A verá la tabla completa.



Mientras que el usuario con el ROLE B verá


Pagina de Login
Tabla.xhtml
http://loca
http://localhost:8080/WebSeguridac

mac
pc