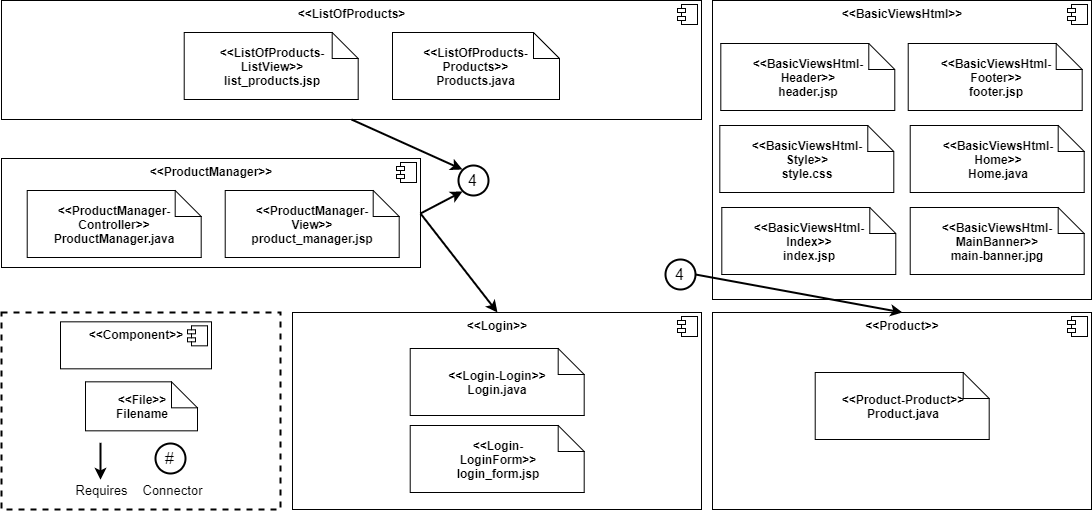
**Implementación de componentes de ClothingStores SPL:**

La figura 1 muestra cómo están divididos los componentes del pool de componentes y sus relaciones.



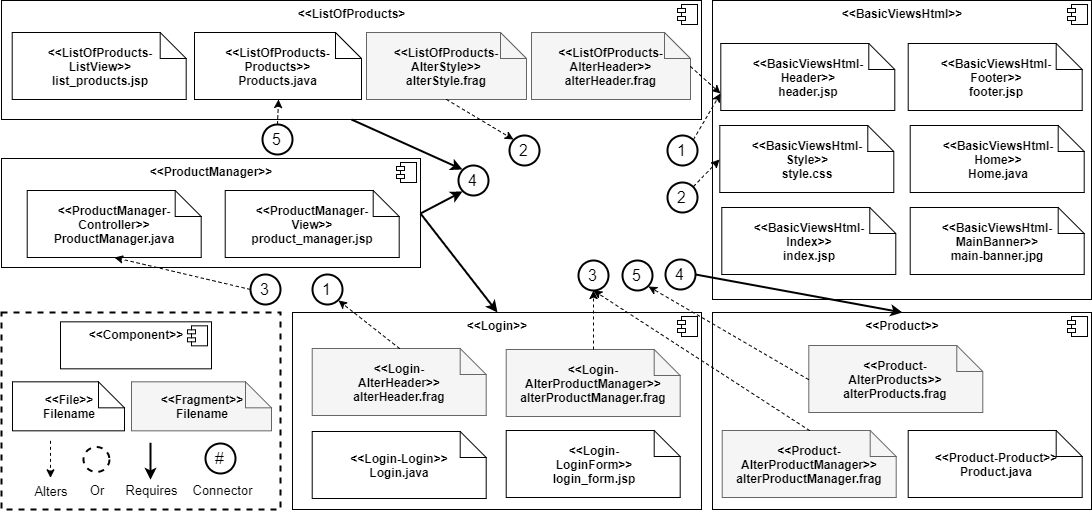
**Figura 1.** Diagrama de componentes de ClothingStores.

Inicie tomando como base los archivos definidos en el “pool de componentes”; más adelante, se presentan una serie de modificaciones que se deberían realizar a los componentes del “pool de componentes” para que se puedan integrar y funcionar para verificar que los componentes a nivel de dominio quedaron bien integrados.

**Si usted hace parte del G2 (DeltaJ) o G3 (VariaMos), modifique estos componentes de tal manera que el proceso de integración se pueda hacer lo más automático posible. Por ejemplo:**

**A) Si hace parte del G2**, cree su pool de componentes como deltas (para aquellos archivos tipo Java) y automatice las modificaciones (que se presentaran más adelante) de estos archivos hasta donde se lo permitan las herramientas y opciones brindadas por DeltaJ.

**B) Si hace parte del G3**, cree el modelo de componentes con VariaMos, le sugerimos crear el modelo tal como se presenta en la Figura 2. Modifique los componentes de tal manera que se puedan automatizar las modificaciones (que se presentaran más adelante); utilice fragmentos y puntos de fragmentación. Luego ejecute la integración de componentes tal como se vio en el documento de “Implementación de componentes mediante FragOP”.



**Figura 2.** Diagrama de componentes de ClothingStores bajo el paradigma FragOP.

**Modificaciones Solicitadas:**

1. Cuando se integran los componentes de ListOfProducts con los de BasicViewsHtml, se deben modificar 2 archivos:
   1. **Style.css:** se debe agregar el siguiente código al final del archivo.

|  |
| --- |
| .box\_image{  padding: 30px;  }  .caption{  text-align: center;  font-weight: bold;  } |

* 1. **header.jsp:** se debe agregar el siguiente código dentro de las etiquetas UL.

|  |
| --- |
| <li><a href="<c:url value='Products'/>">Products</a></li> |

1. Cuando se integran los componentes de Login con los de BasicViewsHtml, se debe modificar 1 archivo:
   1. **header.jsp:** se debe agregar el siguiente código dentro de las etiquetas UL.

|  |
| --- |
| <c:choose>     <c:when test="${sessionScope.admin != '1'}">       <li><a href="<c:url value='Login'/>">Login</a></li>     </c:when>     <c:otherwise>      <li><a href="<c:url value='Login?logout=1'/>">Logout</a></li>     </c:otherwise>  </c:choose> |

1. Cuando se integran los componentes de Login con los de ProductManager, se debe modificar 1 archivo:
   1. **ProductManager.java:** se debe agregar el siguiente código al inicio del método doGet.

|  |
| --- |
| HttpSession session = request.getSession();  String admin = (String) session.getAttribute("admin");    if(admin != "1"){   response.sendRedirect("Home");   return;  } |

1. Cuando se integran los componentes de Product con los de ListOfProducts, se debe modificar 1 archivo:
2. **Products.java:** se debe agregar el siguiente código al inicio del método doGet.

|  |
| --- |
| products = Product.get\_all(); |

Y se debe agregar el siguiente código en la zona de importaciones

|  |
| --- |
| import models.Product; |

1. Cuando se integran los componentes de Product con los de ProductManager, se debe modificar 1 archivo:
2. **ProductManager.java:** se debe agregar el siguiente código antes de la línea *request.setAttribute("title", "Product Manager");*

|  |
| --- |
| products = Product.get\_all(); |

Y se debe agregar el siguiente código en la zona de importaciones

|  |
| --- |
| import models.Product; |

**Proyecto en Eclipse:**

Por último, si lo desea cree un proyecto en Eclipse (de la misma manera como se ejecutaron los ejemplos para todos los grupos), e integre los componentes y ejecute el proyecto resultante. Realice las modificaciones anteriores de manera manual donde lo requiera.