**Proyecto:** Apache Struts Framework

**URL:** El sitio web oficial de Apache Struts es: http://struts.apache.org/

**Información básica del proyecto:**

Apache Struts es un framework open-source de java. Su función principal es facilitar el desarrollo de aplicaciones web basadas en Java proporcionando una arquitectura flexible y extensible. La estructura básica de Apache Struts sigue el patrón de diseño MVC (Modelo vista controlador) cuya estructura será detallada más adelante.

Por ahora cabe destacar que el mayor reto de diseño al que se enfrenta Apache Struts es la gestión eficaz del flujo de control y datos entre los distintos componentes del patrón MVC. Donde se tiene que garantizar una comunicación adecuada entre los componentes al mismo tiempo que se proporciona extensibilidad y capacidad de configuración para dar soporte a una amplia gama de requisitos de aplicaciones web.

**Detalles y Estructura de la sección que implementa el patrón:**

El patrón de desarrollo MVC se ve involucrado en los siguientes componentes del proyecto:

1.”Action”: Actúa como el controlador en el patrón MVC. Recibe las peticiones del usuario, realiza la lógica de negocio necesaria y coordina el flujo de control entre el modelo y la vista. Por lo que este es responsable de asignar las solicitudes entrantes a acciones específicas e invocar la lógica de negocio apropiada.

2.”Model”: Representa la capa de datos y lógica de negocio de la aplicación. También se encarga de encapsular el estado de la aplicación y proporciona métodos para manipular y acceder a los datos. El componente modelo puede interactuar con una base de datos, servicios externos o cualquier otra fuente de datos.

3.”View”: Se encarga de renderizar la interfaz y de presentar la información al usuario. Este recibe las entradas del usuario y muestra los resultados de las acciones realizadas por el modelo. El componente tiene la capacidad de generar archivos de tipo HTML, XML, JSON o cualquier otro formato en función de los requisitos.

El framework Apache Struts también incluye componentes adicionales como validadores, Form Beans, interceptores y archivos de configuración que ayudan a gestionar el comportamiento de la aplicación y a mejorar la implementación del patrón MVC.

**Información general sobre el patrón (MVC):**

El Modelo-Vista-Controlador (MVC) es un patrón de diseño de software que promueve la modularidad y la reutilización de código mientras facilita el mantenimiento, las pruebas y la escalabilidad de la aplicación. Todo esto mediante la separación de las responsabilidades en tres componentes interconectados.

1. Modelo: Representa los datos y la lógica de negocio de la aplicación. Gestiona el estado y el comportamiento de los datos de la aplicación y proporciona métodos para la manipulación de datos.

2. Vista: Presenta la interfaz de usuario al usuario. Muestra los datos del modelo y permite a los usuarios interactuar con la aplicación.

3. Controlador: Maneja las entradas del usuario, procesa las peticiones y orquesta el flujo de control entre el modelo y la vista. Actualiza el modelo basándose en las acciones del usuario y actualiza la vista para reflejar cualquier cambio en el modelo.

**Aplicación del patrón en Apache Struts:**

Apache Struts ejecuta una estructura dividida en los componentes modelo, vista y controlador (“MVC" por su siglas) para lograr la separación de responsabilidades e implementar un desarrollo modular de aplicaciones web. Su implementación es muy organizada a la hora de proporcionar una clara separación de responsabilidades, haciendo que el código base sea más fácil de mantener y extensible. Adicionalmente, permite probar de forma eficiente los componentes individuales y fomenta la reutilización del código. Permitiendo que los desarrolladores puedan centrarse en aspectos específicos de la aplicación sin preocuparse de otros componentes y otros problemas relacionados con el acoplamiento.

En el contexto de Apache Struts, además de los componentes clave del patrón MVC, existen otros componentes cercanos que contextualizan y complementan el patrón. Estos incluyen validadores, Form Beans, interceptores y archivos de configuración. Los validadores se utilizan para validar la entrada del usuario en las acciones o Form Beans, mientras que los interceptores realizan tareas de preprocesamiento y postprocesamiento en las acciones. Los Form Beans actúan como objetos contenedores para los datos de los formularios, y los archivos de configuración proporcionan la configuración necesaria para el framework.

**Desventajas:**

Cabe destacar que existen desventajas potenciales al utilizar el patrón MVC en Apache Struts. La introducción de una capa adicional de abstracción puede aumentar la complejidad de la aplicación y requerir una curva de aprendizaje inicial para los desarrolladores. Además, en proyectos pequeños o con requisitos muy simples, la estructura MVC puede resultar excesiva y generar un exceso de código.

Como alternativa al patrón MVC, se podrían considerar otros enfoques arquitectónicos, como el patrón de dos capas (model-view) o el patrón de tres capas (presentación-lógica de negocio-acceso a datos). Sin embargo, el patrón MVC es ampliamente adoptado y ofrece una estructura sólida para el desarrollo de aplicaciones web en Java.

**Diagrama UML de las principales clases:**

