GUÍA DE DESARROLLO DEL FRAMEWOK ZK, PARA EL DISEÑO ÁGIL DE APLICACIONES WEB, EN LA EMPRESA KRUGER CORPORATION. PROTOTIPO: SISTEMA PARA EL PLAN DE CAPACITACIÓN INTERNA DE EMPLEADOS

Muñoz Onofa Andrés Esteban, Ing. Ramiro Delgado, Ing. Diego Marcillo

1 Escuela Politécnica del Ejército, Quito Ecuador, andreco87@hotmail.com

2 Kruger Corporation S.A, Quito Ecuador, amunoz@kruger.com

RESUMEN

El presente trabajo muestra la investigación de la arquitectura, los componentes, la configuración e instalación del framework ZK. Con esta investigación se desarrolló tres guías de programación y diseño del framework y un prototipo de aplicación construido con el mismo. Se ha documentado ejemplos prácticos reales para la construcción de proyectos dentro de la plataforma JEE que utilicen en su capa web el framework ZK.

Se eligió el framework ZK, por la necesidad de investigar nuevas soluciones tecnológicas, que puedan proveer frameworks ágiles para el desarrollo de aplicaciones web, que sea intuitivo al codificarse, fácil de aprender para los programadores inexpertos, que tenga poca configuración y que responda rápido ante cualquier cambio o requerimiento de la aplicación. ZK ofrece un entorno de desarrollo rápido, y una línea de aprendizaje prácticamente plana .ZK soporta todos los principales patrones de desarrollo, tales como MVC, así como muchas más características propias del framework que satisfacen las necesidades anteriormente mencionadas y pueden ayudar a mejorar la cultura organizacional de desarrollo web en los sistemas que se vayan a ejecutar en un futuro en Kruger Corporation.

Adicional a la investigación del framework se realizó una comparación con el framework Java Server Faces en su versión 2. Se comparó las principales características que tienen los frameworks ZK y JSF. No se pretendió decidir cuál es mejor entre los dos, ya que ambos poseen excelentes características tanto en la arquitectura como en la implementación de aplicaciones dinámicas web. Como se puede ver en la descripción de ambas arquitecturas, los dos frameworks han sido construidos para simplificar la implementación de la tecnología Ajax. Se puede apreciar claramente, que el framework ZK maneja de una mejor manera las peticiones que se realizan al servidor, ya que tiene un componente que reside en el cliente y es un conjunto de código JavaScript y Jquery a diferencia del framework JSF que centraliza sus escuchas y componentes internos en el servidor, lo que muchas veces obliga al programador incluir código JavaScript o Jquery en las páginas, para lograr un mejor rendimiento en las peticiones y actualización de las páginas.

Palabras Clave: Arquitectura, framework, Jasa Server Faces, JavaScript, Ajax, Jquery, Componentes

ABSTRACT

This paper shows the investigation of the architecture, components, configuration and installation of the ZK framework. This research developed three guides programming and design of the framework and a prototype

application built with it. It has been documented real practical examples for the construction of projects within the JEE platform to use in the web tier framework ZK.

ZK framework was chosen because of the need to research new technology solutions that can provide flexible frameworks for developing web applications, that is intuitive to coded, easy to learn for novice programmers who have little configuration and respond quickly to any changes or requirements of the application. ZK provides a rapid development environment, and a nearly flat learning curve. ZK supports all major development patterns such as MVC, and many more features of the framework which meet the above needs and can help improve the culture Web development organizational systems that are executed in the future in Kruger Corporation

.

In addition to the investigation of the framework, a comparison was made with Java Server Faces framework in version 2. Main characteristics were compared with ZK and JSF frameworks. It was not intended to decide which is better between the two, since both have excellent characteristics in both the architecture and implementation of dynamic web applications. As can be seen in the description of both architectures, the two frameworks have been built to simplify the implementation of Ajax technology. It can be seen clearly that the ZK framework of a better way to handle the requests on this server because it has a component that resides on the client and is a set of JavaScript and jQuery unlike the JSF framework that centralizes its listeners and internal components on the server, which often forces the programmer to include code in JavaScript or Jquery pages to achieve better performance in the petitions and updating pages.

KeyWords: Architecture, framework, Java Server Faces, JavaScript, Ajax, Jquery, Components

1. INTRODUCCIÓN

La empresa Kruger Corporation desarrolla sus sistemas empresariales utilizando frameworks como JSF (Java Server Faces) y STRUTS para su capa web en la plataforma JEE (Java Enterprise Edition). Estos frameworks proveen características importantes que permiten un desarrollo web robusto, dinámico y estable. Sin embargo el desarrollo no es completamente ágil si se quisiera implementar una solución web en muy poco tiempo y con un alcance mediano, los componentes que proveen dichos frameworks en ocasiones no cumplen con los requerimientos funcionales que necesita la aplicación.

Es ahí donde se ve la necesidad de investigar nuevas soluciones tecnológicas, que puedan proveer frameworks ágiles para el desarrollo de aplicaciones web, que sea intuitivo al codificarse, fácil de aprender para los programadores inexpertos, que tenga poca configuración y que responda rápido ante cualquier cambio o requerimiento de la aplicación.

Una vez expuestos los puntos más relevantes, se plantea la investigación de un nuevo framework web llamado ZK, ya que ofrece un excelente rendimiento en el desarrollo de aplicaciones empresariales robustas, pero a su vez ágiles de desarrollar. Esto se demuestra al constatar oficialmente que empresas como SONY, SUN, IBM, ROCHE, TOYOTA, ADOBE entre muchas otras más, utilizan este framework para el desarrollo de sus sistemas y aplicaciones.

ZK ofrece un entorno de desarrollo rápido, y una línea de aprendizaje prácticamente plana. ZK soporta todos los principales patrones de desarrollo, tales como MVC, así como muchas más características propias del framework que satisfacen las necesidades anteriormente mencionadas y pueden ayudar a mejorar la cultura organizacional de desarrollo web en los sistemas que se vayan a ejecutar en un futuro en Kruger Corporation.

También se plantea el desarrollo de un prototipo de sistema desarrollado con dicho framework, donde se aplique todo en torno a la investigación realizada y se abstraiga todos los conceptos que se estén por definir en la investigación.

2. METODOLOGÍA

El proyecto de investigación se derivó del hecho de que la investigación incluye la producción u obtención de algo, en su evaluación, y su empleo como parte de una intervención. En este caso en base a la investigación realizada, se obtuvieron dos productos. El primero es una documentación referencial del framework ZK, y el segundo producto es el desarrollo de un sistema prototipo que incluyo dicho framework. El rasgo que tipifica al proyecto de investigación es la existencia de una intención cognoscitiva que prevalece sobre cualquier otro propósito en el proyecto. Conocer quiere decir arribar a proposiciones verdaderas o más completas sobre un objeto de estudio y/o generar, confirmar, refutar o verificar hipótesis en relación con dicho objeto.

2.1 Método de recolección de datos

El análisis de contenido es uno de los procedimientos que más se acerca a los postulados cualitativos desde sus propósitos; busca analizar datos técnicos en este caso, rasgos del objeto de investigación, arquitectura y otros elementos inherentes al objeto que se está investigando. Otras de las características que acercan el análisis de contenido con el paradigma cualitativo es que el examen de los datos se realiza mediante la codificación:

- Lo primero es determinar que contenido se estudiará y por qué es importante. Obviamente, esto se relaciona muy de cerca con la necesidad de tener un tema previo, una o varias preguntas de investigación y objetivos claramente definidos.
- En segundo lugar, debe tenerse claro los elementos que vamos a buscar. Esto nos remite necesariamente a una guía de análisis. Es importante recalcar que esta guía es una propuesta inicial, como en toda investigación cualitativa, y puede, conforme marcha el proyecto, modificarse de acuerdo con la experiencia en la obtención de datos.
- Dependiendo del propósito de nuestra investigación, se debe decidir la forma de recabar la información. Por ejemplo en hojas de reporte de observación o mediante transcripciones. A su vez, cada una de estas formas se analizará.
- Por último, habrá que asegurarse de unificar criterios para la observación y para la codificación, en el caso de que sean varios los observadores o codificadores.

2.2 Metodología para el desarrollo del prototipo

Desde el año 1996 la metodología XP ha sido comprobada en múltiples ocasiones para diversas compañías de varios tamaños alrededor del mundo, el motivo para elegir esta metodología de desarrollo es que en lugar de intentar entregar todo el producto de software en un futuro realiza las entregas de los productos a medida que el cliente los va necesitando ya que se enfoca en la satisfacción del cliente y permite una respuesta adecuada por parte de los desarrolladores a cambios en los requerimientos por parte del cliente.

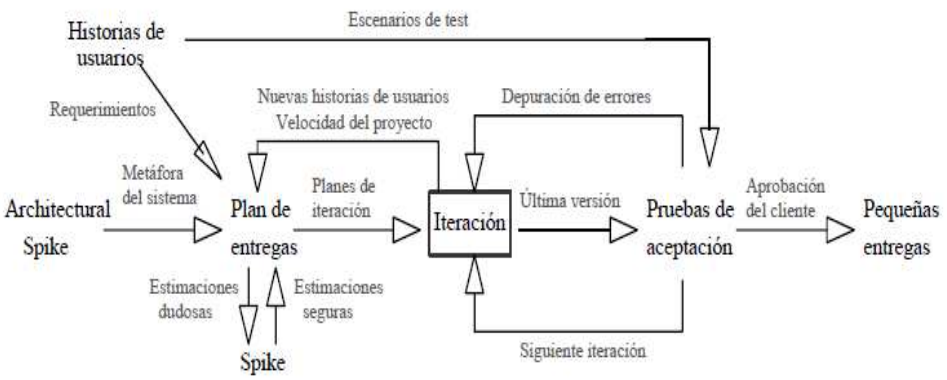

Figura 1. Ciclo de vida de la metodología XP

3. MATERIALES Y MÉTODOS

- Documentación que brinda la página oficial del framework. Dicha documentación consta de libros donde se detalla todo lo relevante al framework.
- Libros de tecnologías similares existentes.
- Programas y tutoriales que brinda la pagina para aprender a utilizar el framework.
- El framework en sí, el cual consta de librerías que serán instaladas en el IDE a utilizar.

4. DISEÑO E IMPLEMENTACIÓN

4.1 Arquitectura del framework

El funcionamiento basado en Ajax del framework ZK, se realiza por tres partes importantes, como se muestra en la Figura 2; el ZK loader, el motor de actualización asíncrona ZK UA, y el motor del cliente ZK. El ZK loader y el motor de actualización asíncrona ZK AU, están compuestos por un conjunto de servlets de Java, y el motor de cliente ZK se compone de códigos JavaScript. La Figura 2 ilustra el mecanismo cuando el ZK loader recibe una solicitud de URL por primera vez.

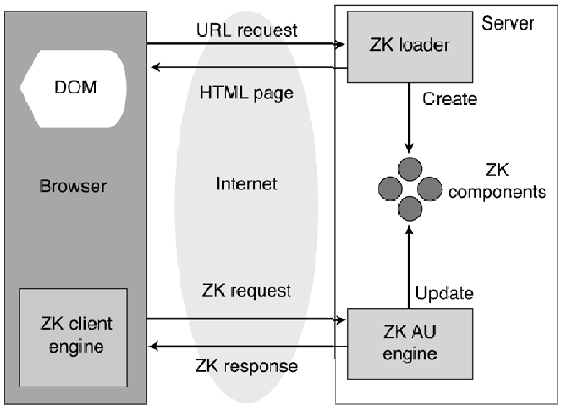

Figura 2. ZK loader, ZK UA y ZK cliente trabajando en una petición Ajax

El mecanismo funciona de la siguiente manera:

- El ZK loader interpreta una solicitud URL y genera la correspondiente página HTML, incluyendo los estándares HTML, los estilos CSS, el código JavaScript, y los componentes ZK en el lado del servidor.
- El ZK loader envía la página HTML al motor del cliente ZK. El motor del cliente ZK reside en el lado del cliente para monitorear los eventos JavaScript encolados en el browser.
- Si los eventos JavaScript se activan, el motor de cliente ZK envía esos eventos es decir las peticiones Ajax devuelta al motor de actualización asíncrona ZK AU sobre el servidor.
- El motor de actualización asíncrona ZK AU recibe las peticiones Ajax, actualiza los componentes ZK, y envía una respuesta Ajax de vuelta al lado del cliente.
- Por último, el motor del cliente ZK recibe la respuesta y actualiza el contenido correspondiente en el browser y el árbol de componentes.

El motor de cliente ZK se compone de una gran cantidad de código JavaScript que se encarga de recibir los eventos y actualizar el contenido de las páginas web, de esta manera es posible lograr una arquitectura muy similar a la que se tiene en las aplicaciones de escritorio construidas con la biblioteca gráfica de java Swing.

4.2 Configuración del framework

El framework ZK se ejecuta como un conjunto de servlets, dentro de un contenedor de servlets Java. El objetivo principal de la configuración del framework, es preparar el ambiente en donde las aplicaciones ZK se desarrollaran y ejecutarán. Para esto, se requiere la instalación de ciertos programas fundamentales para el funcionamiento del framework, esto incluye la instalación de Java Runtime Environment, la instalación de un contenedor de servlets Java y un servidor para desplegar los archivos War.

4.3 Componentes del framework.

Un componente es un objeto de la interfaz de usuario (UI), tal como un botón, una etiqueta o un árbol. El componente define la representación gráfica y comportamiento lógico de un elemento en la interfaz de usuario.

Modificando los atributos del componente o manipulando su estructura, se puede controlar la presentación visual de una aplicación en el cliente. Todos los componentes de ZK implementan la interfaz *org.zkoss.zk.ui.*

Una página es una colección de componentes. Las páginas son miembros de la clase *org.zkoss.zk.ui.* La clase Page, y sus componentes contenidos en la página, serán mostrados en ciertas posiciones en el navegador. Una página es creada automáticamente cuando el ZK loader interpreta una página ZULM.

Un escritorio que hereda de la clase *org.zkoss.zk.ui. Desktop,* es una colección de páginas para servir a la misma solicitud URL.

Se define entonces a un componente como la sinergia de dos mundos ya que además de ser un objeto de Java en el servidor, un componente tiene una parte visual en el navegador embebido en las páginas. Es decir, cuando un componente se une a una página, su parte visual se crea en el navegador. Y cuando un componente se elimina de una página, la parte visual se retira.

4.3.1 El Ciclo de Vida de un Componente

El framework ZK supone algunas secuencias de procesamiento, con respecto a las páginas de carga y actualización, que puedan afectar a la forma de escribir las aplicaciones ZK. En esta sección, vamos a ver el ciclo de vida de los componentes ZK tener una mejor comprensión del mecanismo del framework ZK.

4.3.2 El Ciclo de Vida de una Página ZULM

El ZK loader carga una página ZULM en cuatro fases:

- Inicialización de la Página.
- Creación de los componentes.
- Procesamiento de eventos.
- Representación de los componentes.

4.3.3 Expresiones de Lenguaje (*EL*)

Al igual que en las páginas JSP, se pueden usar expresiones EL en cualquier parte de las páginas ZUML, excepto en los nombres de los atributos, o elementos e instrucciones de procesamiento. ZUML utiliza la misma sintaxis para las EL, que las páginas JSP, es decir, ${expr}. Como se muestra en El Código1:

```
<windowtitle="EL Test" width="200px">
<zscript>  String abc = "ABC";
</zscript>
<buttonlabel="${abc}"/>
</window>
```

Código 1: Ejemplo de EL.

Fuente: Henri Cheng & Robie Cheng ZK Ajax without JavaScript Framework, 2008

Cuando una expresión EL se utiliza como el valor de un atributo, puede devolver cualquier tipo de objeto, siempre que el componente acepte ese objeto como valor de entrada. Por ejemplo, la siguiente expresión será evaluada como un objeto Boolean: <windowif="${some> 10} ">

4.3.4 Eventos

Un evento es de la clase org.zkoss.zk.ui.event.Event y se utiliza para notificar a una aplicación de una acción en el navegador. Cada tipo de evento está representado por una clase distinta. Por ejemplo, org.zkoss.zk.ui.event.MouseEvent denota una actividad del ratón, como hacer clic.

Para responder a un evento, una aplicación debe registrar uno o más detectores al evento. Hay tres formas de registrar un detector de eventos para un componente. Una de ellas es especificando el detector de eventos OnXxx como atributo del componente, donde XXX representa el nombre de la acción, como muestra El Código 2.

```
<window title="Hello" border="normal">
<button label="Say Hello" onClick="alert(&quot;Hello World!&quot;)"/>
</window>
```

Código 2 Ejemplo de creación de un componente

En El Código 3, onClick es el evento que se activa con el objeto botón. El manejo de eventos código para el evento onClick es *alert* ("Hola Mundo!").

La segunda manera de registrar un detector de evento es mediante la definición de la OnXxx () en la clase del componente asociado.

```
<window title="Hello" border="normal">
<zscript> classMyButton extends Button {
public void onClick(MouseEvent event) {
Messagebox.show("Hello World!");
}
}
</zscript>
<button label="Say Hello" use="MyButton"/>
</window>
```

Código 3 Controlador de eventos con un método definido

5. RESULTADOS

5.1 Guía de desarrollo y programación, utilizando el framework ZK.

Una vez expuestos los conceptos más importantes del framework ZK en el Capítulo 2, es necesario ahora tener una guía referencial para desarrollar aplicaciones reales con dicho framework.

Es necesario tener en cuenta que el framework ZK, cubre las necesidades de desarrollo de la capa web en un sistema que adopte la especificación JEE, por lo tanto no se pretende explicar el desarrollo de los otros componentes de esta especificación

En esta guía se implementó desarrollando ejemplos de aplicaciones desarrolladas con el framework ZK, cada implementación tendrá funcionalidades específicas que ayudarán el aprendizaje del framework ZK.

Esta guía contiene también código, útil para el desarrollo de aplicaciones con el framework ZK, el cual puede ser copiado para el uso en diferentes aplicaciones que requieran de necesidades básicas en una aplicación como lo es las operaciones CRUD (crear, leer, actualizar, eliminar) sobre los datos de una base de datos.

5.2 Guía de desarrollo de componentes gráficos.

Esta sección tiene como objetivo el brindar una vista general del desarrollo de componentes en framework ZK. El objetivo de esta guía, es el brindar un conocimiento especifico acerca de cómo se deben programar los componentes del framework ZK, además de proporcionar consejos y directrices de desarrollo de los componentes del framework, para sacar el máximo provecho a los mismos.

Se pretende que esta guía sirva como modelo de programación de componentes, para cualquier aplicación de mediano y corto alcance, realizada con el framework ZK.

5.3 Guía de estilos en las páginas.

La forma más común de personalizar los componentes, es mediante el uso de hojas de estilo en cascada. Esta técnica es conocida desde HTML, por lo que el framework ZK se adapta perfectamente a este recurso.

5.4 Comparación entre el framework ZK y el framework JSF 2.0.

La siguiente es una comparación de las principales características que tienen en común los frameworks ZK y JSF. No se pretende decidir cual es mejor entre los dos, ya que ambos poseen excelentes características tanto en la arquitectura como en la implementación de aplicaciones dinámicas web.

En la actualidad la especificación JEE6, aconseja utilizar como estándar para el desarrollo de la capa web, el framework JSF en su versión actual 2.0. Este framework, está construido para acoplarse fácilmente a cualquier aplicación JEE, es decir no limita al programador, un modelo de arquitectura de aplicaciones estándar. El gran problema a destacar del framework JSF 2.0, son sus excesivas implementaciones, que tratan de parchar y brindar más funcionalidad al mismo. Estas implementaciones, son frameworks aliados que funcionan con la arquitectura de JSF. Entre los más destacados, están las librerías de Richfaces y Primefaces, que se agregan a la aplicación web.

Esta característica puede ser utilizada y aprovechada al cien por ciento, cuando se tiene más experiencia en el desarrollo de aplicaciones con el framework JSF, ya que en sí, estas implementaciones tratan de enriquecer la funcionalidad y la presentación de los componentes del mismo, pero traen consigo muchos problemas de incompatibilidad y bugs inherentes en a la funcionalidad y arquitectura de sus componentes.

Este problema, hace que la programación no sea tan ágil y que el programador se vea confundido al momento de elegir cual componente de estas implementaciones es el mas adecuado, o simplemente utilizar un componente nativo del framework JSF. Es por ello que se requiere de más experiencia y conocimiento de la arquitectura web y tecnologías como JavaScript y Jquery que nos ayuden a corregir estos errores al momento de utilizar el framework JSF.

Por el contrario, el framework ZK, no tiene implementaciones aliadas, y hace que el funcionamiento de sus componentes y la implementación de tecnologías como Ajax Y JavaScript sea totalmente transparente para el desarrollador web. Esta característica del framework es lo que lo hace más ágil en su programación e implementación en aplicaciones dinámicas enriquecidas.

Tabla 1. Tabla de resumen de la comparación del framework ZK y JSF

	JSF	ZK
Lenguaje	Java	Java, Groovy, ZULM
Ajax	Sí	Jquery
MVC	Sí	Sí
MVC push-pull	pull	Push-pull
Internacionalización	Sí	Sí
ORM	Sí, agregando extensiones	Cualquier ORM
Extensiones para pruebas	JUnit	JUnit, ZTL
Extensiones para migración de Bases de Datos		HibernetUtil, SpringUtil
Extensiones de seguridad	Sí	Spring Security
Templates	Facelets, JSP	Macro componentes, composition.
Caching	Sí	Sí
Validaciones	Validaciones nativas	Cliente, Servidor

6. CONCLUSIONES Y TRABAJO FUTURO

- El framework ZK es ideal para realizar prototipos de aplicaciones web escritas en Java, las mismas que pueden ser adaptadas e integradas con cualquier framework adicional como Spring o utilizar tecnologías de soporte en diferentes capas de la plataforma JEE como Hibernate o JPA.

- El framework ZK, reduce el tiempo de desarrollo de las aplicaciones web ya que abstrae la implementación de tecnologías importantes para la construcción de aplicaciones webs dinámicas como Ajax y JavaScript.

- Es más fácil la programación de componentes en el framework ZK en relación a otros frameworks webs, por la facilidad de crear los objetos que representan al componente en las páginas, directamente desde las clases Java.

- La inclusión de código Zscript en las páginas web, agiliza la navegación de las mismas, y nos permite modelar de manera más óptima una aplicación.

- A diferencia de otros frameworks dónde aun es necesario programan detalles de las llamadas Ajax al servidor, ZK protege de estas complejidades y permite al desarrollador centrarse en la lógica de negocio.

- Los componentes del framework ZK se comportan de igual forma en cualquier navegador web, lo que no sucede con componentes del framework JSF que sufre de incompatibilidades al momento de desplegarse en algunos navegadores.

- ZK proporciona las herramientas para suministrar una interfaz de usuario enriquecida, con una complejidad mínima. La capacidad de utilizar ZUL para crear prototipos de una pantalla con retroalimentación de usuarios, es muy útil cuando se implementa aplicaciones web dinámicas. El prototipo puede se puede convertir en la interfaz actual, reduciendo el tiempo de desarrollo inicial.

- Con la lectura de una guía básica del framework ZK como la expuesta en el presente trabajo, es totalmente posible que un programador inexperto desarrolle una aplicación web con el framework ZK, ya que no necesita tener conocimientos previos de tecnologías embebidas como JavaScript y Ajax, más si tener una noción básica de programación Java y HTML.

- En relación al framework JSF, la comunidad que brinda soporte al framework ZK no está segmentada en grupos que hacen implementaciones propias tales como Richfaces o Primefaces en el caso de JSF. Esto potencia al framework al existir una sola implementación del mismo. El framework se enriquece cada día por una comunidad que expone y resuelve desde el problema más sencillo expuesto por el usuario inexperto hasta el caso de uso más complejo.

- En el prototipo construido en el presente trabajo, se explotó al máximo los componentes del framework ZK, combinando los mismos con tecnologías como HTML5 incluidas en las páginas y recibiendo datos de frameworks como Spring.

- En el prototipo se implementó componentes que el framework define como *aplicaciones*, es el caso del calendario y el lector de noticias que son una implementación de las librerías que el framework proporciona.

7. AGRADECIMIENTOS

Agradezco a todas las personas que han guiado este trabajo de manera directa e indirectamente, y a mis profesores por darme excelentes ideas y dejarme exponer esta innovadora tecnología.

8. REFERENCIAS BIBLIOGRÁFICAS

[1] Cheng & Robie Cheng ZK Ajax without JavaScript Framework, 2008.

[2] ZK 5.08 Reference Guide – Equipo ZK 2012.

[3] ZK 5.08 Component Reference – Equipo ZK 2012.

[4] ZK 5.08 Configuration Guide – Equipo ZK 2012..

[5] ZK 5.08 Style Guide – Equipo ZK 2012.

[6] Página oficial del framework http://www.zkoss.org