

Como fuentes de información que se detectaron como base para la generación de este reporte, son:

- Página de la comunidad de zkoss (<http://blog.zkoss.org/>)
- Documentación de la referencia a desarrolladores de zkoss, (<http://www.zkoss.org/documentation#References>)
- Documento de Autentia en donde se explica este tema (<http://es.scribd.com/doc/94736866/CharlaZK>)
- Documento de explicación de cómo implementar un modelo MVZ utilizando el framework de Zkoss
<http://es.scribd.com/doc/82145991/16/Implementing-ZK-MVC>

Zk es un framework de java y javascript, el cual es open source y permite a desarrolladores crear aplicaciones web y móviles ricas y dinámicas en cuanto a sus interfaces, sin necesidad de ningún conocimiento por parte de los mismos de Ajax o JavaScript.

Su fusión de arquitectura cliente-servidor permite escribir aplicaciones a los desarrolladores en Java Puro y diseñar la interfaz de usuarios en xml.

Programación Declarativa

ZK User Interface Markup Language (ZUML) hace el diseño de interfaces ricas de usuario tan simples como escribir páginas html. ZUML es una variante de XUL, el cual hereda todas las características disponibles en XML y separa la definición de interfaz de usuario (UI) de la lógica de ejecución.

ZUML también permite a los desarrolladores la automatización de CRUD's entre componentes de interfaz de usuario y fuentes de datos mediante anotaciones, data bindings y MVVM(Model View ViewModel).

En la siguiente liga, se tiene una referencia completa de cómo funcionan los archivos zul:

http://books.zkoss.org/wiki/ZUML_Reference

En la siguiente liga, se tiene un listado completo de componentes que son soportados en el framework (archivos zul y java):

http://books.zkoss.org/wiki/ZK_Component_Reference

Modelo extensible

La arquitectura de ZK está basada en unidades modulares que son elementos que pueden ser agregados y reemplazados, permitiendo a los desarrolladores extender y personalizar las aplicaciones desarrolladas mediante este framework.

Es permitido a los desarrolladores, agregar código específico a sus aplicaciones desarrolladas mediante zkoss, tal como, comunicación específica mediante Ajax, manejo de eventos, instalaciones de componentes, parsers de ZUML, entre muchas más características.

Seguridad

Por defecto, las aplicaciones en ZK son ejecutadas del lado del servidor, ningún tipo de lógica de negocio es expuesta a la parte del cliente.

La seguridad a nivel de página puede ser implementada con frameworks de terceros, esto es mediante Spring o más específicamente con SWB.

Soporte a Plataforma Nativa

ZK es compatible con los estándares de la W3C, lo que significa que todos los dispositivos que ejecuten navegadores compatibles con la W3C, tienen la capacidad de ejecutar aplicaciones desarrolladas con ZK.

ZK puede ser ejecutado en una variedad grande de dispositivos móviles, en los cuales se aprovechan el despliegue de los componentes nativos de los mismos.

Desarrollo mediante el framework

Es importante el que bajemos el framework que es gratuito, es decir, la versión CE (Community Edition)

Zk tiene un plug-in para eclipse y uno para netbeans, por lo tanto, podremos instalar para cualquiera de estas IDE's y desarrollar nuestras aplicaciones de manera rápida.

La liga en donde podemos bajar el plug-in de zk para eclipse, es la siguiente:

http://books.zkoss.org/wiki/ZK_Installation_Guide/Quick_Start/Create_and_Run_Your_First_ZK_Application_with_NetBeans_and_REM

En la siguiente liga, se explica cómo implementar el modelo MVZ mediante el framework ZK.

<http://es.scribd.com/doc/82145991/16/Implementing-ZK-MVC>

En dicha página se explican los siguientes puntos:

- Introducción a ZK
- Introducción a la arquitectura funcional de ZK
- Interfaz de usuario basada en componentes
- Mapeo de componentes en programación java
- Programación manejada por eventos
- Manejo de eventos en un controlador
- Manejo de eventos de componentes generados dinámicamente
- Un ejemplo implementando MVZ con Zkoss desde su configuración en eclipse.