

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



» Estás en: [Inicio](#) [Tutoriales](#) [MVC y MVVM \(3-5\)](#)



Francisco Ferri Pérez

Consultor tecnológico de desarrollo de proyectos informáticos.

Desarrollador de proyectos informáticos, Microsoft Certified IT Professional - Enterprise Administrator

[Ver todos los tutoriales del autor](#)



Catálogo de servicios Autentia



Fecha de publicación del tutorial: 2012-10-07

Tutorial visitado 1 veces [Descargar en PDF](#)

Framework ZK



MVC y MVVM (3-5)

¿Qué son y en qué se diferencian?

Contenido

1. Introducción
2. ¿Qué son MVC o MVVM?
3. Comparación entre MVC y MVVM
4. ¿Cuál debo usar?
5. Referencias

Introducción

Este tutorial es la segunda parte de una serie de cinco:

1. [Configurar el entorno](#)
2. [Empezar a programar](#)
3. [MVC y MVVM ¿Qué son y en qué se diferencian?](#) (Estás en este)
4. [MVC en ZK \(no publicado todavía\)](#)
5. [MVVM en ZK \(no publicado todavía\)](#)

Está dirigido a desarrolladores con experiencia en la creación de aplicaciones web en Java. A continuación aprenderás qué es MVC y qué es MVVM, también te enseñaremos las diferencias entre ellos. Comprobarás por tí mismo el enfoque que hace ZK sobre ellos..

A lo largo de este documento hacemos referencias a fuentes externas, las encontrarás todas en la sección de Referencias

¿Qué son MVC y MVVM?

Son patrones de diseño o modelos de abstracción utilizados para definir y estructurar los componentes necesarios en el desarrollo de software.

MVC significa **Modelo Vista Controlador**, porque en este patrón de diseño se separan los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. Cuando la lógica de negocio realiza un cambio, es necesario que ella sea la que actualiza la vista.

Una manera muy simple de describirlos sería:

MVVM significa **Modelo Vista VistaModelo**, porque en este patrón de diseño se separan los datos de la aplicación, la interfaz de usuario pero en vez de controlar manualmente los cambios en la vista o en los datos, estos se actualizan directamente cuando sucede un cambio en ellos, por ejemplo si la vista actualiza un dato que está presentando se actualiza el modelo automáticamente y viceversa.

Comparación entre MVC y MVVM

La primera imagen describe la interacción que se produce con MVC en ZK, y la siguiente es la correspondiente al enfoque MVVM que hace ZK.

Síguenos a través de:



Últimas Noticias

» ¡¡¡Terrakas 1x04 recién salido del horno!!!

» Estreno Terrakas 1x04: "Terraka por un día"

» Nuevos cursos de gestión de la configuración en IOS y Android

» La regla del Boy Scout y la Oxidación del Software

» Autentia conquista los Alpes

[Histórico de noticias](#)

Últimos Tutoriales

» Empezar a programar con ZK (2-5)

» Eventos en MySQL

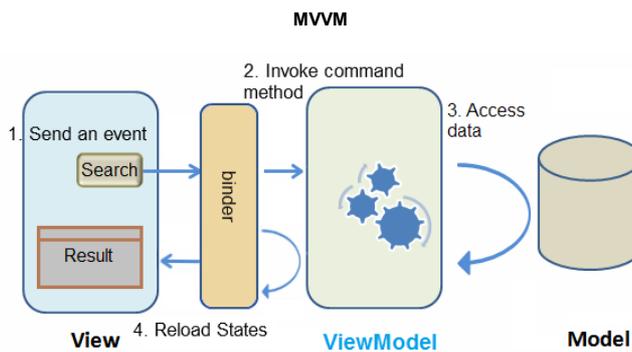
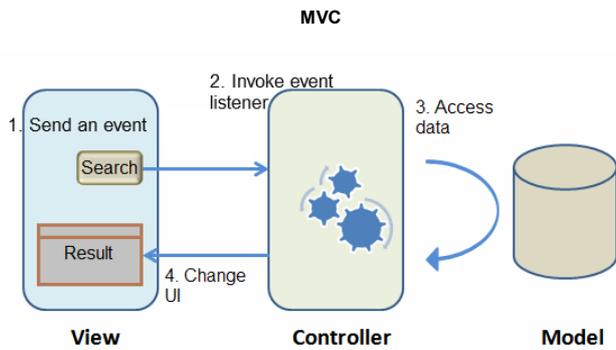
» Plantillas para los métodos equals y hashCode en Eclipse, usando la librería de Apache Commons Lang.

» Como convertir ficheros Flash (.swf) a HTML5

» Tu primer proyecto web con ZK (1-5)

Últimos Tutoriales del Autor

Las principales diferencias entre MVC y MVVM son que en MVVM el "Controller" cambia a "ViewModel" y hay un "binder" que sincroniza la información en vez de hacerlo un controlador "Controller" como sucede en MVC.



Los dos enfoques tienen muchas cosas en común, pero también hay claras diferencias entre ellos. Cada uno de los 2 enfoques tiene su razón de ser/existir.

¿Cuál debo usar?

Construir una aplicación mediante **MVC** puede resultar más intuitivo, porque directamente controlas lo que ves en la vista y su comportamiento, es decir manualmente. MVC se caracteriza porque tienes control total de los componentes, por lo tanto puedes crear componentes hijo dinámicamente ("child"), controlar componentes propios personalizados, o realizar cualquier cosa sobre el componente que este pueda hacer por sí mismo.

En el patrón **MVVM**, puesto que la capa "ViewModel" está débilmente acoplada con la vista (no está referenciada a los componentes de la vista), podemos usarla con múltiples vistas sin tener que modificarla. Por lo tanto los diseñadores y programadores pueden trabajar en paralelo. Si la información y comportamiento no cambian, un cambio en la vista no provoca que se tenga que modificar la capa "ViewModel". Además, como la capa "ViewModel" es un POJO, es fácil realizar tests unitarios sobre ella sin ninguna configuración ni entorno especial. Lo que significa que la capa "ViewModel" tiene una mejor reusabilidad, testabilidad, y los cambios en la vista le afectan menos.

Para resumir, comparamos los 2 patrones de diseño en una tabla:

	MVC	MVVM
Acoplamiento con la vista	Muy poco con plantilla	Muy poco
Acoplamiento con el componente	Un poco	Muy poco
Codificar en la vista	Mediante el ID del componente	A través de una expresión Data binding
Implementación de un controlador	Extendemos ZK's Composer	Es un POJO
Acceso a la información de la UI	Acceso directo	Automático
Acceso a la información desde el backend	Acceso directo	Acceso directo
Actualización de la interfaz de usuario	Manipulamos directamente los componentes	Automático (@NotifyChange)
Nivel de control del componente	Elevado, control total	Normal
Rendimiento	Alto	Normal

Referencias

- [Página web oficial de ZK](#)
- [MVC en Wikipedia](#)
- [MVVM en Wikipedia](#)

Este documento es un extracto de la documentación oficial del Framework ZK, traducido y ampliado por Francisco Ferri. Colaborador de Potix (creadores del Framework ZK). Si quieres contactar con él puedes hacerlo en franferri@gmail.com, en twitter [@franciscoferri](#) o en LinkedIn

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

Share | 0

» Empezar a programar con ZK (2-5)

» Tu primer proyecto web con ZK (1-5)

» NIC Bonding, NIC Teaming, Port Trunking, Etherchannel o Ether bonding, con ifenslave en Ubuntu

» Tutorial básico de bases de datos en Java mediante JDBC

» Introducción a bases de datos SQL en Java.

Últimas ofertas de empleo

2011-09-08
[Comercial - Ventas - MADRID.](#)

2011-09-03
[Comercial - Ventas - VALENCIA.](#)

2011-08-19

[Comercial - Compras - ALICANTE.](#)

2011-07-12
[Otras Sin catalogar - MADRID.](#)

2011-07-06
[Otras Sin catalogar - LUGO.](#)

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» [Regístrate](#) y accede a esta y otras ventajas «



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA

Impulsores

Comunidad

¿Ayuda?

sin clicks

0 personas han traído clicks a esta página

+ + + + + + + +

powered by [karmacracy](#)

Copyright 2003-2012 © All Rights Reserved | [Text & Code](#) | [Condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#)

[W3C XHTML 1.0](#)

[W3C CSS](#)

[XML RSS](#)

[XML RTOM](#)