



Flex Frameworks

Alberto Alcaraz

Índice

- Introducción
 - Definiciones
 - Características comunes
- Comparativa
 - Cairngorm
 - PureMVC
 - Mate
 - Swiz
- Ejemplos
- Conclusiones
- Preguntas

Introducción

Definición I

Es una estructura de soporte definida, mediante la cual un proyecto de software puede ser organizado y desarrollado

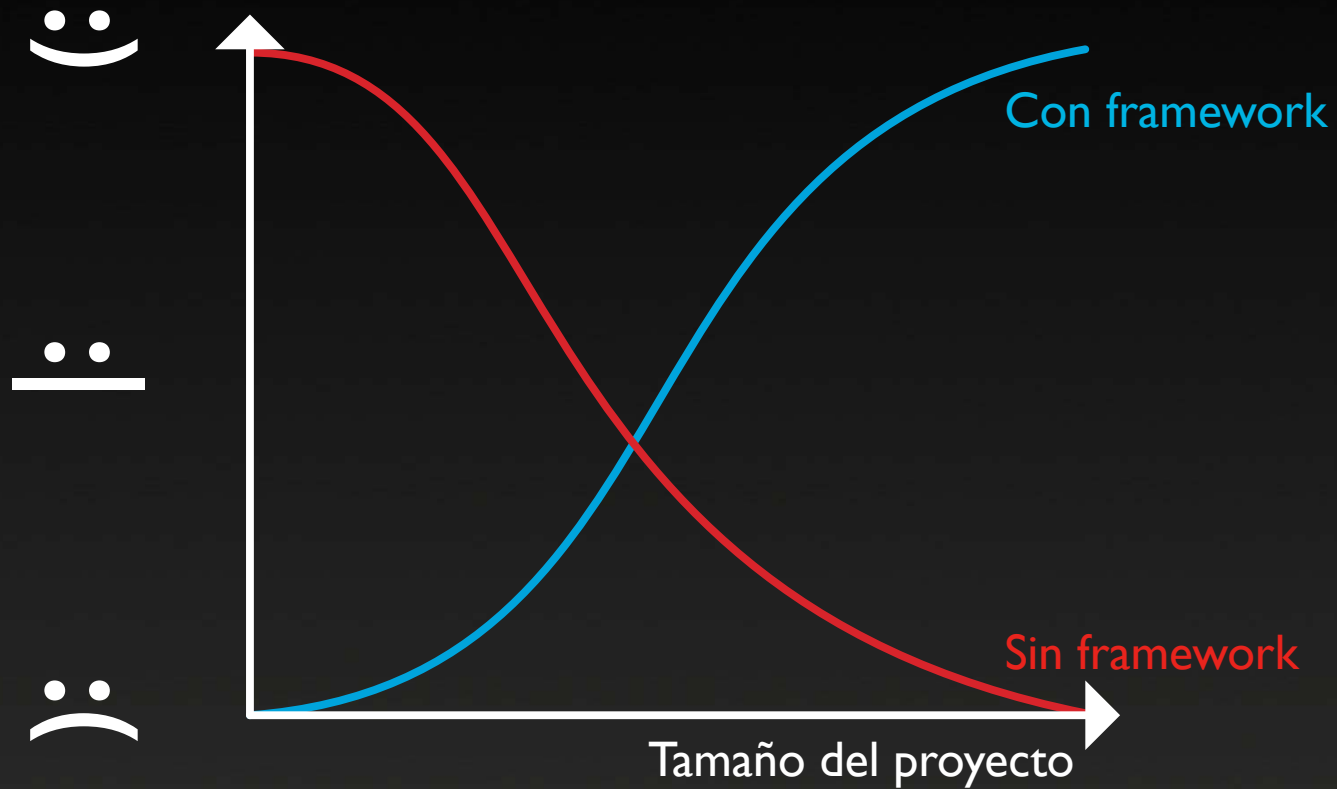
Definición II

Es un conjunto patrones de diseño y buenas prácticas que intentan resolver problemas comunes en el desarrollo de software

Características

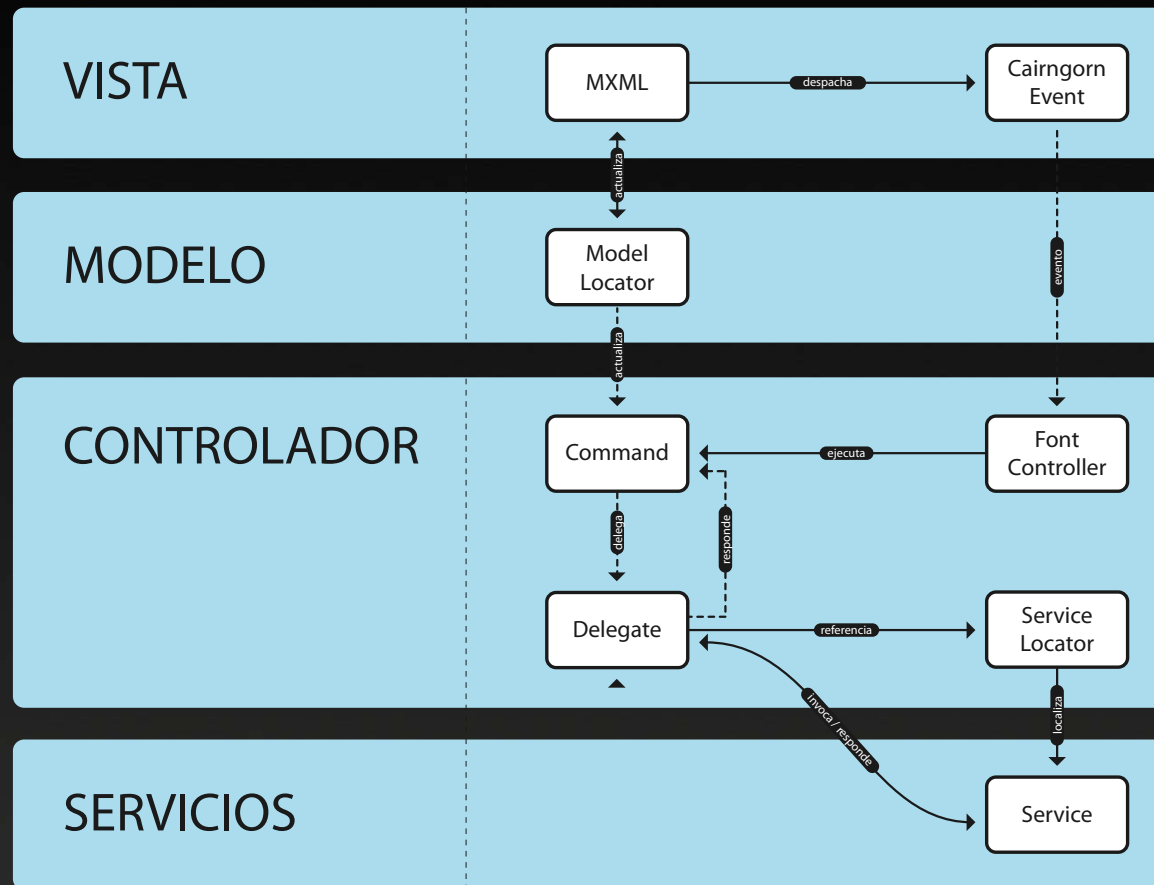
- Ayuda a la organización
- Potencia el desarrollo rápido
- Permite la escalabilidad

Características



Comparativa

Cairngorm



Cairngorm

Características

- Primer framework para Flex
- Apoyado oficialmente por Adobe (OS)
- MVC, Singleton, Command, Delegate

Cairngorm

Lo bueno

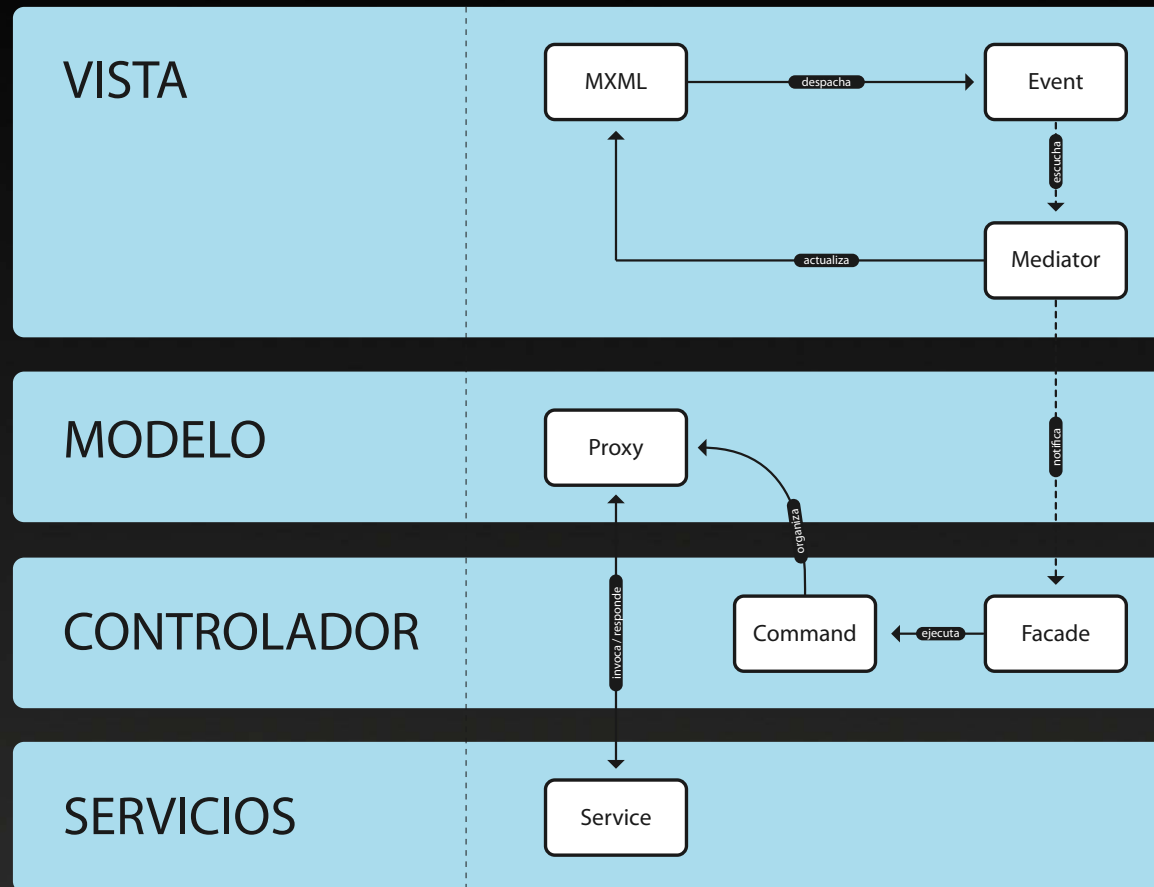
- Usa funciones específicas de flex como los bindings
- Muy extendido
- Buena curva de aprendizaje

Cairngorm

Lo malo

- Mucho código extra
- Cierta intrusión (i.e. CairngormEvent)
- El desarrollo con módulos es complejo

PureMVC



PureMVC

Características

- Framework multi-lenguaje
- Versión standard/multicore
- Façade, Mediator, Proxy, Command, Observer

PureMVC

Lo bueno

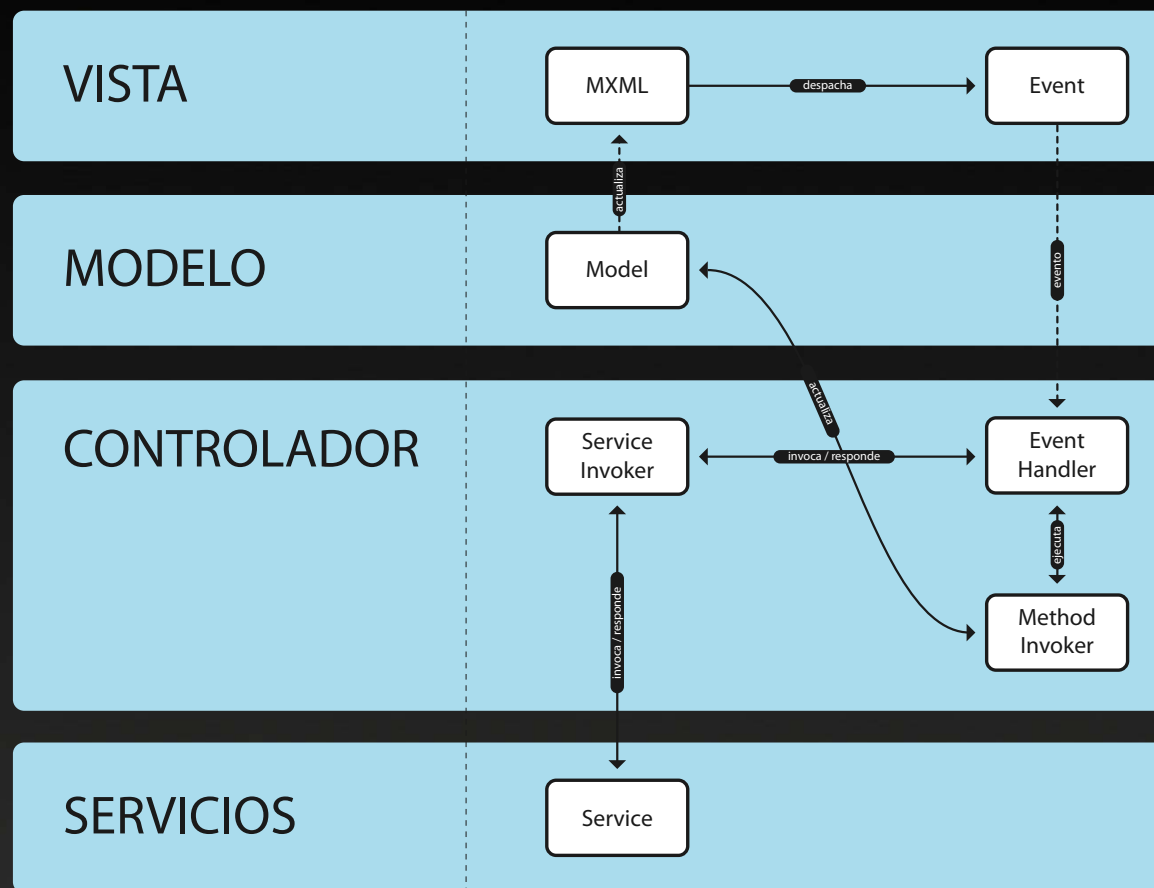
- Buena documentación
- Capas muy desacopladas
- Mucha flexibilidad

PureMVC

Lo malo

- No aprovecha todas las capacidades de Flex
- Mayor grado de complejidad

Mate



Mate

Características

- Hace uso intensivo de MXML
- Posee una librería de plugins
- EventMap, IoC

Mate

Lo bueno

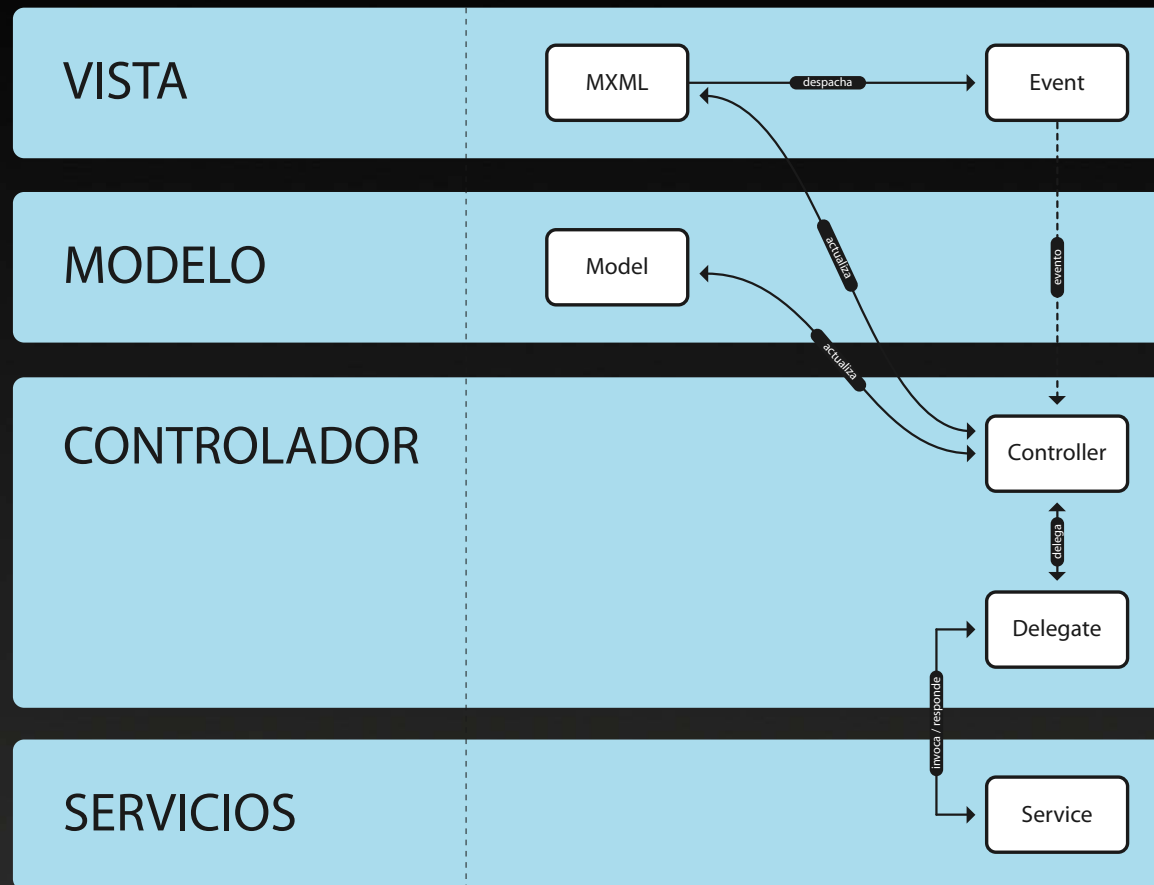
- Poca intromisión del framework
- Muy adaptable según las necesidades
- Sencillo de aprender

Mate

Lo malo

- Escasa documentación
- No define un arquitectura

Swiz



Swiz

Características

- Utiliza metadata para definir dependencias
- Utiliza un EventDispatcher central
- Factory, IoC, DynamicMediators

Swiz

Lo bueno

- “Brutalmente simple”
- Velocidad de desarrollo
- Promueve el desacoplamiento

Swiz

Lo malo

- Posible penalización de rendimiento al utilizar introspección
- No muy extendido

Ejemplos

Conclusiones

- Evaluar la necesidad de un framework
- Probar y aprender todos los posibles
- Mayor equipo → Mayores restricciones

¿Preguntas?

Información de contacto:

E-Mail

alberto.alcaraz@undefined.es

Blog

www.theklue.com

Web

www.undefined.es

www.madeinflex.com

¡Gracias!