Estudio comparativo de Marcos de Trabajo para el Desarrollo Software Orientado a Aspectos

Carlos A. Guerrero⁽¹⁾, Jorge M. Londoño⁽²⁾, Johanna M. Suárez^{(1)*} y Luz E. Gutiérrez⁽¹⁾

(1) Grupo de Investigación en Ingeniería del Software-GRIS, Unidades Tecnológicas de Santander, Calle de los Estudiantes # 9-82 Ciudadela Real de Minas - Bucaramanga - Santander - Colombia

(2) Grupo de Investigación en Desarrollo y Aplicación en Telecomunicaciones e Informática - GIDATI, Universidad Pontificia Bolivariana – UPB, Campus de Laureles Circular 1 No. 70-01, Medellín, Antioquía-Colombia (e-mail: cinving@uts.edu.co; jorge.londono@upb.edu.co; jomasupe@hotmail.com; luzelenagl@live.com)

*Autor a quien debe ser dirigida la correspondencia

Recibido Sep. 6, 2013; Aceptado Nov. 6, 2013; Versión final recibida Dic. 4, 2013

Resumen

Este artículo presenta los resultados de un estudio comparativo que busca identificar los mejores marcos de trabajo para utilizar el paradigma de programación orientado a aspectos POA. Se identificaron marcos de trabajo en 3 lenguajes de programación y se establecieron criterios de evaluación para la comparación. Para cada uno de estos criterios se identificaron las métricas que hicieron posible cuantificarle cumplimiento o no del criterio de comparación. El estudio se realizó con el apoyo de múltiples equipos de desarrollo divididos en dos clases: 1) investigadores con experiencia previa en programación orientada a aspectos y 2) desarrolladores de software sin experiencia previa en programación orientada a aspectos. Los dos equipos implementaron el mismo conjunto de componentes en cada uno de los marcos seleccionados. El estudio evidenció que las ventajas de POA no se hacen tan evidentes en desarrollos que utilicen marcos de trabajo que soporten este paradigma.

Palabras clave: programación orientada a aspectos, marcos de trabajo, componente ortogonal, desarrollo de software

Comparative Study of Frameworks for the Development of Aspect-Oriented Software

Abstract

The results of a comparative study seeking to identify the best frameworks to use the paradigm of aspect-oriented programming (AOP) are presented. Frameworks in three programming languages were identified and the evaluation criteria for comparison were established. For each criterion the metrics that allow quantifying its level of compliance were identified. The study was conducted by multiple development teams divided in two groups: 1) Researchers with previous experience in aspect oriented programming, and 2) development teams with no previous experience in aspect oriented programming. Both teams implemented the same components in each of the selected frameworks. The study showed that the benefits of AOP are not so evident in developments using frameworks that support this paradigm.

Keywords: aspect oriented programming, frameworks, component orthogonal, aspect, software development

INTRODUCCIÓN

El artículo presenta los resultados de un estudio comparativo que busca identificar los mejores marcos de trabajo para utilizar el paradigma de programación orientado a aspectos. El estudio busca proponer al lector una visión clara de la línea de base sobre la cual se sustenta este trabajo y soporta los resultados con base en mediciones en múltiples equipos de trabajo.

La Programación Orientada a Aspectos POA surge como complemento para la programación Orientada a Objetos -POO-y tiene como objetivo aumentar la modularidad y el nivel de reutilización en el proceso de desarrollo de software. Se entiende por aspecto aquel módulo software que no puede ser encapsulado en un procedimiento y no es una unidad funcional en la que se pueda dividir un sistema. Por el contrario son propiedades que cruzan varios módulos y afectan a la ejecución o semántica de los componentes, representan una funcionalidad o propiedad no posible de modularizar usando paradigmas clásicos. Ejemplos de aspectos son: la autenticación, rendimiento, gestión de memoria, acceso a sistemas informáticos, entre otros(Kiczales et al., 1997).

Según los precursores de POA(Kiczales et al., 1997), *“Un aspecto es una unidad modular que se disemina por la estructura de otras unidades funcionales. Los aspectos existen tanto en la etapa de diseño como en la de implementación. Un aspecto de diseño es una unidad modular del diseño que se entremezcla en la estructura de otras partes del diseño. Un aspecto de programa o de código es una unidad modular del programa que aparece en otras unidades modulares del programa”*

El enfoque de POA ha demostrado ser una tecnología potente para manejar las funcionalidades transversales, extrayendo en módulos el código que atraviesa varios componentes del sistema y que de otra manera se repetiría. Esta separación permite la construcción de un software con mayor modularidad facilitando la reutilización, el mantenimiento y la evolución de los sistemas. Sin embargo, según (Mehmood y Jawawi, 2011) *“la Orientación a Aspectos es una zona bastante inmadura”*, lo cual representa un problema para la adopción de estas técnicas en proyectos a gran escala.

CONTEXTUALIZACIÓN

Marcos de trabajo

Marco de Trabajo del inglés Framework, se define como “un conjunto de componentes físicos y lógicos estructurados de tal forma que permiten ser reutilizados en el diseño y desarrollo de nuevos sistemas de información”(Guerrero y Recaman, 2009). Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado, a partir de una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación (Saavedra, 2009). Los marcos de trabajo(Guerrero y Suárez, 2010) contienen patrones(Guerrero et al., 2013) y buenas prácticas que apoyan el desarrollo de un producto y un proceso con calidad. Lo anterior permite vislumbrar la importancia de adoptar un Marco de Trabajo y cómo su selección debe ser una de las actividades relevantes al inicio de todo proceso de desarrollo software.

Arquitectura software -AS-

La AS se define como el diseño de mayor nivel de abstracción en la estructura de un sistema informático, se podría definir como “El arte y la ciencia de diseñar software”(Gutiérrez, 2010). Es la agrupación de patrones y abstracciones que proveen de un marco de referencia que sirve como guía para la construcción de un software. En términos de un sistema o programa, la arquitectura de software se define como la estructura o estructuras del sistema, que comprende elementos de software, las propiedades externamente visibles de esos elementos y las relaciones entre ellos (Bass et al., 20023).

Conceptos fundamentales dela Programación Orientada a Aspectos

POA maneja un conjunto de conceptos propios del paradigma, por lo tanto, es importante tener una contextualización básica para comprender las definiciones de la temática. La Tabla 1 presenta los principales conceptos de POA.

Marcos de trabajo orientado a aspectos

Ahora bien, un Marco de Trabajo Orientado a Aspectos permite coordinar las funcionalidades transversales inherentes a todos los componentes de un software, minimizando notoriamente el código disperso por toda la aplicación. Un ejemplo claro es la función de seguridad (Vidal et al., 2012a) que debe estar presente en

todos los puntos de ejecución del sistema. En cada uno de esos puntos se debe implementar el código necesario para cumplir el requerimiento, lo cual en los paradigmas clásicos conlleva más líneas de código. La solución POA es manejar esta funcionalidad transversal de forma modular como un aspecto para evitar problemas de scattering o esparcimiento de elementos funcionales o no funcionales en diferentes módulos de aplicación (Vidal et al., 2012b)(Vidal et al., 2012c).

Tabla 1: Conceptos Fundamentales de POA

Término	Descripción
Concern	En español incumbencia, hace referencia a todo lo que sea importante para los stakeholders, conocidos como el personal que es afectado directa o indirectamente por un software, ya sea usuario final, patrocinador o persona del equipo de desarrollo. Por ejemplo, una incumbencia puede ser un requisito funcional, un requisito no funcional o una limitación del diseño del sistema. Puede ser más que un requisito del sistema, incluso puede ser un factor de bajo nivel como el manejo de la memoria caché y buffer(Jacobson y Ng, 2004).
Component	Módulo de software que puede ser encapsulado en un procedimiento, son unidades funcionales en las que se descompone un sistema(Asteasuain y Contreras, 2002)
Aspect	Módulo software que no puede ser encapsulado en un procedimiento. Son propiedades que afectan a la ejecución o semántica de los componentes(Vinuesa, 2007).
JoinPoint	Es un punto de la ejecución de un programa bien definido donde se podrá añadir código (funcionalidad). No todos los puntos de ejecución de un programa son puntos de enlace, sólo aquéllos que puedan ser tratados de forma controlada. Cada sistema que soporta POA ofrece una serie de puntos de enlace distintos. Ejemplos de puntos de enlace pueden ser invocaciones a métodos, acceso a campos, etc.(Elrad et al., 2004)
Waver	En español tejedor, es el encargado de realizar la mezcla de los diferentes mecanismos de abstracción y composición que aparecen en los lenguajes de aspectos y componentes, ayudándose de los JoinPoint. (Navasa, 2008)
PointCut	Es un conjunto de instrucciones que se le pasan al tejedor con el fin de que sepa qué código (Advice del Aspecto) se le debe añadir en qué JoinPoint a una aplicación. Un ejemplo podría ser invocar el método X del aspecto cuando se produzca un acceso de lectura al campo W de la clase C(Dong, 2011)
Advice	Es el código del aspecto que se ejecuta en los Joinpoints seleccionados por un PointCut. Normalmente desde su código se puede acceder al contexto de ejecución del punto de enlace (se puede acceder a los valores de variables, etc.)(Sousa, 2008)

METODOLOGÍA PARA EL DESARROLLO DEL ESTUDIO COMPARATIVO

El estudio comparativo de los marcos de trabajo que utilizan el paradigma POA se planteó bajo un esquema progresivo y agrupó las actividades que se describen a continuación.

Identificación de marcos de trabajo

La identificación de marcos de trabajo constituyó una fase importante en la investigación, se realizó una búsqueda rigurosa en diferentes fuentes de información para identificar los marcos de trabajo existentes que soportan el paradigma de la programación orientada a aspectos. Se logró identificar un total de 10 marcos de trabajo, la fuente primaria para cada uno de ellos fue su propia plataforma Web y la documentación asociada, a continuación se relacionan los Marcos de trabajo identificados. 1.- AspectJ; 2.- JBoss AOP ; 3.- Nanning; 4.- Aspectwerkz; 5.- AspectF; 6.- Spring Framework ; 7.- Zend Framework; 8.- PostSharp; 9.- JAC; 10.- CALI

La obtención de la lista de marcos de trabajo permitió realizar el siguiente análisis, de los 10 marcos de trabajo identificados, se consideró pertinente descartar *Nanning*, *JAC* y *CALI*, puesto que los proyectos que los mantenían no registraban actualización en menos de 10 años, por lo cual se consideraron obsoletos. Es importante resaltar que el alcance del presente trabajo fue la comparación de marcos de trabajo orientados a aspectos, independientemente del lenguaje de programación sobre el cual esté construido. Esta razón justifica la no inclusión de lenguajes o extensiones que no brinden herramientas que puedan llegar a ser marcos de trabajo.

Selección y definición de los equipos de desarrollo

Esta fase es clave en la presente investigación puesto que delimita y clarifica el alcance y el procedimiento realizado. Se seleccionaron múltiples equipos de desarrollo divididos en dos clases: 1) Investigadores del proyecto, quienes al iniciar el proyecto solamente contaban con conocimiento y experiencia en desarrollos

bajo el paradigma POA, ésta experiencia y conocimiento se reduce al paradigma como tal y su uso en componentes, más no al conocimiento o uso de marco de trabajo con soporte para POA. 2) Desarrolladores pertenecientes a un colectivo de empresas de desarrollo de software en Colombia que aceptaron apoyar el proyecto, estos desarrolladores contaban con experiencia en el desarrollo de aplicaciones empresariales orientadas a la Web pero sin experiencia en POA, dada la cobertura a nivel nacional y distribución geográfica de las empresas, esta clase se subdividió en 11 subclases y la cantidad mínima de desarrolladores en cada subclase fue de 3.

Definición del producto a desarrollar por parte de los equipos de desarrollo

En esta actividad se establecieron los requerimientos para un conjunto de componentes software de un proyecto específico que incluían funcionalidades transversales, estos requerimientos fueron socializados con las dos clases de equipos de desarrollo. Cada equipo implementó los requerimientos entregados en los marcos de trabajo con soporte para POA seleccionados en la presente investigación. Al finalizar la fase de desarrollo de los requerimientos entregados a los equipos, cada uno de los participantes asignó una valoración cuantitativa a los criterios definidos en esta investigación, teniendo en cuenta las escalas propuestas para cada uno. Esta asignación se realizó con base en la experiencia durante el proceso de implementación de los requisitos entregados. Los criterios y las escalas de cuantificación se presentan en la descripción de la siguiente fase de la metodología.

Es importante resaltar que el estudio presentado en este artículo se centra en la fase de la implementación de componentes de software utilizando marcos de trabajo POA. La consideración de la POA en otras etapas del ciclo de desarrollo, como por ejemplo la toma de requerimientos, el diseño y las pruebas del software será objeto de trabajos futuros.

Definición de criterios de comparación

Para el análisis de los marcos de trabajo identificados se estableció un conjunto de criterios de comparación, para este propósito se realizó un proceso de investigación y revisión de otras investigaciones que permitió identificar 2 estudios comparativos de marcos de trabajo y herramientas POA. El primero fue *AOP@Work: AOP tools comparison* (Kersten, 2005), que realiza un análisis desde la perspectiva de los mecanismos lingüísticos y las interpretaciones de cada herramienta. El segundo fue *Rating of Open Source AOP Frameworks in .NET* (Gnanasekaran, 2008), que presenta una comparación de marcos de trabajo de código abierto sobre .Net que soportan el paradigma POA y en donde se contemplan criterios como el soporte que tienen los marcos de trabajo y la disponibilidad de documentación, entre otros. Los estudios mencionados anteriormente se tomaron como línea de base en la definición de los criterios para la investigación presentada en este documento. Los criterios son: 1) Curva de aprendizaje; 2) Documentación; 3) Facilidad de adopción para desarrolladores; 4) JoinPoints (Puntos de unión); 5) Seguridad; 6) Madurez de la tecnología y soporte; 7) Calidad de la metodología adoptada; y 8) Manejo de transacciones

Curva de aprendizaje: Este criterio de comparación es relevante en el proyecto, pues el alcance de la investigación es la comparación de marcos de trabajo más no la utilización de lenguajes orientados a aspectos. La curva de aprendizaje está determinada por el grado de complejidad para utilizar el marco de trabajo en un proceso de desarrollo. Este grado se presentó en una escala de 4 niveles, de 0 a 3, en donde 0 indica un alto grado de complejidad para aprender a utilizar el marco y 3 que el marco de trabajo ofrece facilidad para el desarrollo de componentes, los otros valores permiten establecer puntos intermedios en esta escala.

Documentación: Este criterio se planteó para evaluar el nivel de utilidad de la documentación disponible para cada uno de los marcos de trabajo. Para este criterio se definieron 2 variables: primero utilidad de la documentación y segundo actualización. Para las dos variables se utilizó una escala de valoración de 0 a 3 que permitiera normalizar los resultados a la escala general. Para la primera variable un valor de 0 indica que la documentación tiene un nivel de utilidad insignificante y un valor de 3 indica que la documentación fue útil. En el caso de la segunda variable, un valor de 0 representa que la documentación se encuentra desactualizada, es decir, que no ayudó a los miembros de los equipos en el uso del marco de trabajo y una valoración de 3 representa que la documentación le permitió solucionar inconvenientes durante la fase de desarrollo. Así como en el criterio anterior, otros valores representan puntos intermedios.

Facilidad de adopción para desarrolladores: Este criterio se relaciona con el nivel de dificultad o facilidad que representa la adopción del marco de trabajo, en términos de las herramientas que ofrece para integrar el marco al proceso de desarrollo. La medición de este criterio se evalúa a partir del nivel de integración que ofrece el marco para adaptarlo a un IDE (Integrated Development Environment) en español entorno integrado de desarrollo, un valor de 3 representa que el marco ofrece un paquete de librerías para integración a entornos de desarrollo y por el contrario 0 que no ofrece soporte para este tipo de integración.

JoinPoints (Puntos de unión): Nivel de soporte del marco de trabajo para la inserción de JoinPoints que permitan agregar código para las funcionalidades transversales. En este caso si el Marco provee soporte para la ejecución de JoinPoint a nivel de métodos, de excepciones y de asignaciones tiene un valor de 3, en caso de solo brindar soporte a nivel de método tendrá un valor de 0 pues es lo más básico que puede ofrecer un marco con soporte POA, si soporta la ejecución de métodos y las excepciones tienen un valor de 1 y si soporta la ejecución de métodos y asignaciones un valor de 2.

Seguridad: En la programación orientada a aspectos existen tres elementos que debe tener en cuenta un marco de trabajo: a) El manejo de excepciones, este elemento es analizado en este estudio en el criterio JoinPoints. b) La persistencia, este elemento no se tuvo en cuenta en este estudio pues existen componentes que no usan POA pero que pueden ser utilizados por los Marcos seleccionados en el presente estudio y c) La Gestión de la seguridad, este criterio busca clasificar los marcos que tenga soporte directo a la gestión de seguridad como componente transversal en un desarrollo, se estableció de la siguiente forma:

Si el marco de trabajo provee objetos para la gestión de sesiones y vistas para la configuración de la seguridad se asignó una valoración de 3, si carece de estos dos elementos se asigna una valoración de 0. Los casos intermedios en este criterio se establecen así: una valoración de 2 para aquellos que provean los objetos para la construcción del componente transversal de seguridad y 1 para aquellos que provean al menos un patrón que permita implantar un componente transversal de seguridad.

Madurez de la tecnología y soporte: Este criterio analiza la madurez del marco de trabajo y soporte al que es posible tener acceso por parte del desarrollador, por ende, las variables que se analizaron en este criterio están orientadas a la existencia y calidad del soporte brindado por los propietarios del marco de trabajo. En la escala definida el valor de 3 representa la existencia de 3 elementos: una fuente primaria de información acerca del marco, un control de versiones actualizado a intervalos de tiempo reciente, tiempos de respuesta directos a las inquietudes realizadas por parte de los desarrolladores. La escala define el valor más bajo cuando el marco no presenta ninguno de los elementos.

Calidad de la metodología adoptada: Este criterio establece el nivel de calidad en la metodología adoptada por el marco de trabajo para lograr soportar el paradigma de la programación orientada a aspectos. Cada marco de trabajo gestiona y expone la programación orientada a aspectos de forma particular, soportado lógicamente en el lenguaje base sobre el cual está construido. En este estudio se contemplaron marcos de trabajo en 3 lenguajes de programación diferentes, por ende era importante definir variables para medir la forma como los marcos exponen todas las funcionalidades de POA al desarrollador.

La escala de valores de este criterio establece un valor de 3 a los marcos de trabajo que permiten la gestión de conflictos, es decir, permiten gestionar las consecuencias de la descomposición de los sistemas software en el desarrollo de sistemas orientados a aspectos (Casas y Vanoli, 2008) y la administración de aspectos genéricos. La escala valora en 0 los marcos que no tienen ninguna de las dos características. Dado que la importancia en la calidad metodológica propuesta por el marco es más relevante en el tema de solución de conflictos se establece el valor de 2 a los marcos que presentan solo esta característica.

Manejo de transacciones: Representa el nivel de soporte del marco de trabajo para la implementación de transacciones transversales a una aplicación. Este criterio es importante porque permite identificar qué tanto soporta el marco de trabajo la implementación de una funcionalidad transversal para el manejo de transacciones. Para la medición de este criterio se definieron dos variables, la primera, el nivel que brinda el marco de trabajo al soporte declarativo para definición de transacciones, se asigna un valor de 3 si el marco cumple con este soporte y un valor de 0 si no ofrece esta característica. La segunda, la facilidad de construir PointCuts que utilicen los métodos transaccionales, para este criterio, se asigna un valor de 3 si la utilización de métodos transaccionales facilitó la construcción del PointCut, se asigna un valor de 0 si la utilización de los métodos transaccionales eleva la complejidad del desarrollo.

Escala de valoración de criterios

Dada la heterogeneidad de los criterios y teniendo en cuenta el proceso de análisis para cada uno de ellos, se recurrió a una escala de valores cuantitativa para representar el nivel de aprobación o desaprobación de cada uno de los criterios por parte de los equipos de desarrollo. La escala normalizada se presenta entre 0 y 3, siendo 3 el nivel de cumplimiento total del criterio y 0 el no cumplimiento del mismo. En la

Tabla 2 se presenta el significado de cada valor en la escala normalizada.

Tabla 2: Escala de valoración

Valoración	Descripción
0	No cumplimiento del criterio
1	Cumplimiento deficiente del criterio
2	Cumplimiento aceptable del criterio
3	Cumplimiento total del criterio

Instrumentos para la consolidación de resultados

La evaluación de Marcos de Trabajo se llevó a cabo en 2 momentos que transcurrieron de forma paralela donde participaron los equipos de desarrollo definidos en la sección *Selección y Definición Equipos de Desarrollo*, cada equipo realizó las actividades descritas en la sección *Definición del Producto a Desarrollar por Parte de los Equipos de Desarrollo*, para la evaluación de los criterios y su tabulación una vez terminado el proceso que debían realizar los miembros de los equipos de desarrollo, se aplicaron instrumentos que permitieron la consolidación de la información. Ver Fig. 1 y Fig. 2.

		Marcos de Trabajo											
		Marco 1				...				Marco N			
		0	1	2	3	0	1	2	3	0	1	2	3
Criterios	Criterio 1												
	...												
	...												
	...												
	...												
	Criterio N												

Fig. 1: Instrumento para evaluación de marcos de trabajo

Para la consolidación de los datos se utilizó como línea de base el instrumento construido en el *estudio Frameworks para el desarrollo de aplicaciones Web que utilizan código abierto* (Guerrero y Recaman, 2009). Ver Fig. 2. Por medio de este instrumento se consolidó la valoración asignada a cada marco de trabajo por los investigadores y por los desarrolladores. La valoración de cada criterio para cada marco se obtiene calculando el promedio entre las valoraciones asignadas por los investigadores y expertos. Como se aprecia en el instrumento, se realiza la sumatoria de las filas y las columnas para tener valores totales de los criterios y los marcos de trabajo.

Valoración del criterio para el tipo de Marco de Trabajo		Tipos de Marcos de Trabajo						Subtotal
		#	...	...	#	#	#	Σ
Criterios	#	...			#		#	Σ
	...	#				#	...	Σ
	...	...	#		...	#		...
	...		#			...		...
	#							...
	...	...	#				#	Σ
Subtotal		Σ	Σ	...	...	...	Σ	Σ

Representa la suma de las escalas de cada criterio

Representa la suma de las escalas del tipo de Marco para cada criterio

Fig. 2: Instrumento para consolidación de resultados

ANÁLISIS DE DATOS

Luego de la consolidación de los datos obtenidos del proceso realizado por los investigadores y expertos, se procedió a realizar un análisis de los resultados, a continuación se presentan los resultados obtenidos en cada equipo.

Equipo de investigadores

En este equipo de desarrolladores el promedio de la sumatoria de cada fila, que hace referencia a los criterios, fue igual a 14.63, el 62.5% de los criterios analizados en los marcos de trabajo superaron este promedio, es decir que 5 de los 8 criterios se cumplen con un alto nivel en los marcos de trabajo analizados. A nivel de columnas, el promedio de la sumatoria es igual a 16.71, el 57.1% de los marcos de trabajo superan este promedio, es decir 4 de los 7 cumplen en un nivel alto los criterios analizados. En la Tabla 3 se aprecia la aplicación del instrumento.

Tabla 3: Aplicación de instrumento - Equipo Investigadores

	AspectJ	JBoss AOP	Aspectwerkz	AspectF	Spring Framework	Zend Framework	PostSharp	Subtotal Filas	
Manejo de transacciones	2,75	2,00	1,25	1,75	2,75	2,75	2,50	15,75	Promedio filas → 14,63
Calidad de la metodología adoptada	2,25	1,25	1,50	1,00	3,00	3,00	2,50	14,50	
JoinPoints (Puntos de unión)	2,75	1,25	1,25	1,75	2,75	2,75	2,50	15,00	
Seguridad	3,00	1,25	1,50	2,00	2,25	2,25	2,75	15,00	
Facilidad de adopción para desarrolladores	2,50	1,50	1,75	1,50	2,25	2,25	2,00	13,75	
Curva de aprendizaje	2,50	1,50	1,50	1,00	2,25	2,25	2,25	13,25	
Madurez de la tecnología y soporte	2,50	1,50	1,25	1,50	2,75	2,75	2,50	14,75	
Documentación	3,00	1,50	1,25	1,50	2,75	2,75	2,25	15,00	
Subtotal columnas	21,25	11,75	11,25	12,00	20,75	20,75	19,25	117,00	
Promedio columnas → 16,71									

Equipo de expertos en desarrollo

El instrumento para la consolidación de los datos de las encuestas aplicadas a los expertos refleja una diferencia significativa en los subtotales de las filas y las columnas, comparados con los resultados del equipo de investigadores, en el equipo de expertos los resultados son menores en un promedio de 3.46 puntos. En la

Tabla 4 el promedio de la sumatoria de las filas es igual a 11.40, el 75% de los criterios superan este promedio. A nivel de columnas el promedio de las sumatorias es igual a 13.02, el 42.9% superan este promedio.

RESULTADOS DEL ESTUDIO COMPARATIVO*Comparación de los resultados obtenidos entre los dos equipos de desarrollo*

Los resultados parciales obtenidos en el análisis de los criterios en cada equipo permiten observar una coincidencia del 60% de los criterios que superaron los promedios. Ver Tabla 5.

Tabla 4: Aplicación de instrumento - Equipo Expertos

	AspectJ	JBoss AOP	Aspectwerkz	AspectF	Spring Framework	Zend Framework	PostSharp	Subtotal Filas	
Manejo de transacciones	2,33	1,33	1,33	1,33	2,67	1,00	2,67	12,67	Promedio filas → 11,40
Calidad de la metodología adoptada	3,00	1,17	1,33	1,33	2,67	1,00	2,00	12,50	
JoinPoints (Puntos de unión)	1,83	1,17	1,17	0,50	2,50	0,67	2,17	10,00	
Seguridad	1,83	1,67	0,83	0,83	2,50	0,67	2,33	10,67	
Facilidad de adopción para desarrolladores	1,33	2,00	0,83	0,83	2,83	0,83	2,17	10,83	
Curva de aprendizaje	3,00	1,33	0,83	0,83	2,17	0,67	2,67	11,50	
Madurez de la tecnología y soporte	2,33	1,33	1,00	1,00	2,00	1,00	2,83	11,50	
Documentación	2,67	0,83	0,83	0,83	2,00	1,33	3,00	11,50	
Subtotal columnas	18,33	10,83	8,17	7,50	19,33	7,17	19,83	91,17	
Promedio columnas → 13,02									

Tabla 5: Comparación resultado criterios

Investigadores		Expertos	
Criterio	Valoración	Criterio	Valoración
Manejo de transacciones	15,75	Manejo de transacciones	12,67
JoinPoints (Puntos de unión)	15,00	Calidad de la metodología adoptada	12,50
Seguridad	15,00	Madurez de la tecnología y soporte	11,50
Documentación	15,00	Documentación	11,50
Madurez de la tecnología y soporte	14,75	Curva de aprendizaje	11,50

Los resultados parciales obtenidos de los marcos de trabajo en cada equipo permiten observar una coincidencia del 85.7% de los criterios que superaron los promedios. Ver Tabla 6.

Tabla 6: Comparación resultado marcos de trabajo

Investigadores		Expertos	
Marco de Trabajo	Valoración	Marco de Trabajo	Valoración
AspectJ	21,25	PostSharp	19,83
Spring Framework	20,75	Spring Framework	19,33
Zend Framework	20,75	AspectJ	18,33
PostSharp	19,25		

Resultados consolidados

La consolidación de los resultados obtenidos por el equipo de investigadores y los expertos es fundamental en este estudio, puesto que permite obtener los resultados finales. En la Tabla 7 se presentan los resultados totales.

Tabla 7: Aplicación de instrumento – Consolidado

	AspectJ	JBoss AOP	Aspectwerkz	AspectF	Spring Framework	Zend Framework	PostSharp	Total Filas	
Manejo de transacciones	2,54	1,67	1,29	1,54	2,71	1,88	2,58	14,21	Promedio filas → 13,01
Calidad de la metodología adoptada	2,63	1,21	1,42	1,17	2,83	2,00	2,25	13,50	
JoinPoints (Puntos de unión)	2,29	1,21	1,21	1,13	2,63	1,71	2,33	12,50	
Seguridad	2,42	1,46	1,17	1,42	2,38	1,46	2,54	12,83	
Facilidad de adopción para desarrolladores	1,92	1,75	1,29	1,17	2,54	1,54	2,08	12,29	
Curva de aprendizaje	2,75	1,42	1,17	0,92	2,21	1,46	2,46	12,38	
Madurez de la tecnología y soporte	2,42	1,42	1,13	1,25	2,38	1,88	2,67	13,13	
Documentación	2,83	1,17	1,04	1,17	2,38	2,04	2,63	13,25	
Total columnas	19,79	11,29	9,71	9,75	20,04	13,96	19,54	104,08	
Promedio columnas → 14,87									

Análisis de criterios (filas)

Las filas de la Tabla 7 indican la valoración obtenida para cada marco de trabajo en cada criterio, a partir del promedio calculado entre los valores asignados por los investigadores y el equipo de expertos. La suma de los totales de cada fila es 104.08 y el promedio de la sumatoria es 13.01. Los valores significativos son los que superan el promedio de las filas, puesto que indican un alto nivel de cumplimiento del criterio en los marcos de trabajo analizados. La Fig. 3 presenta gráficamente los resultados de la sumatoria de las filas, en la gráfica las columnas que se aprecian con trama diagonal representan los criterios que superaron el promedio de 13.01.

Fig. 3: Análisis gráfico resultado filas

En los resultados consolidados, de los ocho (8) criterios utilizados para evaluar los Marcos de Trabajo, el 50%, es decir 4 criterios superaron el valor promedio de la sumatoria de las filas de la Tabla 8. En la Tabla 8 se presentan los criterios con mayor valoración.

Tabla 8: Criterios con mayor valoración

Criterio	Valoración
Manejo de Transacciones	14,21
Calidad de la Metodología Adoptada	13,50
Documentación	13,25
Madurez de la tecnología y soporte	13,13

Análisis de marcos de trabajo (columnas)

En la Tabla 7 la suma de las columnas indica el total de la valoración obtenida por el marco de trabajo en la evaluación de los criterios. La suma de los totales de las columnas es 104.8 y el promedio de esta sumatoria es 14.87. Los valores inferiores al promedio no se consideran relevantes en la investigación, pues reflejan que el marco de trabajo no cuenta con una mayoría de los criterios definidos. En la Fig. 4 se observa la representación gráfica de la sumatoria de las columnas correspondientes a los marcos de trabajo. Las columnas que se aprecian con trama diagonal representan los criterios que superaron el promedio de 14.87.

Fig. 4: Análisis gráfico resultado columnas

El análisis de los marcos de trabajo inició con un grupo de 10 marcos, en el primer análisis se descartaron 3 por falta de actualización del proyecto, por lo que se consideraron tecnologías obsoletas. Con la consolidación de los datos de los instrumentos aplicados a los investigadores y expertos, se obtuvo que el 43% de los marcos de trabajo para el desarrollo de software orientado a aspectos son los que cumplen en mayor nivel con los criterios definidos. Ver Tabla 9

Tabla 9: Marcos de trabajo mayor valoración

Marco de Trabajo	Valoración
Spring Framework	20,04
AspectJ	19,79
PostSharp	19,54

Marcos de trabajo con menor valoración

Finalmente, con los resultados de la

Tabla 4 es posible obtener los marcos de trabajo con menor valoración, siendo una conclusión importante en esta investigación, pues es posible establecer que los tres marcos de trabajo relacionados en la tabla no

deben adoptarse en el desarrollo de un proyecto software al no cumplir con los criterios definidos. Ver Tabla 10.

Tabla 10: Marcos de trabajo menor valoración

Marco de Trabajo	Valoración
AspectF	9,58
Aspectwerks	9,67
JBoss AOP	11,33

CONCLUSIONES

Entre las ventajas teóricas que ofrece POA a los desarrolladores se encuentran: evita duplicidad de código, permite el desarrollo modular y ofrece una mayor reusabilidad, sin embargo, en la práctica y a través del uso de marcos de trabajo, estas ventajas no son tan evidentes pues se presentan conflictos entre el código escrito en POO y el código realizado en POA, dificultando el concepto de reusabilidad característico de POA. Con respecto al desarrollo modular es posible construir con POA dos módulos que funcionen correctamente por separado, sin embargo, al acoplarlos se evidencian problemas entre los módulos.

Los resultados obtenidos en el presente proyecto provienen de dos fuentes, por un lado el equipo de investigadores interesados en adoptar el paradigma POA a los desarrollos de la institución de educación superior a la cual pertenecen, por otro lado un equipo de desarrolladores sin interés en la temática. Los resultados del estudio arrojaron resultados muy similares, lo cual resulta interesante pues las dos fuentes estuvieron aisladas durante el periodo del estudio. Esto permite afirmar que los resultados obtenidos no se ven sesgados por las características propias de cada subgrupo.

Seleccionar un Marco de Trabajo puede llegar a significar el éxito o fracaso de un desarrollo software, independiente del lenguaje de programación o del paradigma que se utilice, es necesario tener un conocimiento profundo del Marco antes de emprender un desarrollo de carácter profesional. La academia brinda la oportunidad de probar soluciones en diferentes contextos y con diferentes herramientas, pero esto es un lujo que no se puede dar el sector productivo. La adopción de POA en la industria requiere una mejora substancial en los marcos de trabajo que soportan POA, de tal forma que las ventajas ofrecidas por el paradigma sean una realidad a la hora de hacer desarrollos comerciales.

REFERENCIAS

Asteasuain, F., y Contreras, B. E., *Programación Orientada a Aspectos Análisis del paradigma*. Tesis de Licenciatura, Depto. de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur. (2002). (En línea) <http://www.angelfire.com/ri2/aspectos/Tesis/tesis.pdf>. Acceso: 5 de agosto (2012).

Bass, L., Clements, P., y Kazman, R., *Software Architecture in Practice*, 3 edición, 63-77, Pearson Educación. Inc., Massachusetts, USA. (2003).

Casas, S., y Vanoli, V., AAC: Agente *Administrador de Conflictos entre Aspectos en AspectJ*. XV Encuentro Chileno de Computación (ECC08). Punta Arenas, Chile, 10 al 15 de Noviembre (2008). (En línea) http://www.researchgate.net/publication/228683186_AAC_Agente_Administrador_de_Conflictos_entre_Aspectos_en_AspectJ/file/9fcfd50ed4cfe5c598.pdf. Acceso: 25 de junio (2012).

Dong, Z., *Aspect Oriented Programming Technology and the Strategy of Its Implementation*. 2011 International Conference on Intelligence Science and Information Engineering (ISIE). 457 - 460, Wuhan, China, 20 al 21 de agosto (2011).

Elrad, T., Aksit, M., y Clarke, S., *Introduction to Aspect-Oriented Software Development*. Magazine Communications of the ACM, 44(10), 29-32 (2001).

Gnanasekaran, V., *Rating of Open Source AOP Frameworks in .NET*. Code Project (2008). (En línea) <http://www.codeproject.com/Articles/28387/Rating-of-Open-Source-AOP-Frameworks-in-NET>. Acceso: 15 de mayo (2012).

Guerrero, A., y Suárez, J., *Patrones de diseño para el desarrollo de aplicaciones web. Desarrollando Software con Buenas Prácticas*, 20-65, (Sic) Editorial Ltda., Bucaramanga, Colombia. (2010).

- Guerrero, C. A., Suárez, J. M., y Gutiérrez, L. E., *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web*. Inf. Tecnol., ISSN: 0718-0764 (en línea), 24(3), 103-114, (2013). http://www.scielo.cl/scielo.php?script=sci_abstract&pid=S0718-07642013000300012&lng=es&nrm=iso&tlng=es. Acceso: 18 de junio (2013)
- Guerrero, C., y Recaman, H., *Marcos de Trabajo (Framework) para soportar el desarrollo de aplicaciones Web de código abierto*, 10-25, (Sic) Editorial Ltda., Bucaramanga, Colombia. (2009).
- Gutiérrez, L. E., *Arquitectura Software, Investigación Aplicada a la Construcción de Marcos de Trabajo*. 10-48, (Sic) Editorial Ltda., Bucaramanga, Colombia. (2010).
- Jacobson, I., y Ng, P.W., *Aspect-Oriented Software Development with Use Cases*. Addison Wesley Publishing Company, California, USA. (2004).
- Kersten, M., *AOP@Work: AOP tools comparison*. Developer Works (2005). (En línea) <http://www.ibm.com/developerworks/library/j-aopwork1/>. Acceso: 16 de mayo (2012).
- Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J. M., e Irwin, J., *Aspect-Oriented Programming*. ECOOP'97—Object-Oriented Programming, Jyväskylä, Finlandia, 9 al 13 de junio (1997).
- Mehmood, A., y Jawawi, D. N. A., *A comparative Survey of aspect-oriented code generation approaches*, 5th Malaysian Conference in Software Engineering (MySEC), 147-152, Johor Bahru, Malasia, 13 y 14 de diciembre (2011).
- Navasa, A., *Marco de trabajo para el desarrollo de arquitecturas software orientado a aspectos*, Tesis Doctoral, Depto. de Ingeniería de Sistemas Informáticos y Telemáticos. Universidad de Extremadura, Cáceres, España, (2008)
- Saavedra, E., *Grails: Framework para el desarrollo de aplicaciones Web*. Revista de Software Libre ATIX, 32-41, (2009). (En línea) <http://osl.ugr.es/descargas/atix08.pdf>. Acceso: 08 de mayo (2012).
- Sousa, I., *Aspect-Oriented Requirements Analysis*. Tesis Doctoral, Faculdade de Ciencias e Tecnologia, Universidad Nova de Lisboa, Lisboa, Portugal. (2008).
- Vidal, C., Del Río, C., y Saens, R., *Extensión y Aplicación de AspectZ a la Administración de un Sistema de Fichas de Salud Electrónicas en Chile*. Información Tecnológica, ISSN: 0718-0764 (en línea), 23(5), 23-32, (2012). http://www.scielo.cl/scielo.php?script=sci_abstract&pid=S0718-07642012000500004&lng=es&nrm=iso&tlng=es. Acceso: 29 de noviembre (2013)
- Vidal, C., Hernández, D., Pereira, C., y Del Río, M., *Aplicación de la Modelación Orientada a Aspectos*. Información Tecnológica, ISSN: 0718-0764 (en línea), 23(1), 3-12, (2012). http://www.scielo.cl/scielo.php?script=sci_abstract&pid=S0718-07642012000100002&lng=es&nrm=iso&tlng=es. Acceso: 28 de noviembre (2013).
- Vidal, C., Schmal, R., Rivero, S., y Villarroel, R., *Extensión del Diagrama de Secuencias UML (Lenguaje de Modelado Unificado) para el Modelado Orientado a Aspectos*. Información Tecnológica, ISSN: 0718-0764 (en línea), 23(6), 51-62. (2012). http://www.scielo.cl/scielo.php?script=sci_abstract&pid=S0718-07642012000600007&lng=es&nrm=iso&tlng=es. Acceso: 28 de noviembre (2013).
- Vinuesa, A., *Separación dinámica de aspectos independiente del lenguaje y plataforma mediante el uso de reflexión computacional*, Tesis Doctoral, Depto. de Informática, Universidad de Oviedo. Oviedo, España, (2007).