

Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web

Carlos A. Guerrero, Johanna M. Suárez* y Luz E. Gutiérrez

Grupo de Investigación en Ingeniería del Software-GRIIS, Unidades Tecnológicas de Santander, Calle de los Estudiantes # 9-82 Ciudadela Real de Minas - Bucaramanga - Santander - Colombia
(e-mail: cinving@uts.edu.co; jomasupe@hotmail.com; luzelenagl@live.com)

* Autor a quien debe ser dirigida la correspondencia

Recibido Sep. 25, 2012; Aceptado Nov. 19, 2012; Versión final recibida Ene. 28, 2013

Resumen

Se presenta el análisis de identificación de Patrones de Diseño definidos por The Gang of Four (GOF) en procesos de desarrollo de software orientados a la Web. Inicialmente se construye un conjunto de criterios para evaluar y seleccionar procesos de desarrollo formales de gran envergadura. Se establece el tamaño de la muestra para aplicar los criterios con estricto rigor metodológico, se realiza la inspección del código fuente para identificar el uso de patrones de diseño y se lleva a cabo un proceso que permite identificar los patrones de diseño que son utilizados por expertos del área de la ingeniería del software. Los resultados permiten concluir que en el sector productivo los patrones de diseño han sido aplicados. Sin embargo, su uso es reducido por falta de conocimiento de la existencia de estos patrones o por falta de experiencia para lograr su correcta utilización.

Palabras clave: patrones de diseño, GOF, procesos de desarrollo, aplicaciones web, ingeniería de software

GOF (The Gang of Four) Design Patterns in the context of Process Development of Web-Oriented Applications

Abstract

This article presents the analysis of identification of Design Patterns defined by The Gang of Four (GOF), in processes of software development orientated to the Web. Initially a set of criteria is constructed to evaluate and to select formal processes of development of great importance. The size of the sample is established to apply the criteria with strict methodological rigor. The inspection of the source code is done to identify design patterns and there is carried out a process that it allows to identify design patterns that are used by experts of the area of the engineering of the software. The results allow concluding that in the productive sector the design patterns have been applied. However, its use is reduced for the lack of knowledge of the existence of these patterns or for the lack of experience to achieve its correct utilization.

Keywords: design patterns, GOF, processes of development, web applications, engineering software

INTRODUCCIÓN

Los resultados y procesos socializados en este documento, se obtienen del trabajo de investigación realizado sobre la temática de patrones de diseño GOF, por los integrantes del Grupo de Investigación en Ingeniería del Software -GRIIS- de las Unidades Tecnológicas de Santander. La investigación toma como línea de base la publicación "Design Patterns: Elements of Reusable Object-Oriented Software" (Gamma et al, 1994), a partir de éste libro se fundamenta la conceptualización principal sobre Patrones de Diseño, puesto que es considerado el principal Catálogo de Patrones desde 1994. El objetivo de este artículo consiste en entregar un precedente del estado actual respecto al uso de patrones de diseño GOF en la industria de producción de software en Colombia. La investigación consiste inicialmente en la construcción de un conjunto de criterios que permiten seleccionar una muestra de procesos de desarrollo software, posteriormente se conforma un equipo de expertos que apoyan el proceso de identificar los patrones de diseño GOF que son utilizados en la praxis, de esta manera se logra la identificación de patrones de diseño usados en la actualidad. El tema abordado en esta investigación es importante en el contexto del desarrollo software, puesto que, aplicar patrones de diseño es considerado una buena práctica que apoya el aumento de la calidad en el proceso de desarrollo y el producto obtenido y estandariza el código fuente de una aplicación para facilitar su mantenimiento.

El término de patrón fue dado por primera vez en el año 1977 por el arquitecto Christopher Alexander, quien dio en su libro *A Pattern Language* la siguiente definición: "Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo ni siquiera dos veces de la misma forma". En la ingeniería del software, un patrón constituye el apoyo para la solución a los problemas más comunes que se presentan durante las diferentes etapas del ciclo de vida del software. Ahora bien, los patrones de diseño según *The Gang of Four -GOF-* "describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos" (Gamma et al, 1994), Eric Braude define los patrones de diseño como "combinaciones de componentes, casi siempre clases y objetos que por experiencia se sabe que resuelven ciertos problemas de diseño comunes" (Braude, 2003). En términos generales es posible decir que un patrón de diseño es una solución a un problema recurrente en el diseño de software.

La línea base en el tema de patrones de diseño la impone el catálogo "Design Patterns: Elements of Reusable Object-Oriented Software" (Gamma et al, 1994), en este libro se presenta un conjunto de 23 patrones de diseño identificados a partir del estudio y la experiencia del grupo GOF, quienes se dedicaron a analizar los problemas recurrentes en el desarrollo de software y realizaron una clasificación y agrupación a partir de dos criterios, su propósito y alcance, las categorías definidas son: creacionales, estructurales y de comportamiento: i) patrones creacionales: Se ocupan del proceso de creación de clases y objetos, son los encargados de "abstraer el proceso de instanciación o creación de objetos, ayudan a que el sistema sea independiente de cómo sus objetos son creados, integrados y representados" (Gracia, 2005). Los patrones que hacen parte de esta categoría son cinco (5): Factory Method, Abstract Factory, Builder, Prototype y Singleton; y ii) patrones estructurales: Tratan de la composición de clases y objetos, "se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes" (Gracia, 2005). Los patrones en esta categoría se encargan de lograr que los cambios en los requisitos de la aplicación no ocasionen cambios en las relaciones entre los objetos (Braude, 2003). Los patrones que hacen parte de esta categoría son siete (7): Adapter, Bridge, Composite, Decorator, Façade, Flyweight y Proxy y iii) patrones de comportamiento: Caracterizan las formas en que las clases o los objetos interactúan y distribuyen la responsabilidad. "Son los encargados de las opciones de comportamiento de la aplicación, permitiendo que el comportamiento varíe en tiempo de ejecución, sin estos patrones cada comportamiento tendría que diseñarse e implementarse por separado" (Gracia, 2005). Los patrones que se agrupan en esta categoría son once (11): Interpreter, Template Method, Chain of Responsibility, Command, Iterator, Mediator, Memento, Observer, State, Strategy, Visitor.

METODOLOGÍA PARA IDENTIFICAR PATRONES GOF

El proceso de identificación de patrones de diseño GOF en procesos de desarrollo software requiere de un conjunto de actividades; inicialmente se construye un conjunto de criterios que permita establecer un punto de análisis para identificar los procesos de desarrollo más representativos, siendo obligatorio cumplir estos criterios para catalogarlos como válidos en este análisis, y obtener la muestra de procesos de desarrollo. El cumplimiento de las condiciones que deben presentar los procesos de desarrollo es estricto e importante, puesto que el objetivo de este trabajo de investigación consiste en obtener resultados a partir del estudio de procesos de desarrollo formales, de esta manera, se selecciona un grupo de procesos de desarrollo que cumplan con un estricto control de calidad en las fases de requerimientos, diseño, desarrollo e implantación.

Ahora bien, es importante tener presente que todo proceso de calidad involucra variables que pueden llegar a determinar la complejidad del mismo, entre las cuales están la cantidad de desarrolladores, diseños en UML, nivel de criticidad del sistema, costos del proyecto, modelo de requisitos empleado y el modelo de calidad utilizado, estas variables han sido identificadas por los investigadores a partir de su experiencia en investigación aplicada en el área del desarrollo software y su participación en procesos de desarrollo, además, cuentan con el aval de un equipo de expertos en el desarrollo de software, profesionales con experiencia mayor a 5 años en procesos de desarrollo de aplicaciones Web de tipo empresarial. Las variables anteriormente enunciadas se convierten en criterios claves para seleccionar los procesos de desarrollo que serán analizados para identificar patrones GOF, la definición de los criterios se lleva a cabo en un trabajo conjunto con los investigadores y el equipo de expertos, estableciendo la importancia y justificación de su aplicación.

(C1) Cantidad de Desarrolladores: En un proyecto con pocos desarrolladores es fácil diferenciar cada una de las etapas a realizar. Actividades como la integración y el control de calidad se pueden medir con mayor celeridad. Si por el contrario el equipo de desarrollo es numeroso, se deben analizar variables y estrategias que permitan construir partes del sistema por componentes, de tal forma que se optimice tiempo y recurso humano en todo el proceso. Es en este punto donde el criterio en mención cobra importancia, puesto que la interacción de una alta cantidad de desarrolladores puede inducir el uso de antipatrones, que si bien dan soluciones de corto plazo a problemas específicos, pueden llegar a hacer colapsar el proyecto (Welicki, 2006).

(C2) Diseños UML - Unified Modeling Language (Schach, 2005): Este criterio permite establecer el uso de un lenguaje formal en la etapa de diseño de los proyectos, es importante porque UML está definido como el lenguaje estándar que debe ser empleado en un proceso de desarrollo para el diseño del software; teniendo en cuenta que los patrones de diseño tienen su estructura definida a partir de un diagrama de clases UML (Gutiérrez, 2010), los proyectos a ser analizados deben cumplir este criterio.

(C3) Criticidad de los Sistemas: Hace referencia al nivel de necesidad de la aplicación para los usuarios o clientes, un sistema es más crítico en la medida en que los usuarios dependen de él para llevar a cabo algún proceso o tarea específica, o cuando en sí misma la aplicación es un sistema de información que necesita mantenerse siempre en funcionamiento. Se define un nivel Bajo, Medio y Alto dependiendo de la criticidad de cada sistema analizado.

(C4) Costos: Corresponde a la cantidad de dinero invertida en el proyecto por el patrocinador. Este criterio es importante, aunque un costo mayor no garantiza el éxito de un proceso de desarrollo, sin embargo, sí garantiza que se destine más dinero a recursos humanos para procesos de calidad. La Tabla 1 presenta los niveles utilizados para mostrar un acercamiento del costo de cada proyecto (A. Guerrero & Suárez, 2010):

Tabla 1: Costo de los proyectos. Los valores de la tabla están expresados en dólares estadounidenses.

Nivel	Tipo	Mayor a	Menor a
0	Pequeño	-	25.000
1	Menor	25.000	40.000
2	Normal	40.000	100.000
3	Grande	100.000	-

(C5) Modelo de Requisitos: Todo proyecto software inicia con la definición de los requisitos, existen algunos modelos y estándares que pueden ser usados para facilitar este proceso y que ayuda a la realización de un proyecto con mayor calidad. Los proyectos que se analizan deben cumplir con el requerimiento de adoptar un modelo de requisitos. Entre algunos de los estándares se encuentran:

i) Estándar IEEE 830 – 1998 (Engineering & Committee, 1998): Estándar definido por la IEEE (Institute of Electrical and Electronics Engineers), representa una plantilla que orienta la forma en que debe presentarse una especificación de requisitos y los apartados necesarios para iniciar la etapa de análisis y diseño.

ii) Plantilla de Elicitación de Requisitos de Amador Duran Toro (Toro & Jiménez, 2000): Documento que define las técnicas para realizar el levantamiento de requisitos y cómo recopilar esta información en una serie de plantillas para la representación de cada requisito.

(C6) Modelo de Calidad: El modelo de calidad adoptado para el desarrollo de un proyecto software garantiza que tanto el proceso como el producto que se está construyendo satisfaga las expectativas del cliente. Algunos de estos modelos son:

i) CMMi (Capability Maturity Model Integration) (Fuensanta, 2010): Modelo para ayudar en la mejora y evaluación de los procesos en proyectos software, durante las etapas de desarrollo, mantenimiento y operación del software.

ii) ISO 9001(ISO, 2000): Es la norma definida por ISO que establece los procesos o actividades que se deben hacer para que los procesos en una organización sean realizados con calidad; especifica los requisitos para un sistema de gestión de calidad, y puede ser aplicada a o todo tipo de organizaciones.

iii) ISO 9126(ISO & IEC, 2001): Estándar ISO específicamente para el control de calidad de los productos software.

Selección de la Muestra

La selección de la muestra es un proceso que permite obtener un valor cuantitativo de los procesos de desarrollo software que se analizan, es importante aclarar que “un proceso de desarrollo software se define como el conjunto de actividades necesarias para construir una aplicación o producto software” (Pressman, 2005), teniendo en cuenta esta descripción, son muchos los procesos de desarrollo que pueden hacer parte de la muestra para ser analizados, puesto que, la industria del software se ha desplegado vertiginosamente en todo el mundo durante las últimas décadas y aunque este estudio es acotado para Colombia, es importante definir la muestra de procesos de desarrollo para direccionar la investigación.

Considerando que el tamaño exacto de la población objeto de estudio es desconocido por el gran número de procesos de desarrollo que pueden ser analizados en Colombia, es conveniente aplicar una metodología para calcular el tamaño de la muestra; se selecciona la metodología denominada: “Tamaño de la población infinito o desconocido” (Snedecor & Cochran, 1984), pues permite establecer el tamaño de la muestra para una población infinita o desconocida como en este caso. La fórmula estadística aplicada para el cálculo de la muestra es;

$$n = \frac{P * Q * Z^2}{e^2} \quad (1)$$

Una característica de esta metodología está dada por los valores que se manejan en cada variable, en el caso del porcentaje de población esperada “P”, es posible asignar un valor pesimista del 50%, por lo que el valor del complemento “Q” que es igual a 1-P será el otro 50% para completar el 100% de la población. El nivel de confianza de la muestra “Z” es la probabilidad de que la medida a estimar se encuentre en el intervalo de confianza, para este estudio se define un nivel de confianza del 90% y el valor nominal para aplicar en la fórmula es de 1.6449; finalmente el porcentaje de error admitido será del 10%. A continuación se presenta la aplicación de la fórmula, el resultado que se obtiene de la fórmula es aproximado a la unidad superior si el primer valor decimal es mayor que 5.

$$n = \frac{0.5 * 0.5 * 1.6449^2}{0.1^2} = 68 \quad (2)$$

El valor de la muestra “n” es igual a 68.

Aplicación de Criterios Propuestos

La siguiente fase del trabajo de investigación consiste en la aplicación de los criterios construidos por los investigadores y el equipo de expertos, a la muestra de proyectos, a continuación se presenta el resultado obtenido y el análisis para cada uno de los criterios.

C1. Cantidad de desarrolladores: Para este criterio se obtiene que la cantidad promedio de desarrolladores en los proyectos de la muestra es 5, manejando un margen de error del 30% se define que la cantidad de desarrolladores que debe tener un proyecto para cumplir con este criterio debe ser igual o mayor que tres (C1=>3). Al evaluar la muestra desde el punto de vista de la cantidad de desarrolladores, se descartan ocho (8) procesos de desarrollo que no cumplieron con el valor mínimo del criterio, es decir, que el 11,8% del total de la muestra salen de la investigación, quedando 60 para continuar con la aplicación del siguiente criterio.

C2. Diseños con UML: La aplicación de este criterio en la selección de procesos de desarrollo software, permite identificar el uso y no uso del estándar UML, concluyendo que los proyectos que superan este criterio son aquellos que si emplean UML en el proceso de diseño. Como resultado se obtiene que once (11) de los 60 procesos de desarrollo resultantes del criterio anterior no aplicaron diseños UML, es decir, que el 16.2% de los procesos de desarrollo de la muestra no continúan en la investigación y quedan 49 elementos en la muestra.

C3. Criticidad del Sistema: La criticidad del sistema se mide en niveles bajo, medio y alto. Los proyectos que superan la aplicación de este criterio deben cumplir con un nivel de criticidad igual o superior a medio. Sistemas de criticidad baja son descartados, ya que para el análisis de la investigación se consideran como sistemas con propósitos irrelevantes, un sistema que represente este nivel de criticidad no está apoyando un proceso que dependa del aplicativo. Si la aplicación colapsa, los usuarios pueden continuar la ejecución de dicho proceso de manera manual. Al aplicar este criterio se obtiene que cuatro (4) procesos de desarrollo no cumplen con el valor del criterio, continúan en el análisis veinte (20) procesos de desarrollo con nivel de criticidad media y veinticinco (25) con alta.

C4. Costos: De acuerdo con la clasificación del tamaño del proyecto en términos monetarios, que se observa en la Tabla 1 de la sección Definición de Criterios, se determina que un proyecto debe estar clasificado como normal o grande para superar este criterio. El análisis obtiene como resultado que tres (3) procesos de desarrollo no cumplen con los valores establecidos para el criterio; la muestra queda conformada por veinticuatro (24) procesos de desarrollo de nivel normal y dieciocho (18) de nivel grande.

C5. Modelo de Requisitos: Se analiza el uso de un modelo de requisitos para la etapa de especificación de requerimientos del software. Los proyectos que no aplican un modelo se descartan durante la aplicación de este criterio, en total seis (6) procesos de desarrollo de los 42 que superaron los cuatro criterios anteriores, no aplicaron un modelo de requisitos, es decir, que la muestra queda conformada por 36 elementos.

C6. Modelo de Calidad: Para continuar en la muestra, un proceso de desarrollo debe utilizar un modelo de calidad para el proceso de desarrollo del software. Los 36 elementos de la muestra aplicaron un modelo de calidad, 5 adoptaron CMM, 8 CMMi y 23 el estándar ISO 9001:2000. La Tabla 2 presenta un resumen del resultado de la aplicación de los criterios a la muestra de 68 procesos de desarrollo.

Tabla 2: Resultados consolidados – Aplicación de Criterios.

Criterio	Cantidad	Porcentaje
C1	8	11,8%
C2	11	16,2%
C3	4	5,9%
C4	3	4,4%
C5	6	8,8%
C6	0	0,0%
Total Descartados	32	47,1%

Como se observa en la Tabla 2, el 47.1% de los procesos de desarrollo se descartan después de aplicar los criterios, es decir, que el 52.9% correspondiente a los 36 procesos de desarrollo restantes, superaron este proceso y se utilizan para continuar con el proceso de inspección del código fuente e identificar la utilización de patrones GOF. Los procesos de desarrollo se llevaron a cabo en los departamentos de Santander y Cundinamarca, el aplicativo software desarrollado en cada uno apoya procesos en todo el territorio colombiano. Ver Fig. 1.

ANÁLISIS DE PATRONES GOF

En esta sección se presenta el proceso de identificación de patrones de diseño GOF que se realiza en los 36 procesos de desarrollo que conforman la muestra.

Identificación de Patrones GOF en Procesos de Desarrollo

En la construcción de aplicaciones Web, los problemas a los que diseñadores y desarrolladores se enfrentan difieren de los que se presentan en otro tipo de procesos de desarrollo. El catálogo de referencia definido por GoF (Gamma et al, 1994) no muestra una tipificación que permita identificar qué patrón puede emplearse específicamente para el desarrollo de aplicaciones Web. En el análisis realizado a procesos de desarrollo se identifican los patrones de diseño que son empleados para el desarrollo de aplicaciones Web,

aunque son 23 patrones de diseño los que se presentan en el catálogo GoF, no todos pueden ser usados para el desarrollo de aplicaciones orientadas a la Web.



Fig. 1: Ubicación geográfica procesos de desarrollo

El proceso de verificación del código fuente de los proyectos se lleva a cabo con el apoyo de un equipo de trabajo conformado por los investigadores del proyecto y profesionales en el área de la ingeniería de sistemas con amplia experiencia en desarrollo de software orientado a la Web, con formación de especialización, maestría y doctorado. La verificación del uso de patrones se realizó de forma manual a partir de la inspección del código fuente de las aplicaciones software construidas, los expertos que participan en esta actividad tienen experiencia en la implementación de patrones GOF y conocen su estructura básica, por lo tanto poseen la destreza para identificarlos. Los resultados de la verificación se organizan de acuerdo a la clasificación propuesta por el GOF. La Tabla 3 presenta el resultado de la revisión de los procesos de desarrollo para identificar patrones de diseño Creacionales.

Tabla 3: Patrones de diseño creacionales identificados en proyectos de desarrollo.

Patrón	Cantidad de procesos de desarrollo que lo utilizaron	Porcentaje sobre el total de procesos de desarrollo (36)
Abstract Factory	0	0%
Builder	15	42%
Factory Method	18	50%
Prototype	0	0%
Singleton	24	67%

Como se observa en la tabla los patrones identificados son: Builder, Factory Method y Singleton, esto equivale al uso del 60% de los patrones posibles en esta categoría, el patrón creacional más utilizado es Singleton, se identificó en el 67% de los procesos de desarrollo analizados. Ahora bien, el estudio permitió hacer un análisis de la cantidad de patrones identificados en los procesos de desarrollo. La tabla 4 presenta los resultados obtenidos.

Tabla 4: Cantidad de Patrones Creacionales identificados.

Cantidad de patrones utilizados	Porcentaje en la categoría	Cantidad de procesos de desarrollo que utilizaron patrones	Porcentaje sobre el total de procesos de desarrollo
0	0%	12	33%
1	20%	6	17%
2	40%	3	8%
3	60%	15	42%
Total		36	100%

Los datos de la tabla anterior permiten inferir que en 12 de los procesos de desarrollo no se identifican patrones GOF, sin embargo, en 24 procesos de desarrollo se identifican patrones GOF creacionales, es decir, en el 67% de la muestra se identifica el uso de los patrones de esta categoría; en el 42% de la muestra se identifica el uso del 60% de los patrones de tipo creacional.

La Tabla 5 presenta el resultado de la revisión de los procesos de desarrollo para identificar patrones de diseño Estructurales.

Tabla 5: Patrones de diseño estructurales identificados en proyectos de desarrollo.

Patrón	Cantidad de procesos de desarrollo que lo utilizaron	Porcentaje sobre el total de procesos de desarrollo (36)
Adapter	0	0%
Bridge	0	0%
Composite	0	0%
Decorator	14	39%
Facade	13	36%
Flyweight	0	0%
Proxy	0	0%

Como se observa en la tabla los patrones identificados son: Decorator y Facade, esto equivale al uso del 29% de los patrones posibles en esta categoría, el patrón estructural más utilizado es Decorator, se identificó en el 39% de los procesos de desarrollo analizados. La tabla 6 presenta los resultados obtenidos de la cantidad de patrones identificados en los procesos de desarrollo.

Tabla 6: Cantidad de Patrones Estructurales identificados.

Cantidad de patrones utilizados	Porcentaje en la categoría	Cantidad de procesos de desarrollo que utilizaron patrones	Porcentaje sobre el total de procesos de desarrollo
0	0%	18	50%
1	14%	9	25%
2	29%	9	25%
Total		36	100%

Los datos de la tabla anterior permiten inferir que en 18 de los procesos de desarrollo no se identifican patrones estructurales GOF, sin embargo, en 9 procesos de desarrollo se identifica el uso de 1 patrón y en 9 el uso de 2 patrones, es decir, que en el 50% de la muestra se identifican patrones estructurales.

Finalmente la Tabla 7 presenta los patrones de diseño de comportamiento identificados en los procesos de desarrollo analizados. Como se observa en la tabla 7 los patrones identificados son: Iterator, Strategy y Template Method, esto equivale al uso del 27% de los patrones posibles en esta categoría, el patrón de comportamiento más utilizado es Iterator, se identificó en el 56% de los procesos de desarrollo analizados.

La tabla 8 presenta los resultados obtenidos de la cantidad de patrones identificados en los procesos de desarrollo.

Tabla 7: Patrones de diseño de comportamiento identificados en proyectos de desarrollo.

Patrón	Cantidad de procesos de desarrollo que lo utilizaron	Porcentaje sobre el total de procesos de desarrollo (36)
Chain of Responsibility	0	0%
Command	0	0%
Interpreter	0	0%
Iterator	20	56%
Mediator	0	0%
Memento	0	0%
Observer	0	0%
State	0	0%
Strategy	15	42%
Template Method	17	47%
Visitor	0	0%

Tabla 8: Cantidad de Patrones de Comportamiento identificados.

Cantidad de patrones utilizados	Porcentaje en la categoría	Cantidad de procesos de desarrollo que utilizaron patrones	Porcentaje sobre el total de procesos de desarrollo
0	0%	12	33%
1	9%	7	19%
2	18%	6	17%
3	27%	11	31%
Total		36	100%

Los datos de la tabla anterior permiten inferir que en 12 de los procesos de desarrollo no se identifican patrones estructurales GOF, sin embargo, en 24 procesos de desarrollo se identifica el uso de patrones estructurales, es decir que en el 67% de la muestra se identifican patrones de esta categoría.

Análisis de resultados de patrones identificados en proyectos software

La revisión de los procesos de desarrollo que cumplieron con los criterios propuestos permite obtener los siguientes resultados. Ver Tabla 9.

Tabla 9: Resultados Generales.

Año	Cantidad de patrones utilizados	Porcentaje
Creacionales	3	60%
Estructurales	2	29%
Comportamiento	3	27%

El análisis realizado permite identificar que en procesos de desarrollo llevados a cabo entre 2002 y 2003 no utilizaron patrones, sin embargo, en procesos de desarrollo realizados entre 2004 y 2010 se identifica la aplicación de entre 1 y 3 patrones GOF, este resultado permite inferir que de acuerdo a la antigüedad de los proyectos, entre más antiguo fue el proyecto, menos patrones se encontraron. Entre las razones factibles propuestas por el equipo que trabajó en la verificación del código fuente de los proyectos se encuentran; desconocimiento de la temática por parte de todo el equipo, ausencia de un modelo de calidad que sugiriera el uso de una estrategia para desarrollar software y el uso de muchos conceptos que eran implícitos, pero algunos de ellos no son comprendidos. De acuerdo a los resultados de la Tabla 9, se puede inferir que los patrones más utilizados son los de tipo Creacional, los que menos se usan son los de Comportamiento y que los patrones de tipo estructural son poco usados, aunque son los que permiten organizar de mejor forma el producto software. En esta categoría sólo los patrones Decorador y Facade fueron utilizados. Como

resultado general de la identificación de patrones de diseño en los procesos de desarrollo de la muestra, se evidencia la aplicación de ocho (8) patrones, lo que equivale al **35%** de los veintitrés (23) patrones de diseño GOF. La Tabla 10 presenta los patrones identificados en los proyectos software analizados.

Tabla 10: Patrones GOF empleados en proyectos software para plataformas web

Categoría GOF	Patrón de Diseño
Creacionales	1. Builder 2. Factory Method 3. Singleton
Estructurales	1. Decorator 2. Facade
De Comportamiento	1. Iterator 2. Strategy 3. Template Method

Juicio de Expertos

Adicionalmente a la identificación de patrones de diseño GOF en procesos de desarrollo de software orientado a las Web, se realiza un análisis teniendo en cuenta la tendencia que se presenta en la comunidad de expertos, que se desempeñan profesionalmente en procesos de desarrollo software, este análisis permite identificar los patrones de diseño GOF que son aplicados en la praxis.

Este análisis es importante teniendo en cuenta que las personas involucradas en un equipo de desarrollo, conocen de las ventajas que ofrecen las técnicas de reutilización basada en el uso de patrones, puesto que han adquirido habilidad para la implementación de los mismos a partir de la experiencia y la necesidad de solucionar problemas a partir de la aplicación de patrones de diseño. Por ende, el juicio de profesionales expertos en el área de la ingeniería del software con experiencia en procesos de desarrollo para la Web, es importante para conocer cuáles de los patrones de diseño GoF son utilizados actualmente. Para el desarrollo de esta fase se realizó un sondeo con la finalidad de conocer los patrones más usados por el equipo de expertos; el equipo fue conformado por expertos con titulación de especialización, maestría y doctorado. A continuación se presentan los resultados obtenidos agrupados de acuerdo a la clasificación GOF. La Tabla 11 presenta los resultados obtenidos en la categoría de patrones creacionales.

Tabla 11: Patrones de diseño creacionales utilizados por expertos

Id.	Expertos				
	Patrones	Exp_01	Exp_02	Exp_03	Exp_04
1	Abstract Factory*	Sí	No	Sí	No
2	Builder*	No	Sí	No	No
3	Factory Method*	Sí	Sí	Sí	Sí
4	Prototype	No	No	No	No
5	Singleton*	Sí	Sí	Sí	Sí

De los cinco (5) patrones Creacionales, cuatro (4) fueron usados por los expertos consultados (*patrones identificados con **), lo que equivale a un 80% de los patrones de esta categoría. La Tabla 12 presenta el resultado de la revisión de los patrones de diseño Estructurales.

Tabla 12: Patrones de diseño estructurales utilizados por expertos

Id.	Expertos				
	Proyectos	Exp_01	Exp_02	Exp_03	Exp_04
1	Adapter	No	No	No	No
2	Bridge	No	No	No	No
3	Composite	No	No	No	No
4	Decorator*	No	No	Sí	Sí
5	Facade*	Sí	Sí	No	Sí
6	Flyweight	No	No	No	No
7	Proxy	No	No	No	No

De los siete (7) patrones estructurales, dos (2) fueron usados por los expertos consultados, lo que equivale a un 29% de los patrones de esta categoría. Finalmente la Tabla 13 presenta los patrones de diseño de comportamiento que son utilizados por expertos.

Tabla 13: Patrones de diseño de comportamiento utilizados por expertos.

Id.	Expertos				
	Patrones	Exp_01	Exp_02	Exp_03	Exp_04
1	Chain of Responsibility	No	No	No	No
2	Command	No	No	No	No
3	Interpreter	No	No	No	No
4	Iterator*	Sí	No	No	No
5	Mediator	No	No	No	No
6	Memento	No	No	No	No
7	Observer*	No	Sí	No	No
8	State	No	No	No	No
9	Strategy*	No	No	No	Sí
10	Template Method*	Sí	No	No	No
11	Visitor	No	No	No	No

De los 11 patrones de comportamiento, 4 fueron usados por los expertos consultados, lo que equivale a un 36% de los patrones de esta categoría.

Análisis de resultados de patrones utilizados por expertos en desarrollo software

Al realizar el análisis de los datos obtenidos del sondeo se concluye que: i) De acuerdo a la clasificación de patrones según GoF, los patrones más usados por el grupo de expertos, son los Creacionales, seguido por los patrones de comportamiento, los menos utilizados son los patrones estructurales; ii) De acuerdo al grupo de expertos, el 100% del grupo utilizan el patrón Factory Method y Singleton; el patrón Facade fue empleado por el 75% del grupo; iii) De acuerdo a la totalidad de los patrones, se encontró evidencia del uso de 10 patrones del catálogo GoF, lo cual equivale al 43% del total de patrones del catálogo. En la Tabla 13 se presentan los patrones utilizados por los expertos que apoyaron el proceso.

Adicional al resultado presentado anteriormente, los expertos expresan un conjunto de apreciaciones de las cuales sobresalen las siguientes:

-Varios patrones de diseño del catálogo GoF pueden emplearse para resolver un mismo problema a pesar que en la definición de su intención describan un objetivo diferente.

-El uso de patrones de diseño es una práctica que se adquiere con la experiencia y participación activa en proyectos de desarrollo software y del constante estudio en la aplicación de técnicas de reutilización basada en el uso de patrones de diseño.

-Un Marco de Trabajo usa en su código base patrones de diseño (Guerrero & Recaman, 2009). Por lo tanto, cuando un desarrollador reutiliza componentes del Marco de Trabajo, aplica inconscientemente los patrones, mejorando el nivel de desempeño de los desarrolladores cuando observan la manera como los Marcos de Trabajo presentan soluciones mediante los patrones GoF.

Tabla 14: Patrones GOF empleados por expertos

Categoría GOF	Patrón de Diseño
Creacionales	1. Abstract Factory 2. Builder 3. Factory Method 4. Singleton
Estructurales	1. Decorator 2. Facade
De Comportamiento	1. Iterator 2. Observer 3. Strategy 4. Template Method

RESULTADOS

Después de realizar las actividades para identificar los patrones de diseño que se emplean en el desarrollo de aplicaciones Web, se obtienen los siguientes resultados: Se identificaron 8 patrones de diseño GoF en los procesos de desarrollo software que superaron los criterios de selección propuestos en esta investigación, en contraste con el sondeo realizado a expertos es posible inferir que son 10 los patrones de diseño GOF que actualmente se utilizan en la práctica por los desarrolladores.

El sondeo realizado a profesionales expertos representa la referencia para conocer los patrones que actualmente son usados en la práctica. Sin embargo, representan un juicio subjetivo. Al realizar el análisis de los procesos de desarrollo de la muestra, el resultado obtenido coincide en un 80% sobre la lista de los patrones usados por expertos. Los patrones *Abstract Factory* y *Observer*, son patrones que están embebidos en un Marco de Trabajo y no son patrones empleados conscientemente por un desarrollador en un proceso de desarrollo, el alcance de esta investigación se limitó al estudio de Procesos de Desarrollo y no a Marcos de Trabajo. Estos dos patrones representan el 20% de los patrones que utilizan los expertos, pero que no son aplicados en los procesos de desarrollo analizados.

A partir de su experiencia directa con Marcos de Trabajo, los expertos argumentan que los patrones son una herramienta para desarrollar aplicativos software, manifiestan que usan los patrones porque hacen parte del Marco de Trabajo que ellos emplean. El proceso de identificación de patrones de diseño GOF arroja como resultado que los patrones GOF que se utilizan en procesos formales de desarrollo en Colombia son: ver Tabla 15.

Tabla 15: Patrones GOF empleados por expertos.

Nro.	Patrón de Diseño	Categoría
1	Builder	Creacionales
2	Factory Method	
3	Singleton	
4	Decorator	Estructurales
5	Facade	
6	Iterator	De Comportamiento
7	Strategy	
8	Template Method	

CONCLUSIONES

El proceso de investigación del el Catálogo GOF, logra demostrar que la publicación no establece una categorización de cuáles de los 23 Patrones de Diseño son exclusivamente para el Desarrollo de Aplicaciones Orientadas a la Web, y no se evidencia por parte de otro autor una identificación de este tipo.

Durante la etapa de inspección para conocer en detalle los Patrones GOF, se encontró gran dificultad debido al alto nivel de abstracción que presenta este catálogo; argumento corroborado por los expertos que colaboraron en el desarrollo de esta investigación. Los expertos son profesionales con más de 5 años de experiencia de trabajo en procesos de desarrollo de aplicativos software de gran envergadura y han usado patrones de Diseño GOF, razón por la que afirman: *"Usar un patrón de Diseño requiere de años de experiencia, a veces se puede leer y leer sobre un Patrón y no entenderlo y otras veces aplicarlo mal"*.

Otro aspecto importante hace referencia a la estructura del Patrón en el Catálogo GOF, esta estructura define el modelo de clases para el Patrón, pero en el apartado de Implementación el ejemplo práctico no se desarrolla a partir de este modelo y el catálogo no propone el modelo para el ejemplo que implementan, por consiguiente se hace confuso entenderlo, un trabajo futuro se puede orientar a la construcción de un catálogo con los patrones de diseño GOF identificados en esta investigación, en el que se explique didácticamente su implementación.

Finalmente, otro aspecto que se considera importante mencionar, es el caso de los diseñadores y desarrolladores que en su afán de emplear "Buenas Prácticas", creen que los Patrones de Diseño son la panacea para solucionar cualquier problema a la hora de desarrollar software, este es un gran error, puesto que no todas las situaciones son candidatas para el uso de patrones de diseño. Se debe tener claridad sobre la utilidad de los patrones, es posible que el problema se haga más grande de lo que realmente es en el momento de usar un patrón de diseño, donde realmente no es necesario.

AGRADECIMIENTOS

El desarrollo de esta investigación se llevó a cabo gracias al trabajo colaborativo entre los integrantes del grupo de investigación en ingeniería de software GRIIS con apoyo de las Unidades Tecnológicas de Santander. Se extiende un agradecimiento al equipo de expertos que proporcionaron sus valiosos aportes.

REFERENCIAS

Braude, E. *Ingeniería del Software una Perspectiva Orientada a Objetos*, 1ª edición, 120-425. RA-MA EDITORIAL. Madrid, España (2003).

Cristian L. Vidal, Rodolfo F Schmal, Sabino Rivero y Rodolfo H. Villarroel. *Extensión del Diagrama de Secuencias UML (Lenguaje de Modelado Unificado) para el Modelado Orientado a Aspectos*. Revista Información Tecnológica. ISSN: 0718-0764. [En línea]. Vol. 23(6), (2012). <http://www.scielo.cl/pdf/infotec/v23n6/art07.pdf>. Acceso: Octubre 2012.

Engineering, S., & Committee, S. *IEEE Recommended Practice for Software Requirements Specifications*. Institute of Electrical and Electronics Engineers, 1-31 (1998).

Fuensanta, M. D. *Marco Metodológico para la Mejora de la Eficiencia de Uso de los Procesos Software*. Tesis de Doctorado, Universidad Carlos III de Madrid, Madrid, España (2010).

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. *Design Patterns Elements of Reusable Object Oriented Software*. 1-358. Addison Wesley Longman Inc. USA (1998).

Gracia, J. *Patrones de Diseño. Ingeniero Software, Análisis y Diseño* [en línea] (2005), <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>. Acceso: Agosto 2 de 2010.

Guerrero, A., & Suárez, J. *Patrones de diseño para el desarrollo de aplicaciones Web - Construyendo software con buenas prácticas*. (Sic) Editorial Ltda. Colombia (2010).

Guerrero, C. A., & Recaman, H. *Marcos de Trabajo (Framework) para soportar el desarrollo de aplicaciones Web de código abierto*. 4-25. (Sic) Editorial Ltda. Colombia (2009).

Gutiérrez, L. *Arquitectura Software - Investigación Aplicada a la Construcción de Marcos de Trabajo*. 10-35. (Sic) Editorial Ltda. Colombia (2010).

ISO 9001:2000, Norma Internacional ISO 9001:2000, 1-34, Ginebra, Suiza (2000).

ISO/IEC 9126, Software Engineering - Product Quality, 1-159, Ginebra, Suiza (2001).

Pressman, R. S. *Ingeniería del software. Un Enfoque Práctico*. 6ª edición, 57-225. McGraw-Hill. México (2005).

Schach, S. *Análisis y diseño orientado a objetos con UML y el proceso unificado*. 35-96. McGraw-Hill (2010).

Snedecor, G., & Cochran, W. *Métodos Estadísticos*. 1ª edición, 158-215. Continental. México. (1984).

Toro, A., & Jiménez, B. *Metodología para la Elicitación de Requisitos de Sistemas Software Versión 2.1*. Informe Técnico LSI-2000-10. Facultad de Informática y Estadística. Sevilla, España (2000).

Welicki, L. *Patrones y Antipatrones: una Introducción-Parte II*. Biblioteca MSDN [en línea] (2006), <http://msdn.microsoft.com/es-es/library/bb972251.aspx>. Acceso: Agosto 25 de 2010.