



# Análisis del impacto del uso de patrones de diseño en la fase de mantenimiento

## Impact Analysis of Using Design Patterns in the Maintenance Phase

Martha de León Correa<sup>1</sup> Ivón Hoyos Beltrán<sup>2</sup> Fabio Quintero Díaz-Granados<sup>3</sup>

**Para citar este artículo:** De León, M.; Hoyos, I.; Quintero, F. (2017). Análisis del impacto del uso de patrones de diseño en la fase de mantenimiento. *TIA*, 5(2), pp. 250-256.

### ARTÍCULO DE REFLEXIÓN

**Fecha de recepción:**  
07-06-2015

**Fecha de aceptación:**  
03-06-2017

ISSN: 2344-8288

Vol. 5 No. 2

Julio - diciembre 2017

Bogotá-Colombia

### Resumen

El proyecto de investigación realizado es un modelo conceptual para determinar el impacto del uso de patrones de diseño en la fase de mantenimiento en proyectos de *software* donde fueron aplicados; en particular, el estudio se dirigió a evaluar mediante una encuesta el esfuerzo y la eficiencia para realizar operaciones de mantenimiento. Como trabajo posterior, se propone la elaboración de un *software* que evalúe las buenas prácticas en el desarrollo de un proyecto de *software*, así, nace la propuesta de un modelo arquitectónico basado en Archimate.

**Palabras clave:** Archimate, DP, GoF, mantenimiento correctivo, mantenimiento, patrones de diseño de *software*, programadores, proyecto de *software*, sistema de *software*, *software*.

<sup>1</sup> Ingeniera de Sistemas; Especialista en Ingeniería de Software, Universidad Distrital Francisco José de Caldas. Ingeniera de Operaciones Senior, Sistemas y Asesorías de Colombia S.A. Correo electrónico: marthadl27@gmail.com

<sup>2</sup> Ingeniera de Sistemas; Especialista en Ingeniería de Software, Universidad Distrital Francisco José de Caldas. Ingeniera Programadora, Winner Group S.A. Correo electrónico: irosho@hotmail.com

<sup>3</sup> Ingeniero de Sistemas, Escuela Colombiana de Ingeniería Julio Garavito. Ingeniero de innovación y desarrollo, Politécnico Grancolombiano. Correo electrónico: neaf1988@gmail.com

## INTRODUCCIÓN

Se necesita mantenimiento para asegurar que un sistema de *software* continúa satisfaciendo los requerimientos de los usuarios; esta fase es importante y aplicable a los desarrollos de *software* usando cualquier modelo de ciclo de vida del *software* o lenguaje de programación. Swanson [4] identifica tres tipos de cambios que se pueden realizar para permitir que un sistema de *software* pueda evolucionar durante el tiempo: (a) correctiva, (b) perfectivo y (c) adaptativo. Así, una operación de mantenimiento correctivo es una modificación reactiva para corregir los problemas descubiertos, mientras que una operación de mantenimiento perfectivo es la necesidad de mejorar el rendimiento o la capacidad de mantenimiento de un sistema de *software* existente, por otro lado, el proceso de adaptación es mantener utilizable para los usuarios un producto de *software* en un ambiente cambiante.

Para realizar mantenimiento de *software*, los programadores pasan mucho tiempo leyendo y comprendiendo el código fuente, este tiempo debe disminuir cuando las opciones de diseño están disponibles; además, la disponibilidad de opciones de diseño debe apoyar mejor a los programadores para lograr operaciones de mantenimiento eficientes [1].

En cuanto a los sistemas de *software* orientados a objetos, las opciones de diseño son a menudo implementadas con patrones recurrentes que son conocidos en la literatura como patrones de diseño. Gamma *et al.* [1] afirman que “design patterns can even improve the documentation and maintenance of existing systems by furnishing an explicit specification of class and object interactions and their underlying intent”. A pesar de esto, son reconocidos solo algunos de los estudios empíricos que evalúan la eficacia y la eficiencia de los patrones de diseño en el mantenimiento del código fuente [2], [3].

El alcance del proyecto de investigación realizado por estudiantes de la especialización en Ingeniería de *Software* de la Universidad Distrital

Francisco José de Caldas, tiene como alcance realizar una estadística del estado actual del uso de los veintitrés patrones de diseño para solucionar problemas comunes en el área de la ingeniería de *software* orientado a objetos, conocido como banda de los cuatro (GoF) [1], en una muestra de empresas de la industria de producción de *software* en Bogotá D.C., con el fin de cuantificar su usabilidad y definir los más comunes en los procesos de desarrollo en las empresas de la muestra seleccionada.

En este artículo, se presenta el trabajo realizado al evaluar los patrones de diseño utilizados en la ejecución de un mantenimiento correctivo y tareas de mantenimiento perfectivo. El resultado se obtendrá de un trabajo de campo tomado una muestra de varias empresas desarrolladoras de *software* de Bogotá D.C., para evaluar de ellas procesos de desarrollo formales de gran envergadura donde, a través de una técnica de recolección de información acerca de la usabilidad de patrones de diseño, se pretende dar una aproximación de los pros y contras al utilizarlos, enfocados en la facilidad de la evolución de los sistemas de *software* en los que se encontró aplicados.

El contenido de este artículo se divide en cuatro secciones: en la primera sección se describe la hipótesis y formulación, la segunda se define la metodología utilizada para la recolección de información, en la tercera se describe la estrategia para la consecución de la muestra de tamaño significativo y en la última sección se presentan los resultados obtenidos.

## HIPÓTESIS Y FORMULACIÓN

### Hn0

El uso de patrones de diseño de GoF en el ciclo de vida del *software* no reduce significativamente los esfuerzos para realizar operaciones validación y evolución de la aplicación.

## Hn1

El uso de patrones de diseño de GoF en las fases del ciclo de vida especificación y desarrollo del *software*, no aumenta significativamente la eficiencia en el proceso de mantenimiento del sistema.

## Objetivo

El objetivo del análisis estadístico es rechazar la hipótesis nula definida y, posiblemente, aceptar las alternativas que se puede derivar de esta investigación.

## Sujeto de control

El objeto por controlar con esta investigación es el código fuente con o sin utilización de patrones, y como sujeto tratado la eficiencia en evolución o mantenimiento de este código fuente (i.e., principal factor), se seleccionarán dos variables dependientes (DP): esfuerzo y eficiencia.

## ASPECTOS METODOLÓGICOS

### Tipo de estudio

El nivel de profundidad que aborda este proyecto es el tipo de exploratorio, ya que permite familiarizarse con la teoría de patrones de diseño de GoF y sirve como punto de partida para la formulación de otras investigaciones con mayor nivel de profundidad que deseen realizarse a futuro.

El nivel en la investigación holística es la perceptual y el holotipo de investigación es exploratorio, ya que se da una aproximación y percibe los aspectos más importantes.

## Método de investigación

En busca de la respuesta a la pregunta planteada en la formulación del problema, es necesario establecer un procedimiento riguroso de manera lógica; por lo anterior, se define como método de investigación el método deductivo, donde, partiendo de lo general, se estudia la teoría de los patrones de diseño de GoF y, posteriormente, serán aplicadas como referencia en la estadística de usabilidad de los patrones en el mundo real en cuanto a la eficiencia en la evolución de los programas durante la fase de mantenimiento.

## Fuentes y técnicas para recolección de la información

Las fuentes a los que se acudió para llevar a cabo este proyecto son presentadas a continuación.

### Documentación

Revisión de la literatura existente de patrones de diseño de GoF [4].

### Encuesta

La recolección de la información se realizó a partir de la aplicación de una encuesta con los formatos de pregunta presentados a continuación

- Escalas de calificación: donde el encuestado seleccionó una única calificación para la pregunta en un continuo espaciado equivalente de posibles opciones.
- Codificación por colores cristal: para la caracterización de los proyectos según tamaño del equipo de trabajo.
- Preguntas de opción múltiple: este tipo de pregunta permitió a las personas que realizaron la encuesta seleccionar una o más opciones de una lista de respuestas definida.

La encuesta contiene veintidós preguntas divididas en el ciclo de vida del proyecto de *software*, estas están divididas de la siguiente forma:

- Especificación: diez preguntas en total divididas en nueve preguntas de escala de calificación y una pregunta basada en la codificación por colores cristal.
- Desarrollo: seis preguntas de escalas de calificación.
- Mantenimiento: siete preguntas en total divididas en tres preguntas de escala de calificación, dos preguntas de opción múltiple, múltiple respuesta y dos preguntas de opción múltiple, única respuesta.
- La distribución de la encuesta se hizo en dos frentes, a saber: papel impreso, de las cuales se distribuyeron 150 paquetes de copias y encuesta en línea.

## DEFINICIÓN DE LA MUESTRA

Considerando que el tamaño exacto de la población objeto de estudio es desconocido por el gran número de proyectos de desarrollo que pueden ser analizados en Bogotá D.C., es conveniente aplicar una metodología para calcular el tamaño de la muestra; así, se selecciona la metodología denominada: tamaño de la población infinito o desconocido (Snedecor y Cochran, 1984) [15], pues permite establecer el tamaño de la muestra para una población infinita o desconocida como en este caso.

A continuación, se muestra el cálculo para el tamaño de la muestra para una población infinita que se tuvo en cuenta para este proyecto.

Ecuación para calcular muestra de población infinita donde:

$n$ : tamaño muestral.

$Z$ : valor correspondiente a la distribución de Gauss,  $Z = 0.05$  es equivalente a 1.96.

$p$ : prevalencia esperada del parámetro a evaluar, en caso de desconocerse ( $p = 0.5$ ), que hace mayor el tamaño muestral.

$q$ :  $1-p$ .

$i$ : error que se prevé cometer, 10%.

$$n = \frac{(Z_{\alpha})^2 * p * q}{i^2}$$

$$Z_{\alpha} = 0,05 = 1,96$$

$$p = 0,5$$

$$q = 1 - p = 1 - 0,5 = 0,5$$

$$i = 10\% = 0,1$$

$$n = \frac{(1,96)^2 * 0,5 * 0,5}{0,1^2}$$

$$n = \frac{3,6864 * 0,25}{0,01} = \frac{0,9216}{0,01} = 92,16$$

La muestra total debe ser al menos de 93 encuestados.

## PROCESAMIENTO DE LA INFORMACIÓN Y PRESENTACIÓN DE RESULTADOS

### Ordenamiento

Para el ordenamiento de la información, se elaboró una tabla de frecuencias con los resultados de las preguntas de la encuesta para determinar.

- Frecuencia.
- Media promedio de todos los valores.
- Medidas de dispersión.
- Varianza mide la dispersión de los valores respecto a la media.
- Desviación estándar.
- Valoración final.

## Procesamiento

El procesamiento de la información se hizo en dos fases que se comentan a continuación.

- Mediante la técnica de frecuencia absoluta: se obtuvo el número de veces que fue seleccionada una opción de respuesta de todas las preguntas de la encuesta.

- Mediante la estadística descriptiva: como herramienta para el estudio y análisis de los datos obtenidos en la encuesta.

## Presentación de resultados

### *Tabla de frecuencias totales y diagrama de barras*

La Tabla 1 muestra la tabulación completa de la encuesta, esto mediante un cuadro de frecuencias en la que se exponen las opciones de respuesta (escala de uno a cinco) y el total contado por cada pregunta; también, se grafica a partir de un diagrama de barras, el cual permite ver la variación de las respuestas diligenciadas en los 100 proyectos.

**Tabla 1.** Tabla de frecuencias

Preg. Núm.	Opción 1	Opción 2	Opción 3	Opción 4	Opción 5
1	26	12	24	27	11
2	6	7	25	36	26
3	7	12	26	29	26
4	12	14	26	36	12
5	6	20	29	29	16
6	9	30	18	25	18
7	12	14	26	25	23
8	10	19	30	32	9
9	9	12	20	42	17
10	6	9	35	35	15
11	13	26	30	21	10
12	11	13	26	23	14
13	15	14	25	36	10
14	10	16	28	35	10
15	25	39	21	9	6
16	13	22	31	24	10
17	5	16	33	33	7
18					
19					
20	31	30	30	9	
21	21	39	29	11	

**Fuente:** elaboración propia.

**Tabla de frecuencias de respuesta**

Este diagrama muestra la frecuencia de respuestas obtenidas en la encuesta para cada pregunta, se ilustran adicionalmente las líneas de desviación estándar y se resaltó en color azul las respuestas con mayor frecuencia (Figura 1).

**ANÁLISIS DE RESULTADOS**

**Clasificación de patrones según su utilización**

De acuerdo con los resultados de los patrones utilizados según su clasificación, se puede inferir que los patrones más utilizados son los de tipo creacional, con un 78%, los que menos se usan son los de comportamiento con un 56%, y que los patrones de tipo estructural son usados con un 66% (Tabla 2).

**Variables de acuerdo con las fases de especificación, desarrollo y mantenimiento del ciclo de vida del software**

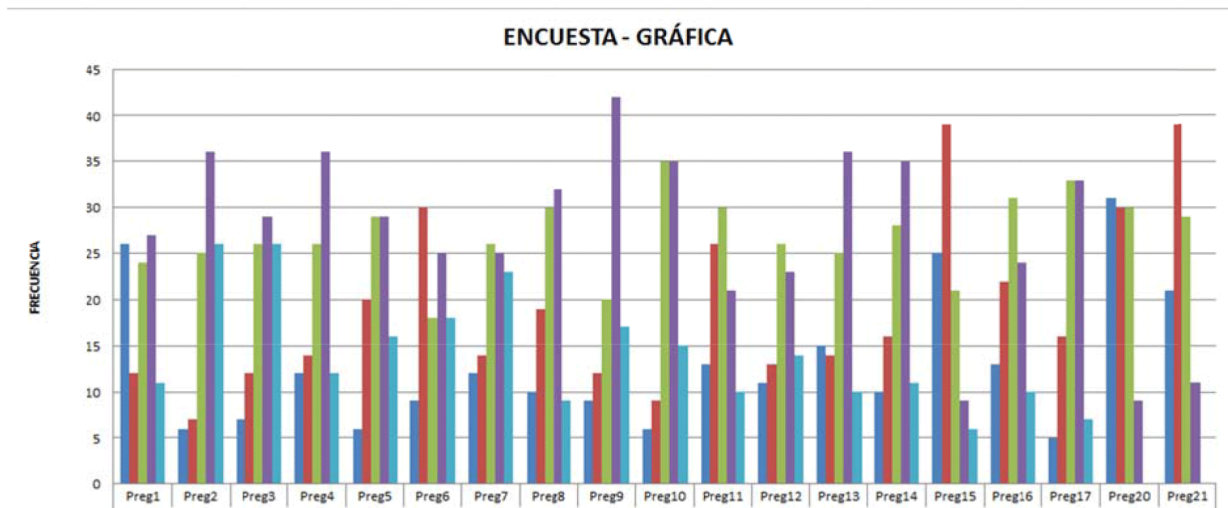
**Fase especificación**

De acuerdo con los resultados de la variable fase especificación, se deduce que un 73% de los encuestados tiene en cuenta desde la fase de especificación de un proyecto los patrones de diseño, el cual representa un total de 46% sobre todo el ciclo de vida de desarrollo de proyectos de software.

**Fase desarrollo**

De acuerdo con los resultados de la variable fase desarrollo, se deduce que un 71% de los encuestados tiene en cuenta desde la fase de desarrollo de un proyecto los patrones de diseño, el cual representa un total de 21,2% sobre todo el ciclo de vida de desarrollo de proyectos de software.

Análisis del impacto del uso de patrones de diseño en la fase de mantenimiento  
De León, M.; Hoyos, I.; Quintero, F.



**Figura 1.** Frecuencias de respuesta

Fuente: elaboración propia.



**Tabla 2.** Clasificación de patrones de uso

Tipo de patrón	Cantidad de patrones utilizados	Porcentaje total de encuestados
Creacionales	78	78%
Estructurales	66	66%
Comportamiento	56	56%

**Fuente:** elaboración propia.

### Fase mantenimiento

De acuerdo con los resultados de la variable fase mantenimiento, se deduce que un 65% de los encuestados tiene en cuenta desde la fase de mantenimiento de un proyecto los patrones de diseño, el cual representa un total de 31,92% sobre todo el ciclo de vida de desarrollo de proyectos de *software*.

## CONCLUSIONES

El proceso de investigación del catálogo GoF, logra demostrar que de los veintitrés patrones de diseño el más utilizado es Singleton, con un 54%, y los menos utilizados son los patrones Memento y Visitor, cada con uno con un 3% del total de los proyectos encuestados; asimismo, el 14% de los encuestados no usan ni contemplan en su proyecto patrones de diseño, además, un 86% utilizan al menos un patrón.

De esta investigación se concluye que el uso de patrones desde las fases iniciales reduce los esfuerzos para realizar operaciones de validación y evolución de la aplicación, y que el desarrollo de un sistema de *software* con patrones de diseño asegura un fácil mantenimiento y evolución de *software* en comparación con un sistema que en su desarrollo no los haya aplicado.

## RECONOCIMIENTOS

Agradecimientos al ingeniero Sandro Bolaños por su dedicación y compromiso en la elaboración del proyecto.

## REFERENCIAS

- [1] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1 edition,
- [2] Jeanmart, S., Gueheneuc, Y., Sahraoui, H. and Habra, N. (2009). Impact of the Visitor Pattern On Program Comprehension and Maintenance. In *Procs. Of International Symposium on Empirical Software Engineering and Measurement*. IEEE CS Press, 69-78.
- [3] Prechelt, L., Unger-Lamprecht, B., Philippsen, M. and Tichy, W. (2002). Two Controlled Experiments Assessing the Usefulness of Design Pattern Documentation in Program Maintenance. *IEEE Trans. Software Eng.* 28(6), 595-606.
- [4] Romero O. (2014). *Notas de Clase Informática I*. Bogotá D.C.: Universidad Distrital Francisco José De Caldas.