

Evaluación de la Calidad de Patrones Arquitectónicos a través de Experimentos Cuantitativos[†]

Indira Hidalgo¹, Anna Grimán¹, Luis Mendoza¹, María Pérez¹

¹ Valle de Sartenejas, Universidad Simón Bolívar, Depto. de Procesos y Sistemas, LISI, Caracas 1080A- Venezuela

<http://www.lisi.usb.ve>

ihidalgo@cantv.net, {[agriman](mailto:agriman@usb.ve), [lmendoza](mailto:lmendoza@usb.ve), [movalles](mailto:movalles@usb.ve)}@usb.ve

Resumen. Para mejorar la calidad de las arquitecturas de software se usan patrones, que capturan existencia, experiencia comprobada en el desarrollo del software y ayudan a promover buenas prácticas de diseño [2]. Sin embargo, no está claro cómo realizar la evaluación del efecto de incorporar uno o más de estos patrones en la arquitectura del software. El objetivo de este artículo es presentar la evaluación de la calidad propiciada por los patrones arquitectónicos de [3] a través del uso de Experimentos Cuantitativos [6]. Para ello fue necesario aplicar un experimento cuantitativo sobre un software colaborativo: un chat. El experimento fue dividido en seis experimentos elementales, los cuales se fundamentaron en la aplicación de métricas arquitectónicas sobre la arquitectura del chat, integrada con las combinaciones de patrones arquitectónicos. Se evidenció que el uso de patrones mejora características que no habían sido consideradas en la hipótesis.

1 Introducción

Para apoyar el trabajo grupal se han desarrollado los sistemas colaborativos. Las herramientas de software desarrolladas para proveer soporte al trabajo grupal se definen como groupware. La calidad de los sistemas colaborativos dependerá en gran medida de la calidad del software que lo constituye [8]. Al respecto, los autores consultados afirman que la arquitectura de software propicia el comportamiento, desempeño y el nivel de satisfacción de los requerimientos del software, permitiéndole a las empresas lograr sus objetivos y expandir las capacidades de su software [1]. Para mejorar la calidad de las arquitecturas de software se usan patrones, los cuales capturan existencia, experiencia comprobada en el desarrollo del software y ayudan a promover buenas prácticas de diseño [2]. En particular, los patrones arquitectónicos expresan un principio de estructuración para organizar sistemas de *software* [3]. Sin embargo, no está claro cómo realizar la evaluación del efecto de incorporar uno o más de estos patrones en la arquitectura del software.

Dado que unas combinaciones iniciales de patrones arquitectónicos fueron propuestas por [3] y actualmente sólo se busca evaluar cada una de ellas más

[†]Esta investigación ha sido financiada por el proyecto FONACIT S1 -2001000794

detalladamente, para así determinar sus efectos en la calidad de los sistemas colaborativos, se entiende que la evaluación incluye proyectar alternativas iniciales seguidas de una evaluación más detallada. El mayor impacto de aplicar estos objetos de evaluación, (combinaciones de patrones arquitectónicos), sobre sistemas colaborativos es la mejora de la calidad a través de la aparición de características que son deseables en estas aplicaciones. Esto denota un carácter cualitativo del impacto, ya que se busca determinar la combinación de patrones que tenga el mayor nivel de apropiación a las necesidades de los usuarios de un sistema colaborativo; sin embargo, esta calidad no puede ser comparada de manera objetiva sin una estimación cuantitativa.

Por otra parte, el alcance del impacto tiene la granularidad del todo, ya que al aplicar una combinación de patrones arquitectónicos en el diseño de un sistema colaborativo se persigue impactar la calidad total del sistema, no de un módulo o parte de éste. Se considera que la aplicación de alguna combinación de patrones arquitectónicos tendrá efectos y éstos podrán *medirse* a partir de la fase de *diseño arquitectónico*.

Para Kitchenham [7], si los objetos que están siendo evaluados son métodos específicos, se pudiera usar un método de evaluación Cuantitativo o un Análisis de Características. Si se disponen de estándares para la comparación de los efectos resultantes de usar las combinaciones de patrones, no se tendría limitaciones para usar experimentos formales y estudio de casos cuantitativo. Además, un estudio cuantitativo es especialmente recomendable para proyectos de 3 ó más meses y puede durar lo que dura el desarrollo del proyecto [7]. Los experimentos usualmente necesitan tareas relativamente cortas y pequeñas que pueden ser ejecutadas en corto tiempo. Esto es útil cuando el *staff* no está disponible por largos lapsos y tareas largas son difíciles de controlar. También los experimentos cuantitativos presentan mu y bajo riesgo, debido a que el método científico soporta la confiabilidad de los experimentos formales. Por su parte, la probabilidad de resultados incorrectos es por si mismo cuantificable cuando se siguen procedimientos formales.

Kitchenham [7] afirma que los Experimentos Cuantitativos son recomendables como método de evaluación cuando: el contexto es independiente de las mediciones del método, los beneficios son claramente cuantificables, existe personal disponible para tomar parte en la experimentación, el objeto de evaluación está relacionado a una actividad sencilla o tarea, existen beneficios directamente mensurables a partir de la salida de la tarea y, se dispone de un tiempo corto de aprendizaje.

En base a lo anterior, el objetivo de este artículo es **presentar la evaluación de la calidad propiciada por los patrones arquitectónicos de [3] a través del uso de Experimentos** Cuantitativos. Para ello fue necesario aplicar un experimento cuantitativo sobre una aplicación colaborativa: un chat. El experimento fue dividido en seis experimentos elementales, los cuales se fundamentaron en la aplicación de métricas arquitectónicas sobre la arquitectura del chat integrada con las combinaciones de patrones arquitectónicos. Conociendo los resultados del experimento, el arquitecto de un sistema colaborativo podrá decidir cuál combinación de patrones arquitectónicos usar, todas las combinaciones son apropiadas para el diseño de un sistema colaborativo, ya que todas propician ciertas características que son deseables en este tipo de aplicaciones. De tal manera, que la selección de alguna combinación en particular dependerá de las prioridades en el aspecto de calidad que

puedan tener los stakeholders.

A continuación se presenta una definición de este tipo de experimentos: sus elementos y fases, para luego describir los aspectos particulares de la investigación y la ejecución de las fases para un caso de estudio. Por último, se discuten los resultados, conclusiones y recomendaciones derivados de la presente investigación.

2 Experimentos Cuantitativos

Según Juristo la experimentación se refiere a relacionar los hechos con las suposiciones, especulaciones y creencias de la Ingeniería de Software [6]. Se pueden identificar dos aproximaciones diferentes para la realización de investigaciones empíricas: cuantitativa y cualitativa. La investigación cuantitativa busca obtener relaciones numéricas entre diversas variables o alternativas bajo estudio. Otras investigaciones son dirigidas a examinar objetos en su configuración original más que buscar una relación cuantitativa o numérica, tratando de darle sentido, o interpretar un fenómeno en término de explicaciones que las personas le den. La tarea principal de la investigación cualitativa es explicar las maneras en que las personas entienden, toman acción y manejan las situaciones del día a día. Por ello, el rol de la investigación es obtener una visión general del contexto de estudio, su lógica, sus arreglos y reglas implícitas y explícitas [6].

La experimentación cuantitativa del presente trabajo de investigación busca determinar los efectos de usar patrones arquitectónicos en el diseño de un sistema colaborativo; en particular, para estos experimentos se usará una aplicación colaborativa (chat) el cual será mostrado más adelante.

2.1 Elementos de la Experimentación

Antes de definir las fases de la experimentación y aplicar cada una de ellas se definen ciertos elementos básicos de todo experimento tal como son planteados en [6]:

1. **Unidad Experimental:** Los objetos sobre los cuales el experimento es ejecutado son llamados unidades experimentales u objetos experimentales. Los experimentos en la Ingeniería del Software involucran sujetar el desarrollo de un proyecto o una parte particular del proceso de desarrollo a ciertas condiciones y luego recolectar un conjunto particular de datos para el análisis.
2. **Sujetos Experimentales:** La persona quien aplica los métodos o técnicas a las unidades experimentales es llamado sujeto.
3. **Variable de Respuesta:** Se refiere a la salida del experimento. Esta salida debe ser cuantitativa.
4. **Parámetros:** Se refiere a alguna característica cuantitativa o cualitativa del proyecto que es invariable a través de la experimentación. Estas son características que no deben influir sobre las variables de respuesta; por lo tanto, no influyen sobre los resultados del experimento. El conocimiento generado por el experimento sólo puede ser generalizado a través de la consideración de parámetros como variables en experimentos sucesivos y estudiando su impacto sobre la variable de respuesta.
5. **Factores:** Se refiere a cada característica del desarrollo de software que afecta la

variable de respuesta. Cada factor tiene diversas alternativas. La experimentación persigue examinar la influencia de esas alternativas sobre el valor de la variable de respuesta.

6. Niveles: Los posibles valores de los factores durante cada experimento son llamados niveles. Esto significa que cada nivel de un factor es una alternativa para el factor.
7. Interacción: Dos factores se dicen que interactúan si el valor de uno se dice influir sobre el valor de otro.
8. Experimentos Elementales: Se refiere a cada ejecución del experimento sobre la unidad experimental. Esto significa que cada aplicación de una combinación de alternativas de factores sobre una unidad experimental por parte del sujeto experimentador es un experimento elemental.

2.2 Fases de la Experimentación

Toda experimentación formal cumple con un conjunto de fases. A continuación se plantea brevemente la definición de cada fase, según son definidas por [6]:

1. Definición de los objetivos de la experimentación: La hipótesis general es transformada en una hipótesis definida en términos de cuáles variables del objeto de estudio serán examinadas. La métrica que será usada se describirá durante la fase de diseño experimental, aunque lo importante durante la definición de los objetivos es asegurar que puede definirse un procedimiento cuantitativo para evaluar la hipótesis. La salida son las hipótesis para pruebas.
2. Diseño de los experimentos: Incluye realizar un plan acorde al experimento que será ejecutado, luego de determinar las condiciones exactas del experimento, esto incluye determinar cuáles variables lo pueden afectar. Los elementos a ser considerados durante el diseño experimental pueden definirse en función de las variables involucradas. El objetivo de un buen diseño experimental es obtener la mayor cantidad de información, con la menor cantidad de experimentos posibles. La salida es el diseño experimental.
3. Ejecución de los experimentos: Los experimentos son ejecutados tal como indica en el diseño. La salida son los resultados experimentales.
4. Análisis de resultados: El objetivo es encontrar relaciones entre los resultados del estudio, entre las variables del experimento. La salida es la hipótesis probada. Cuando se analizan resultados cualquiera de las siguientes relaciones entre variables pueden descubrirse [6]:
 1. Descriptiva: Se produce cuando se detectan patrones en el comportamiento entre las variables.
 2. Correlación: Es aquella que puede describirse a través de una función.
 3. Causal: Determina la aparición determinística o probabilística del valor de una variable cuando otras variables adquieren determinados valores.

3 Descripción de los Elementos de Experimentación

Antes de comenzar con las fases de la experimentación es fundamental describir con

detalle cada uno de los elementos definidos. Se iniciará con la descripción de la unidad experimental; es decir, el sistema colaborativo seleccionado para aplicar sobre él cada uno de los factores y medir las variables de respuesta. Para esta experimentación se seleccionó un chat.

3.1 Unidad Experimental: Chat

Un chat es una aplicación que permite la comunicación en tiempo real de dos o más usuarios quienes pueden enviar y recibir mensajes a través de sus computadores. De tal manera que un chat es una herramienta colaborativa clasificada por [5] en la categoría de herramientas para la comunicación. En la presente sección se analizará la arquitectura de un chat descrito en [4].

La aplicación está constituida fundamentalmente por tres paquetes, un paquete interfaz (Iface), un paquete implementación (Impl) y un paquete cliente (Client), con esta estructura se logra desacoplar la lógica del negocio de su implementación. En la Figura 1 se muestra el conjunto de paquetes que conforman la aplicación, nótese que los paquetes implementación y cliente dependen del paquete interfaz.

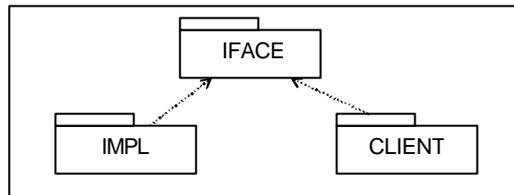


Fig. 1. Diagrama de Paquetes de la Aplicación.

A continuación se presenta una descripción más extensa del chat y su arquitectura, tal cual se define en [4].

- **Paquete Interfaz:** Este paquete consiste de tres clases, una clase interfaz de nombre IChatService y otras dos representan estructuras: ChatMessage y ChatPerson.
- **Paquete Implementación:** Este paquete desacopla la lógica del negocio de su implementación. Las clases que comprende este paquete son: ChatService, SimpleBackendImpl y ChatServerApp, a continuación la descripción de cada una de ellas.
- **Paquete Cliente:** Está constituido por la clase que representa a un cliente del servidor chat, cada instancia de esta clase puede conectarse con el servidor WASP y tener acceso a los métodos de la clase ChatService que permiten enviar y recibir mensajes. La única clase de este paquete se denomina SimpleChatService.

En la Figura 2 se presenta el diagrama de clases, a través del cual se muestra la estructura estática del chat, su organización.

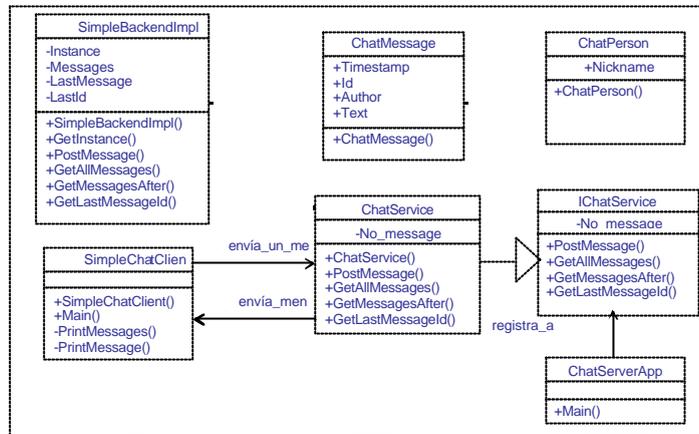


Fig. 2. Diagrama de clases del chat

En la Figura 2 se ilustran relaciones de dependencia y asociación entre las clases. Para describir el comportamiento del Chat se emplearon diagramas de casos de uso y de colaboración. En la Figura 3 se ilustra el diagrama de casos de uso del Chat que está siendo analizado. En este diagrama se presenta un actor que mantiene una interacción con la aplicación para enviar un mensaje al servidor de manera que éste pueda ser reenviado al resto de los usuarios. El primer caso de uso ilustrado es el correspondiente a Enviar Mensaje el cual usa al caso de uso Recibir Mensajes. Por otra parte, se seleccionaron tres actividades fundamentales que realiza la aplicación: la inicialización del sistema, el envío de un mensaje desde un cliente al servidor y, finalmente, la solicitud de recepción de mensajes desde un cliente al servidor.

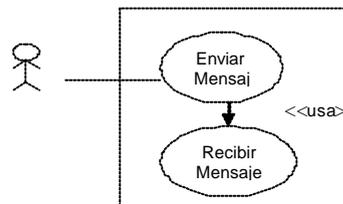


Fig. 3. Diagrama de Casos de Uso del chat

De esta manera el chat descrito será la aplicación colaborativa que constituirá la unidad experimental. Continuando con el detalle de los elementos de la experimentación se presentará la lista de las variables de respuestas.

3.2 Variables de Respuesta

En este caso particular se persigue obtener valores para las características de calidad presentes en el sistema colaborativo (chat) y luego incluyendo en su diseño las combinaciones de los patrones arquitectónicos. En esta sección se presentan las

variables de respuestas que serán consideradas como salidas de cada uno de los experimentos a realizar. Las mismas se fundamentan en el conjunto de características de calidad deseadas en un sistema colaborativo [5], según el Estándar Internacional ISO/IEC 9126. Estas características son:

- Fiabilidad/Madurez, Fiabilidad/Tolerancia a Fallos, Fiabilidad/Recuperabilidad, Funcionalidad/Conformidad, Funcionalidad/Seguridad, Funcionalidad/Interoperabilidad, Eficiencia/Utilización de Recursos, Eficiencia/Comportamiento en el Tiempo, Usabilidad/Entendibilidad, Portabilidad/Coexistencia, Mantenibilidad/Facilidad de Cambio.

De tal manera que se buscará medir cuantitativamente cada una de estas características en cada uno de los experimentos elementales. Se seleccionaron estas variables de salida ya que el interés de esta experimentación es validar el efecto de los patrones arquitectónicos en la calidad de los sistemas colaborativos.

Seguidamente se definen los parámetros de esta experimentación que no deben afectar los resultados obtenidos de la observación de las variables de respuesta.

3.3 Parámetros

Como producto de la traducción previa de las ventajas de cada patrón arquitectónico en características de calidad se determinaron ciertas precondiciones [5]. Se concluyó que algunos patrones propiciaban características que luego aparecían como características de calidad que se veían castigadas si no se cumplían estas precondiciones, constituidas por condiciones externas que hacían que una característica específica se viera perjudicada. La presente experimentación persigue determinar los efectos del uso de patrones, de combinaciones de ellos, en la calidad de un sistema colaborativo. En este sentido, se deben eliminar de la experimentación la influencia de factores ajenos a la estructura del patrón que influyan en la calidad del sistema, factores representados en este caso por ciertas condiciones externas.

La mejor definición de este conjunto de factores (precondiciones) ajenos a la estructura del patrón, dentro de la experimentación, es considerar cada factor como un parámetro, ya que se trata de características cualitativas o cuantitativas que deben mantenerse fijas durante los experimentos elementales y que por ello no interfieren en los valores finales de las variables de respuesta. Los resultados finales además deben ser generalizados dentro de los límites de este conjunto de factores. Para determinar los parámetros de esta experimentación se presenta por patrón un análisis de las condiciones que pueden afectar las variables de respuesta. En la Tabla 1 se presenta este análisis.

Tabla 1. Especificación de Parámetros de la Experimentación [8]

| Patrón | Caract./ Sub-caract | Condición | Parámetros |
|-------------|---|---|---|
| Interceptor | Mantenibilidad / Facilidad de Cambio | Si el desarrollador de la aplicación no tiene suficientes conocimientos de los servicios que puede brindar el sistema se puede llegar a un número inadecuado de despachadores que | Los despachadores de la aplicación se encuentran en la cantidad suficiente y necesaria. |

| | | | |
|-------------|---|--|--|
| | | reducirían tal como lo afirman [3], la facilidad de cambiar la aplicación y agregar nuevos servicios. | |
| Interceptor | Mantenibilidad / Facilidad de Cambio, Mantenibilidad / Facilidad de Análisis | La estructura del patrón no brinda la posibilidad que el framework pueda tomar acciones en caso que algún interceptor no retorne el control en forma oportuna. Por esto puede requerirse de la implementación de mecanismos ajenos a la estructura del patrón que permitan al framework tomar control en caso de interceptores erróneos. | No se implementarán mecanismos de control que puedan dar complejidad al sistema. |
| Reactor | Portabilidad/ Adaptabilidad | En [3] se afirma que la aplicabilidad del patrón está restringida a ambientes de procesamiento síncrono. | El ambiente de aplicación permitirá el procesamiento síncrono de eventos. |
| Proactor | Portabilidad/ Adaptabilidad | En [3] se afirma que la aplicabilidad del patrón está restringida a ambientes de procesamiento asíncrono. | El ambiente de aplicación permitirá el procesamiento asíncrono de eventos. |

3.4 Factores

El factor determinante que afectará las variables de respuesta, (que son en este caso las características de calidad que se desean medir) es la combinación de patrones arquitectónicos seleccionada para cada experimento. Así pues, los factores en esta experimentación están representados por una combinación de patrones sugerida por [3], ya que una combinación representa un elemento que interferirá en las variables de respuesta del experimento que son las características de calidad deseables en los sistemas colaborativos. Las combinaciones se presentan en la Figura 4.

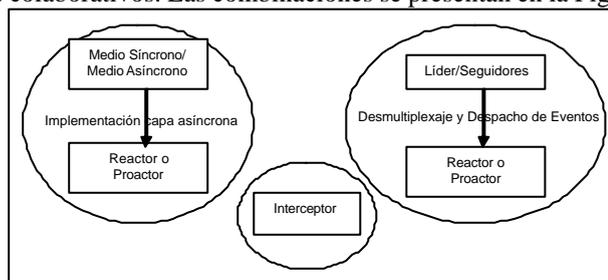


Fig. 4. Combinaciones de patrones arquitectónicos propuestos por [3]

3.5 Niveles

Cada nivel está representado por una combinación de patrones diferente. De esta manera se varía en los factores que interferirán en las variables de respuesta. En este

caso existen cinco alternativas o niveles, cada una representada por una combinación distinta de patrones.

3.6 Interacción

En este caso los resultados obtenidos por el uso de una combinación no debe influir en los resultados al usar otra, ya que las combinaciones serán aplicadas independientemente.

3.7 Experimentos Elementales

Los experimentos elementales serán seis, uno sin la aplicación de factores y cinco por cada nivel del factor o combinación de patrones. Esto permitirá comparar las variaciones de las variables de respuesta.

Una vez definidos los elementos de experimentación, puede comenzarse la aplicación de cada una de las fases propuestas por [6].

4 APLICACIÓN DE LAS FASES DE EXPERIMENTACIÓN

Las fases ilustradas en la Figura 1 serán aplicadas para lograr probar la hipótesis que planteada relativa a los efectos de las combinaciones de patrones al ser aplicadas en el diseño de una aplicación colaborativa. A continuación se presenta la aplicación de las fases de la experimentación de esta investigación.

4.1 Definición de Objetivos

El objetivo general de esta experimentación es determinar el efecto que tiene la aplicación de ciertos patrones arquitectónicos: Interceptor, Reactor, Proactor, Líder/Seguidores y Medio Síncrono/Asíncrono; en la calidad de los sistemas colaborativos. La calidad de un sistema colaborativo se definirá en función de las características determinadas como deseables en estos sistemas, las cuales fueron planteadas en la sección 3.2. En particular el objetivo de esta experimentación será probar la validez de alguna de las siguientes dos hipótesis:

H₀ = La aplicación en el diseño de un sistema colaborativo de una combinación de patrones arquitectónicos, no mejorará los niveles de algunas características de calidad especificadas como propiciadas por alguno de los patrones de la combinación.

H₁ = La aplicación en el diseño de un sistema colaborativo de una combinación de patrones arquitectónicos, mejorará los niveles de algunas características de calidad, que hayan sido especificadas como propiciadas por alguno de los patrones de la combinación.

Más en detalle, la hipótesis H₀ determina que los patrones no afectan positivamente la calidad de los sistemas colaborativos, mientras que H₁ especifica que los patrones, o combinación de ellos si afectan positivamente la calidad del

sistema colaborativo en cuyo diseño se utilice. En tal sentido, la hipótesis H1 plantea que los patrones arquitectónicos mejorarán las siguientes características en un sistema colaborativo:

1. **Interceptor:** Funcionalidad/Seguridad, Fiabilidad/Recuperabilidad, Fiabilidad/Tolerancia a Fallos, Mantenibilidad/Facilidad de Cambio.
2. **Reactor:** Mantenibilidad/Facilidad de Cambio.
3. **Proactor:** Mantenibilidad/Facilidad de Cambio.
4. **Medio Síncrono/Medio Asíncrono:** Fiabilidad/Madurez, Mantenibilidad/Facilidad de Cambio.
5. **Líder/Seguidores:** Eficiencia/Utilización de Recursos, Eficiencia/Comportamiento en el Tiempo, Mantenibilidad/Facilidad de Cambio.

Estas características se obtuvieron de la traducción realizada de beneficios de cada patrón listado por [3] en características de calidad, a su vez en la lista anterior no se mencionan todas las características propiciadas por los patrones, sólo se destacan aquellas características cuya aparición es deseable en los sistemas colaborativos [5].

4.2 Diseño Experimental

La experimentación está basada en detectar cambios cuantificables como resultado de comparar un experimento con otro en la búsqueda de diferencias entre ellos y la razón de estos cambios [6]. Por esta razón serán seis los experimentos a ser aplicados sobre la arquitectura de la aplicación colaborativa, un experimento sobre la arquitectura ya existente, y un experimento por cada combinación de patrones. De esta manera será posible comparar los efectos causados por cada combinación cuando se aplica al diseño de una aplicación colaborativa.

De tal manera que cada experimento elemental consistirá en la aplicación de métricas sobre cada arquitectura:

1. **Primer Experimento:** Evaluación de métricas sobre arquitectura del chat.
2. **Segundo Experimento:** Evaluación de métricas sobre la arquitectura del chat integrada con los patrones: Medio Síncrono/Medio Asíncrono y Reactor.
3. **Tercer Experimento:** Evaluación de métricas sobre la arquitectura del chat integrada con los patrones: Medio Síncrono/Medio Asíncrono y Proactor.
4. **Cuarto Experimento:** Evaluación de métricas sobre la arquitectura del chat integrada con los patrones: Líder/Seguidores y Reactor.
5. **Quinto Experimento:** Evaluación de métricas sobre la arquitectura del chat integrada con los patrones: Líder/Seguidores y Proactor.
6. **Sexto Experimento:** Evaluación de métricas sobre la arquitectura del chat integrada con el patrón Interceptor.

Para lograr aplicar las métricas sobre cada una de las arquitecturas mencionadas será necesario describir cada arquitectura haciendo la integración del chat con las estructuras de cada uno de los patrones que constituyen cada combinación.

A continuación se ilustran diferentes arquitecturas correspondientes a la integración entre el chat con cada combinación de patrones arquitectónicos. Cada estructura será representada a través de un diagrama de clases, sobre los cuales no se agregarán los métodos ni atributos de cada clase para hacer más entendible la notación. En las Figuras 5 a 9 se presentan las arquitecturas a evaluar en cada

experimento elemental.

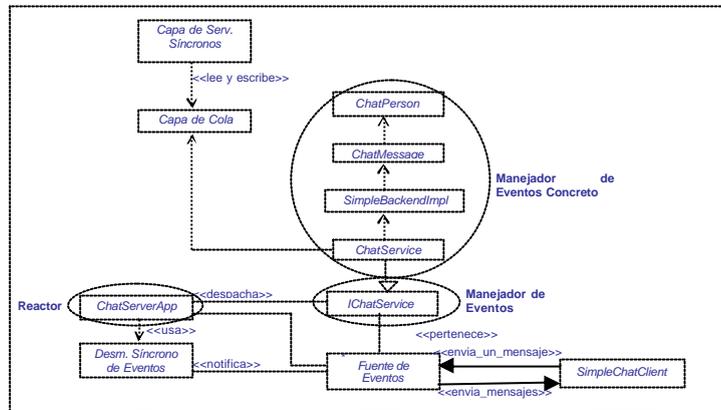


Fig. 5. Experimento 1: Chat integrado con los patrones Medio Sincrono/Asincrono y Reactor

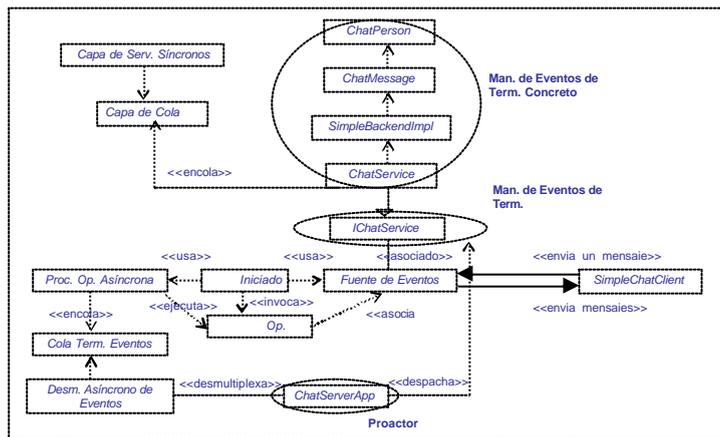


Fig. 6. Experimento 2: Chat integrado con los patrones Medio Sincrono/Asincrono y Proactor.

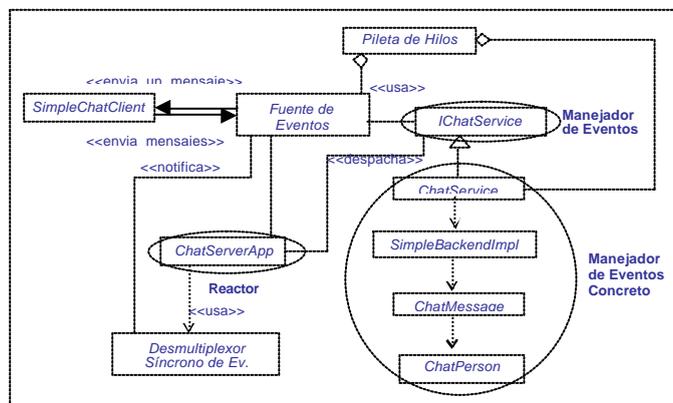


Fig. 7. Experimento 3: Chat integrado con los patrones Líder/Seguidores y Reactor

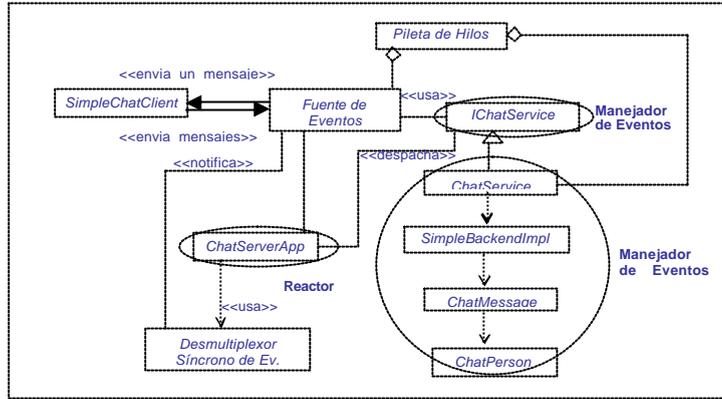


Fig. 8. Experimento 4: Chat integrado con patrones: Líder/Seguidores y Proactor

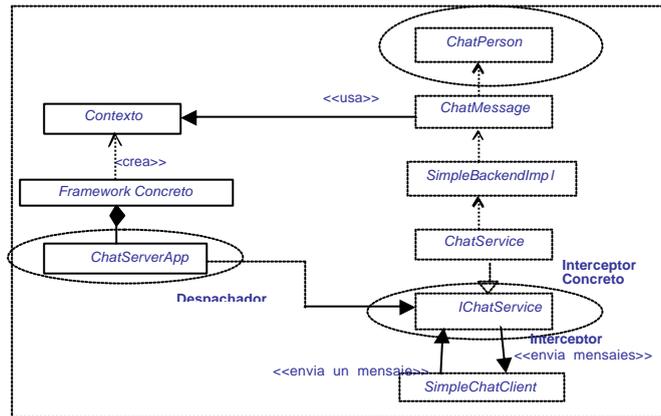


Fig. 9. Chat integrado con el patrón: Interceptor

A continuación se presenta la ejecución de los experimentos elementales, cada uno de ellos consistirá en la aplicación de métricas arquitectónicas sobre las arquitecturas descritas anteriormente. Una vez aplicadas las métricas se hará el análisis de resultados que permitirá especificar conclusiones del trabajo de investigación.

4.3 Ejecución del Experimento

En la Tabla 2 se presenta un ejemplo de las métricas a ser aplicadas sobre las arquitecturas ilustradas, posteriormente se presenta el análisis de resultados, el cual permitirá llegar a conclusiones acerca del efecto que se produce en la calidad de una aplicación colaborativa cuando se emplean patrones arquitectónicos. No se presentan todas las métricas aplicadas (27 en total), por razones de espacio.

Tabla 2. Métricas Arquitectónicas para Fiabilidad/Madurez

| Métrica | Pregunta | Escala |
|--|--|--|
| Verificación de fallas | ¿Posee la arquitectura del software algún módulo para registro y control de fallas? | 5 = Si 1 = No |
| Fallas resueltas | ¿Se tiene en los componentes procedimientos y funciones para manejar y resolver las fallas? | 5 = Tiene 1 = No Tiene |
| Definición de mecanismos para la incorporación de nuevos elementos de software | ¿La arquitectura facilita la incorporación de nuevos componentes de software al sistema? | 5 = Completamente 4 = Casi toda 3 = Medianamente 2 = Poco 1 = Nada |
| Estrategia de mantenimiento | ¿La distribución física/lógica de los componentes obedece a alguna estrategia de mantenimiento para manejar modificaciones y retiros de componentes del sistema? | 5 = Si 1 = No |

Ya definidas las métricas se procederá a aplicarlas en cada experimento elemental.

4.4 Experimentos Elementales

En la Tabla 3 se presentan los resultados de aplicar las métricas sobre cada uno de los experimentos elementales. Para obtener un valor por cada variable de respuesta; es decir, por cada característica-subcaracterística de calidad, se calculó el promedio de las métricas que miden cada una de las características en cada experimento. Para la identificación de cada experimento elemental, recuérdese que:

1. **Experimento 1:** Evaluación de métricas sobre arquitectura del chat.
2. **Experimento 2:** Evaluación de métricas sobre la arquitectura del chat integrada con los patrones: Medio Síncrono/Medio Asíncrono y Reactor.
3. **Experimento 3:** Evaluación de métricas sobre la arquitectura del chat integrada con los patrones: Medio Síncrono/Medio Asíncrono y Proactor.
4. **Experimento 4:** Evaluación de métricas sobre la arquitectura del chat integrada con los patrones: Líder/Seguidores y Reactor.
5. **Experimento 5:** Evaluación de métricas sobre la arquitectura del chat integrada con los patrones: Líder/Seguidores y Proactor.
6. **Experimento 6:** Evaluación de métricas sobre la arquitectura del chat integrada con el patrón Interceptor.

Tabla 3. Valor de las variables de respuesta de los experimentos elementales

| Caracta-Subcaract | E. 1 | E. 2 | E. 3 | E. 4 | E. 5 | E. 6 |
|--|------|------|------|------|------|------|
| Fiabilidad-Madurez | 2.5 | 3.75 | 3.75 | 2.75 | 2.75 | 4 |
| Fiabilidad-Recuperabilidad | 3 | 3 | 3 | 3 | 3 | 5 |
| Fiabilidad-Tolerancia a Fallas | 2 | 2 | 2 | 2 | 2 | 4 |
| Funcionalidad-Conformidad | 4 | 4.5 | 4.5 | 4.5 | 4.5 | 4.5 |
| Funcionalidad-Seguridad | 2.3 | 2.3 | 2.3 | 2.3 | 2.3 | 3.6 |
| Funcionalidad-Interoperabilidad | 1 | 1 | 1 | 1 | 1 | 1 |
| Eficiencia-Utilización de Recursos | 2.6 | 3 | 3.3 | 3.3 | 3.6 | 3 |
| Eficiencia-Comportamiento en el Tiempo | 2.3 | 3 | 3.3 | 3.3 | 3.6 | 3 |
| Usabilidad-Entendibilidad | 3 | 4 | 4 | 4 | 4 | 4 |
| Portabilidad-Coexistencia | 1 | 3 | 3 | 3 | 3 | 3 |

| | | | | | | |
|-------------------------------------|-----|-----|-----|-----|-----|---|
| Mantenibilidad-Facilidad de Cambios | 3.2 | 4.6 | 4.6 | 4.4 | 4.4 | 4 |
|-------------------------------------|-----|-----|-----|-----|-----|---|

La última fase de experimentación consisten en presentar las conclusiones de los resultados obtenidos y de los efectos de la aplicación de cada combinación de patrones arquitectónicos sobre la arquitectura del Chat.

4.5 Análisis de Resultados

Sobre la base de a los resultados de la experimentación mostrados en la Tabla 3, para cada una de las características/subcaracterísticas de calidad deseables en los sistemas colaborativos se presenta la relación con los patrones arquitectónicos.

Fiabilidad / Madurez. Es propiciada por todas las combinaciones: Interceptor, Medio Síncrono/Asíncrono y Reactor, Medio Síncrono/Asíncrono y Proactor, Líder/Seguidores y Reactor y Líder/Seguidores y Proactor. Interceptor propicia la Madurez gracias a uno de sus elementos: Contexto, que permite detectar ciertos estados del framework y tomar acciones para evitar la caída del sistema. Por su parte las combinaciones que contemplan a Medio Síncrono/Asíncrono mejoran la madurez gracias a que este patrón por su estructura divide los servicios de aplicación gracias a lo cual cuando se presente una falla la misma será sectorizada, es decir no debe afectar a todo el sistema. Por último las combinaciones que incluyen a Líder/Seguidores mejoran la Madurez ya que una de las métricas de esta subcaracterística mide la presencia de mecanismos para incorporar elementos de software, en tal sentido los patrones arquitectónicos en general facilitan la inclusión de nuevas funcionalidades a los sistemas colaborativos.

Fiabilidad / Recuperabilidad Para mejorar la Recuperabilidad de un sistema colaborativo se recomienda el uso de Interceptor. Este patrón mejora esta subcaracterística gracias a la presencia del mencionado elemento de nombre Contexto.

Fiabilidad / Tolerancia a Fallas. Análogamente a la Recuperabilidad, en un sistema colaborativo se puede mejorar la Tolerancia a Fallas usando Interceptor.

Funcionalidad / Conformidad. La Conformidad se mide en general con el grado de satisfacción del usuario respecto a las funcionalidades ofrecidas por la aplicación. En este caso incorporar patrones arquitectónicos amplía el conjunto de funcionalidades ofrecidas por la aplicación colaborativa, ya que establece cómo deben ser manejado los eventos, la concurrencia y el acceso y configuración de servicios.

Funcionalidad / Seguridad. La Seguridad en un sistema colaborativo puede ser mejorada con el uso del Interceptor gracias al cual se pueden implementar mecanismos para detectar el acceso no autorizado al sistema.

Funcionalidad / Interoperabilidad. Ninguno de los patrones arquitectónicos permite mejorar la Interoperabilidad del sistema, esto se debe a que no disponen de elementos que medie la interacción con otros sistemas.

Eficiencia / Utilización de Recursos y Eficiencia / Comportamiento en el Tiempo. Para mejorar la Eficiencia de un sistema colaborativo pueden incorporarse al diseño principalmente las combinaciones que incluyen al patrón Líder/Seguidores, esto es gracias a que este patrón dispone que un evento es ejecutado desde su detección por un mismo hilo de ejecución, lo cual evita pérdida de tiempo y recursos en cambios de contexto. Por su parte el uso de las combinaciones que incluyen el

patrón Medio Síncrono/Asíncrono según [3] pueden mejorar la eficiencia debido a que este patrón permite el uso de multiprocesadores. Por su parte, el patrón Interceptor según presenta a la Eficiencia entre las características que perjudica, sin embargo por la experimentación realizada se pudo corroborar que la Eficiencia es mayor con el uso de Interceptor que sin el uso de patrones arquitectónicos.

Usabilidad / Entendibilidad. Todas las combinaciones mejoran la Entendibilidad debido a que los modelos se hacen más entendible con el uso de patrones arquitectónicos puesto que brindan estructura al sistema colaborativo.

Portabilidad / Coexistencia. El uso de combinaciones que incorporen el Reactor y Proactor mejoran la coexistencia dado que de alguna manera estos patrones regulan el acceso a recursos que pueden ser compartidos con otras aplicaciones.

Mantenibilidad / Facilidad de Cambio. Todas las combinaciones mejoran esta subcaracterística, sin embargo, las que incluyen el patrón Medio Síncrono/Asíncrono la incrementan más seguido por Líder/Seguidores e Interceptor. Todos los patrones ofrecen estructuras que facilitan la incorporación de nuevos servicios a un sistema colaborativo.

En síntesis, el experimento cuantitativo permitió validar la hipótesis acerca de las características que propiciaba cada patrón en un sistema colaborativo, pero además, pudo evidenciar que el uso de patrones mejora características que aunque no hayan sido consideradas como beneficios del patrón, son mejoradas cuando éste se aplica a la arquitectura del sistema. Tal es el caso de: Funcionalidad / Conformidad, Usabilidad / Entendibilidad y Portabilidad / Coexistencia. Aunque ninguno de los patrones se destaca por propiciar alguna de estas tres características, se tendrá que, al aplicar los patrones arquitectónicos en un sistema colaborativo ellas pueden mejorar o mantenerse. Conociendo los resultados del experimento, el arquitecto de un sistema colaborativo podrá decidir cuál combinación de patrones arquitectónicos usar, todas las combinaciones son apropiadas para el diseño de un sistema colaborativo, ya que todas propician ciertas características que son deseables en este tipo de aplicaciones. De tal manera, que la selección de alguna combinación en particular dependerá de las prioridades en el aspecto de calidad que puedan tener los stakeholders.

5 Conclusiones y Recomendaciones

Considerando la importancia de los patrones arquitectónicos en el diseño de los sistemas colaborativos, se realizó un análisis que permitió determinar los efectos de los patrones en la calidad de estos sistemas. Para validar si los patrones arquitectónicos de [3] realmente satisfacen en los sistemas colaborativos algunas características de calidad se realizó un experimento cuantitativo, el cual incluyó un conjunto de métricas arquitectónicas que permitieron la evaluación cuantitativa de las características de calidad esperadas. Dicho experimento constó de seis experimentos elementales, cada uno de ellos se basó en la aplicación de métricas arquitectónicas sobre la arquitectura de una aplicación colaborativa, un chat, el cual en el primer experimento se presentó sin patrones y en los restantes fue integrado con combinaciones de patrones arquitectónicos. En el experimento se usaron combinaciones de patrones arquitectónicos sugeridas por [3]: Medio

Síncrono/Asíncrono y Reactor, Medio Síncrono/Asíncrono y Proactor, Líder/Seguidores y Reactor, Líder/Seguidores y Proactor y por último el patrón Interceptor.

Luego de la ejecución de los experimentos cuantitativos es posible responder efectivamente las preguntas formuladas al inicio de la investigación. Permite validar la hipótesis acerca de las características que propiciaba cada patrón en un sistema colaborativo, pero además, se evidenció que el uso de patrones mejora características que aunque no hayan sido consideradas como beneficios del patrón, son mejoradas cuando éste se aplica a la arquitectura del sistema. Tal es el caso de: Funcionalidad/Conformidad, Usabilidad/Entendibilidad y Portabilidad/Coexistencia. Aunque ninguno de los patrones se destaca por propiciar alguna de estas tres características, se tendrá que, al aplicar los patrones arquitectónicos en un sistema colaborativo, ellas pueden mejorar o mantenerse. Después de los resultados obtenidos, surge fundamentalmente como recomendación para futuras investigaciones, que estos experimentos sean extendidos usando métricas que evalúen la calidad del sistema en ejecución y no sólo arquitectónicamente.

6 Referencias Bibliográficas

1. Bass, L., Clements, P. & Kazman, R. (2003). *Software Architecture in Practice*. (2da ed.). Estados Unidos: Addison Wesley.
2. Bosch, J. (2000) *Design and use of Software Architecture*. Inglaterra: Addison-Wesley.
3. Buschman, F., Meunier, R., Rohnert, H., Sommerlad, P. & Stal, M. (2000). *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*, Vol. 2 Estados Unidos: John Wiley & Sons.
4. Dvorak, P. (2003). *A Simple Chat Service for Command Line Development*. Sitio web de Systinet: <http://www.systinet.com>
5. Hidalgo, I (2004) *Patrones Arquitectónicos para Sistemas Colaborativos*, Tesis de Maestría en Ingeniería de Sistemas, Venezuela: Universidad Simón Bolívar
6. Juristo, N., Moreno, A. (2001) *Basics of Software Engineering Experimentation*, Kluwer Academic Publishers
7. Kitchenham, B. (1996). Evaluation Software Engineering Methods and Tools, Part 3: Selecting an appropriate evaluation method-practical issues. Sigsoft Notes.
8. O'Brien, J., (2002). *Management Information Systems: Managing Information Technology in the E-Business Enterprise*. (5^a ed.). Mc Graw Hill.