

# Comparación entre enfoques de prueba en la verificación de una arquitectura de integración

## *Test approaches top-down, bottom-up and hybrids in the verification and validation of architecture*

### **LUIS F. WANUMEN S.**

Ingeniero de Sistemas. Docente de la Universidad Distrital Francisco José de Caldas, Integrante del grupo de investigación METIS. Bogotá, Colombia.

Contacto: [lwanumen@udistrital.edu.co](mailto:lwanumen@udistrital.edu.co)

### **DARIN MOSQUERA PALACIOS**

Ingeniero de Sistemas. Docente de la Universidad Distrital Francisco José de Caldas, director del grupo de investigación ORIÓN. Bogotá, Colombia.

Contacto: [dmosquerap@udistrital.edu.co](mailto:dmosquerap@udistrital.edu.co)

### **EDWIN RIVAS TRUJILLO**

Ingeniero Eléctrico, Doctor en Ingeniería. Director del grupo Interferencia y Compatibilidad Electromagnética. Profesor de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. Contacto: [erivas@udistrital.edu.co](mailto:erivas@udistrital.edu.co)

Fecha de recepción: 30 de Julio de 2012

Clasificación del artículo: Revisión de Tema

Fecha de aceptación: 1 de Octubre de 2012

Grupo de Investigación: ORION y METIS

**Palabras clave:** *Estrategia de Pruebas, Enfoque TopDown, Enfoque BottomUp, Enfoque inside-out, Enfoque outside-in, Método Feature Comparison.*

**Key words:** *Test Strategy, Approach TopDown, Approach BottomUp, Approach inside-out, Approach outside-in, Feature Comparison Method.*

## **RESUMEN**

Este artículo analiza las ventajas y desventajas de los enfoques Top-Down, Bottom-Up e híbridos en la verificación de una arquitectura de interacción entre dos sistemas. Se usa una adaptación del método de “feature comparison”[1] en la realización de esta comparación. Primero se hace una descripción de estos métodos, segundo, se crea un escenario en donde se pueda aplicar estos métodos. Tercero se analizan las ventajas y desventajas de cada método y documentar los resultados.

## **ABSTRACT**

This article discusses the advantages and disadvantages of the approaches Top-Down, Bottom-Up and hybrids in the verification of architecture of interaction between two systems. The comparison between these approaches is done using the «feature comparison». First is a description of these methods, second, it creates a scenario in which to apply these methods. Thirdly we analyze the advantages and disadvantages of each method and document the results.

## 1. INTRODUCCIÓN

Existen enfoques Top-Down, Bottom-Up, e híbridos que han sido utilizados en la ingeniería del software para validar arquitecturas. La literatura sobre estos enfoques reconoce ventajas y desventajas de éstos a nivel general, pero se quiere establecer para un caso específico de integración entre dos sistemas cuáles son las ventajas y desventajas reales del uso de estos enfoques. Para ello, se usa una adaptación del método de “feature comparison”. Este sistema exige hacer una descripción de los métodos a comparar, crear un escenario concreto para aplicar esta comparación, y documentar los resultados. En la descripción se incluyen los enfoques clásicos Top Down, Bottom Up y alternativos (como el de fusión y el que incluye el contexto organizacional).

### 2. DESCRIPCIÓN DEL ENFOQUE CLÁSICO TOP DOWN

El Top Down hace parte de los enfoques clásicos de diseño de sistemas, en donde desde la primera iteración se define como se comportará el sistema y desde este punto de vista, el Top Down permite establecer con anterioridad las funcionalidades generales que tendrá el sistema [2].

#### 2.1 Ventajas del enfoque Top-Down

El enfoque de proceso Top-Down es mundialmente aceptado por la industria y su uso favorece la internacionalización de procesos [5].

De otra parte, usar la filosofía Bottom-Up, permite que el sistema construido no ignore las metas para las cuales fue construido. Una verificación usando enfoque Top-Down produciría información relevante para saber si el sistema cumple con las metas organizacionales para las cuales fue construido [6]. Usar este enfoque en pruebas ayudaría a que no se perdieran estos objetivos.

#### 2.2 Desventajas del enfoque Top-Down

El inconveniente de usar solamente el enfoque TopDown es que permite obtener teorías incompletas. Es decir, la información que se puede obtener sobre el funcionamiento del sistema a partir de un enfoque Top Down es incompleta [3].

A pesar que el enfoque Top Down permite diseñar sistemas como los de redes que por naturaleza tienen el enfoque Top Down, también es innegable que el aprendizaje y verificación de tales sistemas se logra mejor con otros enfoques incluso contrarios al TopDown [4].

### 3. DESCRIPCIÓN DEL ENFOQUE MODERNO BOTTOM UP

El enfoque Bottom-Up, al igual que el enfoque Top-Down, ha sido pensado como una aproximación conceptual que responde a varias tareas en el desarrollo de múltiples sistemas. Es decir, puede usarse este enfoque en la planeación, en el análisis, en el diseño o incluso en la misma implementación. En este caso se hablará más del enfoque Bottom-Up, desde el punto de vista de su contribución en la fase de pruebas de un sistema software.

#### 3.1 Ventajas del enfoque Bottom-Up

La ventaja del enfoque Bottom-Up en la construcción, es que permite construir artefactos de calidad [6]. Se puede derivar entonces que el uso del enfoque Bottom-Up ayuda a identificar con que calidad se hizo un sistema.

#### 3.2 Desventajas del enfoque Bottom-Up

Una de las principales desventajas del enfoque Bottom-Up, aplicado a las pruebas, es su dificultad y complejidad de implantación en sistemas, sobre todo cuando éste tiene gran complejidad y tamaño [6].

## 4. DESCRIPCIÓN DE LOS ENFOQUES ALTERNATIVOS

El uso del enfoque clásico Top-Down y del enfoque moderno Bottom-Up, por separado, ha sido criticado, en parte porque de por sí solos, estos modelos al aplicarse en forma única para verificar un sistema software no arrojan información completa sobre los defectos del sistema [3].

Surgen entonces alternativas para nuevos enfoques de pruebas, entre los que se cuentan, los híbridos que mezclan el Top-Down y el Bottom-Up [3] e incluso enfoques que proponen uno tercero para ser integrado con los dos anteriores.

### 4.1 Enfoque de fusión entre Bottom-Up y Top-Down

El enfoque que plantea la fusión entre el Bottom-Up y el Top-Down, surge de identificar que por separado los dos arrojan información incompleta, en relación con la que arrojan cuando se integran [3]. Es importante aclarar que en muchas situaciones la información arrojada por los dos enfoques es complementaria.

El enfoque Bottom-Up ayuda a reconocer cómo fue construido un sistema, en tanto que el enfoque Top-Down, permite predecir cómo se comportará el sistema [2].

### 4.2 Enfoque de incluir el contexto organizacional

El enfoque que busca incluir una tercera aproximación, el organizacional, a las dos ya existentes, fue planteado fruto de reconocer que la fusión entre las aproximaciones Top-Down y Bottom-Up, es mejor que por separado. También es cierto que el mundo ha seguido cambiando y requiere de enfoques de prueba, de análisis y de desarrollo que se adapten a la realidad altamente cambiante de los sistemas.

Observando la anterior situación, se refleja que el

entorno cambia y que el enfoque Top-Down y el enfoque Bottom-Up verifica algunos aspectos del sistema, pero no permite verificar la relación completa del sistema con el exterior de la organización, o la relación de este con otros sistemas al interior de la organización [7].

## 5. APLICACIÓN DE LOS ENFOQUES DE PRUEBAS

A continuación se muestra como se aplicaron los enfoques de prueba en el escenario de una estrategia de interacción entre un SIG y una SBA.

### 5.1 Aplicación del enfoque Top-Down

Teniendo en cuenta que es posible hacer una descomposición de una arquitectura en forma recursiva hasta llegar a los componentes más detallados que encapsulan lógica de patrones de diseño específicas [8], se propone una verificación Top-Down a partir de atributos de calidad y patrones [9] que parten del estilo arquitectónico orientado a datos y llegan hasta los patrones de diseño detallados tal como muestra la figura 1.

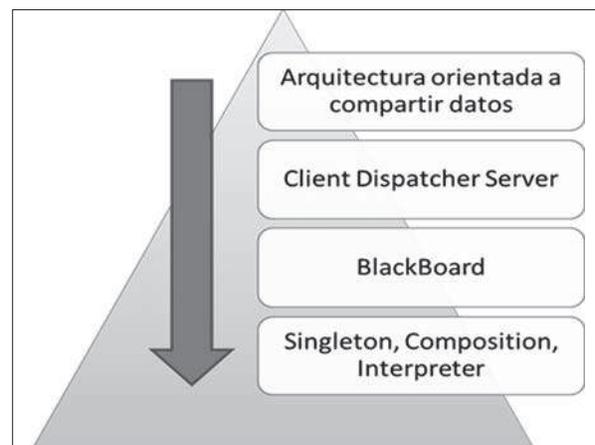


Figura 1. Proceso de aplicación del enfoque Top-Down.

La figura 1 hace que se pueda comprender mejor la arquitectura y que se pueda explicar y justificar éste estilo, sin embargo, no permite ver la coheren-

cia entre los diversos niveles, a menos que se haga una validación de coherencia entre dichos niveles.

## 5.2 Aplicación del enfoque Bottom-Up

Una vez desarrollado el sistema, se procedió a realizar una verificación que parte desde el código fuente hasta llegar a establecer el estilo de arquitectura. Para lograr esta validación se usaron estructuras de código fuente como patrones de bajo nivel y a través de un enfoque iterativo e interactivo se recupero la arquitectura construida sobre tales niveles inferiores extraídos del código fuente. Seguidamente, se observaron las asociaciones entre los casos de patrones extraídos y de esta manera se establecieron los puntos de vista de nivel superior del sistema. Estos puntos de vista proporcionaron información para un refinamiento consecutivo de definiciones de patrones de agregados. Finalmente, con estos patrones de agregados se puede obtener un resumen que permita la descripción de la arquitectura del sistema software.

La aproximación usada para este enfoque Bottom-Up fue una basada en el balanceo de subsistemas que siguiendo un enfoque de éste tipo permite identificar aspectos a mejorar en un proceso [10]. La técnica consiste en verificar si un paquete es abstracto o si es concreto. No se permiten paquetes intermedios, es decir, paquetes que tengan una cantidad similar de clases abstractas y una cantidad similar de clases concretas, porque esto indicaría que el objetivo del paquete no está bien definido. En este proceso se usaron entre otras las siguientes métricas:

- Verificación de extensibilidad de los paquetes (NCC y NCI).
- Verificación basada en la responsabilidad de los paquetes (CA) Afferent Couplings.
- Verificación basada en independencia de los paquetes (EC) Efferent Coupling.
- Verificación del grado de abstracción de los paquetes.

- Verificación de la capacidad de cambio de los paquetes.
- Verificación del balance entre abstracción y estabilidad de los paquetes.

Al final del proceso se hicieron las adecuaciones a los paquetes que no tenían definida su grado de abstracción en alto o bien bajo.

El proceso seguido en este enfoque se muestra en la figura 2.

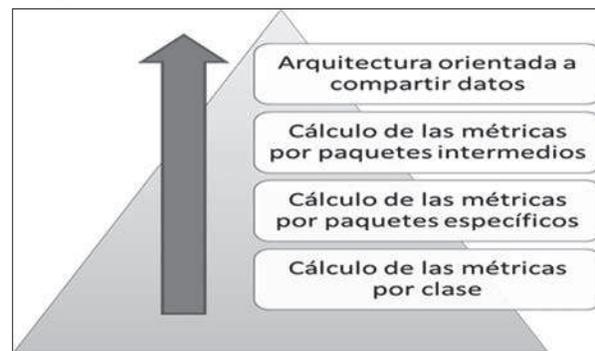


Figura 2. Proceso de aplicación del enfoque Bottom-Up.

La figura 2 refleja, que si el cálculo de las métricas por clase se hace con errores, el proceso de aplicación de pruebas con el enfoque Bottom-Up perderá casi en su totalidad la fiabilidad.

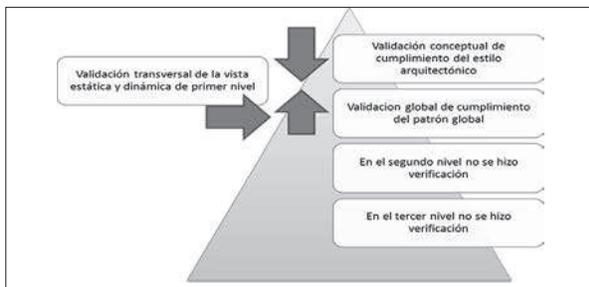
## 5.3 Aplicación del enfoque que fusiona Bottom-Up y Top-Down

Existen fuertes teorías que afirman que el enfoque Bottom-Up complementa el enfoque Top-Down [11]. En esta sección se describe una propuesta de cómo aplicar esta fusión en un caso concreto. Para ello, se realiza una validación conceptual de arriba hacia abajo de cumplimiento del estilo de la arquitectura que parte del estilo y lo va refinando hasta el nivel intermedio a nivel de texto, es decir, a nivel semántico [12]. Luego se hace una validación conceptual Bottom-Up en cumplimiento del patrón global [13], con el fin de evaluar la necesidad de tener un patrón arquitectónico “client dispatcher server” y un estilo “basado en compartición de datos”. Y por último, se verifica conceptualmente que

la vista dinámica de más alto nivel sea coherente con la vista estática de más alto nivel [14].

La figura 3 muestra el proceso seguido que insta el enfoque híbrido en la verificación de la arquitectura propuesta.

En la figura 3 queda claro que no se hicieron verificaciones híbridas para los niveles 2 y 3. La razón fue la complejidad en el manejo del texto al expresar exactamente cada patrón. La ventaja de la aplicación de este enfoque es haber logrado solucionar problemas de incoherencia entre las vistas estáticas y dinámicas del primer nivel de la arquitectura.



**Figura 3.** Proceso de aplicación del enfoque que fusiona Bottom-Up y Top-Down

## 5.4 Aplicación del enfoque que incluye el contexto organizacional

En la aplicación de estos enfoques se hicieron dos verificaciones. Una usando el modelo de aceptación tecnológica [15] y la otra haciendo una verificación de la arquitectura a partir del cumplimiento de escenarios [16]. Los escenarios se muestran en la tabla 1.

**Tabla 1.** Escenarios de prueba del sistema.

ID Escenario	Explicación
E1	La SBA inicia primero que el SIG
E2	El SIG inicia primero que la SBA
E3	Se tienen imágenes tipo Raster
E4	Se tienen imágenes tipo Vector
E5	Se tiene una gran cantidad de agentes enviando información al SIG
E6	Se quiere verificar que la georeferenciación sea correcta

El resultado de estas verificaciones y validaciones

demonstró que la arquitectura necesita, en próximos proyectos de investigación, mejoras en la parte de georeferenciación, sobre todo cuando se sigan añadiendo elementos de la lógica de negocio al SIG.

## 6. RESULTADOS

Los enfoques anteriores fueron aplicados en la integración entre un sistema de información geográfico y una simulación para visualizar emergencias por fuego. Los resultados de esta aplicación se muestran en la tabla 2, en donde se aprecian las ventajas y desventajas de la aplicación de cada enfoque de pruebas para el escenario inicialmente planteado.

**Tabla 2.** Ventajas y desventajas iniciales encontradas en la aplicación de los tres enfoques de prueba.

Enfoque	Ventaja	Desventaja
Top-Down	Ampliamente aceptado Permite enfocarse en las metas organizacionales Más sencillo de aplicar	No permite detectar la mayor cantidad de errores.
Bottom-Up	Permite detectar la mayor cantidad de errores.	Es menos general y por tanto está menos documentado que el Top-Down,
Fusión entre Bottom-Up y Top-Down	Permite detectar la mayor cantidad de errores que si se usa por separado el enfoque Top-Down y el enfoque Bottom-Up.	No toma en cuenta lo cambiante de la organización.
outside-in	Ayuda a verificar los cambios al exterior de la organización. Ayuda a encontrar errores de funcionalidad.	No ayuda a realizar verificaciones estructurales, es más para la parte funcional.
inside-in	Ayuda a verificar los cambios al interior de la organización Ayuda a encontrar errores de funcionalidad.	No ayuda a realizar verificaciones estructurales, es más para la parte funcional.

## 7. VALIDACION

La verificación Top-Down, a partir de atributos de calidad y patrones, tuvo la ventaja de ser el enfoque de pruebas más fácil de realizar y uno de los

que permitió comprender conceptualmente la arquitectura.

La verificación basada en el balanceo de subsistemas, fue una de las que más errores encontró en la arquitectura e incluso permitió encontrar errores en la implementación de los patrones de diseño, sin embargo, es bastante larga y requiere mucho tiempo para su ejecución.

Las tres validaciones de los enfoques que fusionan Bottom-Up con Top-Down, encontraron errores, aunque en menor proporción que la validación Bottom-Up. Sin embargo, a pesar que a nivel textual no es larga de ejecutar, si es compleja debido al manejo del texto que se requiere para poder realizar análisis semánticos basados en escritos. Para los niveles superiores de la arquitectura es una buena estrategia de pruebas, pero para niveles inferiores genera muchas incoherencias.

Los enfoques de validaciones que incluyen el contexto organizacional, son las validaciones que menos incoherencias generan y aunque no dejan en evidencia una gran cantidad de errores, los fallos que encuentran son de gran impacto y podrían ocasionar el rediseño de la arquitectura.

## 8. CONCLUSIONES

El enfoque TopDown es el más intuitivo y sencillo de usar, pero uno de los que menos errores de codificación logró obtener en la arquitectura de integración. Por su parte, el enfoque Bottom Up, a pesar de su dificultad para usar, fue el que más éxito tuvo para el caso de encontrar errores en la codificación. Finalmente, los enfoques híbridos demostraron mayor robustez que los enfoques Top Down y Bottom Up por separado. Sin embargo, el esfuerzo para su aplicación no compensa con la cantidad de errores encontrados.

---

## REFERENCIAS

- [1]. Kasutani E., Yamada, A., “An adaptive feature comparison method for real-time video identification”, Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on , vol.2, no., pp. II-5-8 vol.3, 14-17 Sept. 2003.
- [2]. Faubel C., Schoner G., “A neuro-dynamic architecture for one shot learning of objects that uses both bottom-up recognition and top-down prediction”, Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on , vol., no., pp.3162-3169, 10-15 Oct. 2009.
- [3]. Idestam-Almquist P., “A method to integrate top down and bottom up induction for augmentation of incomplete theories”, CompEuro ‘92 . ‘Computer Systems and Software Engineering’, Proceedings. , vol., no., pp.107-108, 4-8 May 1992.
- [4]. Koo S.G.M., Kwong S.W. , “Teaching Computer Communication Networks: Top-down or Bottom-up?”, Frontiers in Education, 2005. FIE ‘05. Proceedings 35th Annual Conference , vol., no., pp.S2H, 19-22 Oct. 2005.
- [5]. Thomas M., McGarry F., “Top-down vs. bottom-up process improvement”, Software, IEEE , vol.11, no.4, pp.12-13, Jul 1994.
- [6]. PizkaM., Bauer, A., “A brief top-down and bottom-up philosophy on software evolution”, Software Evolution, 2004. Proceedings. 7th International Workshop on Principles of , vol., no., pp. 131- 136, 6-7 Sept. 2004.
- [7]. O’Leary D.E., “Developing multiple-agent systems is more than top-down vs. bottom-up”, Intelligent Systems and their Applications, IEEE , vol.13, no.2, pp.2-5, Mar/Apr 1998.
- [8]. Bosch J., “Specifying frameworks and design patterns as architectural fragments”,

- Technology of Object-Oriented Languages, 1998. TOOLS 27. Proceedings , vol., no., pp.268-277, 22-25 Sep 1998.
- [9]. Bachmann F., Bass, L., Klein M., Shelton C., “Designing software architectures to achieve quality attribute requirements”, Software, IEE Proceedings - , vol.152, no.4, pp. 153- 165, 5 Aug. 2005.
- [10]. Jakobsen A.B., “Bottom up process improvement tricks”, Software, IEEE , vol.15, no.1, pp.64-68, Jan.-Feb. 1998.
- [11]. Yuhua Zheng, Yan Meng, Yaochu Jin, , “Fusing bottom-up and top-down pathways in neural networks for visual object recognition” Neural Networks (IJCNN), The 2010 International Joint Conference on , vol., no., pp.1-8, 18-23 July 2010.
- [12]. Kotb, Y., “Improving the UML consistency using Text Semantic Similarity approach”, Computer Technology and Development (ICCTD), 2010 2nd International Conference on , vol., no., pp.90-94, 2-4 Nov. 2010.
- [13]. Litvak B., Tyszberowicz S., Yehudai A, “Behavioral Consistency Validation of UML Diagrams”, Software Engineering and Formal Methods, 2003.Proceedings. First International Conference on , vol., no., pp.118-125, 22-27 Sept. 2003.
- [14]. J. Muskens, R.J. Bril, M.R.V. Chaudron, “Generalizing Consistency Checking Between Software Views” Software Architecture, 2005. WICSA 2005. 5th Working IEEE/IFIP Conference on , vol., no., pp.169-180, 2005
- [15]. Chiou Wen-Chih, Perng Chyuan, Lin Chin-Chao , “The Relationship Between Technology Acceptance Model and Usability Test - Case of Performing E-learning Task with PDA”, Information Engineering, 2009. ICIE '09. WASE International Conference on , vol.1, no., pp.579-582, 10-11 July 2009.
- [16]. Masse, J., Saehwa Kim; Seongsoo Hong, “Tool set implementation for scenario-based multithreading of UML-RT models and experimental validation”, Real-Time and Embedded Technology and Applications Symposium, 2003. Proceedings. The 9th IEEE, vol., no., pp. 70- 77, 27-30 May 2003.