

# Patrones y Arquitectura de Software: Una Revisión Integrativa

María José Murcia Martínez

Neiva, Huila

mariajosemurciamartinez85@gmail.com

Centro de la Industria, la Empresa y los Servicios

## Resumen

Este artículo presenta una revisión integrativa de 34 artículos relacionados con patrones de diseño y arquitectura de software. Se discuten las tendencias actuales y la importancia de la usabilidad, la flexibilidad y la adaptabilidad en el desarrollo de software. Se analizan diferentes enfoques y metodologías, resaltando la necesidad de herramientas y técnicas que faciliten la implementación de arquitecturas eficientes.

**Palabras clave:** Patrones de diseño, arquitectura de software, usabilidad, metodologías ágiles, integración de sistemas.

## 1. Introducción

La arquitectura de software juega un papel crucial en el desarrollo de aplicaciones, ya que establece la estructura fundamental que guiará la implementación y el mantenimiento del sistema. Este artículo revisa 34 estudios que exploran diversas perspectivas sobre patrones de diseño y su impacto en la arquitectura de software.

## 2. Revisión de la Literatura

La literatura muestra un creciente interés por la integración de la usabilidad desde el diseño inicial del software [1]. La arquitectura de software, definida como la organización de un sistema, es fundamental para asegurar un bajo acoplamiento entre componentes [2]. Además, se evidencia la necesidad de

adaptarse a cambios constantes en los requisitos del negocio [3].

## 3. Metodología

Se realizó una revisión sistemática de la literatura, recopilando artículos relevantes sobre patrones de diseño y arquitectura de software. Los artículos fueron clasificados según su enfoque y contribuciones al campo.

## 4. Resultados

A continuación, se presentan los resúmenes de los 34 artículos revisados:

1. **Patrones de usabilidad:** Mejora la usabilidad del software desde el momento arquitectónico, integrando ajustes continuos en el diseño [1].
2. **Arquitectura de software:** Describe la organización de un sistema, enfatizando un acoplamiento bajo entre componentes para facilitar el mantenimiento [2].
3. **Introducción a la Arquitectura de Software:** Explora cómo esta disciplina crítica se adapta a cambios y requisitos sociales y de negocio [3].
4. **Atributos de calidad:** Analiza cómo la arquitectura organiza el sistema, enfocándose en atributos como mantenibilidad y seguridad [4].
5. **Integración en metodologías ágiles:** Examina la relación entre arquitectura y metodologías

- ágiles, destacando la importancia de requisitos arquitectónicos [5].
6. **Patrones de Diseño GOF:** Estudia la aplicación de estos patrones en el desarrollo web en Colombia, subrayando la necesidad de capacitación [6].
  7. **Análisis comparativo de Patrones de Diseño:** Compara cinco patrones, enfatizando su contexto específico de aplicación [7].
  8. **Revisión sobre generadores de código:** Destaca su papel en la eficiencia del desarrollo de software, especialmente en aplicaciones web [8].
  9. **Teoría para el diseño de software:** Propone que el diseño sea un proceso continuo, anticipando cambios y dividiendo en componentes manejables [9].
  10. **Perfiles UML:** Examina cómo pueden mejorar la representación de patrones de diseño, facilitando su personalización [10].
  11. **Lenguajes de Patrones:** Explica la importancia de estos lenguajes en la arquitectura de software, permitiendo soluciones reutilizables [11].
  12. **Arquitectura para comunidades académicas:** Propuesta para facilitar el acceso a la educación mediante televisión digital interactiva [12].
  13. **Herramienta tecnológica de costos:** Simula costos y programación de obras para estudiantes de Ingeniería Civil [17].
  14. **Modelado y Verificación en la Nube:** Propuesta de herramienta para modelar arquitecturas en la nube, asegurando adherencia a mejores prácticas [26].
  15. **Arquitectura guiada por el Dominio:** Enfoque que centra el desarrollo en el conocimiento del negocio, facilitando flexibilidad [?].
  16. **Sistema de visualización médica Vismedic:** Mejora del sistema mediante una arquitectura por capas, optimizando extensibilidad [?].
  17. **Desarrollo de Intranet Industrial:** Aplicación de patrones de diseño orientados a objetos para mejorar la eficiencia [17].
  18. **Arquitectura en desarrollo ágil:** Propuesta de un modelo que integra la arquitectura con metodologías ágiles [18].
  19. **Herramienta para aprender patrones:** Desarrollo de una herramienta web que facilita la comprensión de patrones de diseño [19].
  20. **Aplicación de patrones para flexibilidad:** Muestra cómo los patrones permiten crear software adaptable a cambios en reglas de negocio [20].
  21. **ReusMe:** Herramienta que facilita la reutilización de código JavaScript para mejorar la usabilidad [21].
  22. **Arquitectura de Microservicios:** Propuesta para superar limitaciones de arquitecturas monolíticas, mejorando escalabilidad [22].
  23. **Arquitectura del robot Lázar:** Descripción de la arquitectura para gestionar sensores y actuadores, facilitando control [23].
  24. **Arquitectura N-capas:** Mejora de la modularidad y reutilización mediante separación de responsabilidades en componentes [24].
  25. **Módulo de recomendación para EGPat:** Herramienta para gestionar y recomendar patrones de diseño en recursos educativos [25].
  26. **Arquitectura reutilizable:** Propuesta para desarrollar aplicaciones web rápidamente mediante la reutilización de patrones [26].
  27. **Integración de objetos de aprendizaje:** Propuesta de arquitectura basada en servicios web para mejorar gestión de contenidos [27].
  28. **Lineamientos para cajero automático:** Aborda patrones GRASP, asegurando alta cohesión y bajo acoplamiento [28].

29. **Arquitectura en proceso de desarrollo:** Importancia de la arquitectura en el ciclo de vida en espiral, integrando MDA [29].
30. **Modelo "4+1":** Divide la arquitectura en cinco vistas complementarias, facilitando el enfoque en aspectos específicos [30].
31. **Reingeniería de software:** Mejora sistemas heredados mediante ingeniería inversa y rediseño [31].
32. **APIs de IoT usando C++:** Explora la creación de APIs eficientes utilizando patrones de diseño y metaprogramación [32].
33. **Framework web para aplicaciones:** Estandariza y optimiza la creación de aplicaciones dinámicas [33].
34. **Coffee Challenge:** Juego educativo que enseña patrones de diseño mediante simulaciones prácticas [34].

## 5. Discusión

Los resultados sugieren que la arquitectura de microservicios ofrece ventajas significativas sobre las arquitecturas monolíticas, permitiendo una mejor escalabilidad y mantenimiento. Asimismo, el uso de metodologías ágiles y la integración de requisitos arquitectónicos son cruciales para el éxito del desarrollo de software [5].

## 6. Conclusiones

La revisión de los 34 artículos muestra que los patrones de diseño y la arquitectura de software son componentes esenciales para la creación de sistemas eficientes y adaptables. La implementación de enfoques modernos y herramientas adecuadas puede facilitar el desarrollo y mejorar la calidad del software.

## 7. Agradecimientos

Agradezco a mi instructor, Jesús Ariel, por la revisión de los artículos, así como por su apoyo y orientación en la elaboración de este trabajo.

## Referencias

- [1] Moreno, A. M., & Sánchez-Segura, M. (2003, November). Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el Momento Arquitectónico. In JISBD (pp. 117-126).
- [2] Romero, P. Á. (2006). Arquitectura de software, esquemas y servicios. *Ingeniería Industrial*, 27(1), 1.
- [3] Cristiá, M. (2008). Introducción a la Arquitectura de Software. Research-Gate. Recuperado de: <https://www.researchgate.net/publication/251932352>
- [4] Bastarrica, M. C. (2005). Atributos de Calidad y Arquitectura del Software.
- [5] Navarro, M. E., Moreno, M. P., Aranda, J., Parra, L., Rueda, J. R., & Pantano, J. C. (2017, September). Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles. In XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires).
- [6] Guerrero, C. A., Suárez, J. M., & Gutiérrez, L. E. (2013). Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. *Información tecnológica*, 24(3), 103-114.
- [7] Alvarez, O. D. G., Larrea, N. P. L., & Valencia, M. V. R. (2022). Análisis comparativo de Patrones de Diseño de Software. *Polo del Conocimiento: Revista científico-profesional*, 7(7), 2146-2165.
- [8] Casas, M. R. H. (2020). Revisión sistemática sobre generadores de código fuente y patrones de arquitectura (Master's thesis, Pontificia Universidad Católica del Perú).

- [9] Cristiá, M. (2021). Una Teoría para el Diseño de Software.
- [10] Garis, A. G., Riesco, D. E., & Montejano, G. A. (2006). Perfiles UML para definición de Patrones de Diseño. In VIII Workshop de Investigadores en Ciencias de la Computación.
- [11] Jimenez-Torres, V. H., Tello-Borja, W., & Rios-Patiño, J. I. (2014). Lenguajes de patrones de arquitectura de software: una aproximación al estado del arte. *Scientia et technica*, 19(4), 371-376.
- [12] Campo, W. Y., Chanchí, G. E., & Arciniegas, J. L. (2013). Arquitectura de software para el soporte de comunidades académicas virtuales en ambientes de televisión digital interactiva. *Formación universitaria*, 6(2), 03-14.
- [13] Cárdenas-Gutiérrez, J. A., Barrientos-Monsalve, E. J., & Molina-Salazar, L. (2022). Arquitectura de software para el desarrollo de herramienta Tecnológica de Costos, Presupuestos y Programación de obra. *I+ D Revista de Investigaciones*, 17(1), 89-100.
- [14] Blas, M. J., Leone, H. P., & Gonnet, S. M. (2019). Modelado y Verificación de Patrones de Diseño de Arquitectura de Software para Entornos de Computación en la Nube.
- [15] Cambarieri, M., Difabio, F., & García Martínez, N. (2020). Implementación de una arquitectura de software guiada por el dominio. In XXI Simposio Argentino de Ingeniería de Software (ASSE 2020)-JAIIO 49 (Modalidad virtual).
- [16] Rodríguez Peña, A. D., & Silva Rojas, L. G. (2016). Arquitectura de software para el sistema de visualización médica Vismedic. *Revista Cubana de Informática Médica*, 8(1), 75-86.
- [17] Lazo, L., & Paul, J. (2004). Desarrollo de sistemas de software con patrones de diseño orientado a objetos.
- [18] Navarro, M. E., Moreno, M. P., Aranda, J., Parra, L., & Rueda, J. R. (2018). Arquitectura de software en el proceso de desarrollo ágil: una perspectiva basada en requisitos significantes para la arquitectura. In XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste).
- [19] Ferrandis Homsí, A. (2021). Desarrollo de una herramienta para el aprendizaje de patrones de diseño software (Doctoral dissertation, Universitat Politècnica de València).
- [20] Escalante, L. C. (2014). Aplicación de patrones de diseño para garantizar alta flexibilidad en el software. *Tecnología y Desarrollo (Trujillo)*, 12(1), 77-82.
- [21] Benigni, G., Antonelli, O., & Vásquez, Y. (2009). TOOL FOR REUSE OF JAVASCRIPT CODE ORIENTED TO INTERACTION PATTERNS. *SABER. Multidisciplinary Journal of the Research Council of the University of the East*, 21(1), 60-69.
- [22] López, D., & Maya, E. (2017). Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web.
- [23] García, J. M., Gil, Á. E., & Sánchez, E. A. (2018). Development of a software architecture for the Lázaro mobile robot. *Ingeniare. Chilean Journal of Engineering*, 26(3), 376-390.
- [24] Cardacci, D. G. (2015). Arquitectura de software académica para la comprensión del desarrollo de software en capas (No. 574). *Serie Documentos de Trabajo*.
- [25] Alfonso Azcuy, R., & Llull Céspedes, L. Á. (2021). Módulo de recomendación de patrones de diseño para EGPAT. *Revista Cubana de Ciencias Informáticas*, 15(2), 118-137.
- [26] Murillo, M. M. (2019). Arquitectura software reutilizable basada en patrones de diseño y patrones de interacción para el desarrollo rápido de aplicaciones web.
- [27] Rojas, M., & Montilva, J. (2011). Una arquitectura de software para la integración de objetos

- de aprendizaje basada en servicios web. In Ninth LACCEI Latin American and Caribbean Conference. Engineering for a Smart Planet, Innovation, Information Technology and Computational Tools for Sustainable Development.
- [28] Ortega, G. A. V. (2021). Lineamientos para el diseño de aplicaciones web soportados en patrones GRASP. *Ciencia e Ingeniería: Revista de investigación interdisciplinar en biodiversidad y desarrollo sostenible, ciencia, tecnología e innovación y procesos productivos industriales*, 8(2), 4.
  - [29] Meaurio, V. S., & Schmieder, E. (2013). La arquitectura de software en el proceso de desarrollo: integrando MDA al ciclo de vida en espiral. *Archivo de la Revista Latinoamericana de Ingeniería de Software*, 1(4), 142-146.
  - [30] Kruchten, P. (1995). Planos Arquitectónicos: El Modelo de 4+1 Vistas de la Arquitectura del Software. *IEEE software*, 12(6), 42-50.
  - [31] Contreras, T. B. Introducción a la Reingeniería de Software Mediante Patrones de Diseño.
  - [32] Pacheco, A., Escobar, J., & Trujillo, E. Uso de patrones de diseño y metaprogramación para construir APIs de IoT usando C++.
  - [33] Villalobos, G. M., Sánchez, G. D. C., & Gutiérrez, D. A. B. (2010). Diseño de framework web para el desarrollo dinámico de aplicaciones. *Scientia et technica*, 16(44), 178-183.
  - [34] Castaño, L. E. G., Soto, S. V. M., & Maturana, G. V. Coffee Challenge: un juego para el aprendizaje de patrones de diseño de software.