# Enfoques y Herramientas para el Diseño y Mejora de Arquitecturas de Software en Aplicaciones Web y Otros Contextos Tecnológicos

Mayra Alejandra Tamayo Perdomo
Neiva, Huila
mairatp2005@gmail.com
Centro de la Industria, la Empresa y los Servicios

7 de diciembre de 2024

## Resumen

Este artículo ofrece una revisión exhaustiva de 30 estudios enfocados en patrones de diseño y arquitecturas de software. Se analizan las tendencias contemporáneas y se subraya la relevancia de aspectos como la usabilidad, flexibilidad y adaptabilidad en el desarrollo de software. A través de diferentes enfoques y metodologías, se destaca la necesidad de herramientas que faciliten la implementación de arquitecturas eficientes y efectivas.

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

## 1. Introducción

La arquitectura de software es fundamental en el desarrollo de aplicaciones, ya que determina la estructura y organización del sistema, influyendo en su implementación y sostenibilidad. Este artículo revisa un conjunto de 34 investigaciones que analizan diversas perspectivas sobre cómo los patrones de diseño impactan en la arquitectura de software, especialmente en el contexto de aplicaciones web y entornos de computación en la nube.

## 2. Revisión de la Literatura

La literatura actual refleja un interés creciente en integrar la usabilidad desde las etapas iniciales del diseño de software [**?**]. La arquitectura de software, entendida como la disposición de sus componentes, es crucial para mantener un bajo acoplamiento y una alta cohesión [**?**]. Además, se observa la imperiosa necesidad de adaptarse a cambios constantes en los requisitos empresariales [**?**], lo que resalta la importancia de patrones de diseño flexibles y escalables.

## 3. Metodología

Se llevó a cabo una revisión sistemática de literatura, recopilando y analizando artículos relevantes sobre patrones de diseño y arquitectura de software. Los estudios se clasificaron según sus enfoques, contribuciones y la evolución de las metodologías empleadas en el ámbito del desarrollo de software.

## 4. Resultados

A continuación, se resumirán los hallazgos y aportes clave de los 34 artículos revisados, destacando las contribuciones más significativas en el campo de la arquitectura de software.

## 5. Conclusiones

La recopilación de estudios presentados en este documento pone de manifiesto la importancia de los patrones de diseño y la arquitectura de software en el desarrollo de aplicaciones modernas. A medida que el entorno tecnológico avanza, es esencial que los desarrolladores se mantengan actualizados sobre las mejores prácticas y enfoques que optimicen la calidad y eficiencia del software.

## 6. Modelado y Verificación de Patrones de Diseño de Arquitectura de Software para Entornos de Computación en la Nube

Este artículo presenta un enfoque integral para diseñar arquitecturas de software en aplicaciones web,

enfatizando la importancia de patrones genéricos que proporcionan estructura. Se introduce un metamodelo de componentes arquitectónicos y una herramienta gráfica para su implementación. La verificación de patrones es clave para asegurar su correcta aplicación, contribuyendo a la calidad del software. Se utiliza UML para describir aspectos arquitectónicos, destacando la experiencia del arquitecto en la selección de instancias adecuadas para cumplir con los requisitos funcionales.

**Referencia:** Blas, M. J., Leone, H., & Gonnet, S. (2019). Modelado y verificación de patrones de diseño de arquitectura de software para entornos de computación en la nube. RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação, 35, 1-17. https://doi.org/10.17013/risti.35.1-17

# 7. Desarrollo de una herramienta para el aprendizaje de patrones de diseño software

Se desarrolla una herramienta orientada a estudiantes para facilitar el aprendizaje de patrones de diseño de software. La aplicación se centra en patrones clave y busca mejorar la comprensión de conceptos complejos. Se emplean diversas tecnologías para su creación y se realizan encuestas para identificar dificultades en el aprendizaje. La herramienta tiene como objetivo preparar a los estudiantes para el mercado laboral, integrando teoría y práctica de manera interactiva.

**Referencia:** Ferrandis, (2021). Desarrollo de una herramienta para el aprendizaje de patrones de diseño software. RIUNET. https://riunet.upv.es/bitstream/handle/10251/174122/Ferrandis%20-%20Desarrollo%20de%20una%20herramienta%20para%20el%20aprendizaje%20de%20patrones%20de%20diseno%20software.pdf?sequence=1&isAllowed=y

# 8. Patrones de diseño de software aplicado a las aplicaciones web

El documento revisa patrones de diseño aplicados al desarrollo de aplicaciones web, resaltando la complejidad y expectativas de los usuarios. Se establece la Ingeniería Web como subdisciplina de la Ingeniería de Software. A través de un caso de estudio, se evidencian deficiencias por la ausencia de patrones de diseño. Se propone el uso de patrones como MVC para mejorar la funcionalidad y estructura del software, subrayando la importancia de buenas prácticas en el desarrollo.

**Referencia:** Universidad Señor de Sipán (USS). (2020). Patrones de diseño de software aplicado a las aplicaciones web. Repositorio USS. https://repositorio.uss.edu.pe/handle/20.500.12802/6783

# 9. Desarrollo de una arquitectura de software para el robot móvil Lázaro

Este artículo presenta una arquitectura de software diseñada para optimizar el control del robot móvil Lázaro. La arquitectura incluye tres niveles que gestionan actuadores y sensores, utilizando C y comunicación remota a través de módulos XBee®. Se abordan ventajas de arquitecturas deliberativas y reactivas, además de herramientas de inteligencia artificial para mejorar la navegación. La evaluación muestra resultados positivos en flexibilidad y facilidad de uso, aunque se identifican áreas de mejora.

**Referencia:** Scielo. (2018). Desarrollo de una arquitectura de software para el robot móvil Lázaro. Revista de Ingeniería. https://www.scielo.cl/scielo.php?pid=S0718-33052018000300376&script=sci_arttext

# 10. Arquitectura de software, esquemas y servicios

Se discute la arquitectura orientada a servicios (SOA) y su relevancia en aplicaciones empresariales, destacando su capacidad para facilitar la integración de sistemas. Se identifican desafíos en la gestión de conexiones y la necesidad de diseñar servicios autónomos. El documento aboga por soluciones estandarizadas y reusabilidad entre componentes, además de propuestas para gestionar dependencias en sistemas distribuidos, lo que podría mejorar la productividad en el desarrollo de software.

**Referencia:** Uniroja. (2006). Arquitectura de software, esquemas y servicios. Revista Dialnet. https://dialnet.unirioja.es/servlet/articulo?codigo=4786655

## 11. Arquitectura de software para el desarrollo de herramienta Tecnológica de Costos, Presupuestos y Programación de obra

Este artículo se centra en un software académico para la gestión de costos en Ingeniería Civil. Se organiza información sobre materiales y procesos, buscando aumentar la productividad. Se implementan metodologías como la Estructura de División del Trabajo (EDT). La herramienta es gratuita y accesible, con el fin de modernizar la enseñanza y preparar mejor a los estudiantes para el mercado laboral.

**Referencia:** Redalyc.org. (2022). Arquitectura de software para el desarrollo de herramienta tecnológica de costos, presupuestos y programación de obra. Revista Tecnológica. https://www.redalyc.org/journal/5337/533774788007/533774788007.pdf

## 12. Lenguajes de Patrones de Arquitectura de Software: Una Aproximación Al Estado del Arte

El artículo examina el estado de los lenguajes de patrones en arquitectura de software, analizando su evolución y aplicaciones. Se menciona la necesidad de un enfoque arquitectónico desde los años 60 para mejorar la calidad del software. Se categorizaron patrones y se presentaron ejemplos de aplicación en áreas como gestión de identidades. Se concluye que estos lenguajes son herramientas valiosas para resolver problemas arquitectónicos.

**Referencia:** Redalyc.org. (2014). Lenguajes de patrones de arquitectura de software: Una aproximación al estado del arte. Redalyc. https://www.redalyc.org/pdf/849/84933912003.pdf

## 13. Implementación de una Arquitectura de Software guiada por el Dominio

Se presenta un enfoque de arquitecturas guiadas por el dominio, enfatizando el diseño dirigido por el dominio (DDD). Propone la transformación de arquitecturas típicas en arquitecturas hexagonales, separando la lógica técnica de la complejidad del negocio. Se discuten conceptos como contextos delimitados y lenguaje ubicuo, validando el enfoque con un caso de estudio. Se concluye que esta transformación mejora la mantenibilidad y escalabilidad del software.

**Referencia:** Universidad Nacional de La Plata (UNLP). (2022). Implementación de una arquitectura de software guiada por el dominio. Repositorio UNLP. https://sedici.unlp.edu.ar/bitstream/handle/10915/115198/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

## 14. Arquitectura de software basada en microservicios para desarrollo de aplicaciones web

Este artículo aborda la transición de una arquitectura monolítica a una basada en microservicios en la CGTIC de Ecuador. Identifica tecnologías y metodologías para facilitar la transición, destacando la flexibilidad y autonomía de los microservicios. Se discuten las ventajas y desafíos de esta arquitectura, así como la importancia de servicios web para la integración. Se enfatiza la necesidad de adoptar arquitecturas modernas para mejorar la calidad del desarrollo.

**Referencia:** (2019). Arquitectura de software basada en microservicios para desarrollo de aplicaciones web. Repositorio Institucional. https://acortar.link/6oUGHO

## 15. Análisis comparativo de Patrones de Diseño de Software

Se analiza la importancia de los patrones de diseño como soluciones estandarizadas en el desarrollo de software. Se evalúan cinco patrones clave y su aplicabilidad en diferentes contextos. La investigación revela que no hay un patrón superior, cada uno cumpliendo un propósito específico. Se concluye que los patrones son fundamentales para mejorar la organización y calidad del código, ofreciendo un recurso valioso para desarrolladores.

**Referencia:** Danilo, O., Patricia, N., & Vinicio, M. (2022). Análisis comparativo de patrones de diseño de software. Polo del Conocimiento: Revista Científico-Profesional, 7(7), 2146–2165. https://dialnet.unirioja.es/descarga/articulo/9042927.pdf

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

El estudio examina la aplicación de patrones de diseño GoF en proyectos de desarrollo de software. Se identifican patrones creacionales y estructurales utilizados en la mayoría de los proyectos. A pesar de su uso, se destaca la falta de conocimiento que limita su implementación. Se propone una metodología para ayudar a identificar estos patrones y se recomienda crear un catálogo accesible para su comprensión.

**Referencia:** Scielo. (2021). Patrones de diseño GOF (The Gang of Four) en el contexto de procesos de desarrollo de aplicaciones orientadas a la web. Revista INFOTEC. https://www.scielo.cl/pdf/infotec/v24n3/art12.pdf

## 17. Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el momento Arquitectónico

El artículo analiza la relación entre arquitectura de software y usabilidad, proponiendo un método que incluye aspectos de usabilidad desde el inicio del desarrollo. Se presentan patrones de usabilidad para mejorar la eficiencia y satisfacción del usuario. Este enfoque busca disminuir cambios costosos en el futuro, haciendo el desarrollo más efectivo y centrado en el usuario.

**Referencia:** Moreno, A. (2024). Patrones de usabilidad: Mejora de la usabilidad del software desde el momento arquitectónico. ResearchGate. https://www.researchgate.net/publication/221595496_Patrones_de_Usabilidad

## 18. Revisión sistemática sobre generadores de código fuente y patrones de arquitectura

El estudio se centra en los generadores de código y su relación con patrones arquitectónicos, analizando su uso en desarrollo de software. Se evalúan las ventajas y desventajas de estas herramientas, ofreciendo una perspectiva de tendencias actuales. Se concluye que los generadores de código pueden ayudar a automatizar tareas repetitivas, mejorando la consistencia y precisión en proyectos grandes.

**Referencia:** ProQuest. (2024). Revisión sistemática sobre generadores de código fuente y patrones de arquitectura. ProQuest. https://acortar.link/CYBZpK

## 19. Perfiles UML para definición de Patrones de Diseño

El artículo explora cómo los perfiles UML ayudan a definir y documentar patrones de diseño en programación orientada a objetos. Se enfoca en patrones estructurales y propone que las herramientas UML existentes son suficientes para su uso. Se sugiere especificar más patrones en el futuro y buscar herramientas que simplifiquen esta tarea.

**Referencia:** Universidad Nacional de La Plata (UNLP). (2024). Perfiles UML para definición de Patrones de Diseño. Repositorio UNLP. https://acortar.link/0enQyk

## 20. Arquitectura de software académica para la comprensión del desarrollo de software en capas

El texto presenta un modelo académico basado en un enfoque de desarrollo en capas, destacando su relevancia para prevenir problemas. Se propone una estructura de tres capas y el uso del patrón MVC para gestionar interacciones. Se concluye que la adopción de arquitecturas bien estructuradas favorece sistemas flexibles y sostenibles.

**Referencia:** Google Scholar. (2024). Arquitectura de software académica para la comprensión del desarrollo de software en capas. Google Scholar. https://www.econstor.eu/bitstream/10419/130825/1/837816424.pdf

## 21. Introducción a los patrones de diseño

Este artículo contextualiza históricamente los patrones, organizándolos en creacionales, estructurales y de comportamiento. Se enfatiza la importancia de evitar la reinvención de soluciones, promoviendo el uso de patrones comprobados. Se incluyen ejemplos

prácticos y recursos adicionales, haciendo que el contenido sea accesible para principiantes y programadores experimentados.

**Referencia:** Academia.edu. (2019). Introducción a los patrones de diseño. Academia.edu. https://www.academia.edu/download/60132311/Introduccion-a-los-patrones-de-diseno20190727-56910-x3jn57.pdf

## 22. Identificación y clasificación de patrones de diseño de servicios web para mejorar QoS

Se investiga la identificación y clasificación de patrones de diseño que mejoran la Calidad de Servicio (QoS) en servicios web. Se presentan metodologías para evaluar la efectividad de estos patrones, analizando casos de estudio que evidencian mejoras en la QoS. La investigación concluye que la adecuada selección de patrones es crucial para optimizar el rendimiento de servicios web.

**Referencia:** Scielo. (2024). Identificación y clasificación de patrones de diseño de servicios web para mejorar QoS. Repositorio UNLP. https://sedici.unlp.edu.ar/handle/10915/94512

## 23. Desarrollo de sistemas de software con patrones de diseño orientado a objetos

Este artículo se centra en el desarrollo de software utilizando patrones de diseño orientado a objetos. Se discuten las ventajas de aplicar estos patrones para mejorar la modularidad y reutilización del código. Se presentan ejemplos prácticos que ilustran cómo los patrones facilitan el desarrollo y mantenimiento de sistemas complejos, promoviendo buenas prácticas en programación.

**Referencia:** "Desarrollo de sistemas de software con patrones de diseño orientado a objetos". (2024, 11 de noviembre). Cybertesis UNMSM. https://acortar.link/RX308S

## 24. Módulo de recomendación de patrones de diseño para EGPat

El documento desarrolla un módulo que recomienda patrones de diseño para el software EGPat. Se analizan las necesidades de los usuarios y se propone un sistema que facilite la selección de patrones adecuados según el contexto. Este enfoque busca mejorar la eficiencia en el desarrollo y garantizar la calidad del software mediante la aplicación de patrones relevantes.

**Referencia:** Scielo. (2024). Módulo de recomendación de patrones de diseño para EGPat. Revista Scielo. https://acortar.link/rhUG8Q

## 25. Arquitectura de software para el desarrollo de videojuegos sobre el motor de juego Unity 3D

Se presenta una arquitectura de software específica para el desarrollo de videojuegos en Unity 3D. Se discuten patrones de diseño que son particularmente útiles en este contexto, como el patrón de observador y el patrón de estado. Se concluye que una buena organización de la arquitectura puede facilitar la escalabilidad y mantenibilidad de los videojuegos desarrollados.

**Referencia:** Martínez, J. (2023). Arquitectura de software para el desarrollo de videojuegos sobre el motor de juego Unity 3D. Revista de Tecnología y Diseño. https://revistatecnologiaydiseo.com/unity3d

## 26.Revisión de elementos conceptuales para la representación de las arquitecturas de referencias de software

Este documento revisa conceptos clave para representar arquitecturas de referencia en software. Se subraya la importancia de tener representaciones claras que faciliten la comprensión y comunicación entre equipos. Se analizan diferentes enfoques y se proponen mejoras para la representación visual de arquitecturas, lo que podría beneficiar el desarrollo colaborativo.

*Scielo.* (2019). https://acortar.link/zeeKAc

## 27. Atributos de Calidad y Arquitectura de Software

El artículo examina la relación entre atributos de calidad y arquitectura de software. Se identifican factores que contribuyen a la calidad del software y se discuten métodos para evaluarlos. Se concluye que una buena arquitectura puede mejorar significativamente la calidad del software, y se proponen prácticas para su implementación efectiva.
*UPM.* (1905). http://www.grise.upm.es/rearviewmirror/conferencias/jiisic04/Tutoriales/tu4.pdf

## 28.Herramientas para reuso de código JavaScript orientado a partir de interacción

Este documento aborda herramientas que facilitan el reuso de código en JavaScript, centrándose en la interacción del usuario. Se analizan diversas herramientas y su impacto en la eficiencia del desarrollo. Se concluye que el reuso efectivo de código no solo ahorra tiempo, sino que también mejora la calidad y mantenibilidad de las aplicaciones web.
*Redalyc.* (2024). https://www.redalyc.org/pdf/4277/427739438009.pdf

## 29. Una Teoría para el Diseño de Software

Se presenta una teoría que busca establecer principios fundamentales para el diseño de software. Se discuten conceptos clave que deben ser considerados en el desarrollo, promoviendo un enfoque más estructurado. La teoría propone una serie de directrices que pueden ser aplicadas en diversos contextos de desarrollo, favoreciendo la calidad del software.
*UNR.* (2024). https://www.fceia.unr.edu.ar/ingsoft/intro-diseno.pdf

## 30. Lenguajes de patrones de diseño de software bajo una perspectiva cognoscitivista

El artículo investiga los lenguajes de patrones de diseño desde una perspectiva cognoscitiva. Se analizan cómo estos lenguajes pueden ayudar a los desarrolladores a entender y aplicar patrones de manera más efectiva. Se concluye que una mejor comprensión de los lenguajes de patrones puede facilitar el aprendizaje y la implementación de soluciones en el desarrollo de software.
*Redalyc.* (2023). https://www.redalyc.org/articulo.oa?id=66640712

## 31.Patrones de diseño para mejorar la accesibilidad y uso de aplicaciones sociales para adultos mayores

Este artículo propone patrones de diseño que mejoran la accesibilidad de aplicaciones sociales dirigidas a adultos mayores. Se analizan las necesidades específicas de este grupo demográfico y se presentan soluciones de diseño que facilitan el uso de tecnologías. Se concluye que aplicar estos patrones puede contribuir significativamente a mejorar la experiencia de usuario de las aplicaciones.
*Redalyc.* (2015). https://www.redalyc.org/articulo.oa?id=15839609009

## 32.Una arquitectura basada en software libre para archivos web

Se discute una arquitectura de software libre orientada a la gestión de archivos web. Se presentan las ventajas de usar software libre, como la flexibilidad y la personalización. El artículo enfatiza la importancia de construir una arquitectura que se adapte a las necesidades específicas de los usuarios, promoviendo la colaboración y el desarrollo comunitario.
*Redalyc.* (2013). https://www.redalyc.org/articulo.oa?id=82326270005

## 33. Arquitectura de Software para el Soporte de Comunidades Académicas Virtuales en Ambientes de Televisión Digital Interactiva

El documento presenta una arquitectura diseñada para soportar comunidades académicas en entornos de televisión digital interactiva. Se discuten las características necesarias para facilitar la interacción y el aprendizaje en estas comunidades. Se concluye que una arquitectura bien diseñada puede mejorar la

comunicación y el acceso a recursos educativos.
*Scielo.* (2024). https://www.scielo.cl/scielo.php?pid=S0718-50062013000200002&script=sci_arttext

## 34. Arquitectura de software de una aplicación móvil para desarrollar un sistema de identificación por radiofrecuencia

Este artículo aborda la arquitectura de software para una aplicación móvil enfocada en la identificación por radiofrecuencia. Se analizan los componentes necesarios para un funcionamiento eficiente y se discuten los desafíos técnicos. Se concluye que una arquitectura adecuada es crucial para garantizar el rendimiento y la usabilidad de la aplicación en entornos prácticos.
*Redalyc.* (2015). https://www.redalyc.org/articulo.oa?id=512251501004

## 35.Uso de patrones de diseño de software: un caso práctico

El artículo presenta un caso práctico que ilustra la aplicación de patrones de diseño en un proyecto real. Se analiza cómo la implementación de estos patrones mejoró la calidad del software y facilitó el mantenimiento. Se concluye que el uso sistemático de patrones puede transformar significativamente el desarrollo de software, haciendo que los proyectos sean más sostenibles.
*Redalyc.* (2012). https://www.redalyc.org/articulo.oa?id=44170527004

# Resultados

La recopilación de estudios subraya la relevancia continua de los patrones de diseño y la arquitectura de software en un entorno tecnológico en constante evolución, destacando la necesidad de mantener a los desarrolladores informados sobre mejores prácticas.