

# Optimización de la Arquitectura de Software en Entornos Ágiles y la Nube: Un Enfoque Integral para el Diseño y Selección de Patrones Arquitectónicos

Mariana González Calderón

<sup>1</sup>Servicio Nacional de Aprendizaje (SENA), Neiva-Huila, Colombia

CORRESPONDING AUTHOR: Mariana González Calderón (e-mail: mariana.gonzalez7@soy.sena.edu.co).

**ABSTRACT** This paper presents a detailed analysis of a comprehensive environment for the design and selection of software architectures oriented to web applications, with a special focus on cloud computing contexts and agile methodologies. The main objective is to provide a framework for architects and developers that facilitates the creation and verification of architectural design patterns, guaranteeing correct implementation and ensuring software quality. The article also addresses the importance of integrating patterns such as MVC, MVP, Microservices and Cloud Architecture, and how these can be selected according to the specific characteristics and requirements of the project. Through an agile methodology and a practical approach, graphical tools for the creation of flexible and adaptable architectures are explored. The results highlight how previous architectural decisions can guide the creation of more efficient and maintainable solutions, promoting productivity and ensuring software quality in dynamic environments. It is concluded that the combination of agile methodologies with well-defined design patterns is key to successful software development in technologically demanding environments.

**INDEX TERMS** Arquitectura de Software, Patrones de Diseño, Computación en la Nube, Metodologías Ágiles, Microservicios, MVC, Mantenimiento de Software

## BACKGROUND

La arquitectura de software es crucial en el desarrollo de aplicaciones, especialmente en entornos dinámicos como la computación en la nube y las metodologías ágiles. Un enfoque adecuado en la selección y el diseño de arquitecturas afecta directamente la calidad, el rendimiento y la mantenibilidad del software. Este artículo busca explorar cómo un metamodelo de componentes arquitectónicos puede ayudar a los arquitectos a enfrentar los desafíos del diseño y verificación de arquitecturas, especialmente en la nube, mediante el uso de patrones de diseño como MVC, MVP y Microservicios.

## COLLECTION METHODS AND DESIGN

La investigación se centró en la integración de patrones de diseño en la arquitectura de software utilizando metodologías ágiles. Se utilizó un metamodelo de componentes arquitectónicos que permite un enfoque flexible y adaptable. A través de la metodología ágil, las decisiones arquitectónicas fueron sometidas a validaciones periódicas con herramientas gráficas como Canva, lo que facilitó la colaboración entre arquitectos y desarrolladores en el diseño de arquitecturas.

## VALIDATION AND QUALITY

Se validaron las arquitecturas propuestas mediante un caso práctico de diseño de software en la nube. Los patrones seleccionados (MVC, Microservicios, etc.) fueron evaluados según su capacidad para generar soluciones escalables y mantenibles. Los equipos de desarrollo utilizaron herramientas gráficas para asegurar que las decisiones arquitectónicas cumplieran con los requisitos de calidad, como el rendimiento y la seguridad.

## RECORDS AND STORAGE

Los datos relacionados con los patrones de diseño arquitectónico, las herramientas utilizadas, y las arquitecturas implementadas se organizaron en archivos digitales. Estos archivos contienen detalles sobre las decisiones arquitectónicas, la selección de patrones y las configuraciones de las herramientas gráficas empleadas. Los archivos están almacenados en un repositorio accesible para los equipos de desarrollo.

## INSIGHTS AND NOTES

El uso de metodologías ágiles junto con patrones de diseño arquitectónicos ha demostrado ser efectivo para la adaptación

rápida a cambios durante el ciclo de vida del proyecto. Las herramientas gráficas contribuyeron significativamente a la validación y verificación de las decisiones arquitectónicas. Sin embargo, el estudio debe expandirse para pruebas a gran escala en entornos reales, lo cual es una limitación en cuanto a la generalización de los resultados.

#### **SOURCE CODE AND SCRIPTS**

El código fuente utilizado para simular escenarios de diseño de software está disponible en el repositorio GitHub: <https://github.com/Mari-anita/complementario-3125033>. También se utilizaron plataformas como Canva para la creación de diagramas, y su versión actual fue incorporada en el proceso de diseño.

#### **REFERENCES**

- 1) Blas, M., et al. (2019). "Modelado y Verificación de Patrones de Diseño de Arquitectura de Software para Entornos de Computación en la Nube."
- 2) ProQuest. (s. f.). "Marco de Trabajo para Seleccionar un Patrón Arquitectónico en el Desarrollo de Software."
- 3) Redalyc. (2008). "Lenguajes de Patrones de Arquitectura de Software: Una Aproximación al Estado del Arte."
- 4) Revista de Tecnología y Software. (2020). "Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles." 15(3), 45-60.