

Resumen de Artículos

Artículo 1: Guía para la descripción y documentación de arquitecturas de software utilizando lenguajes de descripción de arquitectura; este artículo es una tesis de la UNAM que presenta la guía para documentar arquitecturas de software, usando lenguajes de Descripción de Arquitectura (LDA). La autora propone esta guía basándose en el estándar ISO/IEC/IEEE 42010:2011, ya que actualmente no existen guías detalladas que faciliten su aplicación en la industria.

El objetivo es mejorar la documentación de la arquitectura de Software, que debido a una buena documentación se facilitara la comunicación entre los interesados, el análisis del sistema y su mantenimiento.

Artículo 2: Metodología de desarrollo de Software y documentación de procesos en IPSOFT. S.A.

El objetivo principal de este artículo es identificar, describir, mejorar e implementar procesos de desarrollo de software que permitan reducir la desorganización en los desarrollos nuevos productos o aplicaciones en IPSOFT, S.A. La metodología propuesta busca ser adaptable de distintos tamaños y complejidades.

Artículo 3º Estrategia Para la estandarización de la documentación y las actividades para el desarrollo de software en la universidad de las ciencias informáticas. Este artículo representa una estrategia para la estandarización de la documentación en el entorno del software de la universidad de las Ciencias Informáticas (UCI), la motivación detrás de la profunda investigación de este artículo surge de la necesidad de mejorar la calidad y la organización en los proyectos de software de la UCI, que busca posicionarse en el mercado global.

Artículo 4º Propuesta de Metodología y Documentación en el desarrollo de Software.

El artículo es una tesis de grado que presenta una propuesta de Metodología y documentación Para el desarrollo de Software. Se enfoca en la necesidad de combinar las Ventajas de metodologías tradicionales de análisis y diseño estructurado con las metodologías ógiles actuales, Para mejorar la eficiencia y calidad en el desarrollo de software.

Artículo 5º Documentación y adopción de buenas prácticas para el desarrollo de software basado en los fundamentos Devops

El Artículo se enfoca en la implementación de DevOps como una metodología para mejorar los procesos de desarrollo de software en las empresas. Se exploran las buenas prácticas y herramientas necesarias para adoptar Devops. Con el objetivo de aumentar la colaboración entre los equipos de desarrollo y operaciones, la propuesta incluye la creación de manuales de procesos y una evaluación para determinar la preparación de las empresas para Devops, haciendo énfasis en la importancia de la documentación y la automatización en cada etapa del ciclo del software.

Artículo 6º Principios básicos para documentar la arquitectura de software.

Este artículo se basa en el desarrollo de sistemas en la arquitectura de software. Principalmente se enfoca en el modelo 4 + 1 Vista de Kruchten las cuales son Vista lógica: representa los objetos que será necesario modelar, Vista de desarrollo: esta muestra la organización del software, Vista de procesos: representa como se ejecutan sus funciones y como se manejan los recursos.

Vista física: Muestra los dispositivos en los que se ejecuta el sistema.

Vista de escenarios: Muestra como interactúan las otras vistas en situaciones de uso.

Artículo 7º: El artículo resalta como una documentación bien estructurada y detallada puede mejorar la comunicación entre los interesados, el mantenimiento del sistema y la evaluación de sus cualidades, también menciona que aunque existen muchos LDAS, su adopción en la industria es limitada debido a su complejidad y la falta de guías prácticas.

Artículo 8º: Formalización del proceso de elaboración de una arquitectura de software.

Es una tesis que tiene como objetivo principal la formalización del proceso de documentación de arquitecturas de software, específicamente en el contexto de las empresas virtuales asociadas a la Universidad Peruana de Ciencias Aplicadas, los autores buscan mejorar la calidad y eficiencia de los proyectos de software a través de la implementación de metodologías y herramientas de documentación basadas en prácticas reconocidas a nivel internacional.

Artículo 9: Desarrollo de una guía metodológica para mejorar la documentación del capítulo técnico de los trabajos de grado de la carrera de Software que utilicen la metodología Scrum.

El propósito central de este artículo es abordar la problemática de la falta de estandarización en la documentación técnica de los proyectos de titulación que se desarrollan bajo la metodología Scrum, lo cual es común en la carrera de Software de la Universidad. La guía propuesta tiene como objetivo ayudar a los estudiantes a organizar y estructurar mejor la documentación de sus proyectos, facilitando su compresión y aumentando la calidad del trabajo final.

Artículo 10: Importancia de la documentación de la arquitectura de software para el desarrollo de sistemas: Caso práctico Sistema de conciliación de Indeval.

El artículo se enfoca en demostrar la importancia de la documentación de la arquitectura de software en el desarrollo de sistemas complejos. A través del análisis del sistema de conciliación de Indeval, la tesis ilustra cómo una arquitectura bien documentada puede mejorar la comunicación entre los stakeholders, asegurar que se cumplan los requisitos del sistema, y facilitar tanto su desarrollo como su

mantenimiento. La investigación explora diversos enfoques y modelos de documentación, subrayando su aplicabilidad en ~~8%~~ proyectos reales dentro del sector financiero.

Artículo 11º: Una herramienta para asistir la documentación de arquitecturas de software.

Este artículo aborda la importancia de documentar la arquitectura de Software para el éxito de proyectos de desarrollo. La documentación de la arquitectura, conocida como SAD (Software Architecture Document), organiza la documentación clara sobre la estructura y decisiones de diseño del sistema, adaptándose a las necesidades de los diferentes stakeholders.

Artículo 12º: Software Architecture Documentation: the road ahead.

Este artículo discute las direcciones futuras de la documentación de arquitecturas de Software, haciendo referencia en la importancia de herramientas automatizadas y métodos formales para mejorar la precisión y eficiencia de la documentación.

Artículo 13: Evaluating the Documentation of Software Architectures: A Systematic Review

Este estudio revisa sistemáticamente los métodos para evaluar la documentación de arquitecturas de Software, proporcionando una visión sobre las mejores prácticas y los desafíos comunes en este campo.

Artículo 14: Architectural styles in Software Architecture Documentation.

Este artículo explica cómo la elección de un estilo arquitectónico afecta la manera en que se estructura y presenta la documentación. Se destacan los beneficios de seguir un enfoque estandarizado, donde los estilos arquitectónicos guían la organización y la claridad de la documentación, facilitando la comunicación entre los miembros del equipo y otros interesados.

Artículo 15: An ontology-Based Approach to Software Architecture Documentation.

Este artículo propone el uso de ontologías como una herramienta para mejorar la consistencia y la exhaustividad en la documentación de arquitecturas de Software. Las ontologías permiten una representación formal de los conceptos y las relaciones dentro de un sistema, lo que facilita la generación de documentación que es tanto comprensible como estructurada.

el artículo discute cómo las ontologías pueden integrarse en el proceso de desarrollo de software.

Para garantizar que todos los aspectos clave de la arquitectura estén adecuadamente documentados.

Artículo 16: Automated Support for Architecture Documentation. Este artículo examina diversas herramientas y técnicas que permitan la automatización del proceso de documentación arquitectónica. Se enfoca en cómo la automatización puede reducir el esfuerzo manual requerido para mantener la documentación actualizada y precisa. Entre las herramientas se incluyen generadores de documentación basados en modelos y herramientas de análisis estético que extraen información directamente del código fuente. Se discuten también los desafíos asociados con la automatización, como la necesidad de personalización y la integración con flujos de trabajo existentes.

Artículo 17: Challenges in Documenting Software Architectures.

Este artículo identifica y analiza los principales desafíos que enfrentan los equipos al documentar arquitecturas de software, especialmente en sistemas grandes y complejos. Entre los desafíos mencionados se encuentran la dificultad para mantener la documentación, arquitectónica

y la falta de estándares claros para lo que debe incluirse en la documentación arquitectónica. El artículo propone varias estrategias para superar estos desafíos, incluyendo la incorporación de prácticas ágiles y el uso de herramientas colaborativas.

Artículo 18: the role of documentation in Software Architecture Recovery.

Este artículo explora la importancia de la documentación, en la recuperación de arquitecturas de software, especialmente en sistemas heredados donde la arquitectura original puede haberse perdido o distorsionado con el tiempo. La documentación existente, incluso si está incompleta o desactualizada, puede ser una valiosa fuente de información para comprender y reconstruir la arquitectura de un sistema, también se nos habla de métodos para validar y actualizar la documentación durante el proceso de recuperación arquitectónica.

Artículo 19: Software Architecture Documentation.

Practices and Problems.

Este artículo revisa las prácticas comunes en la documentación de arquitecturas y destaca los problemas recurrentes que los equipos enfrentan. Se mencionan los errores comunes como la sobre-documentación, la falta de mantenimiento, y la dependencia excesiva

en un solo tipo de vista arquitectónica. El artículo sugiere mejores prácticas para evitar estos problemas como la utilización de herramientas de documentación colaborativo y la definición clara de los objetivos y el alcance de la documentación desde inicio del proyecto.

Artículo 20: A Framework for Documenting Product-Line Architectures.

Este es un estudio que propone un marco estructurado para documentar arquitecturas de líneas de productos de software, un enfoque que permite gestionar variabilidades y reusabilidad en el desarrollo de productos relacionados. El marco incluye recomendaciones sobre qué aspectos de la arquitectura deben documentarse, cómo gestionar, y cómo estructurar la documentación, para facilitar la reutilización en diferentes productos dentro de la línea.

Artículo 21: Improving Software Architecture Documentation with Viewpoints.

Artículo 21: Improving Software Architecture Documentation Viewpoints

Este artículo presenta el enfoque de 'Viewpoints' o Puntos de Vista como una manera de mejorar la claridad y usabilidad de la documentación arquitectónica. Cada Viewpoint aborda una preocupación específica, como la seguridad, el rendimiento, o la escalabilidad, proporcionando una representación clara y detallada de cómo la arquitectura aborda estos preocupaciones. El uso de Viewpoints permite a los arquitectos de software segmentar la documentación de manera que sea más accesible y relevante para diferentes audiencias.

Artículo 22: Documenting Software Architecture for Agile Projects

Este artículo explora los desafíos y estrategias para documentar arquitecturas de software en entornos ágiles, donde la documentación suele ser vista como secundaria frente a la entrega rápida de funcionalidades. Se discuten enfoques para equilibrar la necesidad de documentación con la agilidad, tales como la documentación evolutiva, que se actualiza a medida que el sistema crece y cambia, y la incorporación de documentación ligera, para poder agilizar

el proceso.

Artículo 23: Consistency checking in software Architecture Documentation

El artículo propone el uso de técnicas automáticas y semi-automáticas para verificar que la documentación esté alineada con el código y que diferentes partes de la documentación no se contradigan entre sí. Esto es especialmente importante en proyectos grandes donde múltiples equipos trabajan en diferentes partes del sistema.

Artículo 24: The Evolution of Software Architecture Documentation Practices.

Este artículo rastrea la evolución de las prácticas de documentación en arquitecturas de software durante las últimas dos décadas, destacando cómo han cambiado las herramientas y metodologías en respuesta a la evolución de la industria. Se examinan los cambios en la documentación debido a la adopción de metodologías ágiles, la introducción de nuevas herramientas de modelado, y la creciente importancia de la automatización.

Artículo 25: A case study on documentation in Microservices Architecture.

El estudio presenta un caso práctico sobre la documentación de la arquitectura de un Sistema basado en micro servicios; el artículo sugiere mejores prácticas para documentar sistemas de micro servicios, incluyendo la automatización de la generación de documentación y la integración de la documentación con herramientas de monitoreo y despliegue.

Artículo 26: Guidelines for effective Software Architecture Documentation.

Este artículo proporciona un conjunto de directrices destinadas a mejorar la efectividad de la documentación de arquitecturas de software. Las directrices incluyen recomendaciones sobre cómo estructurar la documentación, qué niveles de detalle son apropiados para diferentes audiencias, y cómo mantener la documentación alineada con la arquitectura actual del sistema. También se discuten los beneficios de la documentación continua, que se actualiza a medida que el sistema evoluciona, en lugar de ser creada al final del ciclo del desarrollo.

Artículo 27: Using UML Software Architecture Documentation

El artículo discute cómo UML (lenguaje de Modelado Unificado) puede ser utilizado de manera efectiva para documentar arquitecturas de software. Se explica cómo diferentes diagramas UML, como diagramas de clase, de secuencia, y despliegue, pueden capturar aspectos clave de la arquitectura y cómo pueden integrarse para proporcionar una visión completa del sistema. También se discuten las limitaciones de UML y cómo pueden superarse.

Artículo 28: Documenting Software Architectures

Este artículo enfatiza la importancia de la documentación en la arquitectura de software, abordando cómo utilizar estilos bien conocidos como SOA, cliente-servidor, y modelo multicapa para representar elementos arquitectónicos. Proporciona directrices y plantillas para documentar arquitecturas de manera que sean comprensibles para todos los interesados, basándose en modelos como RUP y UML.

Artículo 29: ontology-based Software Architecture Documentation.

Este artículo propone el uso de ontologías para mejorar la documentación de la arquitectura de software. Presenta un enfoque que facilita la recuperación del conocimiento arquitectónico mediante la organización estructurada de la documentación, lo que mejora la gestión y reutilización de la información.

Artículo 30: How Organization of Architecture Documentation Affects Architectural Knowledge Retrieval

Este estudio del artículo investiga la forma en que se organiza la documentación arquitectónica afecta la recuperación de conocimiento. Ofrece estrategias para estructurar la documentación de manera que mejore la usabilidad y eficiencia en la recuperación de información arquitectónica, abordando los problemas comunes en la organización de grandes volúmenes de datos.

Artículo 31: Enriching Software Architecture Documentation.

Este artículo presenta métodos para mejorar la calidad de la documentación de la arquitectura de software. Propone la integración de detalles

adicionales que no solo clarifiquen las decisiones arquitectónicas, sino que también hagan que la documentación sea más valiosa para los futuros mantenedores del sistema.

Artículo 32: Empirical Study of Architectural Knowledge Management Practices.

Este es un estudio empírico que analiza cómo las prácticas de gestión del conocimiento arquitectónico influyen en el éxito de los proyectos de Software. Enfatiza la importancia de la documentación como un medio para mitigar los desafíos de la comunicación y garantizar la continuidad del conocimiento en todo el ciclo de vida del software.

Artículo 33: Applying Bayesian Analysis Guidelines to Empirical Software Engineering Data

Este artículo se centra en el análisis bayesiano. Pero también explora cómo los datos empíricos de la ingeniería de software, incluida la documentación arquitectónica, pueden ser medidos, evaluados y mejorados mediante directrices analíticas avanzadas.

Artículo 34: Experimental Analysis of Textual and Graphical Representations for Software Architecture Design.

Este artículo compara la eficacia de las representaciones textuales y gráficas en la documentación y diseño de la arquitectura de software, concluyendo que una combinación de ambas mejora la comprensión y la comunicación de las decisiones arquitectónicas.

Artículo 35: Software Engineering Whispers? the Effect of Textual vs Graphical Software Design
Este artículo explora cómo las descripciones textuales y gráficas afectan la transmisión de información en el diseño de software, subrayando la importancia de seleccionar el formato adecuado para la documentación en función del público y el contexto.

Artículo 36: the 4+1 View Model of architecture
Este artículo introduce el modelo 4+1 de vistas, que ofrece un marco para documentar arquitecturas de software mediante la combinación de diferentes perspectivas: lógica, de desarrollo, de procesos, física y de escenarios. Este modelo ayuda a capturar

la arquitectura desde varios ángulos, asegurando que todas las partes interesadas puedan comprender los aspectos relevantes del sistema.

Artículo 37: Disseminating Architectural Knowledge on Open-Source Projects.

Este artículo analiza cómo se distribuye el conocimiento arquitectónico en proyectos de código abierto, utilizando como ejemplo "Architecture of Open Source Applications". Se enfoca en los desafíos y las mejores prácticas para documentar y compartir conocimiento en comunidades distribuidas de desarrollo.

Artículo 38: A Comprehensive Approach to Software Architecture Documentation

Este artículo propone un enfoque integral para la documentación de arquitecturas de software, integrando múltiples vistas y estilos para capturar completamente el diseño arquitectónico de un sistema complejo. Se enfoca en cómo la documentación puede ser adaptada para diferentes audiencias.

Artículo 39: Principles of Sound Documentation in Software Architecture.

Este artículo discute los principios fundamentales para una documentación sólida en la

arquitectura de software, como la calidad, precisión, y adoptabilidad a diferentes audiencias y escenarios. También resalta la importancia de la documentación no solo sea completa, sino también útil y fácil de mantener.

Artículo 40: Viewpoints And Views in Software Architecture Documentation

Este artículo explora el concepto de Puntos de Vista en la documentación arquitectónica, y cómo estos pueden ayudar a organizar y presentar la información de manera coherente y útil para diferentes tipos de stakeholders. El artículo destaca la importancia de adaptar la documentación a las necesidades de cada grupo de interés.