

1

Artículos de Investigación

Realizado por:
Marriana Cherry Prada

Jesús Ariel González Bonilla

Servicio Nacional de Aprendizaje
Centro de la Industria, la Empresa y los Servicios
Tecnólogo en Análisis y Desarrollo de Software

Nº de Ficha: 2694667

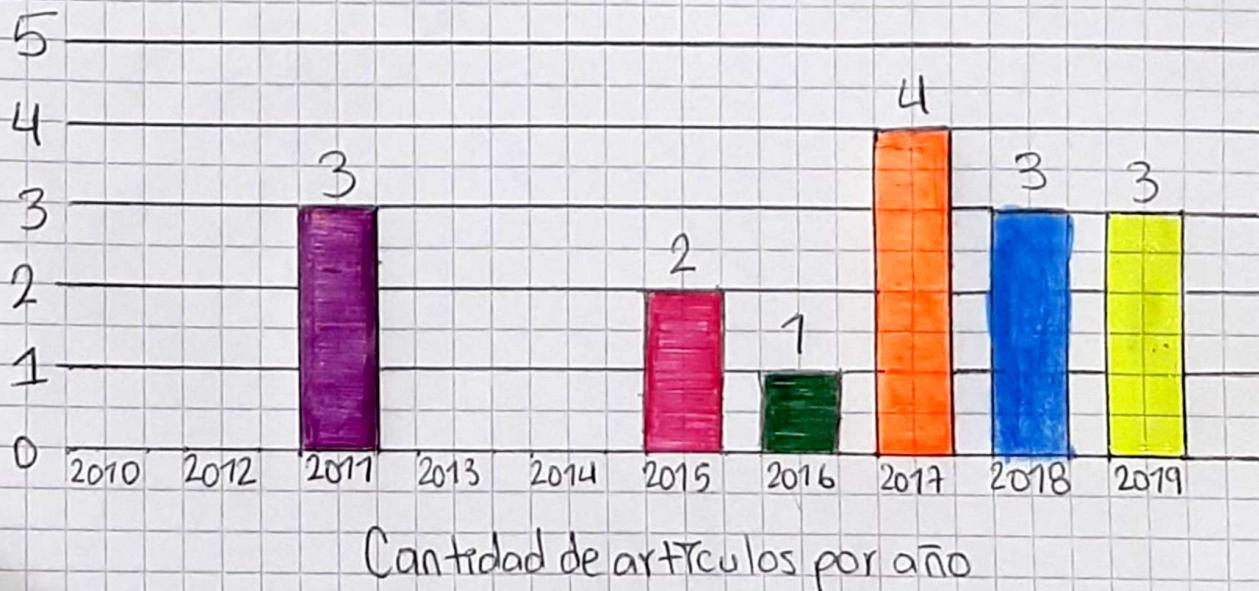
Neiva - Huila
2024

2

Análisis comparativo de patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad: Una revisión sistemática de la literatura

Resumen:

En el inmenso mundo de las aplicaciones móviles existen diversos patrones de diseño que nos brindan facilidades de crear un desarrollo más eficiente y organizado. El objetivo de la revisión es el de encontrar los principales estudios sobre patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad, y posteriormente determinar criterios de identificación que servirán como herramienta de selección de patrones de diseño de calidad. Para poder implementar la más conveniente según las necesidades del proyecto.



Reflexión:

Los patrones de diseño han mostrado un mayor impacto en el desarrollo de software. Dado que los patrones de diseño proporcionan soluciones comprobadas, el desarrollo de aplicaciones móviles puede ser un proceso tedioso, ya que cada aplicación debe pasar por los ciclos de desarrollo para garantizar que la aplicación se ajuste a los atributos de calidad estándar. Luego de que el desarrollador compare los principales patrones de diseño de software, finalmente podrá implementar la más conveniente para el proyecto.

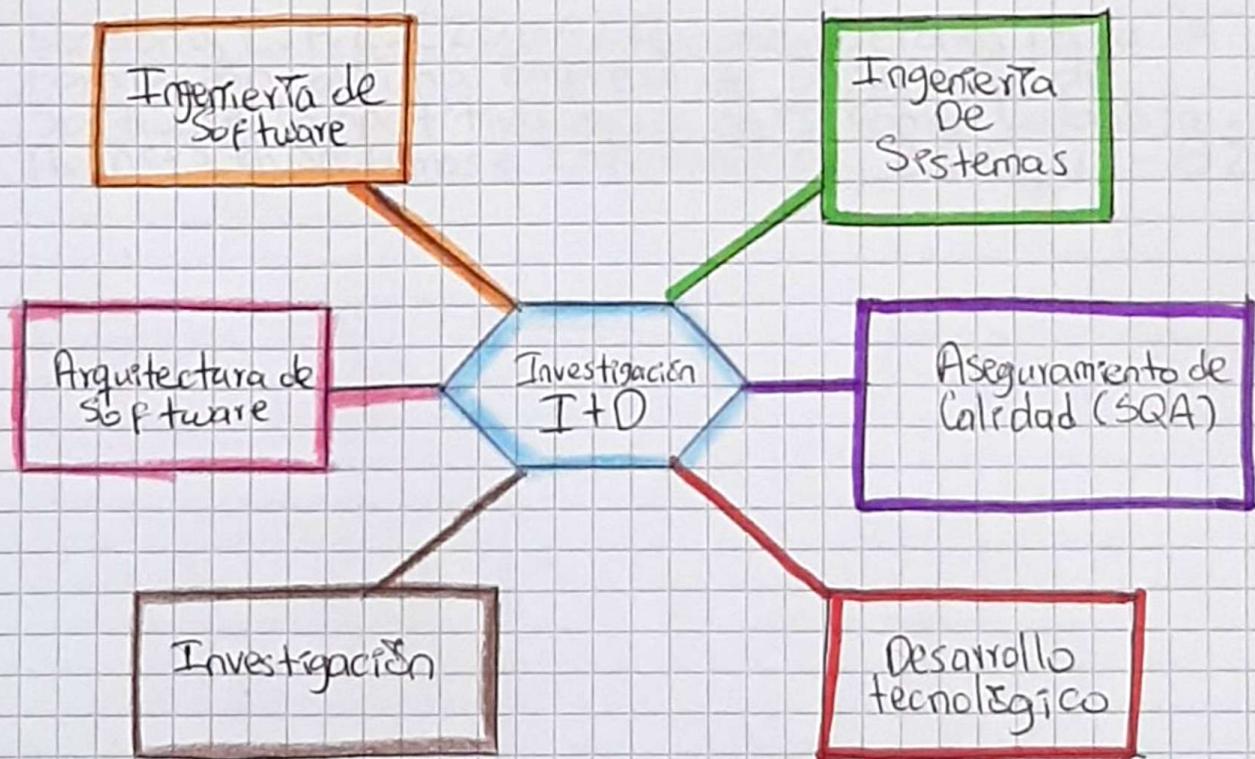
Cita APA - Bibliografía:

Abanto Cruz, J.-A., & Gonzales Ramírez, O.-F. (2019). Análisis comparativo de patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad: Una revisión sistemática de la literatura.

Recomendaciones para la Formación de una Empresa de Desarrollo de Software Competitiva en un país como Colombia

Resumen:

Algunas de las recomendaciones para ser una empresa competitiva, son, buscar que la organización sea reconocida como una compañía transparente con los clientes, la estructura de la organización, debe estar fortalecida, con un organismo que oriente el desarrollo tecnológico, tener mecanismos que les permitan medir y saber si se están cumpliendo las metas establecidas en los procesos, los productos y proyectos de la compañía. De esta manera podrá ser una empresa competitiva en Colombia.



Estructura I+D empresa modelo

Reflexión:

Al tener una empresa dirigida al desarrollo de Software, o al querer formarla, es necesario tener ciertos criterios importantes para de esa manera poder ser una empresa competitiva en todo este campo del desarrollo de Software, más en un país como lo es Colombia, el cual en estos tiempos a tomado fuerza en el tema de la tecnología y sus componentes. Todo el artículo hablaba sobre recomendaciones, las nombraban y luego explicaban cada una de ellas, para que la organización de una compañía sea buena y resalte.

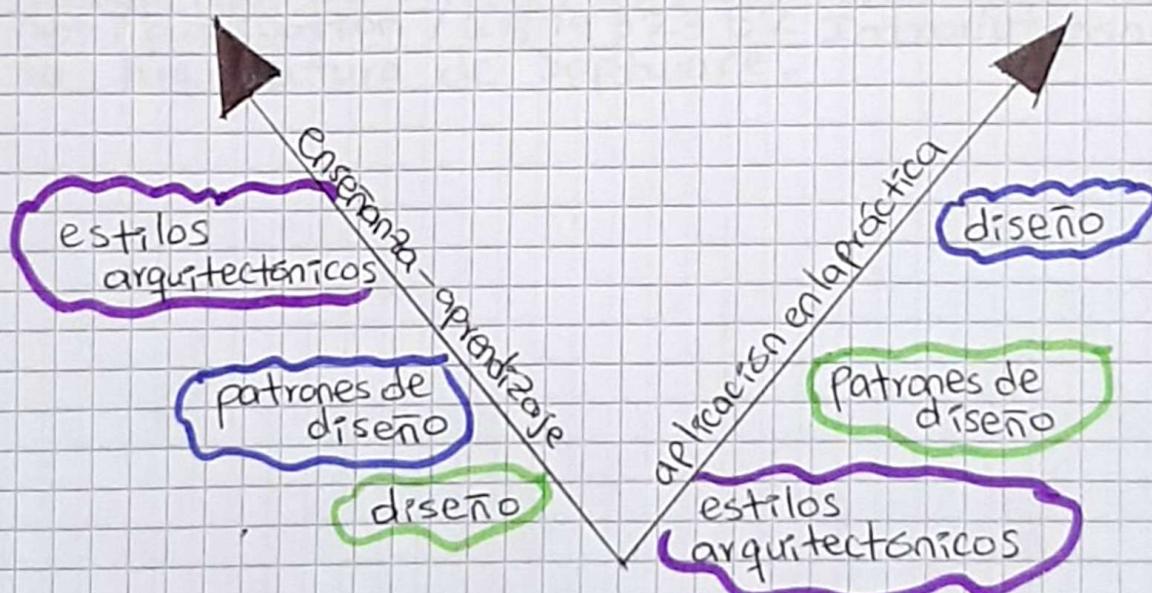
Cita APA - Bibliografía:

Londono, L. F. L. (2005). Recomendaciones para la Formación de una Empresa de Desarrollo de Software Competitiva en un país como Colombia - Avances en Sistemas e Informática, 2 (1), 41-52.

Introducción a la Arquitectura de Software

Resumen:

En los primeros años de la construcción de software no existía el diseño del sistema como una etapa independiente la programación. A principios de la década del 90 algunos investigadores, comenzaron a ver la necesidad de investigar y desarrollar un nivel de abstracción superior al del diseño, al que llamaron "Arquitectura de software". La arquitectura del sistema afecta a su entorno y este también afecta a la arquitectura. La arquitectura de software de un sistema es el resultado de combinar decisiones técnicas, sociales y del negocio.



Reflexión:

Apartir de la creación de la arquitectura de software se crearon de igual manera diferentes y variadas modelos de desarrollo, como el "Modelo de Cascada", la arquitectura del Sistema, la cual es la que le sigue a la Ingeniería de Requerimientos. La fase de la Arquitectura del sistema, se subdivide en tres etapas: Elección del estilo arquitectónico, Selección de los patrones de diseño y diseño de componentes; lo cual es importante de aprender para llevarla a la práctica, también es importante conocer los estilos arquitectónicos, estos son una generalización de los patrones de diseño, como abstracción.

Cita APA - Bibliografía:

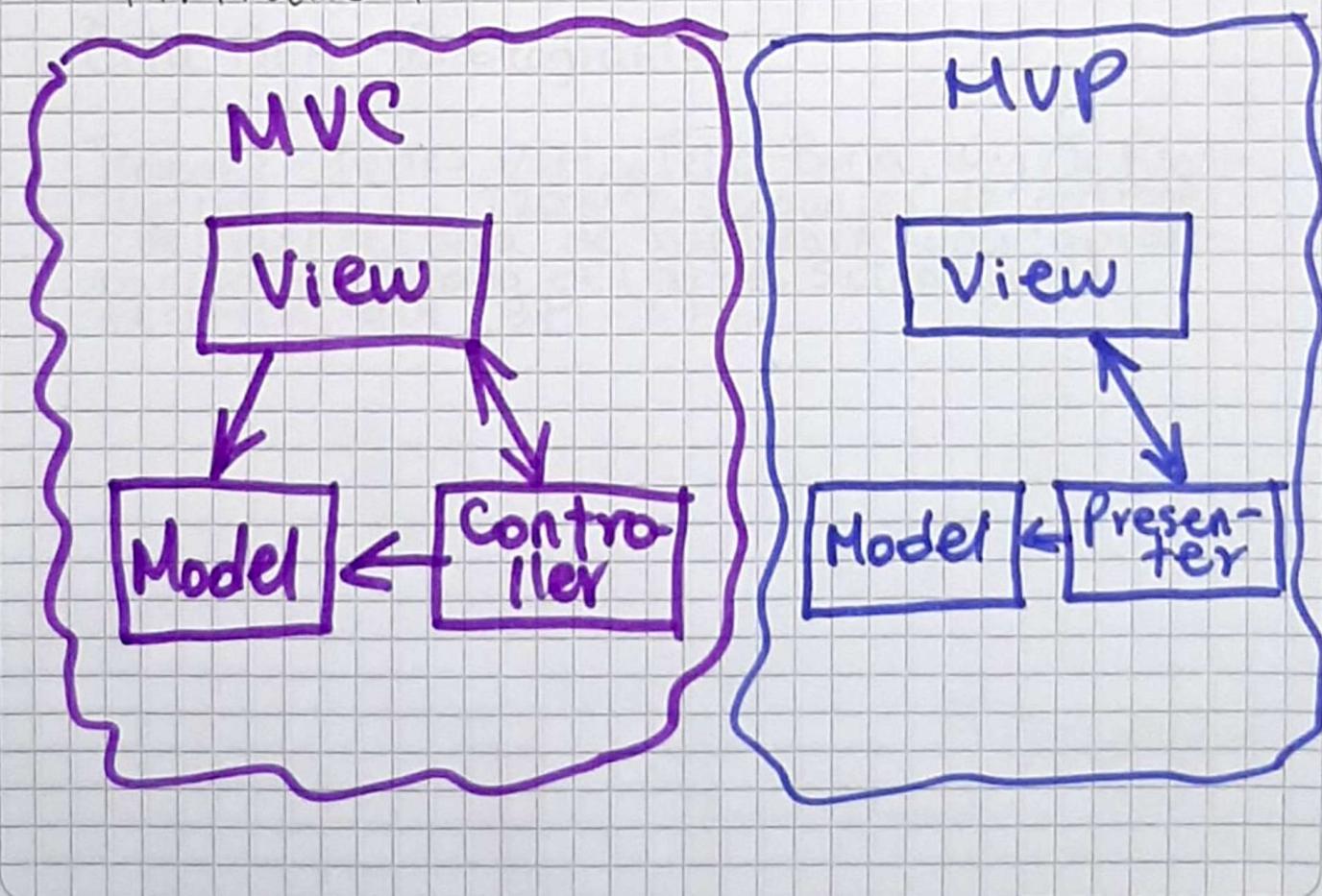
Cristiá, M. (2008). Introducción a la Arquitectura de Software. Research-Gate. [Online]. Recuperado de: <https://www.researchgate.net/publication/251932352> Introducción a la Arquitectura de Software.

Lenguajes de Patrones de Arquitectura de Software: Una Aproximación al Estado del Arte

Resumen:

Mostrar el estado del arte en un área de la arquitectura de software llamada "Lenguajes de Patrones" desde sus orígenes los avances actuales y sus aplicaciones en la construcción de arquitecturas de software en diferentes dominios de aplicación. La extensibilidad y aplicabilidad de los Lenguajes de patrones se convierte en una herramienta importante para diseñadores e implementadores de todo tipo de sistemas de información. Es importante evitar la programación al estilo vaquero. Parches de último minuto o trabajo de última noche que pueden traer graves consecuencias.

Artículo 4:



9

Reflexión:

Para que un software sea bueno y tenga una funcionalidad correcta, es importante evitar hacer las cosas demasiado rápido, porque se pueden presentar especificaciones incompletas, lo cual puede resultar en un re-proceso, haciendo perder tiempo en la elaboración.

La creación de nuevas formas de hacer las cosas ayuda en la productividad y una mayor calidad en los productos de software.

Cita APA - Bibliografía:

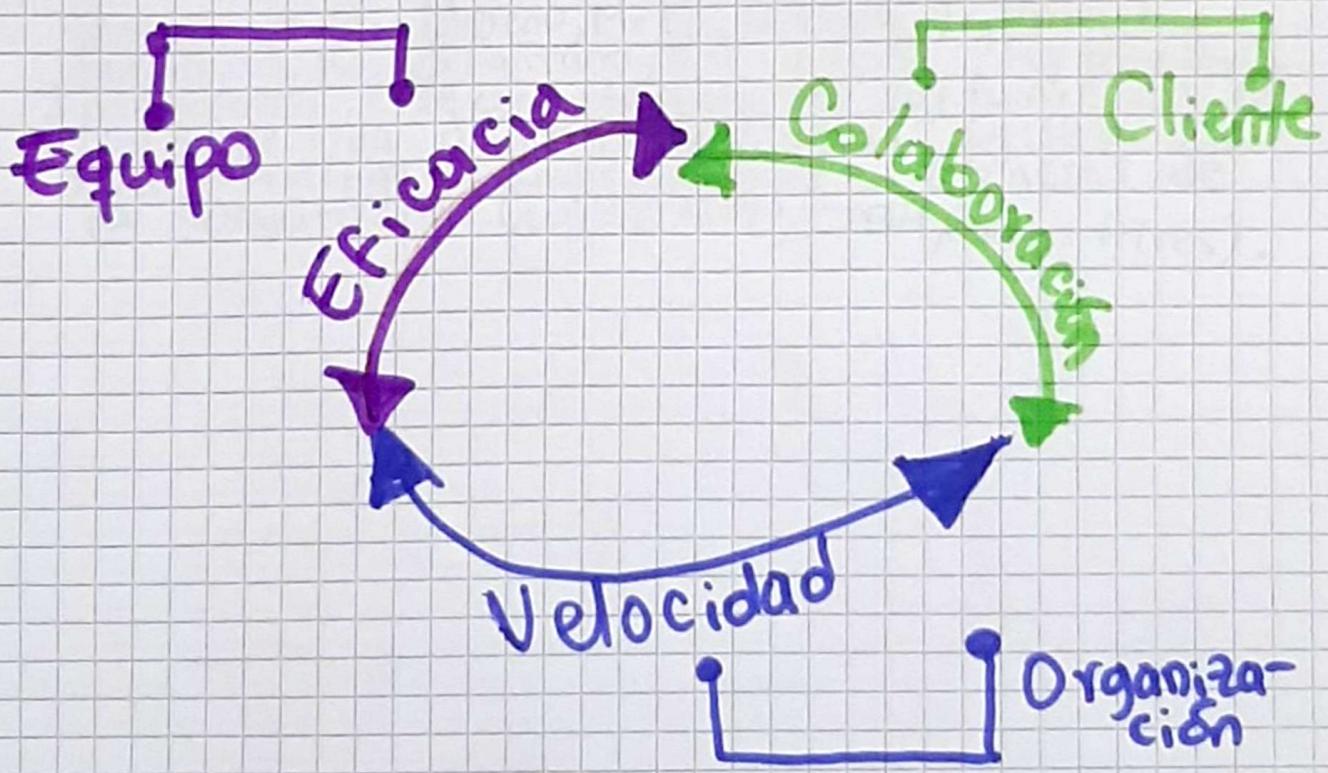
Jiménez-Torres, V.H., Tello-Borja, W., & Ríos-Patiño, J.-I. (2014). Lenguajes de patrones de arquitectura de software: una aproximación al estado del arte. *Scientia et Technica*, 19(4), 371-376.

Integración de arquitectura de software en el ciclo de vida de las metodologías Ágiles

Resumen:

Las metodologías Ágiles se centran en el trabajo en equipo, la adaptabilidad y colaboración dentro del grupo de software y también entre los miembros del grupo y los usuarios finales. El uso de las Metodologías (MA), ha marcado una tendencia en su adopción al desarrollo de proyectos de software (AR), en tanto, es una manifestación de decisiones de etapas muy tempranas del diseño sobre un sistema.

El tratamiento con enfoques diferentes en las primeras etapas, ha sido uno de los factores que ha causado la sensación de las MA y la AS.



Reflexión:

Las metodologías ágiles (MA) han transformado la manera en que se desarrollan proyectos de software, al centrarse en la adaptabilidad, la colaboración y el trabajo en equipo. Su enfoque en las primeras etapas del diseño permite ajustar el rumbo del proyecto de manera temprana, lo cual es crucial en entornos dinámicos donde los requisitos pueden cambiar rápidamente. La flexibilidad de las MA permite que los equipos respondan de manera más eficiente a las necesidades del cliente, lo que genera productos finales más alineados con las expectativas del usuario.

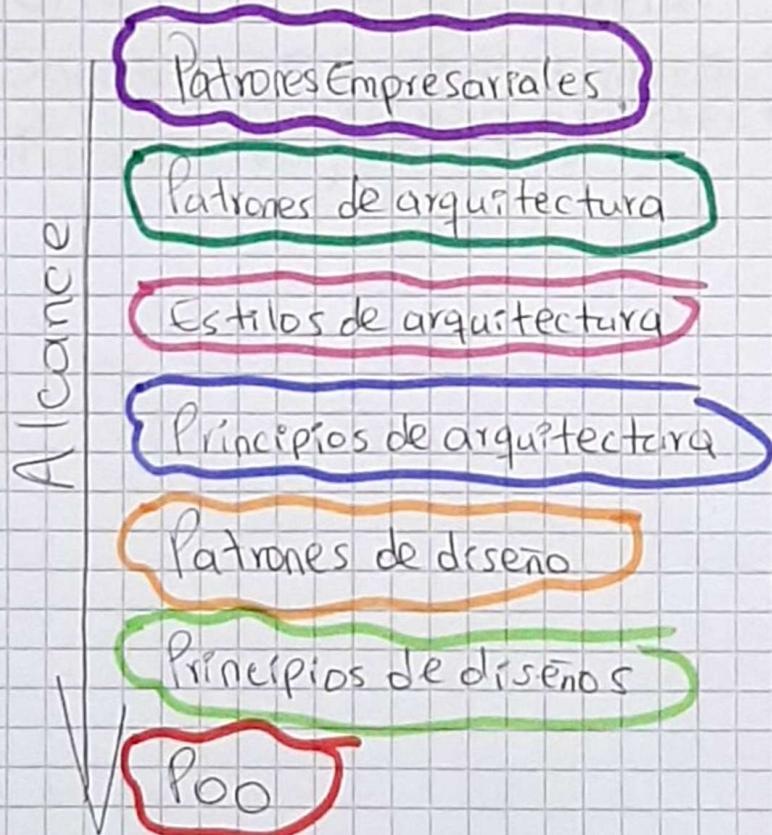
Cita APA - Bibliografía:

Navarro, M.-E., Moreno, M. P., Aranda, T., 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).

Evaluación de una Arquitectura de Software

Resumen:

Una arquitectura de software es clave para que las organizaciones puedan avanzar y concentrarse en su función misional, solo si esta está bien diseñada. Es importante realizar evaluaciones tempranas a la arquitectura mediante algún método, porque eso ayuda a la elección de una arquitectura, mejora la comunicación y permite una mejor interpretación de las historias de usuario. También tener en cuenta el análisis para el atributo de calidad modificabilidad.



Reflexión:

La arquitectura de software es el nacimiento sobre el que se construye el sistema, y su calidad tiene un impacto directo en la capacidad de una organización para alcanzar sus objetivos de negocio. Una buena arquitectura no solo asegura que los sistemas sean funcionales, sino que también facilita la adaptación, el mantenimiento y la escalabilidad a medida que las necesidades evolucionan. En este contexto, realizar evaluaciones tempranas y continuas de la arquitectura es crucial.

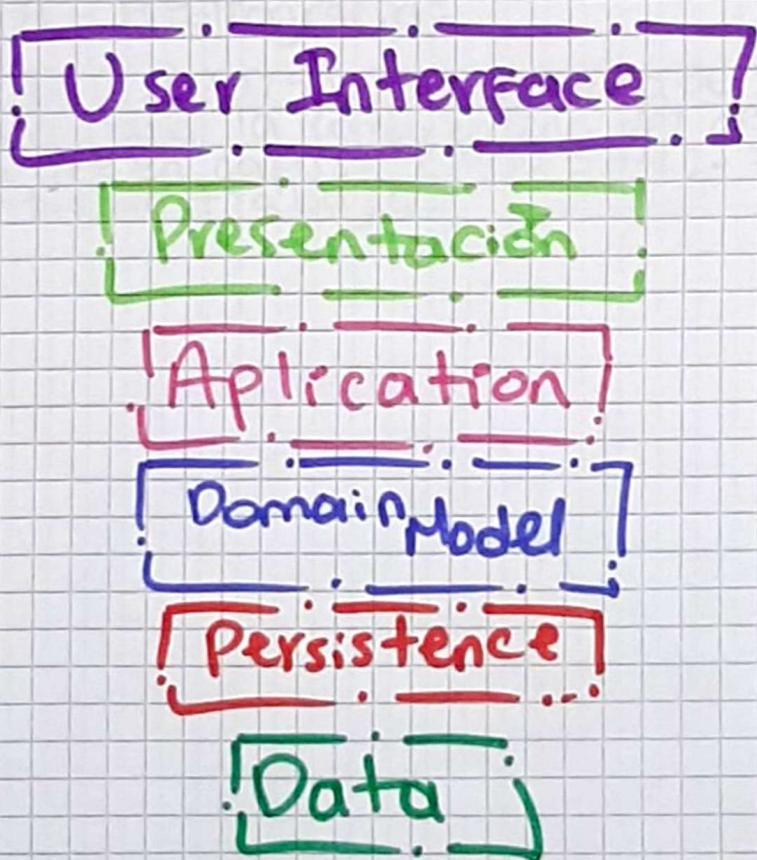
Cita APA - Bibliografía:

Sanabria, E.R., & Rodríguez, S.U. (2021). Evaluación de una arquitectura de Software-Prospectiva, 19 (2).

Arquitectura de Software académica para la comprensión del desarrollo de software en capas

Resumen:

El desarrollo de software implica considerar una cantidad variada de aspectos tecnológicos. Entre los más destacados podemos mencionar los relacionados con el acceso a datos, las interfaces, los procesos funcionales, el control de las transacciones, la accesibilidad y la seguridad. Lograr un diseño coherente con los requerimientos planteados, niveles aceptables de flexibilidad, extensibilidad y usabilidad, así como facilitar las actividades de mantenimiento lleva a pensar la concepción del software en capas.



Reflexión:

El diseño en capas permite separar claramente los diferentes responsabilidades del sistema, como la Capa de acceso a datos, la capa lógica de negocio y la capa de presentación (interfaz de usuario). Esta separación facilita la modificación y evolución de cada capa de forma independiente. Por ejemplo, si se necesita cambiar el tipo de base de datos o actualizar un framework, podemos hacerlo sin alterar la lógica de negocio ni la interfaz de usuario.

Cita APA - Bibliografía:

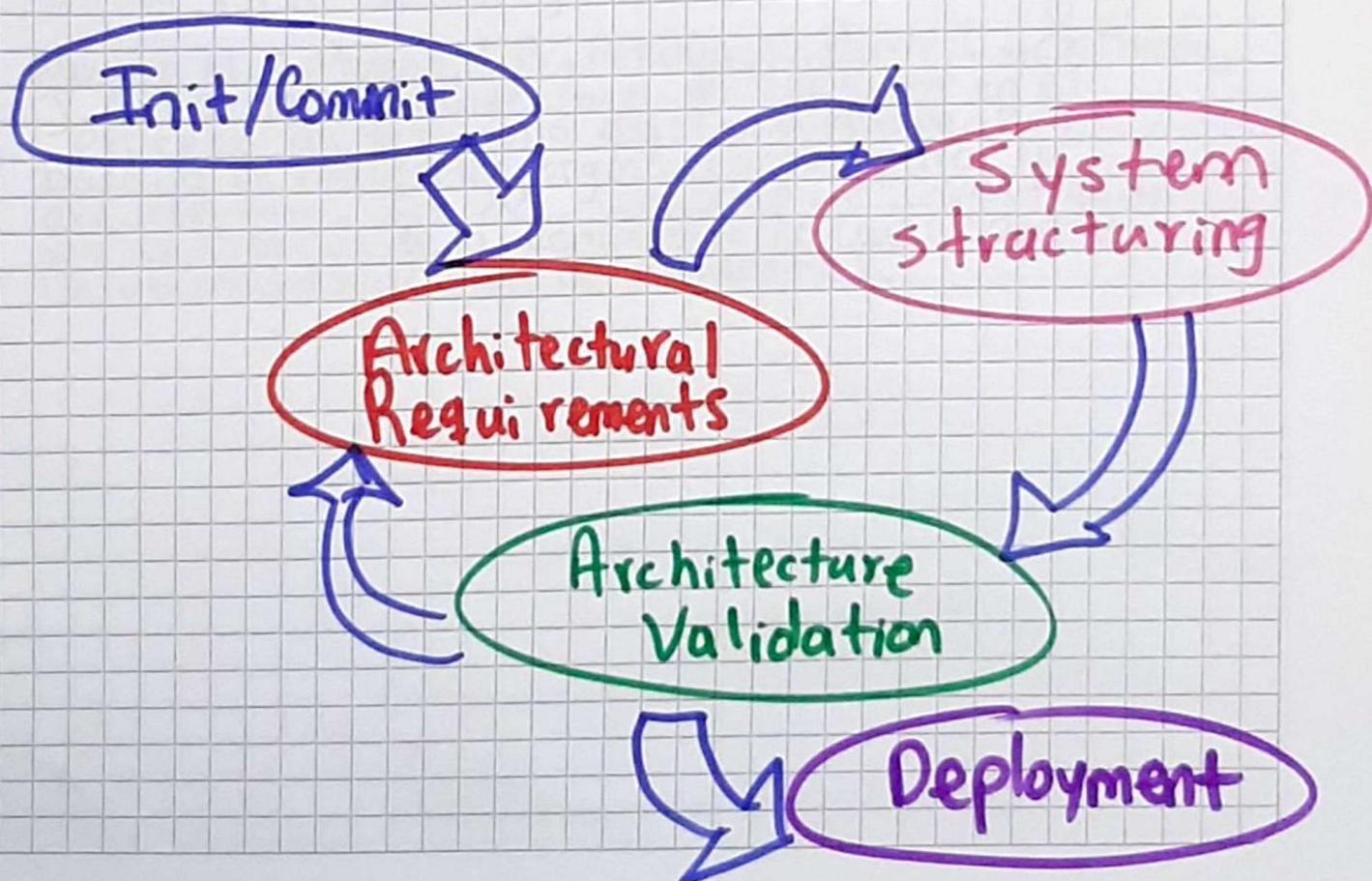
Carducci, D.G. (2015). Arquitectura de Software académica para la comprensión del desarrollo de software en capas. (No. 574). Serie Documentos de trabajo.

Arquitectura de Software en el proceso de desarrollo ágil: una perspectiva basada en requisitos significantes para la arquitectura

Resumen:

Recopilar, comprender y gestionar los requisitos es un aspecto crítico en todos los métodos de desarrollo. Esto también es cierto para las Metodologías Ágiles en la que la captura de requisitos es realizada en todo el proceso de desarrollo, con requisitos que van evolucionando y cambiando a lo largo del ciclo de vida.

Este proceso es opuesto al enfoque de la arquitectura del software, donde los requerimientos deben ser identificados, recabados y comprendidos.



Reflexión:

La reflexión sobre la gestión de requisitos en las metodologías ágiles frente al enfoque tradicional de la arquitectura de software revela una de las tensiones más fundamentales en el desarrollo de software. En un enfoque tradicional o "en cascada", la captura y definición de requisitos es una fase crítica inicial. Una vez que estos requisitos son comprendidos y documentados, la arquitectura del software se diseña y ajusta.

Cita APA - Bibliografía:

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 Investigador rest en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste).

La Arquitectura de Información (AI) en el proceso de desarrollo de Software

Resumen:

La AI se ha convertido para la producción de software en un proceso determinante con vistas a que los productos alcancen la calidad requerida. El objetivo de la investigación fue demostrar la importancia del rol del arquitecto de información en el proceso de desarrollo de software para la cual se analizan teóricamente los elementos que identifican el proceso de AI y se describen las Metodologías RUP y XP en cuanto a roles.

Flujos de trabajo

→ Identificación del negocio

Levantamiento de requisitos

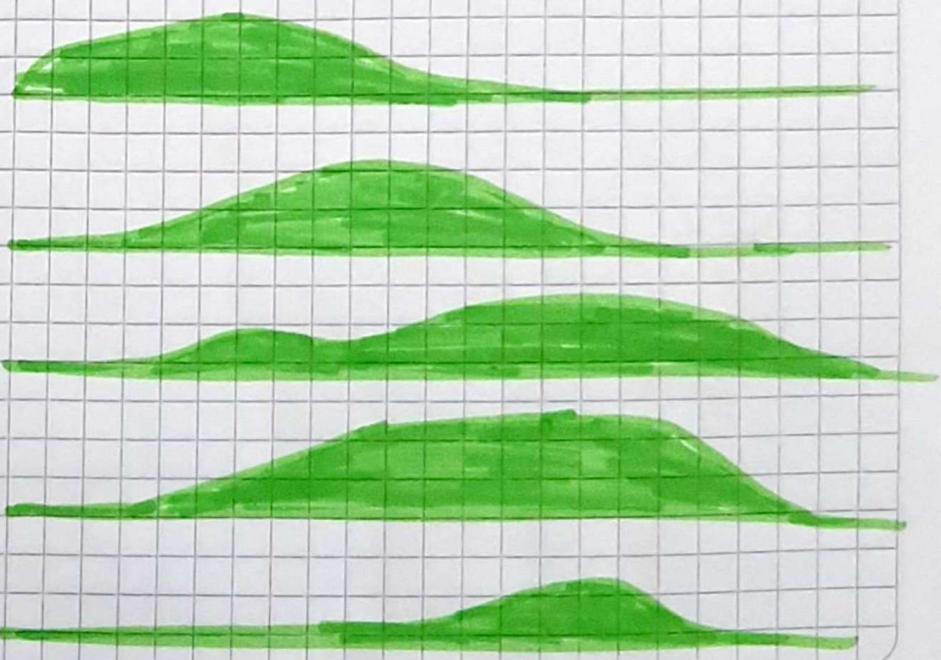
Análisis y Diseño

Interpretación

Prueba

Fases AI

Inicio Consolidación Validación



Reflexión:

En un entorno cada vez más dominado por la automatización y el uso de la inteligencia artificial, el rol del arquitecto de información no solo sigue siendo relevante, sino que se ha intensificado. Este profesional no solo se encarga de estructurar la organización del conocimiento y la información dentro del sistema, sino que también es esencial para garantizar que el software sea escalable, sostenible y funcional.

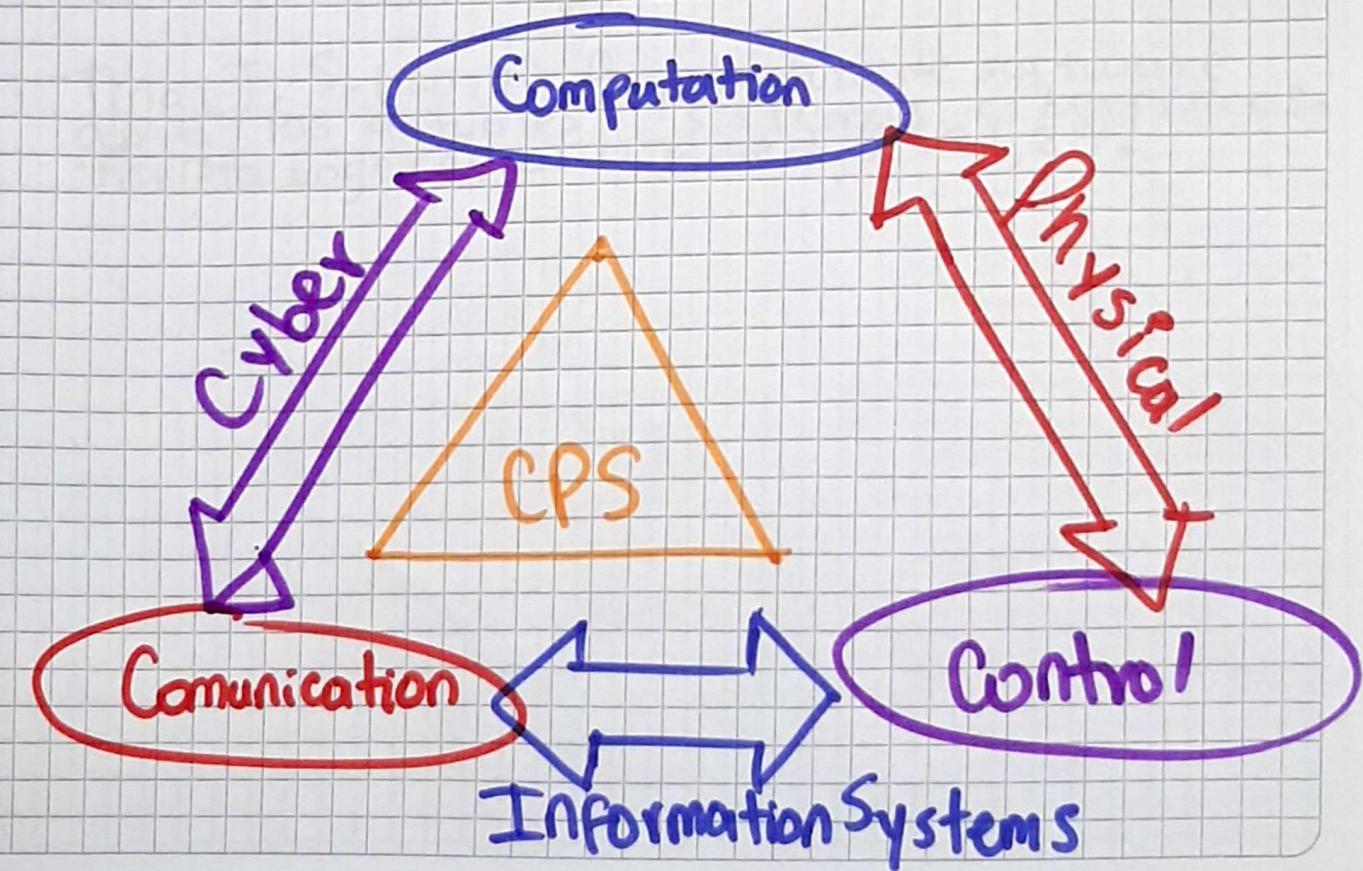
Cita APA - Bibliografía:

Moyares, Y., & Lorenzo, D.B., (2021). La Arquitectura de información (AI) en el proceso de desarrollo de software. *Bibliotecas. Anales de investigación*, 6, 97-102.

Arquitectura de software para los actuales sistemas ciberfísicos

Resumen:

La próxima generación de sistemas ciber-físicos, plantea grandes desafíos en el diseño de software. No se trata sólo de diseñar un sistema entorno a los plazos de ejecución, lo más importante es maximizar la utilización de recursos. En el artículo se propone un sistema base para la arquitectura de software en el que los servicios se puedan diseñar e implementar y componer fácilmente de acuerdo con la demanda.



Reflexión:

Los CPS, al integrar componentes computacionales y físicos de manera estrecha, no solo requieren sistemas de software robustos sino que deben maximizar la eficiencia en el uso de los recursos disponibles. Esta tarea se complica aún más cuando se deben tener en cuenta factores como la latencia, confianza, la seguridad y la escalabilidad.

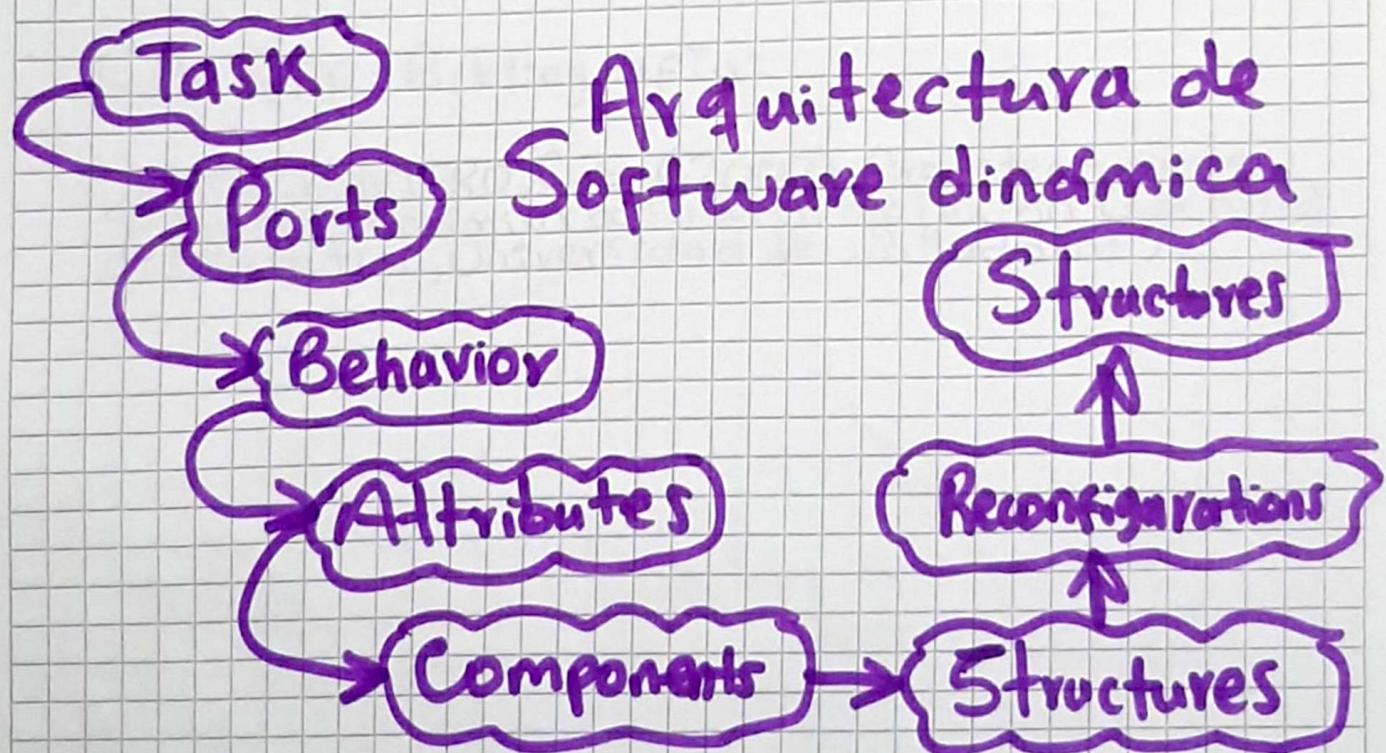
Cita APA-Bibliografía:

Ting, J. S. (2011). Arquitectura de software para los actuales sistemas ciberráfricos. Revista Ingenierías USBmed, 2(1), 29-

Arquitectura de software dinámica basada en reflexión

Resumen:

La tesis sobre el artículo, se centraba en el campo de la Arquitectura de Software, una rama de la Ingeniería de Software dedicada al estudio de la estructura de los sistemas software complejos. Se trata y estudia concretamente uno de los problemas pendientes del campo: la especificación y descripción de arquitecturas de software dinámicas es decir, aquellas cuya estructura puede variar. Con esto se elabora de manera informal un modelo reflexivo de descripción arquitectónica de software.



Reflexión:

La tesis sobre el artículo que aborda el campo de la arquitectura de software, una rama crucial dentro de la Ingeniería de Software, resalta la importancia de la estructura de los sistemas software complejos. Este campo se encarga de analizar cómo se organizan y se interrelacionan los componentes dentro de un sistema para garantizar su efectividad, escalabilidad y flexibilidad.

Cita APA - Bibliografía:

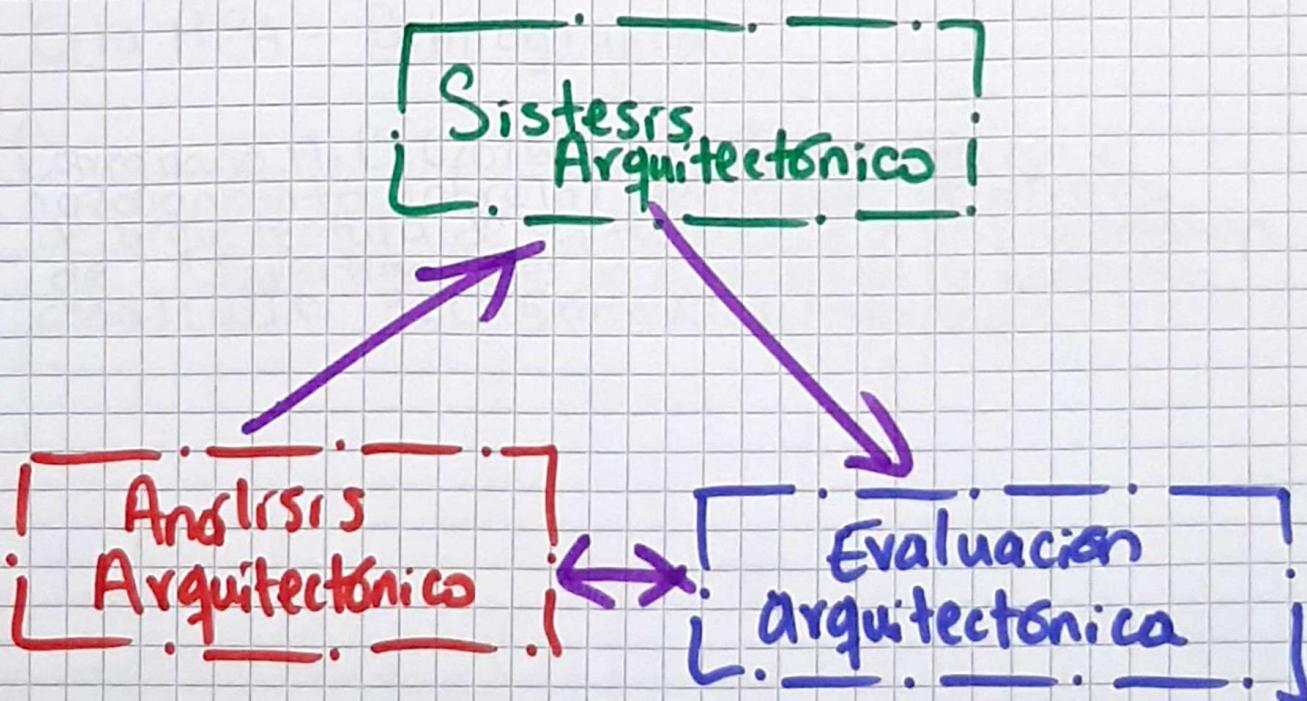
CUESTA QUINTERO, C.E. (2002). Arquitectura de Software dinámica basada en reflexión (Doctoral dissertation, Universidad de Valladolid).

Representación y razonamiento sobre las decisiones de diseño de arquitectura de software

Resumen:

El objetivo general de la tesis estipulada en el artículo, estaba directamente vinculado a brindar soporte al arquitecto de software durante el diseño arquitectónico.

Partiendo de la hipótesis de que los arquitectos de software reutilizan soluciones conocidas en nuevos diseños, se identifica la necesidad de contar con una herramienta, tanto conceptual como computacional, que pueda colaborar con los arquitectos de software.



Reflexión:

La reutilización de soluciones conocidas. Esta práctica, que consiste en aplicar patrones, diseños o componentes previamente probados y validados, es una estrategia fundamental para mejorar la eficiencia, reducir los costos de desarrollo y aumentar la calidad del software. La hipótesis planteada es completamente válida y refleja una realidad observada en la mayoría de los proyectos de desarrollo de software a gran escala.

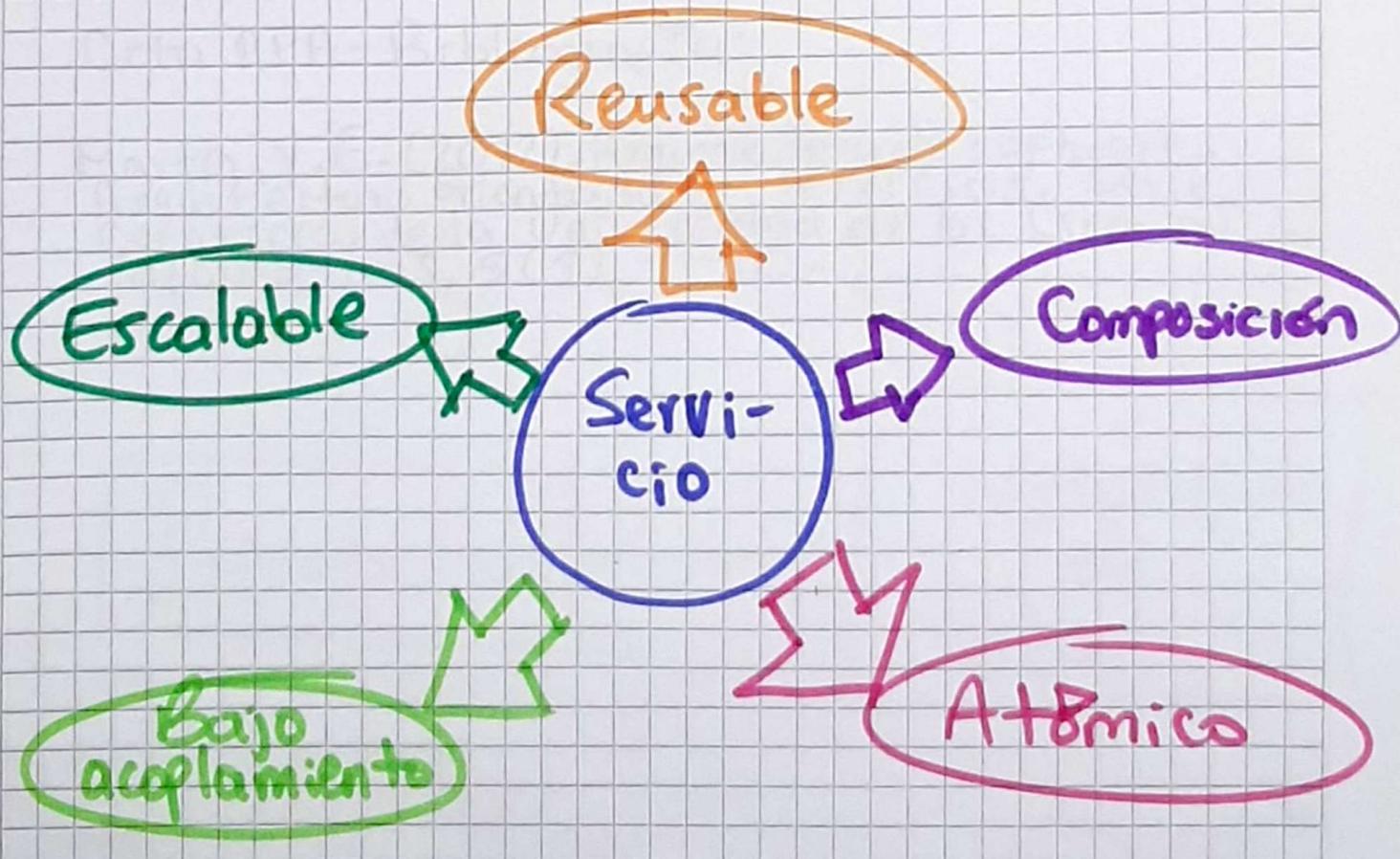
Cita APA - Bibliografía:

Carrignano, M. C. (2016, June). Representación y razonamiento sobre las decisiones de diseño de arquitectura de software. In XVIII Workshop de Investigadores en Ciencias de la Computación (WICC) 2016, entre Ríos, Argentina).

Arquitectura de Software. Arquitectura orientada a Servicios.

Resumen:

Dentro del proceso de desarrollo de software de un sistema uno de los temas importantes a tratar es la arquitectura de software, es la parte de la ingeniería de software que se dedica al estudio, análisis y descripción para formalizar el esquema global de un sistema, poniendo énfasis en el estudio de las interacciones entre sus elementos básicos, denominados componentes. En el artículo se realizó un estudio general de la arquitectura de software haciendo énfasis en la arquitectura orientada a servicios.



Reflexión:

La arquitectura de software es un componente fundamental dentro del proceso de desarrollo de software, ya que establece las bases para la estructura general del sistema. Su objetivo es organizar los diversos elementos que conforman el sistema de forma eficiente, asegurando que interactúen correctamente entre sí y con otros sistemas externos. Se encarga de describir, analizar y formalizar el esquema global del sistema, lo que incluye la distribución de tareas.

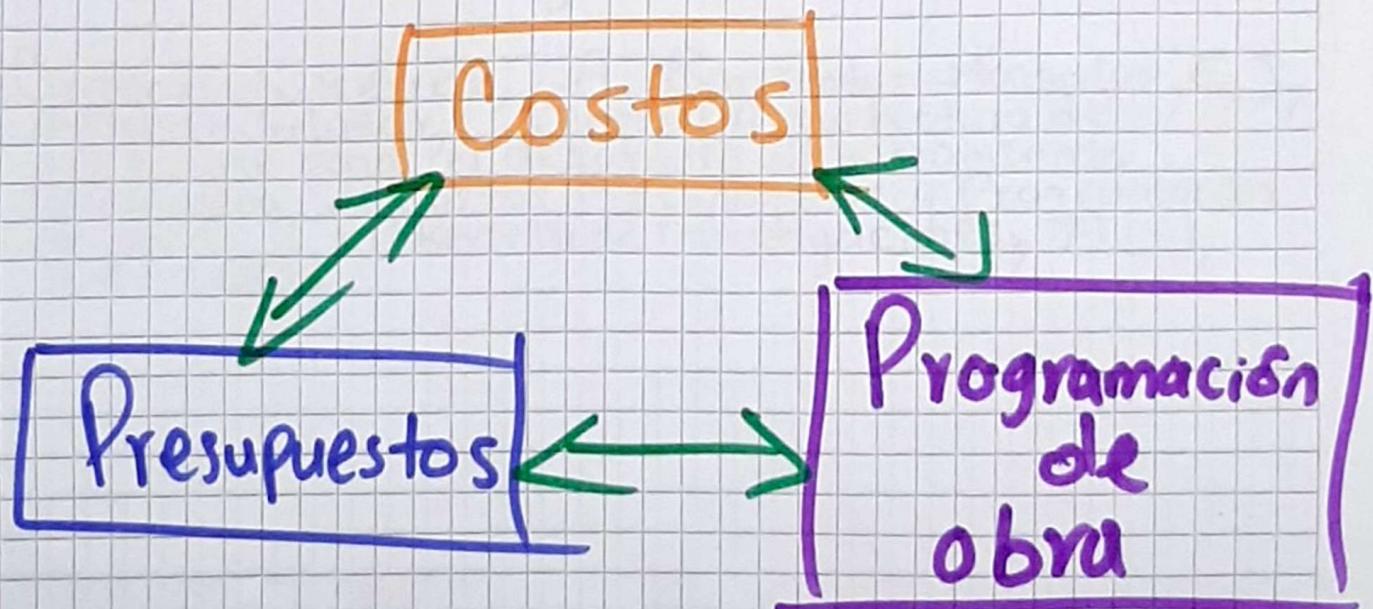
Cita APA - Bibliografía:

Martín, Y. E. (2012). Arquitectura de software - Arquitectura orientada a servicios. Serie Científica de la Universidad de las Ciencias Informáticas, 5(1), 1-10.

Arquitectura de Software para el desarrollo de herramienta tecnologica de Costos, presupuestos y programacion de obra

Resumen:

Durante todo el artículo trataban temas relacionados a la arquitectura de software, como los casos de uso, diagramas de flujo del software, diagramas de despliegue, diagramas de paquetes del software, por último nombraron todo sobre la cronología y metodología del desarrollo de costos, presupuestos y programación de obra, que se deben llevar a cabo, calcular, establecer listas, establecer valores de los costos indirectos, calcular el valor total del proyecto y realizar la programación de obra.



Reflexión:

La arquitectura de software es la base sobre igual la cual se construye el sistema, y como tal, debe ser cuidadosamente diseñada y documentada. Los diagramas de flujo, los diagramas de despliegue, los casos de uso y los diagramas de paquetes permiten representar de forma visual y estructurada como interactúan los diferentes componentes del sistema. Estos artefactos son fundamentales para garantizar que todos los elementos del software estén bien definidos.

Cita APA - Bibliografía:

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.

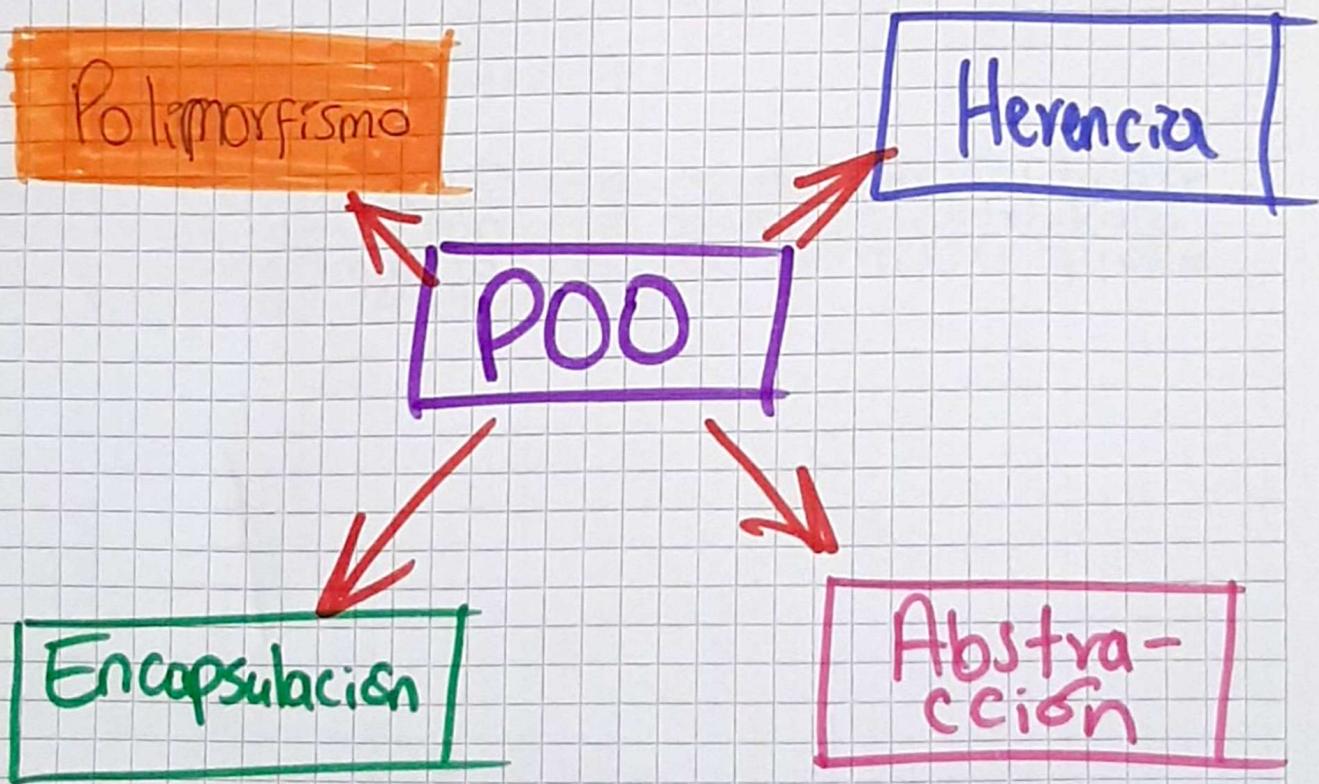
Arquitectura de software con programación orientada a objetos

Resumen:

El objetivo del artículo fue describir la arquitectura de software con programación orientada a Objeto apartir de la revisión de fuentes documentales actualizadas.

Se encontró que la POO agrupa un conjunto de técnicas que permiten desarrollar y mantener mucho más fácilmente programas de una gran complejidad.

La evolución de la arquitectura de software, escala cada vez más posiciones superiores, que permiten a los profesionales resolver diversos problemas..



Reflexión:

La arquitectura de software con programación POO ha demostrado ser una de las metodologías más poderosas y efectivas para el desarrollo de software, especialmente en el contexto de sistemas complejos. A través de la revisión de fuentes documentales actualizadas, se ha comprobado que la POO agrupa un conjunto de técnicas y principios que faciliten no solo el desarrollo, sino también el mantenimiento y evolución de programas con un alto grado de complejidad.

Cita APA - Bibliografía:

Vera, J.B.V. (2023). Arquitectura de Software con Programación orientada a objeto. *Polo del Conocimiento: Revista científico-profesional*, 8(12), 1497-1508.