

ARTICULO 1

ANÁLISIS COMPARATIVOS DE PATRONES DE DISEÑO DE SOFTWARE

RESUMEN

Los patrones de diseño son herramientas que ayudan a resolver problemas comunes en el desarrollo de software, promoviendo la organización y la reutilización del código. En este artículo se exploran en detalle cinco patrones: Template Method, Model-View-Controller (MVC), Model-View-Presenter (MVP), Front Controller y Model-View-View Model (MVVM). La investigación, basada en una revisión de bibliografía científica, analiza estos patrones comparándolos a través de métricas específicas.

El análisis muestra que no hay un patrón "mejor" de forma general, ya que cada uno está diseñado para resolver problemas específicos y aportar ventajas en ciertos contextos. Así, corresponde al desarrollador evaluar cuál patrón es más adecuado para cada situación. Se concluye que los patrones de diseño ayudan a estructurar el código de forma organizada, facilitando su comprensión y mantenimiento, y contribuyen al desarrollo de software de calidad gracias a su modularidad.

REFLEXIÓN

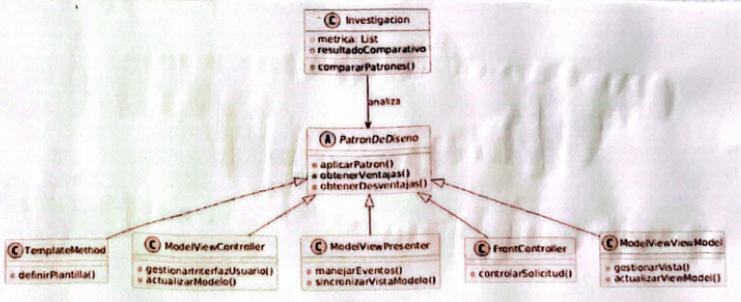
Este análisis sobre los patrones de diseño nos invita a reflexionar sobre cómo la arquitectura de software es más que solo un conjunto de reglas técnicas; es una forma de pensamiento que busca ordenar y hacer más claro el desarrollo de sistemas complejos. Los patrones como MVC, MVP y MVVM, entre otros, no son recetas infalibles, sino guías que se adaptan y evolucionan según las necesidades de cada proyecto. Nos recordó que el diseño de software no tiene soluciones universales. En cambio, requiere una mirada crítica del desarrollador para elegir la estructura que mejor potencie la funcionalidad, claridad y mantenimiento del sistema.

La conclusión de que no existe un patrón superior nos

sobre a la flexibilidad y la responsabilidad de comprender bien cada problema antes de elegir una solución. Esta libertad puede parecer desafiante. Pero también enriquece el trabajo creativo en software: cada proyecto nos da la oportunidad de probar y ajustar estos patrones, aprendiendo a valorar el equilibrio entre teoría y práctica. En última instancia, el uso de estos patrones refleja una ética profesional en la programación donde buscamos construir código que no solo funcione bien, si no que también sea claro, sostenible y comprensible para otros.

BIBLIOGRAFÍAS

Alvarez, O. D. G., Larrea, N. R. L., & Valencia, M. V. R. (2022). Análisis comparativo de patrones de diseño de software. *Polo del Conocimiento: Revista Científico-Profesional*, 7(7), 2746 - 2765.



ARTÍCULO 2

UNA ARQUITECTURA PARA UNA HERRAMIENTA DE PATRONES DE DISEÑO

RESUMEN

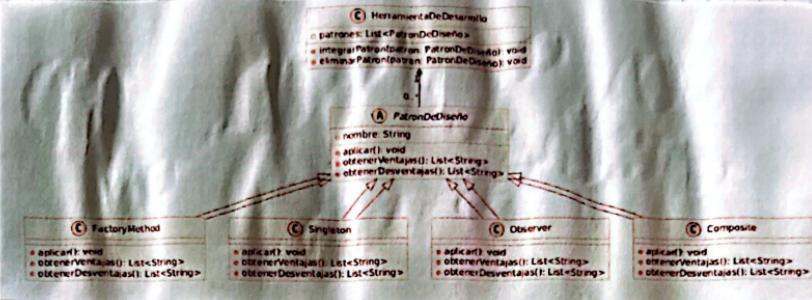
El artículo propone una arquitectura para herramientas de desarrollo que integren patrones de diseño, lo cual facilitaría su uso en la creación de software. Aunque los patrones son soluciones probadas y estructuradas para problemas comunes, actualmente pocas herramientas los incluyen de forma integrada. Existen puntos de vista distintos sobre esta integración: algunos creen que los patrones deben guiar a los programadores sin automatizar el código, mientras que otros apoyan su implementación directa en herramientas. La arquitectura presentada permite representar y manipular patrones de manera flexible, con el objetivo de mejorar la eficiencia en el desarrollo orientado a objetos y facilitar el diseño de código modular y reutilizable.

REFLEXIÓN

La propuesta resalta la importancia de contar con herramientas que faciliten el uso de patrones de diseño, lo cual podría revolucionar la forma en que los desarrolladores abordan problemas comunes en el software. Si bien es cierto que los patrones de diseño no deben reemplazar la creatividad y el análisis humano, su inclusión en las herramientas puede potenciar el trabajo de los programadores al reducir esfuerzos y mejorar la calidad del código. En última instancia, el uso adecuado de los patrones dentro de un entorno de desarrollo podría ayudar a crear software más limpio, estructurado y mantenable.

BIBLIOGRAFÍA

Martínez, J. S., Molina, J. G., & García, P. J. (1999). Una Arquitectura para una Herramienta de patrones de diseño. Depto. de Informática y Sistemas, Universidad de Murcia, Murcia.



ARTÍCULO 3

PLINKER: Relaciones con patrones de diseño

RESUMEN

El artículo analiza como los patrones de diseño aunque muy útiles de forma individual, pueden complementarse entre si para resolver problemas más complejos de manera integrada. Sin embargo, una de las dificultades actuales es que la documentación de estos patrones rara vez detalla como deben combinarse o relacionarse dentro de un mismo diseño. Este trabajo realiza una clasificación formal de los tipos de relaciones entre patrones, identificando seis tipos principales como patrones que se dan en conjunto patrones que se desglosan en otros o que los crean. Además se presenta la herramienta Plunker, un editor gráfico que permite modular clases en una interfaz de usuario, en diseños ya existentes y aprovechar las relaciones entre patrones de manera automática. La herramienta facilita la construcción y gestión de diseños orientados a objetos, permitiendo a los diseñadores incorporar patrones de forma consistente y mediante la consistencia entre ellos, especialmente en arquitecturas más grandes como UML.

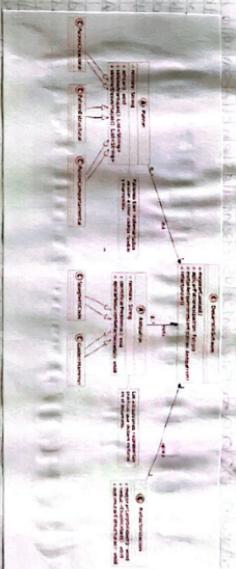
REFLEXIÓN

La propuesta de este artículo nos recuerda que en el desarrollo de software, los patrones de diseño no solo resuelven problemas具体情况 sino que también son bloques de construcción que pueden enriquecer y extender el uso de los patrones individuales. Pueden abordar problemas más amplios que crear soluciones más flexibles y adaptadas. La herramienta Plunker ofrece una asistencia clave al automatizar la integración de patrones mostrando cómo estos complementarse o modificarse individualmente sin perder su naturaleza original. Esto permite optimizar el diseño de software y favorecer una implementación más eficiente y reusable, evitando que los desarrolladores tengan que lidiar con cada patrón de manera aislada. En cierta instancia, este

enfoque contribuye a la creación de software modular y cohesionado, facilitando su mantenimiento y escalabilidad.

BIBLIOGRAFÍA

Liebener, L., Rossi, H. M., & Marcos, C. (2003). Relaciones con patrones de diseño. *Revista de Ciencias Exactas*, 1(1).



APÓDULO 4 ERRORES DE DISEÑO, REFACTORACIÓN Y ANTIPATRONES

Resumen

El apartado exploró tres elementos esenciales en el desarrollo de Software orientados a objetos: patrones de diseño, refactorización y antipatrones. Los errores de diseño ofrecen soluciones efectivas para problemas recurrentes en la construcción de sistemas, aportando flexibilidad y facilidad de mantenimiento al Software por otro lado. Los errores documentaron prácticas que en lugar de ayudar, han demostrado ser perjudiciales y deben evitarse. A través de ejemplos y clasificaciones se presentan patrones antipatrones como **Singleton**, **Factory Method**, **Hammer**, cada uno representando prácticas que evitan que se cometa el desarrollo. Además, la refactorización se describe como un proceso que recopila mejoras para mejorar su diseño sin alterar su funcionamiento. La synergia entre estos elementos permite al equipo mejorar su software más sólido, permitiendo mayor progreso a errores, mejorando el producto final y facilitando su evolución.

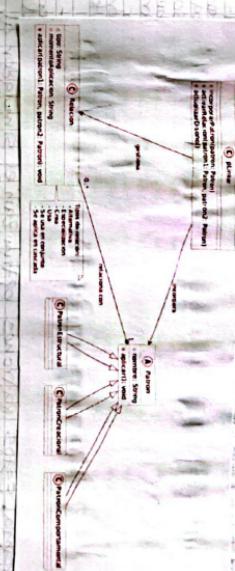
REFLEXIÓN

Este análisis sobre patrones de diseño y antipatrones nos muestra que el desarrollo de Software no se trata solo de escribir código funcional. Si no de aplicar conocimientos y experiencia de forma estratégica. Los patrones de Diseño no son simplemente soluciones genéricas para estructura y organización del Software. Sino guías para estructura y diseño que sea escalable y fácil de entender. Por su parte los antipatrones actúan como recordatorios de prácticas ineficaces que al contrario de lo que sucede con los patrones, permiten mejorar el flujo de trabajo y la calidad del código. Además, la refactorización desempeña un papel fundamental ya que permite que el código evolucione sin perder su calidad o legibilidad. Compartiendo una cultura de mantenimiento y mejora continua. En conjunto estos prácticos nos subrayan el proceso de desarrollo. Esto que también permiten a los desarrolladores evolucionar a

cambios y realizar ajustes de forma más eficiente, produciendo software de alta calidad y optimizando recursos.

BIBLIOGRAFÍA

Camgo, G.D. (2009). Errores de diseño, refactorización y antipatrones.



Resumen

USO DE PATRONES DE DISEÑO: UN CASO PRÁCTICO.

RESUMEN

Este artículo describe cómo los patrones de diseño agoran soluciones prácticas a problemas comunes en el desarrollo de software, mejorando su valor para mayor flexibilidad y la funcionalidad del código. Se muestra un simulador de un procesador multinivel. En este caso se estudiaron varios patrones, como el patron de polimorfismo (biblioteca Singleton), una estrategia para aumentar la reutilización y minimizar el mantenimiento del código. Esta sección expone las ventajas y desventajas de los patrones: el uso más eficiente permite el desarrollo más rápido y eficiente de sistemas. Los patrones optimizan el trabajo de desarrollo y facilitan modificaciones y adiciones sin afectar el núcleo de su funcionamiento.

Resumen

La aplicación de patrones de diseño en este proyecto demuestra cómo las soluciones bien estructuradas mejoran la organización y la flexibilidad del software. Los patrones permiten a los desarrolladores trabajar con un código más limpio y comprensible, simplificando la incorporación de cambios o nuevas funcionalidades. Sin embargo, como sin comprometer la estabilidad. Sin embargo, como se sugiere, no es solo cuestión de aplicarlos. Algunos patrones, siendo muy útiles en el contexto adecuado para ciertas complejidades innecesarias. Este balance entre diseño y eficiencia es clave en el desarrollo de software. Es deseable mantener la extensión práctica con patrones que también es valiosa para cumplir las necesidades de un cliente. Un ejemplo es la creación de soluciones dinámicas y adaptativas en proyectos reales.

Bibliografía

Germíndez, E.S.; Jiménez, I.M.; Rodríguez, L.R. (2012) USO de patrones de diseño: un caso práctico. Ingeniería, 22(2), 45-59.



Artículo 6

PARCERIAS DE DISERTO: EJEMPLOS DE APLICACIONES EN LOS GENERATIVOS LEARNING OBJECT

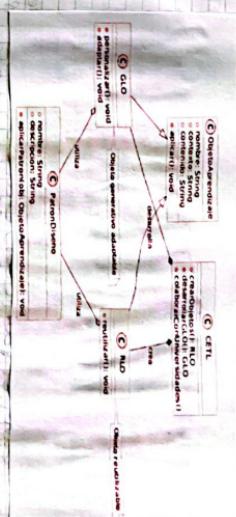
Este artículo describe cómo los factores de diseño pueden mejorar la creación y revisión de objetos de aprendizaje en el ámbito del e-learning. Al definir y seleccionar recursos para problemas comunes en la enseñanza virtual, estos factores facilitan la creación de recursos adaptables, uno de los objetivos establecidos en el centro de excelencia CenCe, secundaria en la creación de objetos de aprendizaje generativos (GA) que permiten a los alumnos practicar y consolidar sus conocimientos en función de sus necesidades educativas. Los GA se construyen a partir de objetos de aprendizaje reutilizables (RA) logrando una mayor eficiencia en la creación de recursos adaptables y eficientes para el aprendizaje en línea.

REFEXION

Este trabajo destaca el valor de los software de diseño en la educación donde la capacidad para revitalizar las vivencias de los recursos humanos es fundamental. Los resultados muestran que los software educativos tienen la posibilidad de crear diccionarios de aprendizaje que sirven para aumentar la comprensión de los procesos de desarrollo de los contenidos. No solo optimizan el proceso de desarrollo de los contenidos, sino que también permiten una enseñanza más personalizada y efectiva. Al incorporar soluciones robadas de software, los educadores pueden enfocarse en la calidad pedagógica, combinando en herramientas interactivas y touchpads que el aprendizaje interactivo sea accesible en entornos digitales.

BIBLIOGRAFÍA

Telco. (2009). Raíones de crecimiento en los sistemas de aprendizaje en línea (e-learning object). Revista de Educación a Distancia CREDO.



Artículo 7:

PATRONES DE DISEÑO GOF (THE GUIDE OF FOO) EN EL CONTEXTO DE PROCESOS DE DESARROLLO. APLICACIONES ORIENTADAS A LA WEB

RESUMEN

Este artículo analiza la implementación de patrones de diseño GoF (The Guide of Foo) en el desarrollo de aplicaciones web. Se analiza su uso en procesos formales de desarrollo de software en donde los patrones de diseño GoF son clasificados en creaciones, estructuras y patrones de comportamiento ayudando a resolver problemas comunes en la industria. La calidad del software es un problema importante que se ha tratado en parte debido a la falta de conocimiento o experiencia. En general, el patrón Singleton es uno de los más utilizados en los proyectos orientados a la web, mientras que otros patrones como los adjuntos o los observadores, que tienen una menor eficiencia y adaptabilidad, optimizan el tiempo y recursos a largo plazo.

DEFINICIÓN

Este trabajo destaca la importancia de los patrones de diseño como herramientas que ayudan a optimizar el desarrollo de software, especialmente en aplicaciones complejas como las aplicaciones web. La investigación muestra que algunos patrones se usan con frecuencia, otros quedan poco utilizados y la familiarización de los desarrolladores con todos los patrones del catálogo GoF puede facilitar la calidad y productividad de los proyectos de software en el futuro. Al tratar del uso adecuado de estos patrones, los errores de desarrollo pueden ser detectados más eficientes y adaptables, optimizando tiempo y recursos a largo plazo.

BIBLIOGRAFÍA

Gutierrez, M., & Gutierrez, L. E. (2013). Patrones de diseño GoF (The Guide of Foo) en el contexto de procesos de desarrollo de aplicaciones orientadas a la web.

Información tecnológica, 24 (3), 103-114.



ANÁLISIS DE PATRONES DE SEGURO

INTENCIÓN DE PATRONES DE SEGURIDAD

EN RODAS DE ISSEY

RESUMEN

Este trabajo analiza dos de los catálogos de Patrones de diseño más conocidos: el "core security patterns" (CSP) y el "core SRE pattern catalogue" (CSP). Los dos sistemas de patrones de la experiencia de Sun Microsystems, tienen como objetivo principal es integrar diferentes tipos de patrones de segundo (CSP) con los patrones multicasa (CSP), para así facilitar su uso. Se muestra para esto el espacio de examinar los patrones de Seguridad en el contexto de una arquitectura de microservicios, creando una secuencia de Patrones de Seguridad en aquellas multicasa. Se desarrollan dos casos prácticos para mostrar la implementación conjunta de ambos catálogos. El primero tiene una doble utilidad: en el ámbito industrial ayuda a implementar patrones de seguridad en arquitecturas multicasa de forma más accesible y lógica, y en el ámbito educativo establecer un modelo de aprendizaje para comprender los patrones de manera secuencial, proponiendo un subsecuente de factores multicasa necesarios para entender y evaluar los patrones de segundo.

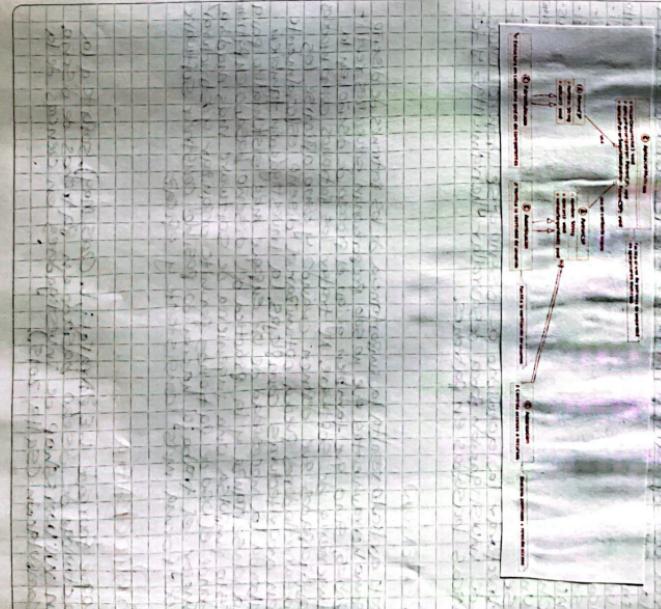
REFLEXIÓN

Este trabajo refleja la importancia de no ver los patrones que se observan como elementos aislados, sino como piezas de una colección que se integran para fortalecer la arquitectura de aplicaciones conjuntas. La investigación resulta como los factores de Seguridad y multicasa no solo coexisten, sino que deben complementarse para mayor la efectividad y la Seguridad de las aplicaciones. Esto es particularmente relevante en el contexto de la industria donde la protección de datos y la Seguridad son clave. La integración de factores permitirán abordar estos factores de manera expeditiva, des de una perspectiva holística. La propuesta de un enfoque secuencial de requisitos trae facilidad y agilidad, ayudando a los estudiantes y desarrolladores a progresar desde los

conceptos básicos hasta la implementación de patrones de Seguridad en todos un contexto multicasa. Esto integrará de manera más robusta y segura las fases de desarrollo, lugar en anticipadas más robustas y seguras y fases de monitoreo.

BIBLIOGRAFÍA

Perez Valdes, M.E. (2010). Integración de sistemas de seguridad y patrones de diseño. SRE



ARTÍCULO 9

SOPORTE A LA ACTIVIDAD DE DISEÑO BASADO EN PATRONES DE DISEÑO

RESUMEN

Este trabajo expresa como los patrones de diseño pueden servir como soporte para el proceso de diseño en ingeniería de software, permitiendo una reutilización sencilla y la que se presentan en otras disciplinas de ingeniería, la investigación destaca que los patrones de diseño en software un objeto (La Salud) como un refugio (Casa), dando apoyo a ofrecer un manejo más eficiente de las decisiones de diseño y optimizar el desarrollo. El artículo revisa las diferentes de diseño de autores como Ralph Lunde y Boschmann que consisten en el diseño tanto una actividad como un objeto que consideran el impacto del Software. Se plantea un modelo conceptual y contextual para estructurar la actividad de diseño y se proponen líneas de investigación para desarrollar herramientas y técnicas basadas en patrones.

REFLEXIÓN

Este artículo resalta la importancia de los patrones de diseño como herramientas que no solo proporcionan soluciones sencillas, sino que también guían el proceso de diseño en las ingenierías de Software. Al tratar los patrones como módulos que asumen cuando se aplican los desarrollos pueden optimizar el diseño específicamente en arquitecturas complejas. La investigación también aborda la necesidad de un enfoque más sistemático para la enseñanza y la práctica del diseño basado en patrones. Esto impulsa una ingeniería de software más madura, donde el uso de patrones permite optimizar las estructuras de diseño de forma más eficiente y efectiva.

BIBLIOGRAFÍA

ROQUE, FAVRILE. LEÓN, MATEO. (2015). Soporte a la actividad de diseño basado en patrones de diseño. In: FUNDACIÓN WILFRED KOPP DE INVESTIGADORES EN CIENCIAS DE LA CONSTRUCCIÓN. GAITÍN, 2015).

ARTÍCULO 10

ELABORACIÓN DE DISEÑO Modelo-Vista-controlador CHNC Y SU IMPLEMENTACIÓN EN JAVAFX SWING

BUSCAR

RESUMEN

El artículo se engloba en el patrón de diseño Modelo-Vista-Controlador CHNC y su implementación de Java Swing. Este patrón busca dividir una aplicación en tres componentes: el Modelo, la Vista y el controlador.

1. **Modelo:** Representa la lógica de los datos de la aplicación, desconociendo cómo será visualizada o controladas.
2. **Vista:** Se encarga de presentar la información del modelo al usuario y se actualiza automáticamente cuando el Modelo cambia.
3. **Controlador:** Gestiona la entrada del usuario y actualiza el Modelo y la vista según sea necesario.

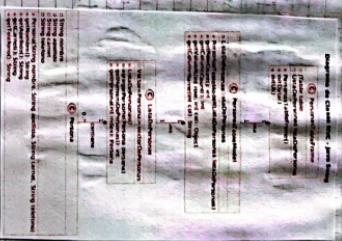
El artículo también explora la implementación de MVC en Java Swing, donde el patrón presenta una arquitectura dividida entre la vista y la lógica en el delegado de interfaz de usuario (el delegado). Esto permite que cada componente visual esté ligado directamente con su modelo, facilitando la actualización y adaptación de los datos.

REFLEXIÓN

El patrón MVC aporta organización y claridad a los sistemas complejos, promoviendo el desarrollo modular. Aunque aumenta el tiempo inicial de desarrollo, ofreciendo a largo plazo una mayor mantenibilidad y facilidad para actualizar o extender una aplicación. En Java Swing, el MVC mejora la experiencia de desarrollo al proporcionar un marco estructurado para conseguir buenas prácticas, aunque requiere una curva de aprendizaje.

BIBLIOGRAFIA

Pantaleo, E. (2004) El Patrón de diseño Modelo-Vista-controlador CHNC y su implementación Java Swing. ACTA (ppm), 2(4), 493-519.



ARTÍCULO 19

UN NUEVO PATRÓN DE DISEÑO DE SOFTWARE: PROGRAMMER'S ATTITUDE

RESUMEN

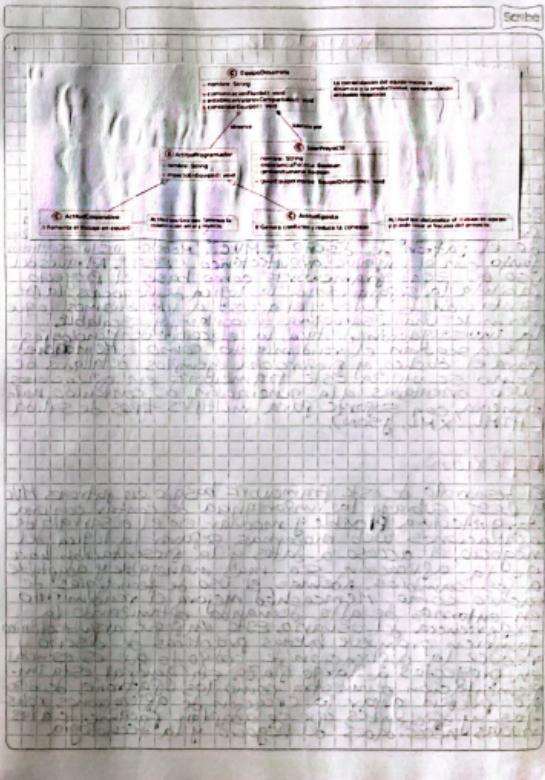
Este artículo introduce un nuevo patrón de diseño llamado "Programmer's Attitude" o "Actitud del Programador". Dard aborda problemas de comportamiento que surgen en equipos de desarrollo de software debido a diferentes actitudes de los programadores. Estas actitudes incluyen la falta de cooperación entre programadores experimentados, hasta la sobreestimación de habilidades por parte de programadores novatos. También aborda como la mala comunicación y la falta de claridad en roles y objetivos puede afectar negativamente la dinámica del equipo. La solución propuesta se centra en la consolidación de equipos de desarrollo con una comunicación fluida y valores compartidos, guiados por los principios experimentados en sectores técnicos y de gestión humana.

REFLEXIÓN

Este artículo subraya una dimensión clave en la ingeniería de Software: la influencia del factor humano en el éxito de los proyectos. Aunque los patrones de diseño suelen aplicarse a la estructura y funcionalidad del software, "Programmer's Attitude" ilustra los principios de diseño para resolver problemas de ese sector. Se recuerda que para construir software de calidad es tan importante tener un buen código como centrarse en un equipo cohesionado y motivado. La actitud de los desarrolladores y su relación con los líderes del proyecto son fundamentales para lograr resultados consistentes. Este enfoque también destaca la necesidad de fomentar gerencias técnicas y de promover la consolidación de equipos donde la cooperación, la comunicación efectiva, permitan mitigar conflictos y enfocarse en el objetivo común.

BIBLIOGRAFÍA

Pantaleo, G.G., & Poitras, D. A. "Un nuevo patrón de diseño de software: Programmer's Attitude".



NOTICIO 12

IMPLEMENTACION DE UN FRAMEWORK PARA EL DESARROLLO DE INTERFAZES WEB UTILIZANDO PREGONERES DE SILENO Y JAVASCRIPT EN MVC / REST API

RESUMEN

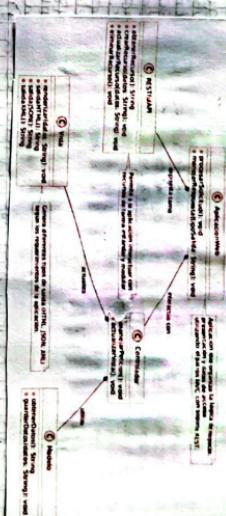
Este trabajo se centra en la implementación de un framework para desarrollar aplicaciones web utilizando el patrón MVC. Se describe la arquitectura propuesta, la ejecución de la aplicación, la presentación de datos y el acceso a los mismos, lo que permite una estructura flexible y escalable, la ejecución de tecnologías que aseguran el rendimiento óptimo, así como la ejecución de diferentes tipos de servicios en diferentes entornos, así como la generación de contenido multimedia y la ejecución de múltiples tipos de salida (HTML, XML, JSON).

REFLEXIÓN

El desarrollo de este framework basado en patrones MVC nos subraya la importancia de contar con una arquitectura flexible y modular en el desarrollo de aplicaciones web modernas. Proporciona la base del negocio, el acceso a datos y la presentación hace que las aplicaciones sean más manejables y adaptables a cambios. Además, el uso de tecnologías conocidas como Node.js mejora el rendimiento en entornos de alta demanda, optimizando la experiencia del usuario. Este enfoque arquitectónico también proporciona buenas prácticas de diseño que reduce la complejidad del código, lo que es esencial en proyectos grandes. En general, este desarrollo promete simplificar la creación de aplicaciones robustas y eficientes que se adaptan fácilmente a las necesidades del negocio y la tecnología.

BIBLIOGRAFÍA

- Bullman, E. (2011). Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño de arquitectura MVC y pregones. Doctoral dissertation, Universidad de tecnologia informatica, Belgrano. Facultad de tecnologia informatica.



APARTADO 4.3

PRUEBAS DE REGRESIÓN FUNCIONAL MEDIANTE USO DE PATRONES DE DISEÑO

RESUMEN

Este artículo propone un método innovador para automatizar la creación de casos de prueba funcionales mediante regresión, utilizando patrones de diseño. La propuesta se basa en una implementación de generación automática de los casos de prueba C-test-case (método TCT). La cual consiste en la creación de patrones de diseño que se adaptan a la funcionalidad y trae beneficios de la creación de software en el tiempo. Los resultados muestran que las pruebas de regresión se han vuelto más eficientes y económicas, generando casos de prueba nuevos y relevantes para los sistemas de negocio. Los resultados demuestran que el uso de patrones de diseño es un factor clave en la eficiencia y calidad de las pruebas de software.

REFLEXIÓN

Este trabajo demuestra el valor de los patrones de diseño más allá de su aplicación en la estructura del software, mostrándolos como herramientas clave en la automatización y eficiencia de las pruebas de software. Patrones como "Abstract factory", "Builder", "Composite", "Factory method" y "Strategy" permiten generar casos de prueba de manera efectiva, según los cambios en la funcionalidad. El efecto de usar estos patrones es que las pruebas de diseño se convierten en procedimientos más fáciles. No solo incrementa la eficiencia, sino que también permite a los equipos de desarrollo trabajar más eficientemente, cumpliendo con los requisitos, mejorando la calidad del software.

BIBLIOGRAFÍA

Letam, O. N. Generación de Pruebas de Regresión Mediante el uso de Patrones de Diseño. 2023.

NOTICIOLO 14**HEMANTICA DE GENERACION DE CODIGO A
SINTETIZAR RENDIDOS DE DISEÑO**

RESUMEN:

El documento resume EXEMPLUM, una herramienta desarrollada para generar código Java a partir de diagramas de clases UML utilizados como diseño orientados a objetos. EXEMPLUM facilita el diseño e implementación de software integrando una interfaz gráfica para crear diagramas y permitiendo la generación automática de código basado en patrones UML. Esta herramienta es especialmente útil para diseñadores novatos ya que proporciona plantillas reutilizables y fáciles de usar que facilitan el desarrollo y refactoring. Además, permite exportar modelos en formato XMI para su uso en otras herramientas de modelado, aumentando la interoperabilidad y flexibilidad en diseño libre.

REFLEXION:

EXEMPLUM resalta el valor de los patrones de diseño en la programación orientada a objetos, pues ofrecen soluciones estandarizadas para problemas comunes, mejorando la calidad y eficiencia del código. Al ofrecer plantillas y ejemplos detallados, la herramienta no solo facilita la generación rápida de soluciones robustas sino que también optimiza el tiempo de desarrollo y reduce errores. Esto permite la creación de software más robusto, fácil de mantener y escalable sin necesidad de comenzar desde cero. Además, al exportar modelos en formato XMI, permite que las arquitecturas de software creadas puedan ser utilizadas en diferentes plataformas y herramientas, lo que aumenta la interoperabilidad. Así, EXEMPLUM es una herramienta tecnológica que simplifica los procesos de diseño, permitiendo a los desarrolladores trabajar con más rapidez, calidad y accesibilidad, así como manejar más equipos de trabajo y diferentes lenguajes de programación.

NIVELES DE EXPERIENCIA.**BIBLIOGRAFIA.**

Garcés Rodríguez, G.-J., Martínez Gómez, D., R. Morales Pérez, C.-D., & Sánchez Santoni, J. (2019). Herramientas de generación de código a partir de patrones de diseño.

ASPIRACIÓN 15

ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE
AGILIZANDO PATRONES DE DISEÑO DURANTE EL PROCESO DE DESARROLLO

RESUMEN

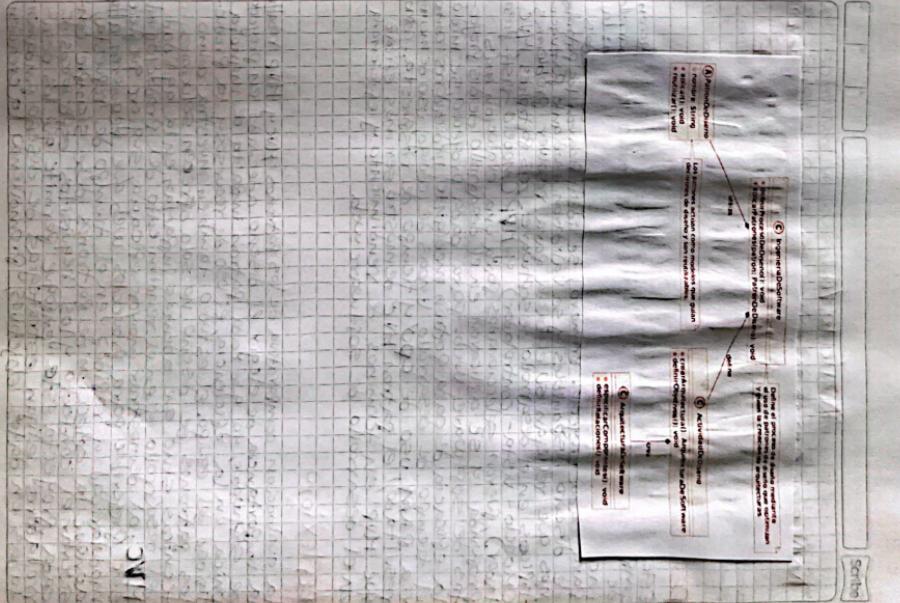
El documento analiza el uso de patrones de diseño en Masters S.A., una empresa dedicada al desarrollo de software financiero. La implementación de patrones de diseño busca mejorar la calidad del software y optimizar el proceso de desarrollo. Se establece un plan que garantiza el uso de estos patrones, los cuales son clasificados en creacionales, estructurales y de comportamiento. La investigación sugiere que los patrones de diseño mejoran la calidad del código, facilitan la colaboración, reducen errores, aumentando la satisfacción del cliente y disminuyendo los costos de mantenimiento y corrección.

REFLEXIÓN

El uso de patrones de diseño en Master S.A. no solo estructura el proceso de desarrollo sino que también ayuda a los stakeholders de desarrollo tanto al equipo como al cliente. Al promover la reutilización de código, los patrones mejoran la eficiencia y la claridad en el desarrollo. Implementar estos patrones promueve un ciclo de mejora continua en el que cada proyecto se convierte en una lección en el que cada proyecto. Se convierte en una lección en el que cada proyecto. En esencia, este enfoque no solo permite crear productos más robustos y confiables sino que también establece una base sólida para la evolución tecnológica y la satisfacción en el trabajo en equipo.

BIBLIOGRAFÍA

Pérez, M. P. (2005). Aseguramiento de la calidad del software aplicando patrones de diseño durante el proceso de desarrollo. C.



ARTÍCULO 16

ANÁLISIS COMPARATIVO DE PATRONES DE DISEÑO MVC Y MVP PARA EL RENDIMIENTO DE APLICACIONES WEB

Resumen

Este trabajo de investigación se centra en la evaluación empírica de patrones de diseño aplicando al desarrollo de software web avanzado un problema común: la incertidumbre en la selección de los patrones adecuados para cada proyecto. Primero se identificaron los patrones de diseño disponibles y se evalúaron mediante puntuaciones que reflejan su eficacia para resolver problemas típicos en el desarrollo de software. Los patrones **MVC** (Modelo - vista - controlador) y **MVP** (Modelo - vista presentador) fueron seleccionados para una evaluación más profunda, y se aplicaron en el desarrollo de dos proyectos de software web, para cada proyecto, se midió la eficacia de cada patrón mediante indicadores como el tiempo de desarrollo, líneas de código, uso de memoria RAM, uso de CPU y tiempo de respuesta. Como resultado, el patrón **MVC** demostró ser más eficaz que **MVP** en términos de rendimiento y eficiencia en el desarrollo de software web.

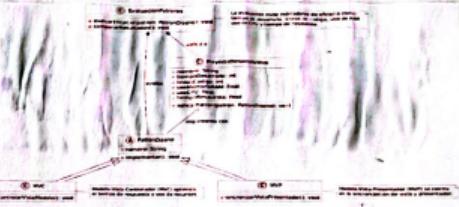
REFLEXIÓN

La evaluación comparativa de los patrones **MVC** y **MVP** resalta la importancia de elegir el diseño adecuado en proyectos de software, ya que cada patrón impacta de manera única en la eficiencia, el uso de recursos y la productividad del equipo. El estudio también muestra cómo los patrones de diseño no solo ofrecen soluciones técnicas, sino que además impactan en la agilidad del desarrollo y en el rendimiento del producto final. **MVC**, al demostrar mayor eficacia, sugiere que puede ser preferible para proyectos web que requieren alta modularidad y rapidez en el tiempo de respuesta, aunque la selección de patrones debe siempre considerar el contexto y los requisitos específicos de cada proyecto. Este enfoque permite que los desarrolladores elijan estrategias de diseño fundamentadas en datos objetivos, promoviendo la creación

de software más optimizado y adaptado a las necesidades actuales del desarrollo ágil y eficiente.

BIBLIOGRAFÍA

González González, C. E. (2023). Análisis comparativo de patrones de diseño MVC y MVP para el rendimiento de aplicaciones web.



Artículo 17

DESEARROLLO DE COMPONENTES DE SOFTWARE EN BASE AL DISEÑO MEDIANTE LA CREAÇÃO D'ICHA FAVOR. GRÁFICO INTERACTIVO

RESUMEN

El documento trata sobre la aggiugación del patrón de diseño Mediator en el desarrollo de una herramienta educativa llamada organizador gráfico interactivo (OGI). Se organizó el desarrollo en 4 fases: planificación, diseño, desarrollo y evaluación. La fase de planificación se realizó para mejorar la interacción del software. Para la fase de diseño se utilizó el patrón Mediator. Dentro de los distintos componentes del OGI, dentro las etapas de programación, herramientas de observación se comunican a través de un medidor cultural. Este permite facilitar la interacción con otros componentes permitiendo al OGI funcionar de manera independiente en conjunto con otras aplicaciones.

REFLEXIÓN

El uso del patrón Mediator en el OGI muestra como un patrón de diseño puede resolver problemas futuros y mejorar la calidad del desarrollo de un software. Al revisar el desarrollo menciono entre componentes el patrón Mediator permitió que cada parte de la aplicación se modifique y se mejore sin afectar al sistema completo. Además el uso que en una arquitectura de componentes tiene la posibilidad de la reutilización y extensión del OGI en otros contextos mejorando su longevidad y valor educativo. Esta experiencia resalta la importancia de elegir la técnica de diseño que sea óptima para la necesidad de la mayoría de los proyectos o de desarrollo concreto.

BIBLIOGRAFÍA

Almarza, F. A.; Ponce, H. R. & López, H. G. (2010) desarrollo de componentes de software en base al patrón de diseño mediador: el caso del organismo gráfico interactivo. In: J. Sánchez, congreso iberoamericano de informática. Educutora CRR. 571-578.

Aviñón 18

a objetos (CMoos). Un Congreso Argentino de
desarrollos de la construcción.

RESUMEN

ESTRUCTURAS DE DISEÑO PARA MUNDOS VIRTUALES ORIENTADOS A OBJETOS (CMoos)

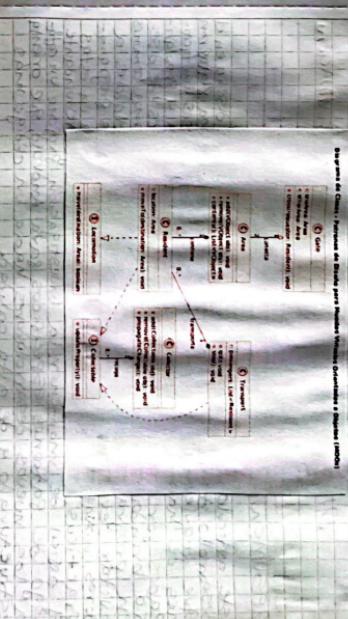
El documento explica el uso de portales de diseño para construir mundos virtuales orientados a objetos (CMoos) que proporcionan un lenguaje de programación específico para poder programar comportamientos comunes en estos portales. Los portales ofrecen servicios como crear, borrar, organizar espacios virtuales y manejar la interacción de los objetos y definir interacciones y relaciones entre estos objetos. Estos portales permiten estructurar la ejecución, la interacción y la disposición espacial en un mundo virtual y permitir la creación, la simulación y el entretenimiento.

REFLEXIÓN

La propuesta de un lenguaje de programación específico para mundos virtuales orientados a objetos proporciona una forma de diseñar aplicaciones que se ajustan a las necesidades inherentes en estos entornos. La utilización de soluciones estructuradas permite a los desarrolladores mejorar la ejecución, la claridad y la conexión en la construcción de estos mundos. Además, proporciona un lenguaje común los desarrolladores facilitan la comunicación entre equipos y permiten que los desarrolladores se beneficien del conocimiento colectivo. Esto permite lograr una mayor eficiencia del desarrollo y optimizar la construcción de mundos virtuales más accesibles y funcionales.

BIBLIOGRAFÍA

Hernández, F., Martínez, A., Fernández, A. y Pérez, A. (2009). Estructuras de diseño para mundos virtuales orientados a objetos de diseño para mundos virtuales orientados a objetos.



Análisis 19

CLASIFICACIÓN DE LOS PATRONES DE DISEÑO IDÓNEOS EN PROGRAMACIÓN ANDROID

RESUMEN.

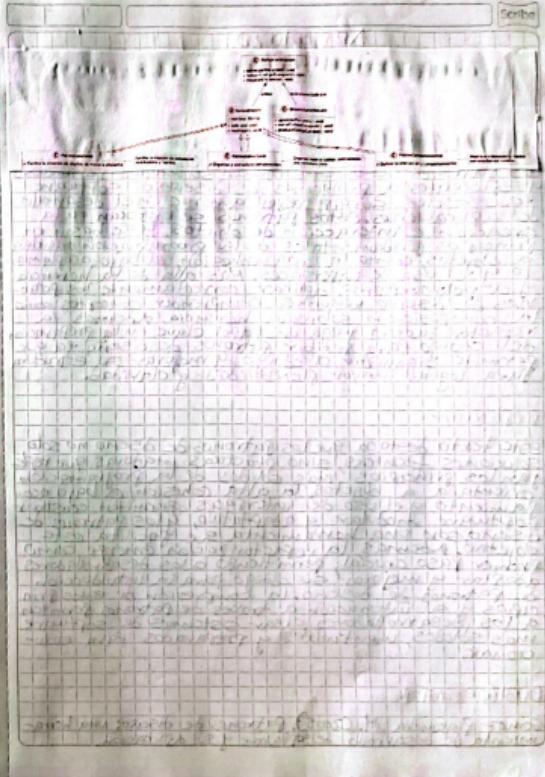
Este artículo esfuerza el valor de los patrones de diseño en el desarrollo de aplicaciones móviles en android, una plataforma que requiere soluciones eficientes debido a sus limitaciones de hardware. Inspirados en la arquitectura y popularizados por el grupo Gang of Four, los patrones de diseño brindan soluciones probadas para problemas comunes en programación, ayudando a los desarrolladores a evitar errores, como el código spaghetti. Los programadores novatos suelen caerse en errores en estas prácticas, lo cual puede resultar en código redundante y difícil de mantener. La investigación propone un catálogo de patrones específicos para android, que orienta a los nuevos desarrolladores hacia un código más estructurado y de alta calidad.

REFLEXIÓN

Este trabajo muestra cómo los patrones de diseño son esenciales para garantizar que los desarrolladores se adapten a los nuevos en la creación de aplicaciones robustas y eficientes. Los patrones no solo resuelven problemas técnicos, sino que también fomentan buenas prácticas de programación, evitando errores comunes y facilitando el mantenimiento. En una plataforma como android, la implementación de patrones de diseño permite optimizar el uso de recursos y crear aplicaciones que cumplen con los estándares de calidad necesarios en un entorno de rápida evolución.

BIBLIOGRAFÍA

- Astorga, J. A., L. Tirado, J. L. O. Ramírez, P. U. P. Mendoza, J. C. L. S. Garzón, & P. (2019). Clasificación de los patrones de diseño idóneos en programación android. Revista digital de tecnologías informáticas y sistemas, 3(1).



APUNTES

PATRONES DE DISEÑO: UNA HERRAMIENTA DE DESARROLLO DE SOFTWARE Y SU APLICACIÓN

Objetos

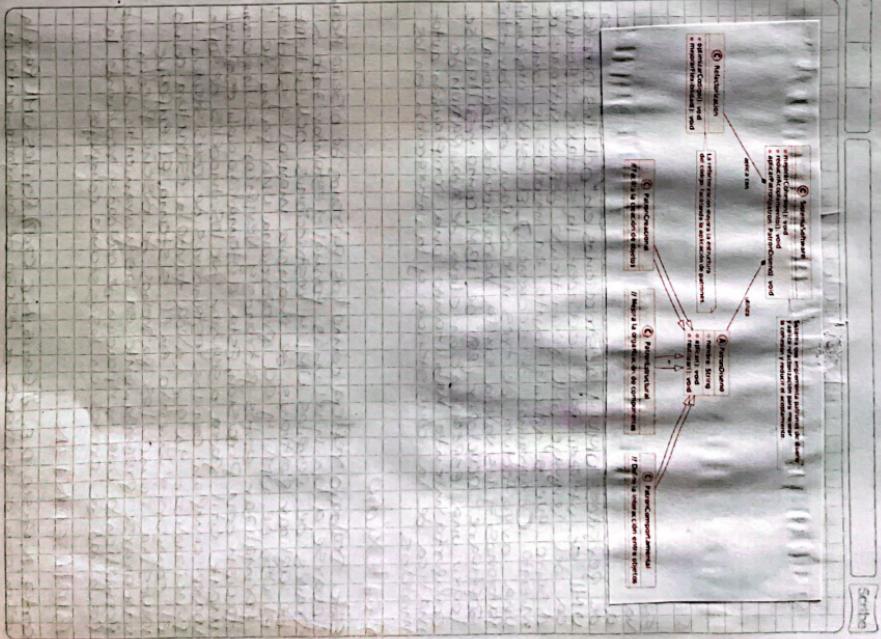
los patrones de diseño ofrecen soluciones reutilizables y eficientes a problemas de diseño de software que suelen surgir regularmente en el desarrollo y aplicaciones. Estos patrones se enfocan en la creación e interacción de objetos, y se basan en muchos fundamentales de programación orientada a objetos como la alta coherencia y bajo acoplamiento y uso de interfaces. Mas allá de su relevancia en el polimorfismo, aplicar correctamente los patrones de diseño requiere comprender conceptos como la composición sobre la herencia, la cual juega un papel clave en la implementación de refactoring con la ayuda de patrones de diseño que permiten organizar el código y mejorar su estructura para lograr mayor flexibilidad y claridad.

Reflexión

Este texto destaca que los patrones de diseño no solo son soluciones técnicas, sino también modos que nos sirven para abordar a efectos la alta cohesión y bajo acoplamiento que el uso de interfaces permite construir software flexible y adaptativo. Los patrones de diseño son una herramienta que facilita este objetivo. Ademas, la refacción de código, conocida como refactor paso crucial, permitiendo a los desarrolladores mejorar su código para la introducción de cambios de diseño. La compresión de estos principios y su utilización a través de patrones permiten a los desarrolladores crear sistemas de software más sólidos, mantenibles y preparados para evolucionar.

BIBLIOGRAFIA

Gómez, F. (2005). Patrones de diseño: una herramienta de desarrollo de software y su aplicación.



ARTÍCULO 21

RESPONSOS UML KNOWN: DISEÑO DE PRIMEROS MODELOS DE COMPORTAMIENTO

COMPORTAMIENTO

REFLEXIÓN

Este trabajo supone un enfoque para especificar y validar patrones de diseño de conocimiento que acuerdo con la clasificación se define mediante el uso de servicios UML y restricciones en OCL. Los servicios UML se utilizan para extender la sintaxis y semántica del lenguaje UML permitiendo definir subtipos de diseño que se aplican tanto en modelos de comportamiento (subtipos de secuencia). El estudio implementa una arquitectura de tres niveles para organizar estos factores en los servicios en la herramienta Rational Software Developer (RSD) de IBM. Como resultado se logra una especificación clara de cada patrón de comportamiento facilitando una verificación automática de la consistencia y la interacción en los modelos.

REFLEXIÓN

Este trabajo refleja cómo los patrones de diseño deben labrarse a un nivel más formal y estandarizado o nuevos de patrones UML y restricciones OCL. En lugar de seguir factores de manera manual y subjetiva, el uso de UML y OCL proporciona una forma de formalizar estos factores y verificar su implementación con precisión. Esta autómata hace que los modelos de software cumplan con los estándares de consistencia y funcionalidad ya que validan tanto la estructura como el comportamiento en el sistema. El enfoque muestra cómo los patrones de diseño pueden integrarse progresivamente en las herramientas de modelado permitiendo una verificación de calidad y reduciendo la posibilidad de errores en los sistemas complejos.

BIBLIOGRAFÍA

CONTRERAS ESCO, D. E. Y GARCÍA, M. E. 2012. Resúmenes UML para la definición de patrones de diseño de comportamiento. In:

ARTÍCULO 22

LOS PATRONES DE DISEÑO COMO HERMANAS
EN UNA LINEA: UNA REDACCIÓN DEL PROFESORADO

RESUMEN

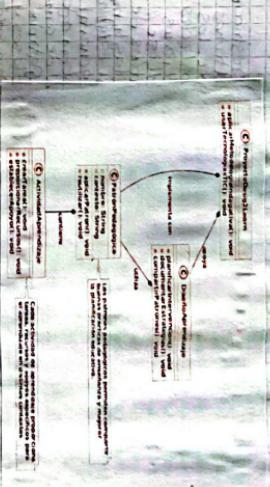
Este artículo explica el uso de patrones pedagógicos en el diseño de actividades con objetivos específicos y conseguir buenas prácticas educativas que se adapten a diferentes contextos. Los patrones pedagógicos también conocidos como patrones de diseño de aprendizaje, buscan documentar el conocimiento y las estrategias utilizadas por docentes para mejorar la planificación y facilitar la creación de interacciones con las más efectivas. Inspirados en la metodología de los patrones de diseño software, estos patrones ofrecen una estructura revisable que sirve a los educadores a desarrollar actividades de aprendizaje adaptadas a contextos específicos. Un investigación describe el protocolo "Design Learning" que selecciona estos patrones en el contexto universitario usando tecnologías y metodologías basadas en el aprendizaje nádativo.

REFLEXIÓN

Este trabajo resalta la importancia de adoptar los principios de los patrones de diseño al ámbito educativo permitiendo a los docentes avanzar soluciones efectivadas para situaciones comunes de enseñanza. Los patrones pedagógicos no solo mejoran la eficiencia en la creación de actividades, sino que también fomentan la colaboración entre profesores quienes pueden compartir y reutilizar escenarios de aprendizaje en comunidades de práctica. Además, el centro común es lograr combinar y una estructura formal los docentes ofrecen regularmente experiencias exitosas y ajustar sus interacciones. Según el contexto y las necesidades de los estudiantes. Esto demuestra como los patrones de diseño pueden trasladar el desarrollo de software ofrecer un nuevo brillar en otros campos, como la educación, promoviendo una enseñanza más organizada, efectiva, y colaborativa.

BIBLIOGRAFÍA

Gros, Salvat, B.; Escrivet Roig, A.; Martínez Martí, M. (2016). Los patrones de diseño como herramientas para diseñar la práctica del profesorado. *Revista de la Facultad de Tecnología Educativa - RELETIC*, 2016, Vol.15, núm. 3, p.77-95



ARTÍCULO 23

ESTANDARIZACIÓN DE PATRONES DE DISEÑO DE COMPORTAMIENTOS

RESUMEN

Este trabajo explora el uso de "perfíles UML" y "restricciones UML" para especificar y estandarizar patrones de comportamiento en el desarrollo de software. Los patrones de diseño de comportamiento, según la clasificación de COF, ayudan a gestionar interacciones complejas entre objetos. Mediante perfiles UML se propone un marco para definir estos patrones de manera precisa y reusable, aprovechando las herramientas UML ya existentes. La investigación establece una arquitectura de tres niveles: un nivel general para todos los patrones, un nivel intermedio (catalogo de patrones, estructurales, creacionales, de comportamiento) y un nivel específico para cada patrón. La metodología incluye la identificación de clases y objetos clave, la definición de estereotipos para estos y la aplicación de restricciones para asegurar el comportamiento deseado.

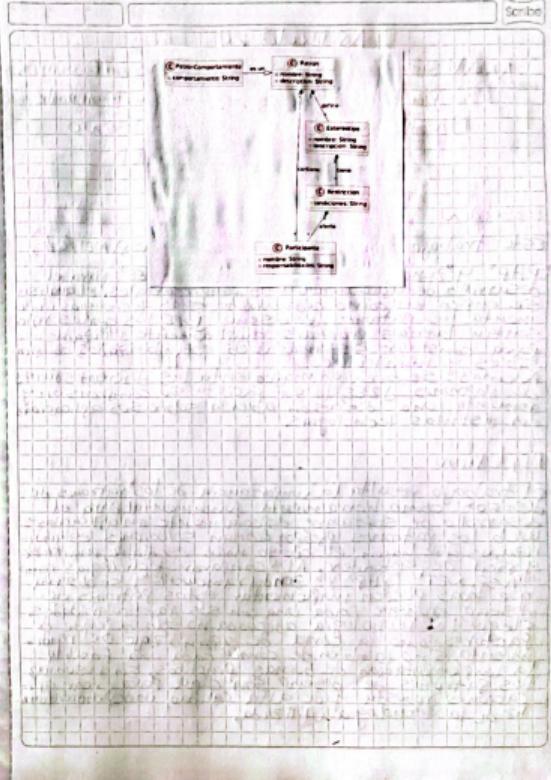
REFLEXIÓN

Este trabajo muestra cómo los perfiles UML y las restricciones UML pueden elevar los patrones de diseño a un nivel más estandarizado y formal. A través de esta metodología, los patrones de comportamiento se vuelven más accesibles y precisos, ayudando a los desarrolladores a aplicarlos de manera consistente.

Este enfoque también permite evitar la necesidad de crear nuevas herramientas aprovechando el poder de UML para representar y validar patrones complejos. La arquitectura en niveles propuesta no solo permite especificar patrones, sino que también permite su reutilización y claridad, facilitando la creación de software con una estructura sólida y organizada.

BIBLIOGRAFÍA

Cortés, L., Gómez, A. G., & Silesco, D. E. (2011). Formalización de patrones de diseño de comportamiento. In XII Workshop de Investigadores en Ciencias de la Computación.



ARTÍCULO 24

ANÁLISIS DE LA ADECUACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN AL NUEVO DESARROLLO.

BASADO EN PATRONES DE DISEÑO

J2EE DE ALTO NIVEL

RESUMEN

Este trabajo compara tecnologías como ASP.NET/C#, PHP y Zend Framework frente a J2EE para el desarrollo de aplicaciones web empresariales, el análisis se enfoca en cómo cada una de estas tecnologías implementa los patrones de diseño J2EE, los cuales representan buenas prácticas ampliamente utilizadas para construir sistemas web estandarizados confiables y mantenibles. Además, se crea un mapeo entre las diferentes plantas de patrones J2EE, lo que facilita su comprensión y fomenta una discusión abierta sobre sus aplicaciones en distintas tecnologías.

REFLEXIÓN

El trabajo resalta la importancia de los patrones de diseño como herramienta fundamental para el desarrollo de software, especialmente en aplicaciones Web. Los patrones proporcionan soluciones estandarizadas que hacen el código más escalable, mantible y fácil de entender. Al comparar tecnologías como ASP.NET, PHP y Zend Framework, se observa que todos pueden implementar estos patrones de diseño, aunque cada una con sus particularidades. Elegir la tecnología adecuada no solo depende de factores como la curva de aprendizaje del rendimiento, sino también de cómo cada una facilita la implementación de buenas prácticas. En resumen, los patrones de diseño son clave para crear aplicaciones robustas y de alta calidad, independiente de la tecnología utilizada.

BIBLIOGRAFÍA

Morales Franco, E. M. (2009). Análisis de la adecuación de lenguajes de programación Web a un desarrollo basado en patrones de diseño J2EE de alto nivel.



ARTÍCULO 25

ANÁLISIS DEL IMPACTO DEL USO DE PATRONES DE DISEÑO EN LA FASE DE MONITOREO Y MANTENIMIENTO

RESUMEN

El mantenimiento de software es esencial para asegurar que los sistemas sigan cumpliendo con las necesidades de los usuarios a lo largo del tiempo. Se identificaron tres tipos de mantenimiento: **correctivo, preventivo y predictivo**. El uso de **patrones de diseño** que son soluciones genéricas para problemas comunes facilita el mantenimiento, al implementar la compresión del código y prevenir el tiempo dedicado a las modificaciones. La investigación de estudiantes de la Universidad distrital analizó el uso de **12 patrones de diseño de GoF** en empresas de software. Mediante una encuesta aplicada a 10 profesionales se buscó cuáles fueron efectuadas en términos de **escalabilidad, evolución y soporte**, además de proponer un modelo jerárquico para un servicio web de evaluación de buenas prácticas.

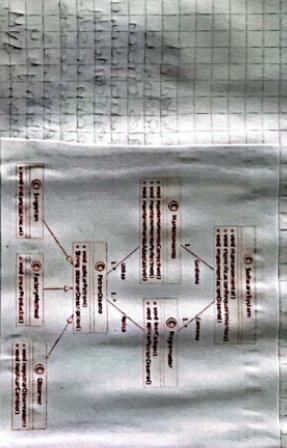
REFLEXIÓN

Los patrones de diseño juegan un papel fundamental en la mejora de la calidad del software especialmente cuando se refiere a su mantenimiento a lo largo del tiempo. Algunos de los patrones de diseño más utilizados son: **Strategy**, **Observer**, **Composite**, **Factory**, **Singleton**, **Decorator**, **Adapter**, **Facade**, **Bridge**, **Proxy**, **Command**, **Visitor**, **Iterator**, **Mediator** y **State**. Los patrones permiten que el código sea más sencillo, sencillo de modular y fácil de modificar. Esto no solo permite a los programadores cuando se requiere hacer cambios o mejoras en el sistema. Simplemente también refuerza el tiempo necesario para entrar en contacto con las empresas.

La investigación que se llevó a cabo en este estudio resaltó la importancia de los patrones de diseño en la **escalabilidad y evolución** de los sistemas, ya que entornos de software cambiantes o cambiando de adaptarse a nuevas necesidades o tecnologías cruciales.

BIBLIOGRAFÍA

De León Correa, M.: *Hijos de Bernain I.-P. 3º Cuatrimestre año 2019-2020. Análisis del impacto del uso de patrones de diseño en la fase de mantenimiento.*



Artículo 26

GUÍA PRÁCTICA PARA EL USO DE PATRONES DE DISEÑO EN EL DESARROLLO DE SOFTWARE

RESUMEN

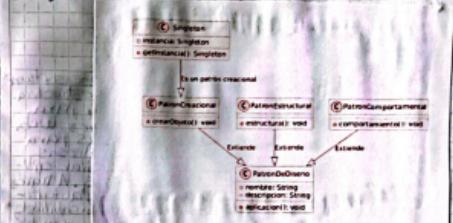
En la búsqueda continua por mejorar la calidad del software, una de las estrategias más efectivas es el uso de patrones de diseño, los cuales ofrecen soluciones reutilizables a problemas recurrentes en el desarrollo. Estos patrones no solo se enfocan en la reutilización del código, sino en la creación de soluciones estructurales que resuelven problemas comunes de manera más eficiente. Grandes plataformas como Microsoft con su Enterprise Library y Java con sus complementos de la plataforma J2EE, han desarrollado sus propios patrones de diseño. Sin embargo, estos patrones, aunque útiles a menudo se convierten en cajas negras, difíciles de comprender para los usuarios. El proyecto de titulación busca resolver este problema al ofrecer una guía comprensible sobre los patrones de diseño basada en los principios del libro "Design Patterns" de Erich Gamma y otros autores. Para exemplificar la aplicación de estos patrones, se utilizó UML, Java y MySQL.

REFLEXIÓN

Los "patrones de diseño" son una de las mejores herramientas para asegurar que el software sea más flexible, fácil de mantener y escalable, en un mundo donde los problemas comunes se repiten una y otra vez, estos patrones permiten a los desarrolladores tomar soluciones probadas, y adaptarlas a sus necesidades, lo que mejora tanto la calidad como la eficiencia del trabajo. El uso de patrones de diseño también permite la fácil comprensión entre los desarrolladores, facilitando la colaboración y la comprensión del código. Sin embargo, es fundamental que existan ocajas normas como a veces ocurre con las implementaciones de plataformas populares.

BIBLIOGRAFÍA

Aldas Mena, D. E., & Andrade Cadena, M. A. (2011). Guía práctica para el uso de patrones de diseño en el desarrollo de software (Bachelor's thesis, Quito/2011).



ARTICLE 27

INCIDENCIA DE LOS PATRONES DE DISEÑO EN SUS VALORES EN LOS SEGUIMIENTOS RE

NONCARNES WEB

13W015337

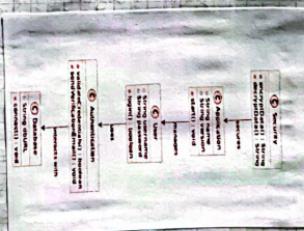
En conclusión, se observa que el manejo de los portales de discusión se sitúa entre la moderación y la administración. Los moderadores son las personas que tienen la responsabilidad de garantizar la seguridad de los visitantes y de las páginas web. Los administradores son las personas que tienen la responsabilidad de mantener las páginas web funcionando y seguras para los visitantes. Los administradores también tienen la responsabilidad de garantizar que los visitantes sigan las reglas establecidas por el sitio web. Los administradores también tienen la responsabilidad de responder a las preguntas y problemas de los visitantes. Los administradores también tienen la responsabilidad de mantener las páginas web actualizadas y relevantes. Los administradores también tienen la responsabilidad de mantener las páginas web seguras y protegidas. Los administradores también tienen la responsabilidad de mantener las páginas web accesibles para todos los visitantes.

Reflection

Los softones de diseño no solo contribuyen a la eficiencia y a la organización del codijo sino que también son parte esencial en la ejecución y desarrollo de las instrucciones. Al estructurar el codijo mediante tipos y subtipos de diseño, el programador puede trabajar en un entorno más amigable, donde las reglas de diseño se convierten en guías claras y realistas. Estos softones proveen los equipos de desarrollo a crear software buenas que sean más resistentes a los errores. Los softones de diseño, tan solo tienen la función de suministrar sugerencias o mejoras o novedades de calidad al código. Por lo tanto la responsabilidad de los softones de diseño no se les nombra por calidad del software. Si no que también, hace que las soluciones sean más seguras y menos propensas a fallos de seguridad.

BÍBLICA

MERIBEL, Valenciano J. J. - 3 caballos - Munoz, T.D. - G.R.P.M.
y Sindicato de los Profesionales de Software - Seguridad de operaciones web - REINVESTIGAR, B.C.T.,
2360 - 259.



ARTÍCULO 28

APLICACIÓN DE PATRONES DE DISEÑO ESTRUCTURADO PARA EL MODELLAMIENTO DE CLASES DE LOS SISTEMAS EMPRESARIALES

RESUMEN

En este trabajo de investigación se examinan los patrones de diseño creados y aplicados específicamente en el modelamiento de clases para sistemas empresariales. La investigación destaca la creciente complejidad de los sistemas modernos y la necesidad de patrones más específicos, ya que los patrones generales existentes no abordan de manera eficiente todos los problemas de diseño, especialmente en el contexto de sistemas de información administrativa. El estudio de enfoque cuantitativo, correlacional y preexperimental se realizó con 76 alumnos de la Universidad Católica de Santa María. Se utilizaron encuestas y análisis estadísticos para validar la relación positiva entre la aplicación de estos nuevos patrones de diseño y la mejora en el modelamiento de datos en los sistemas empresariales. Los resultados muestran que los patrones desarrollados y validados son efectivos en el diseño y modelamiento de sistemas.

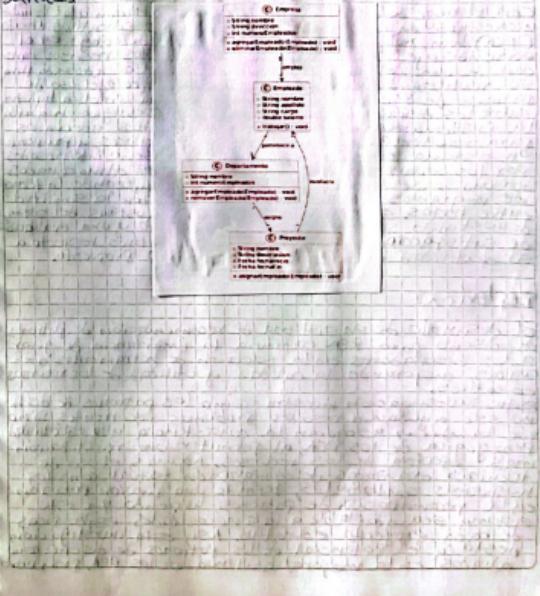
REFLEXIÓN

Este estudio refleja la necesidad de evolucionar y adaptar los patrones de diseño para resolver problemas más específicos que surgen en el desarrollo de software. Aunque los patrones generales son útiles, la especificidad y la adaptación a contextos particulares, como los sistemas empresariales, son fundamentales para crear soluciones más eficientes. El trabajo muestra, cómo los patrones creados no solo tienen un impacto directo en la mejoría del modelado de datos, sino que también validan su utilidad práctica en aplicaciones reales. Esto resalta la importancia de seguir investigando y adaptando los patrones de diseño para abordar desafíos emergentes en la industria del software.

especialmente en áreas tan críticas como los sistemas de información administrativa, donde la precisión y la optimización del diseño son claves para el éxito a largo plazo.

BIBLIOGRAFÍA

Zúñiga Carrasco, M.M. (2020). Aplicación de patrones de diseño estructurados para el modelamiento de clases de los sistemas empresariales.



Artículo 29

PATRÓN DE DISEÑO, RECURSOS EDUCATIVOS ABIERTOS; SISTEMAS DE RECOMENDACIÓN

Resumen

Con el auge de las tecnologías de la información y las comunicaciones en la educación, los recursos educativos abiertos (REDA) han gozado de como herramientas de apoyo en el aprendizaje. Sin embargo, muchos de estos recursos carecen de una calidad pedagógica y didáctica adecuada para su utilización. Para mejorar el diseño y el uso de patrones de diseño se ha considerado en una práctica habitual entre los desarrolladores de REDA, esta investigación analiza la problemática de encontrar los patrones adecuados para resolver problemas de diseño específicos y propone un sistema para resolver problemas de diseño interresados por los usuarios con problemas similares al implementado en una base de datos de patrones, empleando la teoría de n-gramas contextuales para ampliar la similitud entre textos. Se realizan pruebas exhaustivas del sistema de satisfacción mostrando resultados positivos respecto a su efectividad en alcanzar el objetivo propuesto.

Replikation

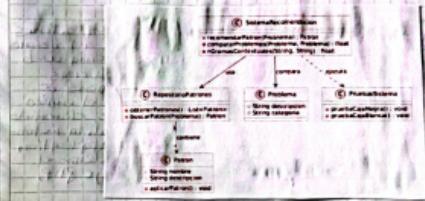
El desarrollo de un sistema de recomendación de patrones de diseño para REDA evidencia la importancia de garantizar a los desarrolladores en la selección de herramientas adecuadas, mejorando la calidad y utilidad de los recursos educativos.

Esta investigación muestra cómo los patrones de diseño no solo ayudan a resolver problemas recurrentes sino que también aseguran la consistencia y la efectividad pedagógica de los REDA al permitir una reutilización más efectiva de los recursos. Además, el uso de técnicas avanzadas de comparación textual, como los n-gramas contextuales, permite una mayor precisión en la recomendación de patrones, haciendo que este sistema sea adaptable a las necesidades específicas de cada proyecto educativo. Así, esta investigación destaca el potencial de los patrones de diseño como avances en la mejora continua de los REDA, promoviendo

una educación más accesible y de calidad.

BIBLIOGRAFÍA

Arteaga Gómez, J., & Menéndez Martínez, J. (2015). Sistema de recomendación de patrones de diseño para Recursos Educativos Abiertos (Coachelos tesis), Universidad de las ciencias informáticas, facultad 4).



ARTÍCULO 30

INTEGRACIÓN DE PROYECTOS DE SOFTWARE

MEDIANTE EL USO DE ENTORNOS Y ARQUITECTURAS DE DESARROLLO

RESUMEN

Este trabajo busca servir como una guía para quienes desean adentrarse al desarrollo y diseño de aplicaciones web, utilizando los pasos de establecimiento previamente desarrollados para implementar las prácticas. Algunas de las técnicas de diseño que se presentan incluyen el uso de interfaces de usuario y técnicas avanzadas para efectuar aplicaciones cliente-servidor. Una de las principales ventajas para el desarrollo es configurar un servidor HTTP local, la cual facilita el desarrollo de aplicaciones. Una vez configurada la aplicación, se puede acceder a través de la interfaz gráfica.

REFLEXIONES

La investigación destaca cómo los patrones de diseño facilitan la estructuración y optimización del código, lo que resulta especialmente útil en el desarrollo de aplicaciones web. Al usarlos en conjunto con técnicas como RAD, se logra una integración armónica de código y buena arquitectura en el diseño de software. Los patrones no solo resuelven problemas comunes en la programación, sino que también ofrecen un marco estructurado para la colaboración y el mantenimiento a largo plazo. El uso de máquinas virtuales como su parte fundamental los desempeña bien, centrándose exclusivamente en el código y en el diseño del sistema. Sin embargo, sería una configuración de servicios en conexión con esta combinación de patrones de diseño. RAD y virtualización ofrecen un enfoque moderno y eficiente que facilita la creación de aplicaciones robustas, escalables y rápidamente desarrrollables.

BIBLIOGRAFÍA

Moreno López, E.A. (2013) integración de proyectos de

Software mediante el uso de patrones y arquitecturas de desarrollo (Doctoral dissertation).

