

Patrones de Software arquitectura ITS

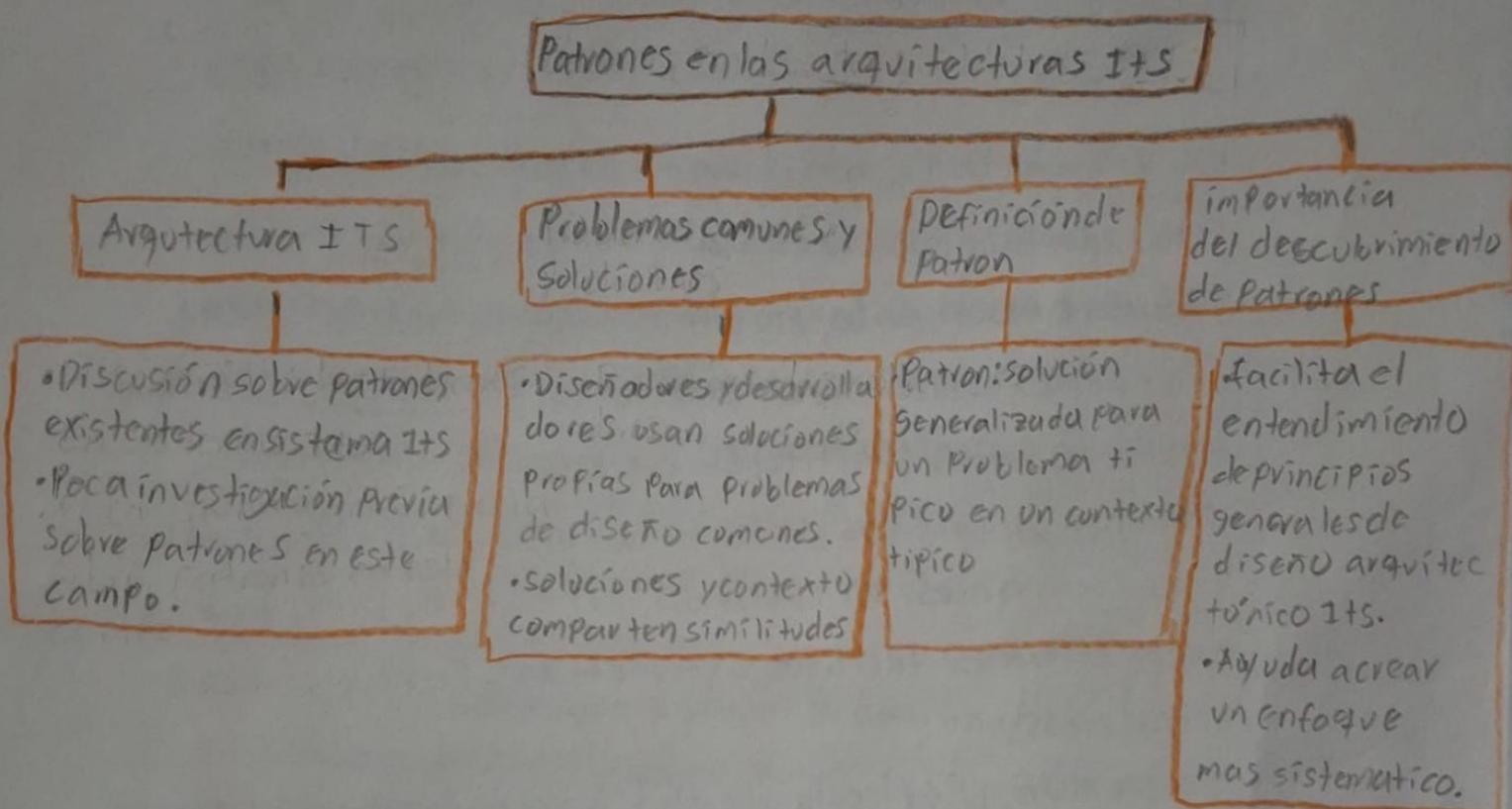
1) Resumen: El artículo analiza el uso de patrones en las arquitecturas de sistemas tutores inteligentes (ITS), destacando su escasa atención por los investigadores. A pesar de los problemas comunes entre sistemas, diseñadores suelen crear soluciones únicas. Sin embargo, estas comparten similitudes que permiten identificar patrones, entendidos como soluciones generales a problemas recurrentes en contextos específicos. Descubrir estos patrones facilita sistematizar principios arquitectónicos.

2) Reflexión: El artículo resalta la importancia de identificar patrones en el diseño sistemas tutores inteligentes (ITS) para optimizar su desarrollo. Aunque los diseñadores suelen crear soluciones únicas para problemas comunes, reconocer patrones permite reutilizar enfoques efectivos y mejorar la calidad y coherencia de los sistemas. La falta de atención a estos patrones limita el potencial de los ITS. Incorporarlos al diseño arquitectónico sistematiza el proceso, haciéndolo más eficiente y efectivo.

3) Bibliografía

Autor: Devedzic, Vladan / Harrer Andreas
Año: 2005

4) Gráfico:



• Una arquitectura de componentes para el modelado de redes en SIG

1) Resumen: Nos habla de la arquitectura de dominio RED. Permite modelar redes de distribución en sistemas de información geográfica, aprovechando el concepto de grafo espacio-temporal. RED se enfoca en subdominios como redes hidráulicas, eléctricas y oleoductos, y cuenta con componentes para modelado, cálculo y optimización. Permite agregar nuevos atributos a las entidades en tiempo de ejecución, y utiliza una arquitectura en capas que integra elementos middleware y visualización. La implementación está organizada en paquetes que agrupan componentes genéricos y específicos. RED fue evaluada mediante un prototipo y aporta una solución genérica para el desarrollo de aplicaciones SIG en este dominio. El artículo también describe el método utilizado para su desarrollo.

2) Reflexión: La arquitectura de dominio RED aborda la necesidad de contar con soluciones integrales para el desarrollo de aplicaciones SIG en redes de distribución. Su enfoque en subdominios específicos y el uso de componentes reutilizables para modelado, cálculo y optimización, la convocaría en un valioso aporte.

El método de desarrollo aplicado, que separa la ingeniería de dominio de la ingeniería de aplicación, destaca la importancia de contar con procesos adecuados para el diseño de arquitecturas de software. Esto facilita el reuso y evolución de la arquitectura. En general, el artículo ilustra cómo una arquitectura de dominio bien concebida puede simplificar significativamente el desarrollo de aplicaciones en áreas específicas. Esto representa una lectura valiosa para la ingeniería de software orientada a dominios.

Sistemas de información geográfica (SIGs)

- capacidades:
modelar y manipular
redes espaciales
- Definir objetos con
figuras geométricas,
imágenes y tablas de datos
- Limitaciones:
fallan al modelar redes

Aplicaciones especializadas

- Problemas:
No modelan
aspectos temporales
y especiales de redes

Arquitectura RED

- Solución:
Permite modelar
y manipular redes
especiales y
temporales.
- facilita el desarrollo
de aplicaciones SIGs
reutilizables.

Palabras clave

- modelado
de redes de
distribución.
- Arquitecturas
de componentes
reutilizables

• Patrones de diseño GOF (the Gang of Four) en el contexto de Procesos de desarrollo de aplicaciones orientadas a la web

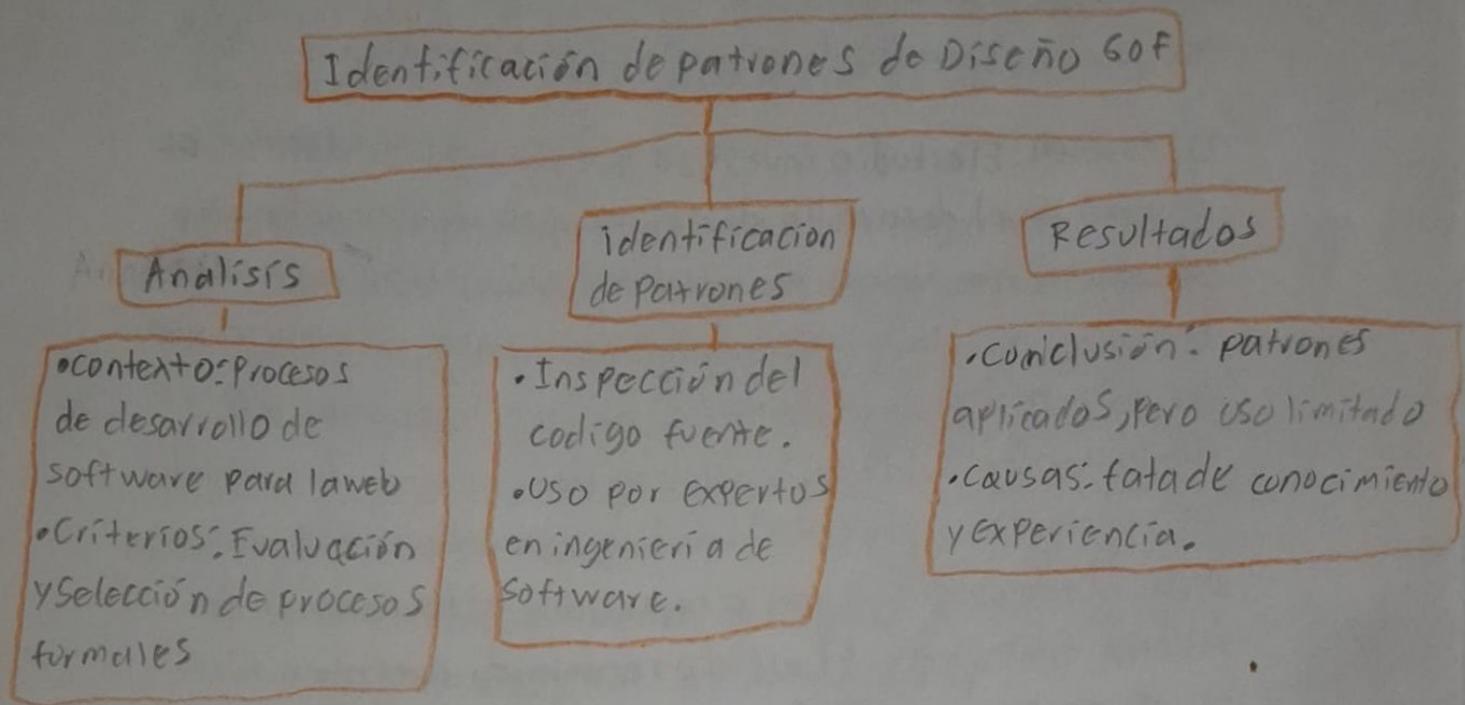
1) Resumen: El estudio investigo que patrones de diseño GOF se usan en el desarrollo de aplicaciones web en colombia. Se analizaron varios proyectos y se encuestó a expertos del tema. Los resultado mostraron que los patrones creacionales son los más populares, seguidos por los estructurales y de comportamiento. El uso de estos patrones mejora la calidad y eficiencia del software. Sin embargo, se encontró que muchos desarrolladores desconocen o no aplican estos patrones de manera consistente. El estudio recomienda fomentar la educación en patrones de diseño y el uso de herramientas que faciliten su implementación.

2) Reflexión: Los patrones de diseño son como los manuales de construcción que usan los mejores arquitectos para crear edificios super resistentes y bonitos. En la programación, estos manuales nos ayudan a crear videojuegos y APPS increíbles. ¡Es como tener un super kit de construcción para crear videojuegos y APPS increíbles. Al igual que tu castillo de lego, tus programas serán más fuertes y divertidos si usas estos manuales especiales. Este estudio nos dice que muchos programadores colombianos ya están usando estos manuales y que hacen que sus programas colombianos ya están usando estos manuales y que hacen que sus programas sean mejores. Así que la próxima vez que crees algo en la computadora, recuerda que puedes usar estos trucos para hacerlo aún más genial!

3) Bibliografía:

Autor: Carlos Guerrero, Johanna Suárez y Luz Gutiérrez
Año: 2013

HD Gráfica



• Aplicación de patrones de diseño estructurales para el modelamiento de clases de los sistemas empresariales

1) Resumen: El artículo se centra en la mejora de la calidad del diseño de software a través de la aplicación de Patrones de diseño estructurales.

Estos patrones son como plantillas o recetas predefinidas que ofrecen soluciones probadas para problemas comunes en el desarrollo de software. Al utilizar estos patrones los programadores pueden construir sistemas más robustos, flexibles y fáciles de mantener.

Con el objetivo principal del estudio es determinar si el uso de estos patrones de diseño en sistemas empresariales.

2) Reflexión: Este estudio nos muestra como los patrones de diseño pueden hacer una gran diferencia en la calidad software. Basicamente, estos patrones funcionan como soluciones prácticas para problemas comunes, permitiendo que el código sea más robusto, flexible y fácil de mantener. La investigación destaca que aplicar estos patrones ayuda a que los modelos de datos sean más claros y coherentes, facilita encontrar y corregir errores y hace que, en el futuro, sea más sencillo modificar el sistema.

3) Bibliografía:

Autor: Manuel Mariano Zúñiga

Año: 2020

Patrones de diseño y modelamiento de datos

interés en patrones de diseño

- Aumento debido a sistemas más complejos
- Problema: patrones generales, falta de específicos para modular datos en sistemas administrativos.

• Investigación

- Enfoque: cuantitativo, aplicada, correlacional, pre-experimental.
- Participantes: 76 alumnos de ingeniería de sistemas.

• Resultados:

- Patrones creados y validados.
- Relación positiva en el modelamiento de datos en sistemas empresariales.

• Una ontología para la representación de conceptos de diseño de software.

1) Resumen: Este trabajo apunta a crear una herramienta poderosa para los desarrolladores de software, proporcionando una base sólida para la toma de decisiones de diseño y la creación de sistemas de alta calidad, lo que facilitará la colaboración entre equipos de desarrollo, estandarización para solo utilizar un lenguaje común para hablar del diseño del software, utilizando la ontología permitiendo una estructura formal que relacione todos estos conceptos, permitiendo búsquedas y consultas precisas.

2) Reflexión: La propuesta de crear una ontología para el desarrollo. Al unificar los conceptos y las mejores prácticas en un solo lugar, se facilita la comunicación entre desarrolladores, se promueve la reutilización de código y se asegura una mayor calidad en los proyectos. Esta iniciativa no solo beneficiaría a los programadores experimentados, sino que también sirve como una herramienta de software para principiantes.

3) Bibliografía:

Autors: Gloria Giraldo, Juan Acevedo, David Moreno
Año: 2017

Nuevos conceptos:

Diagramas de modelado
y Patrones.

Problema: Exceso de
información dígital
demasiado

• falta de estructura;

poca iniciativas
para organizar el
conocimiento.

• Ejemplo: Éxito de
GOF basado en una
pilasión.

- Crear una ontología
para estructurar y
organizar conceptos
de diseño
• Enfoque en diagramas
y Patrones

. Paquete java para la integración de herramientas de pruebas de software basado en patrones de software

1) Resumen: Este artículo de investigación explora varios patrones de diseño que ayudan a estructurar y organizar mejor el desarrollo de software. Los patrones creacionales facilitan la creación controlada de objetos, como el singleton, que asegura una única instancia, o el factory, que permite crear objetos de una familia de clases. Los patrones estructurales como el adapter y el facade optimizan las relaciones entre clases para crear sistemas más flexibles y extensibles.

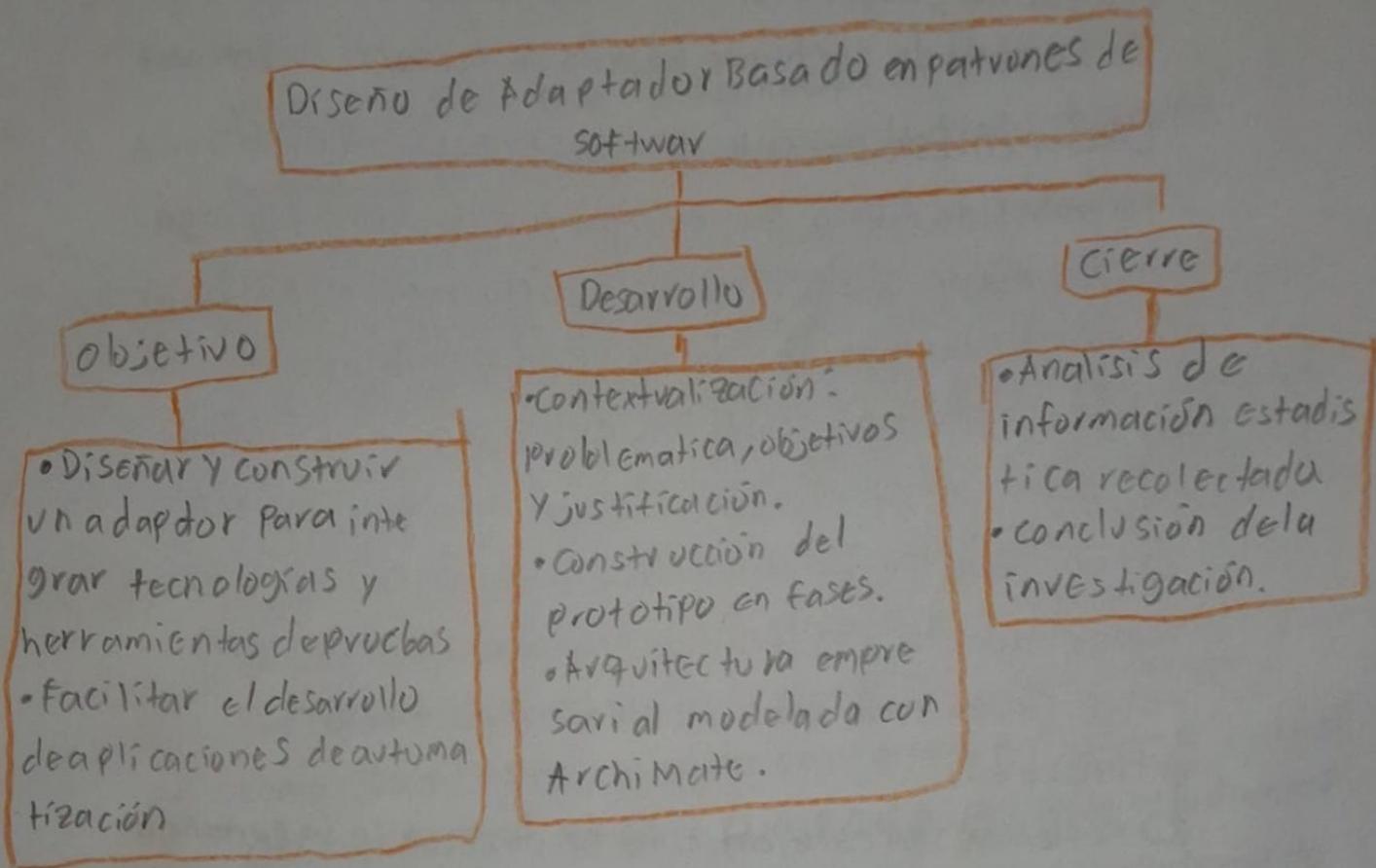
2) Reflexión: En este artículo se destaca la importancia de los patrones de diseño para optimizar el desarrollo de software, abordando la creación, estructuración y comportamiento de objetos en un sistema. Los patrones creacionales, como el singleton o el factory, ofrecen mecanismos para manejar la creación de instancias de manera controlada, lo cual resulta útil en contextos donde la creación directa puede generar problemas de rendimiento o estructura.

3) Bibliografía:

Autor: Yhan Edinson Estevez Mendoza

Año: 2079

4D Graficas:



Entorno virtual para la formación de tecnologos e ingenieros de sistemas en patrones de diseño de software

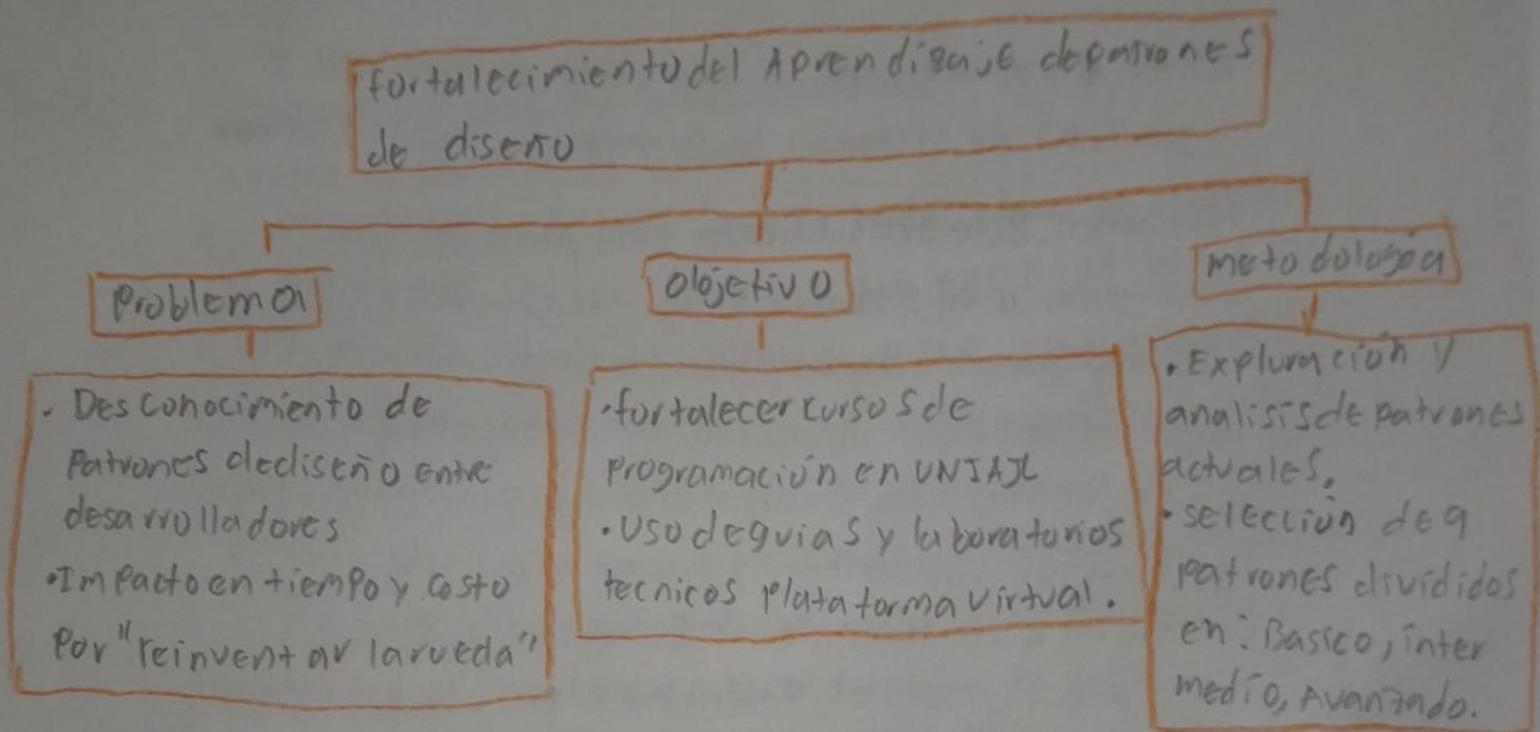
1) Resumen: Este artículo está dirigido a estudiantes de ingeniería en sistemas de la UNIAJC que desean profundizar en patrones de diseño de software. A lo largo del texto, se presentan nueve patrones esenciales del reconocido libro "gang of four", organizados en tres niveles: básico, intermedio y avanzado. Cada nivel se enfoca en distintos tipos de patrones de diseño: los creacionales, como el "Factory Method", los estructurales, como "Composite", y los de comportamiento, como "observer".

2) Reflexión: Este artículo es una guía útil para estudiantes de ingeniería en sistemas que desean aprender patrones de diseño. Estos patrones ayudan a resolver problemas comunes en el desarrollo de software, mejorando la calidad y facilitando el trabajo en equipo. Usar javascript en los ejemplos hace que los conceptos sean aplicables en proyectos actuales, sobre todo en el desarrollo web. La inclusión de moodle como plataforma permite un aprendizaje flexible y autónomo. Este enfoque práctico, basado en resolver problemas reales, prepara a los estudiantes para enfrentar desafíos en la industria y fomenta la innovación en tecnología.

3) Bibliografía:

Autor: Ramiro Andres Escobar y Tania Isidora Mora Pedreros
Año: 2027

4) Gráfico:



.Aplicación web para recomendación de libros para apoyar la elección de compra.

1) Resumen: Este artículo aborda temas clave en el desarrollo de software, como las API REST, que facilitan la comunicación entre sistemas mediante HTTP, y los principios SOLID, que ayudan a estructurar código más mantenible. Se discuten también patrones de diseño, útiles para resolver problemas comunes, y bases de datos no relacionadas eficaces para manejar grandes volúmenes de datos.

Herramientas como Git y Jira mejoran la gestión de versiones y tareas en equipo, mientras que Node.js facilita la creación de aplicaciones en tiempo real.

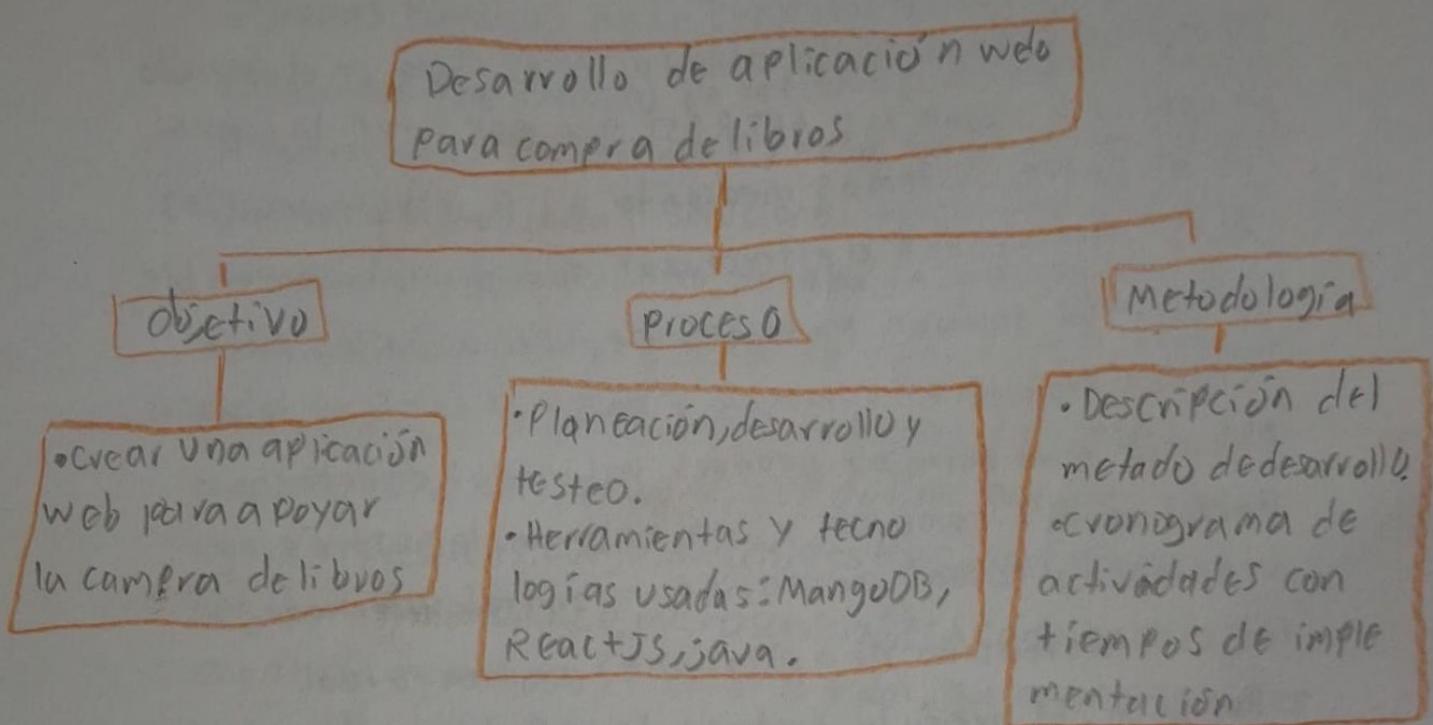
2) Reflexión: El artículo destaca la importancia de aplicar buenas prácticas y herramientas en el desarrollo de software. Las API REST y los principios SOLID permiten crear sistemas escalables y fáciles de mantener, mientras que los patrones de diseño ayudan a resolver problemas comunes de forma efectiva. Las bases de datos no relacionadas ofrecen flexibilidad para manejar grandes volúmenes de datos de manera ágil. Además, herramientas como Git y Jira facilitan la organización y colaboración en equipo.

3) Bibliografía:

Autor: Juan Pablo Gómez Quintero

Año: 2027

4) Grafica:



9

•Buenas Prácticas en la construcción del Software

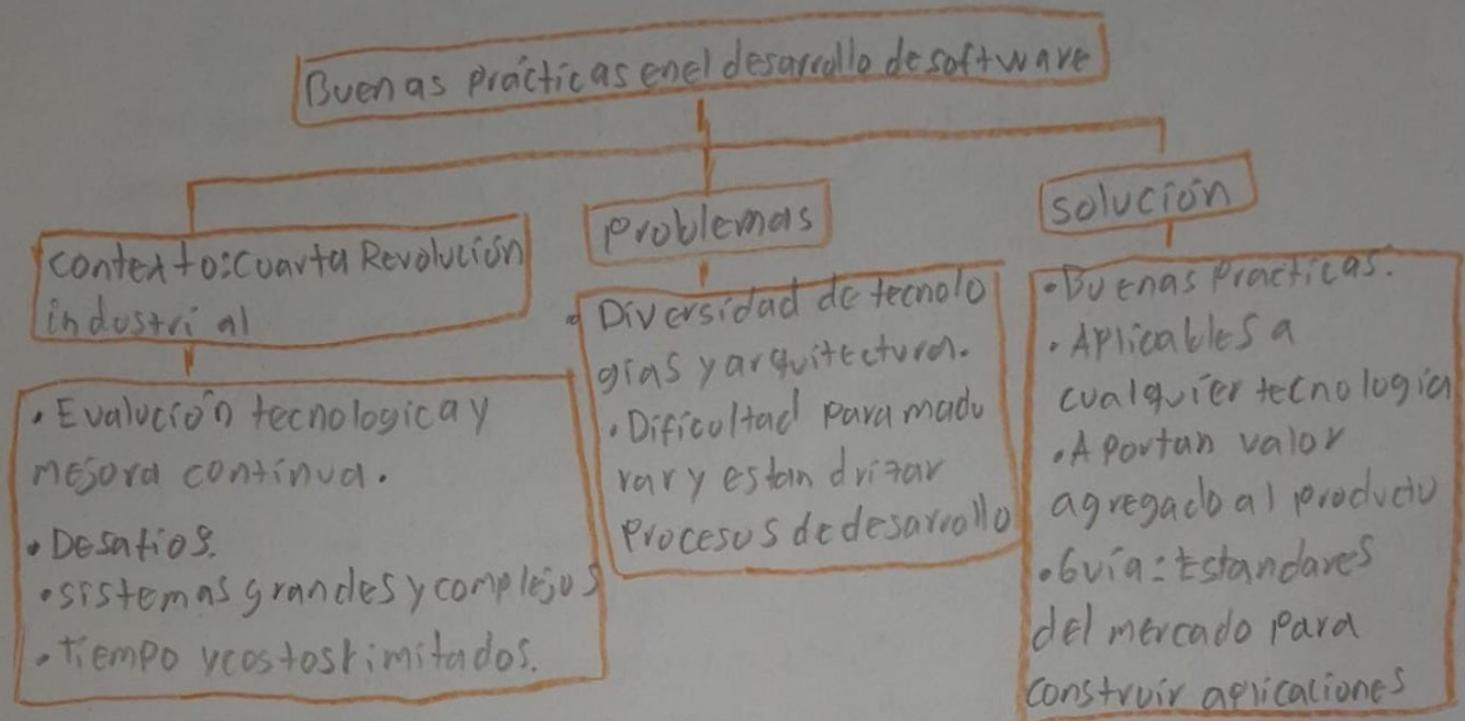
1) Resumen: Este artículo de reflexión explora diferentes arquitecturas de software y metodologías de desarrollo utilizadas para crear soluciones eficientes y flexibles en sistemas de TI. Primero, detalla la arquitectura de solución, que guía el diseño de estructuras integradas para responder a necesidades actuales y futuras, y la arquitectura de software, que define patrones y tecnologías específicas. Luego, describe estilos arquitectónicos como capas, monolítico, microservicios, EDA (arquitectura orientada a eventos) y cliente-servidor, que organizan la comunicación y estructura de aplicaciones.

2) Reflexión: este artículo destaca la importancia de elegir bien la arquitectura y la metodología de desarrollo para crear software que sea adaptable, modular y eficiente. Cada estilo arquitectónico - como capas, monolítico o microservicios - ofrece ventajas según las necesidades de la aplicación, ya sea flexibilidad, independencia o escalabilidad. La metodología XP, por su parte, fomenta la comunicación constante entre desarrolladores y clientes, permitiendo cambios rápidos y manteniendo el equipo enfocado en el cliente.

3) Bibliografía:

Autor: Camilo Andrés Prieto y Diego Alejandro Madrid
Año: 2022

4D Gráfica



• construcción de un repositorio de activos de software para el desarrollo agil de aplicaciones aplicando un método para el reuso

1) Resumen: Este artículo aborda la problemática de la reutilización de software en la industria del desarrollo, enfocándose en cómo los repositorios de activos pueden mejorar este proceso. A lo largo del tiempo, metodologías como la programación estructurada y la programación orientada a objetos se han desarrollado para mejorar la eficiencia. La investigación incluye un repositorio llamado "Activos" que almacena artefactos reutilizables, como código y esquemas de base de datos, y permite el acceso a estos recursos para proyectos futuros.

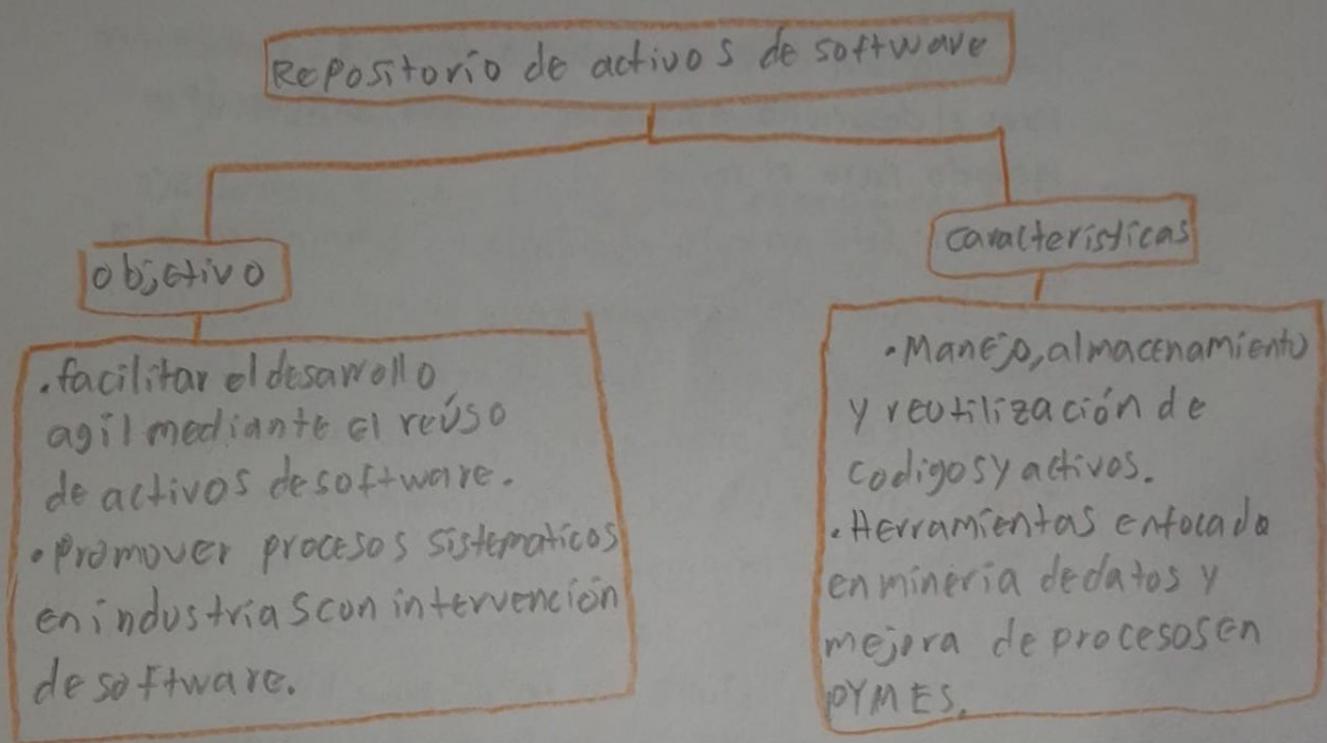
2) Reflexión: El reuso de activos de software es clave para ahorrar tiempo y recursos en el desarrollo. Al reutilizar patrones y componentes, los equipos logran aplicaciones más consistentes y eficaces, además de fomentar la colaboración. Los repositorios de software organizados facilitan este proceso, evitando empezar desde cero y permitiendo construir sobre el trabajo de otros. Esta práctica no solo mejora la productividad, sino que impulsa una cultura más innovadora y colaborativa en la industria del software.

3) Bibliografía

Autor: Yeimar Alonso Castro, Javier David Fernández, Julián Alberto Rivera, Eder Acevedo Marín.

Año: 2017

4) Gráfico



• Protección contra ataques de red para el software
segurmatita antivirus para Linux.

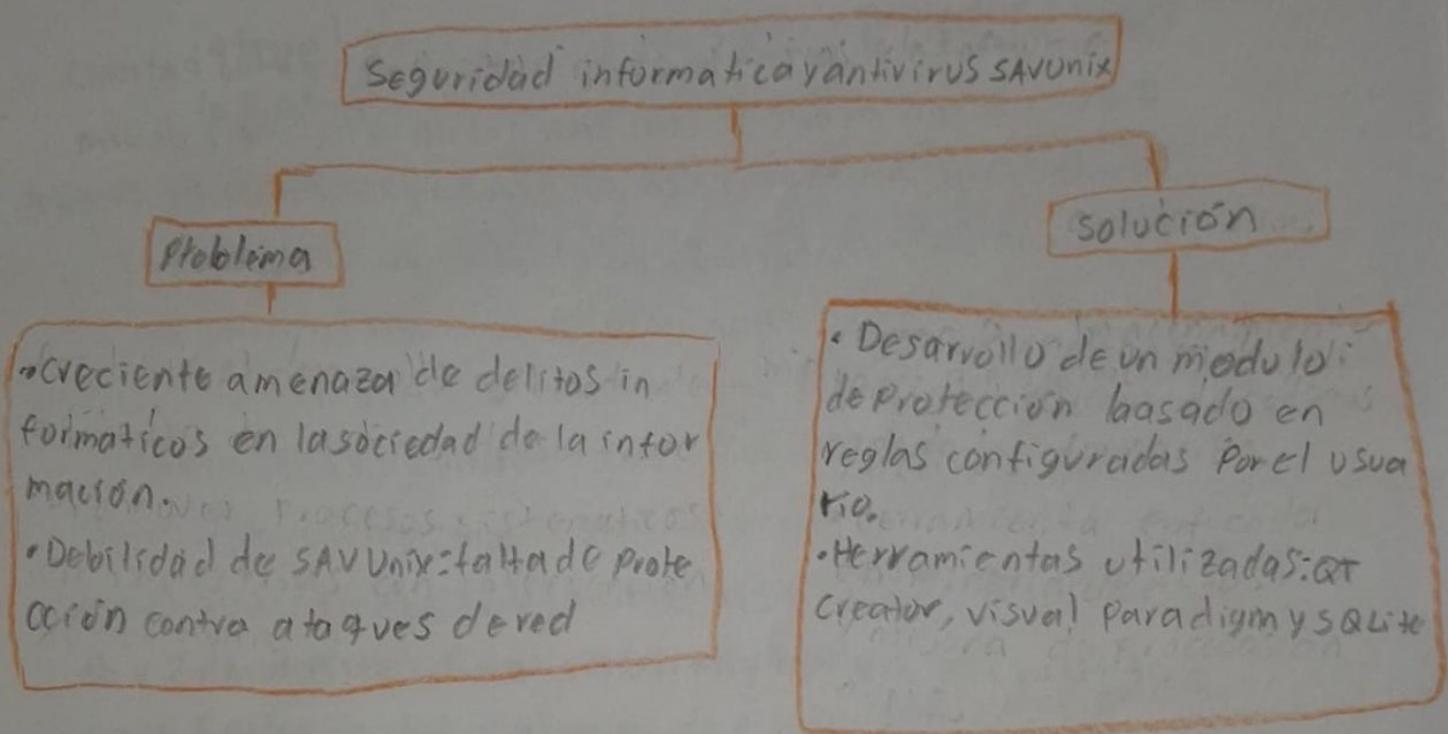
1) Resumen: En el artículo 3 nos especifican los patrones que utilizan donde se realizan las validaciones de una herramienta en específico desarrollada mediante pruebas funcionales y unitarias, con el objetivo de asegurar su correcto funcionamiento. Se utilizan patrones de diseño para estructurar el sistema, lo que permite un diseño más eficiente y modular. Los patrones GRASP ayudan a asignar responsabilidades a las clases, mejorando cohesión y bajo acoplamiento; patrones GOF (gang of four) se dividen en creacionales, estructurales y de comportamiento, y facilitan la organización del sistema.

2) Reflexión: La validación de herramientas de software mediante patrones de diseño asegura sistemas robustos y mantenibles. Las pruebas funcionales y unitarias verifican el correcto funcionamiento de cada componente, mientras que los Patrones GRASP organizan la asignación de responsabilidades, promoviendo cohesión y bajo acoplamiento entre clases. Los Patrones GOF complementan el diseño al estructurar la creación y el comportamiento de objetos, permitiendo que el sistema sea escalable y reutilizable.

Bibliografía:

Autor: Dennis Lázaro Álvarez Medina
Año: 2018

4) Gráfica:



Diseño de un prototipo de arquitectura de software integral que permita gestionar y controlar eficazmente las etapas del ciclo de desarrollo.

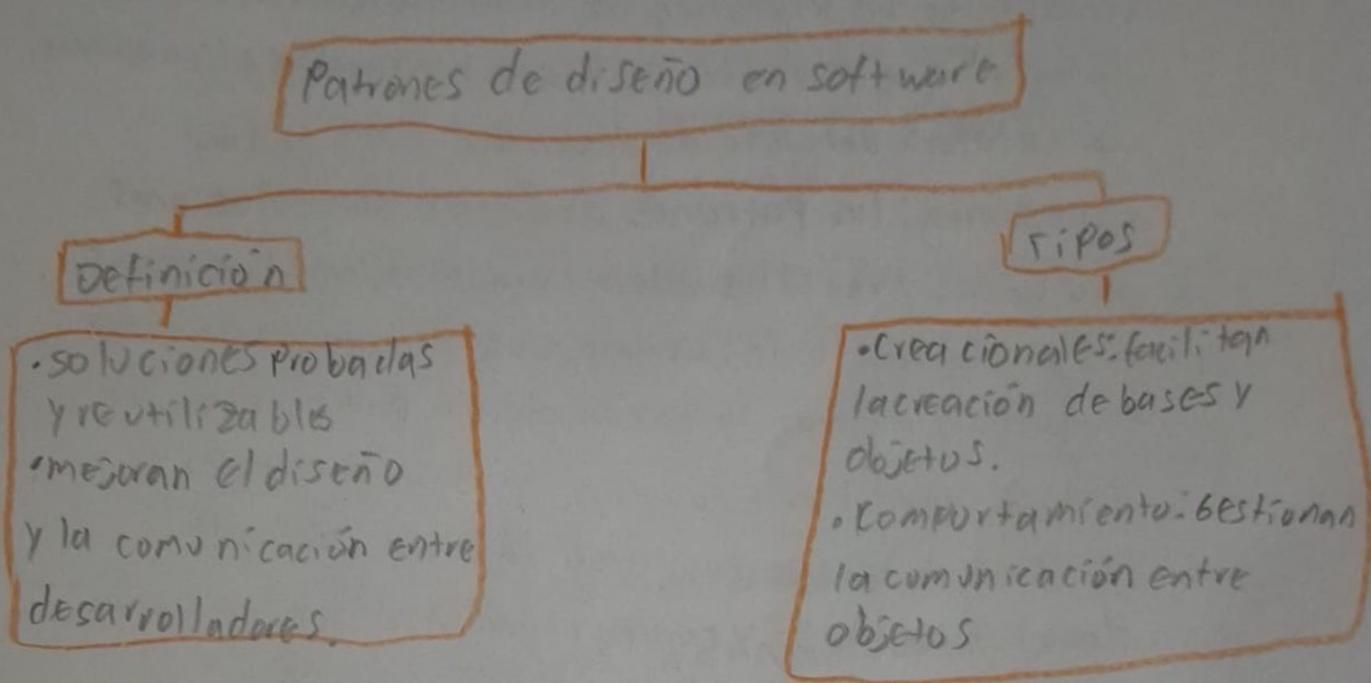
1) Resumen: los patrones de diseño son soluciones probadas y reutilizables para problemas comunes en el desarrollo de software, que permiten optimizar el diseño y mejorar la comunicación entre desarrolladores. Existen tres tipos principales de patrones: creacionales, que facilitan la creación de clases y objetos; y comportamiento, que gestiona la comunicación entre objetos.

Reflexión: Los patrones de diseño son soluciones probadas que ayudan a los desarrolladores a enfrentar problemas comunes de manera eficiente. Al usarlos, no solo ahorramos tiempo, sino que también mejoramos la flexibilidad, escalabilidad y mantenibilidad del software. Sin embargo, es esencial adaptarlos a las necesidades específicas de cada proyecto, ya que son directrices más que soluciones rígidas. Dominar su uso facilita la creación de sistemas más robustos y eficaces.

3) Bibliografía:

Autor: Alvaro Javier Torres y Oscar Javier Vargas
Año: 2024

4) Gráfica



Diseño de un sistema de monitoreo semafórico low cost para ciudades emergentes de Colombia como producto de yune xtraffic

1) Resumen: Este artículo analiza la arquitectura de software y los patrones de diseño, esenciales para construir sistemas robustos y eficientes. Describe tres tipos de arquitecturas: de capas (organización en niveles), cliente-servidor (interacción y procesamiento de datos), y micro servicios (servicios independientes). Además, los patrones de diseño, agrupados en creacionales, estructurales y de comportamiento, ofrecen soluciones probadas para problemas comunes de desarrollo.

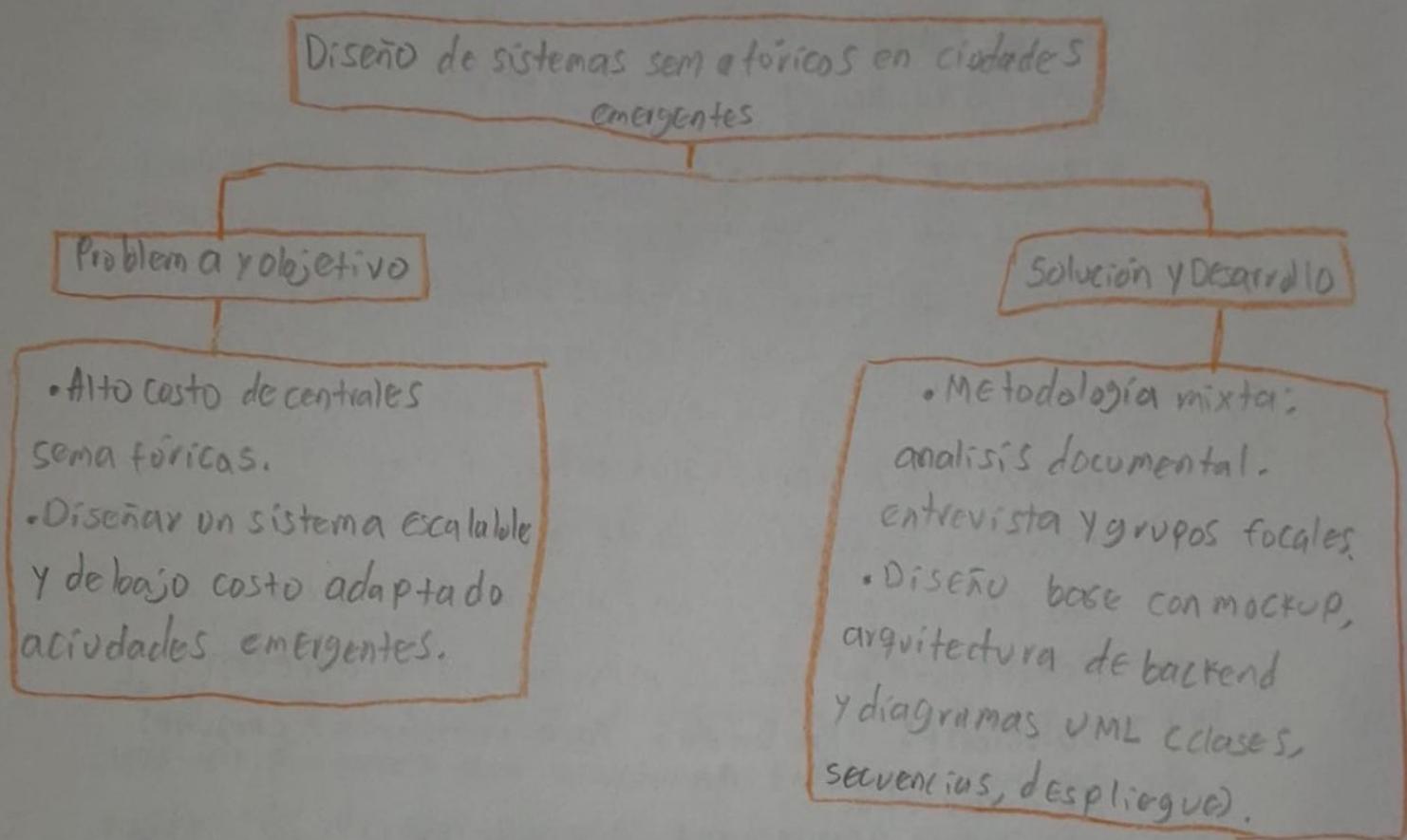
2) Reflexión: Este artículo nos recuerda la importancia de planificar cuidadosamente la arquitectura de software y usar patrones de diseño como herramientas estratégicas para mejorar la eficiencia, escalabilidad y organización de un sistema. Al implementar arquitecturas como la de capas o microservicios, se facilita la adaptación y el crecimiento del software. Los patrones de diseño, además, nos brindan soluciones probadas y permiten una comunicación clara entre los desarrolladores.

3) Bibliografía:

Autor: Diego Amaya Deltran, Jeisson Sierra Camargo,
luz: Dominguez Guzman.

Año: 2023

4) Gráfica



• Método de predicción automático para evaluar

1) Resumen: este artículo se centra en conceptos clave de algoritmos de clasificación y toma de decisiones, además de prácticas esenciales para desarrollar modelos predictivos eficientes. Se introducen patrones y anti-patrones de diseño de software, enfatizando los creacionales, estructurales y de comportamiento, que solucionan problemas comunes en desarrollo. También se analizan los anti-patrones, errores comunes que afectan negativamente los proyectos y cómo evitarlos.

2) Reflexión: este artículo nos ayuda a resaltar la importancia de las buenas prácticas en el desarrollo de software, como el uso de patrones que solucionan problemas comunes y la evitación de anti-patrones que generan errores recurrentes. Destaca el análisis de código estático como herramienta para identificar vulnerabilidades y mejorar la calidad del software desde el inicio. También, explica cómo los algoritmos de clasificación, como Naive Bayes, permiten construir modelos predictivos eficaces con datos limitados.

Bibliografía:

Autor: Juan Pablo Duque Alfonso

Año: 2078

- Evaluar calidad en etapas tempranas mediante modelos estadísticos.
- Detectar problemas y proporcionar retroalimentación eficiente.

- Usa API de Git para analizar código fuente.
- Aplica detección de patrones de diseño y análisis estático de código.
- Predice errores usando Naïve Bayes con más del 90% de precisión

• Extensión de la arquitectura Docker para el despliegue automático de contenedores

1) Resumen: Este artículo propone una extensión de Docker que automatiza el despliegue de contenedores, simplificando su uso para usuarios sin conocimiento técnico. La solución incluye una plataforma web que gestiona contenedores y proporciona una interfaz amigable. Utiliza un servidor Ubuntu, Docker CE y un Docker registry, con contenedores personalizados para MySQL y Apache. Ofrece funcionalidades como autenticación de usuarios, gestión de imágenes y creación de contenedores, permitiendo el despliegue de sitios web.

2) Reflexión: El artículo nos habla sobre la extensión de Docker resaltando la importancia de simplificar tecnologías complejas para hacerlas accesibles a usuarios sin conocimientos técnicos. La plataforma propuesta demuestra cómo la abstracción de conceptos técnicos facilita la adopción de herramientas avanzadas como Docker. Además, destaca la relevancia de la automatización en la gestión de contenedores, optimizando tareas como la creación y administración de bases de datos o sitios web.

Bibliografía:

Autor: Luz Elena Gutiérrez López, Carlos Andrés Guerrero Alarcón.

Año: 2020

Arquitectura para despliegue de contenedores

Importancia de los contenedores

- Separan aplicaciones e interactúan con sus partes sin afectar el todo.
- Comparten procesos similares a la arquitectura orientada a servicios.

Resultados de la navegación

- Directo: propuesta arquitectura para despliegue automático basada en Docker.
- Indirecto: plataforma web para verificar y validar la arquitectura.

• Desarrollo de las funcionalidades para el diseño visual de consultas y archivos de datos en la herramienta ORMapping

1) Resumen: En el artículo utilizan los Patrones de diseño GRASP aplicados en el desarrollo de una solución. Se destacan tres Patrones: Experto, que asigna responsabilidades a la clase con más conocimiento para lograr cohesión y encapsulamiento; controlador, que separa la lógica de negocios de la presentación, mejorando el control y la reutilización del código; y creador, encargado de instanciar objetos, reduciendo el acoplamiento.

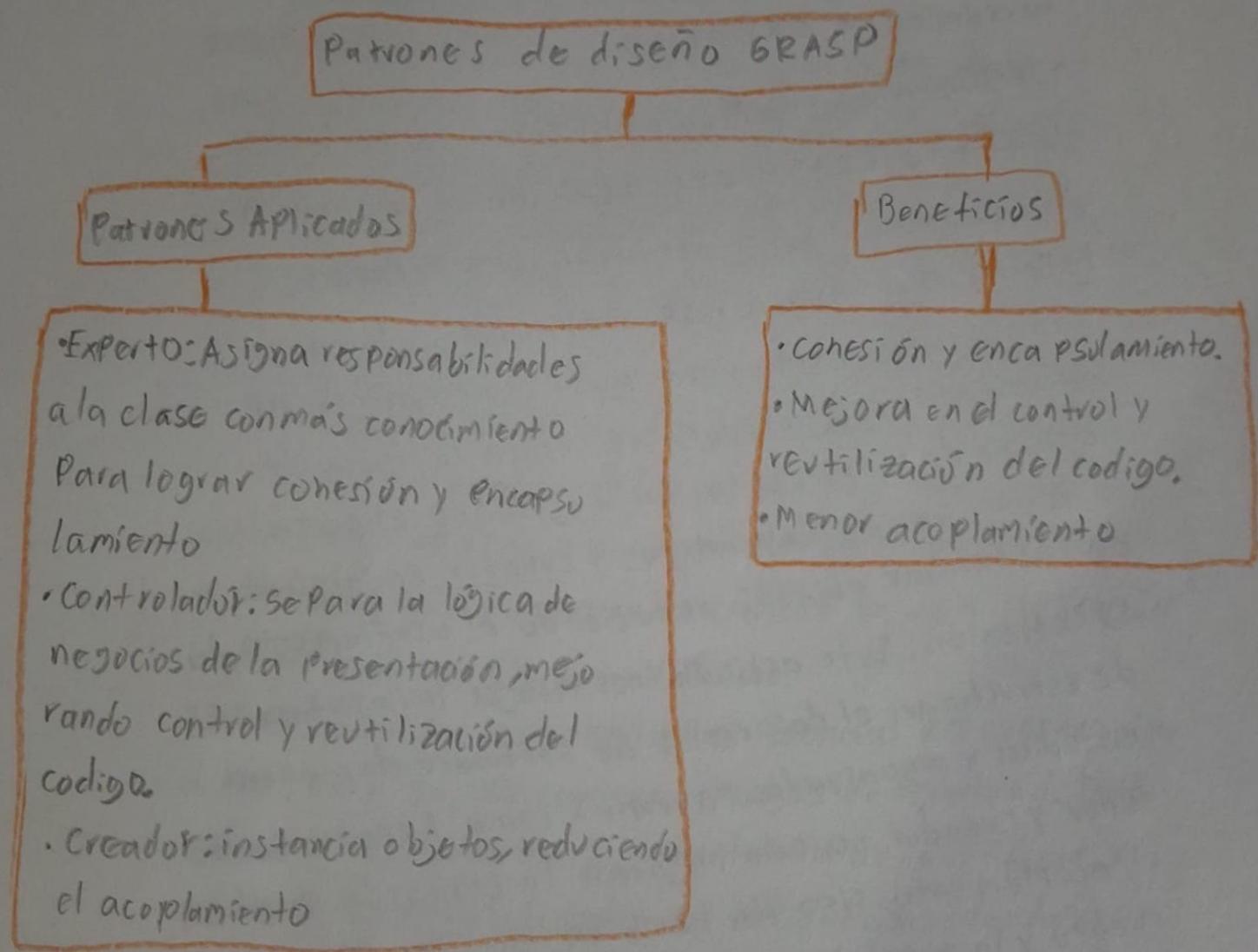
2) Reflexión: Este artículo nos refleja la importancia de estructurar el desarrollo de software de manera metódica y organizada. Patrones como Experto, Controlador y Creador no solo mejoran la cohesión y reducen el acoplamiento, sino que también facilitan el mantenimiento y la escalabilidad de las aplicaciones. Estos principios permiten asignar responsabilidades de forma lógica, promoviendo diseños claros y reutilizables. Además, el uso de metodologías ágiles y herramientas como las tarjetas CRC complementa este enfoque.

Bibliografía:

Autor: Yoandrys León Batista

Año: 2016

4) Gráfica



• Incidencia de los patrones de diseño de software en la seguridad de las aplicaciones web

1) Resumen: El artículo explora el impacto de la tecnología, en especial de internet, como herramienta clave para la vida cotidiana y su relación con la seguridad web. Destaca la importancia de OWASP en la identificación de riesgos como fallos de control de acceso, inyecciones, configuraciones inseguras y componentes desactualizados. Se utiliza la metodología en cascada para analizar vulnerabilidades, seleccionar patrones de diseño adecuados y aplicar correcciones en una aplicación desarrollada en C# con .NET Core.

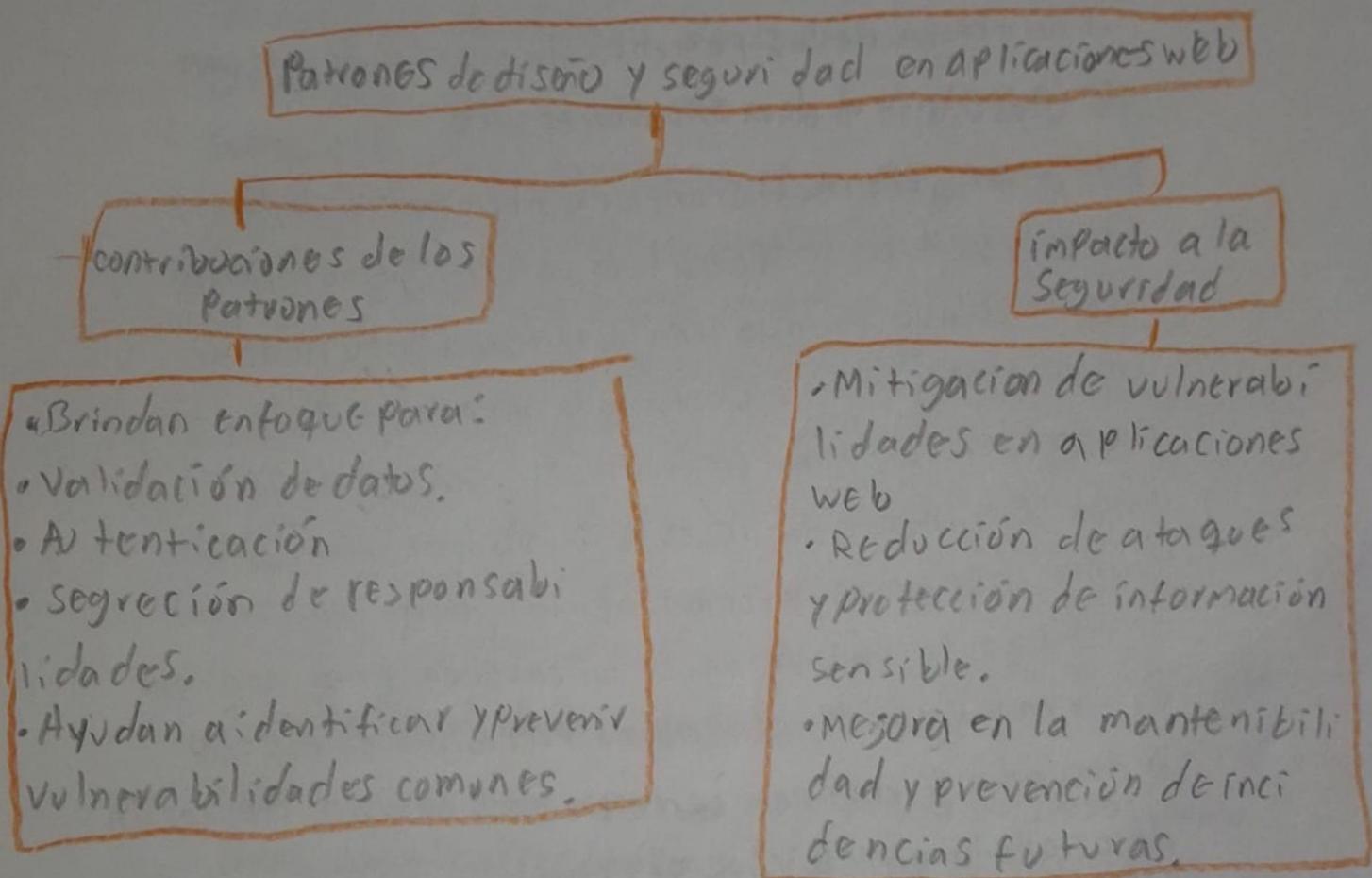
2) Reflexión: La tecnología es esencial en nuestra vida diaria, pero su uso conlleva riesgos de seguridad. Este artículo nos recuerda la importancia de identificar y mitigar vulnerabilidades en el software, destacando herramientas y patrones de diseño como aliados clave para crear aplicaciones más seguras y confiables. Implementar estas buenas prácticas no solo fortalece la seguridad, sino también la confianza de los usuarios en nuestras soluciones.

3) Bibliografía:

Autor: Mesias Valencia, Juan José

Año: 2024

4) Gráfica:



• COFFEE CHALLENGE: un juego para el aprendizaje de patrones de diseño de software

1) Resumen: El artículo describe el diseño de un juego educativo basado en el método de Gomez para enseñar patrones de diseño de software. El objetivo es aplicar estos patrones en el desarrollo de una página web de venta de café. Se seleccionó la técnica "El dorado" para la estructura del juego, que incluye la resolución de problemas con patrones como GOF y de interfaz de usuario. A través de pruebas piloto, se ajustaron detalles como la traducción y la calidad de las imágenes.

2) Reflexión: Este artículo como el uso de juegos educativos, como "coffee challenge", facilita la enseñanza de patrones de diseño de software. Mediante una narrativa y metaforas, los estudiantes pueden aplicar conceptos teóricos en situaciones prácticas. El proceso de prueba y ajuste del juego demuestra la importancia de la retroalimentación para mejorar el aprendizaje. Este enfoque dinámico promueve una comprensión más profunda y práctica de los contenidos, adaptándose a las nuevas formas de enseñanza.

3) Bibliografía

Autor: Luz Eliana Gonzalez castaño, sara Valentina Marroquin Soto, Grissa Vianney Maturana Gonzalez y Roberto Antonio Manjarres Betancur

Año: 2027

Patrones de diseño y seguridad en aplicaciones web

Características de los patrones

- Ofrecen enfoque para:
- Validación de datos.
- Autenticación.
- Segregación de responsabilidades comunes.

Impacto en la seguridad

- Mitigación de vulnerabilidades en aplicaciones web.
- Reducción de ataques y protección de información sensible.
- Mejora en la mantención y prevención de incidentes futuros.

• Aplicación de Perfiles UML en la especificación de patrones de comportamiento

1) Resumen: El artículo explora el uso de Perfiles UML para especificar patrones de diseño de comportamiento (PDC) en diagramas de clases y secuencia. Presenta la arquitectura APPD, que organiza los perfiles en tres niveles, utilizando estereotípos y restricciones OCL para modelar patrones específicos, como command. Esto permite validar consistencias entre diagramas y aplicar patrones en herramientas UML existentes, alineándose con el enfoque de arquitectura dirigida por modelos (MDA).

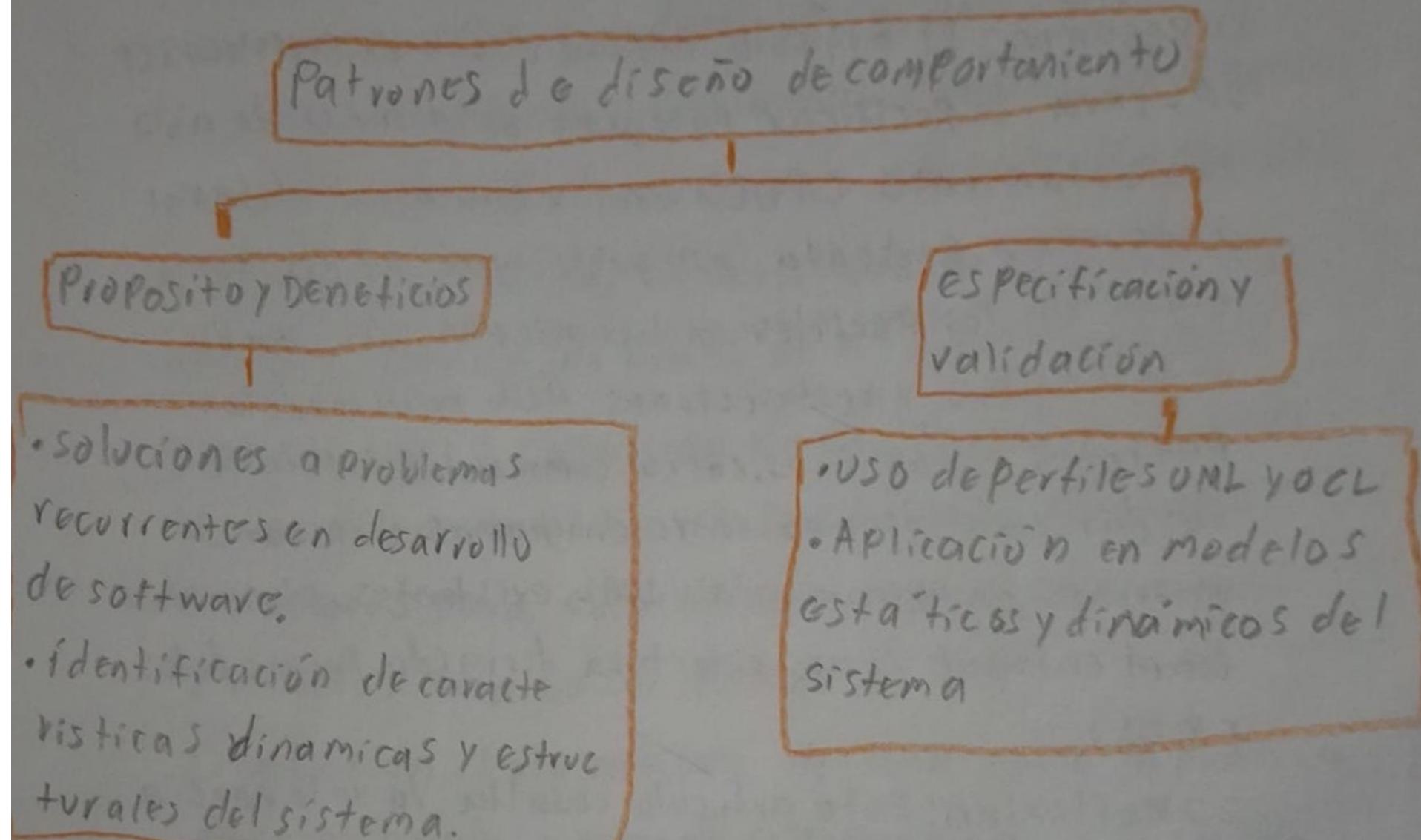
2) Reflexión: Este artículo resalta la relevancia de los patrones de diseño de comportamiento y su especificación con perfiles UML, mejorando la claridad y validación en diagramas de clases y secuencia. Este enfoque, alineado con la arquitectura dirigida por modelos (MDA), permite usar herramientas estándar y fomenta la reutilización eficiente. La incorporación de estereotípos y restricciones OCL aporta precisión y uniformidad, facilitando diseños más claros y consistentes.

3) Bibliografía:

Autor: Alberto Cortez y Ana Gariz

Año: 2012

4) Grafico:



• Minería de Datos con Busqueda de Patrones de Patrones de comportamiento

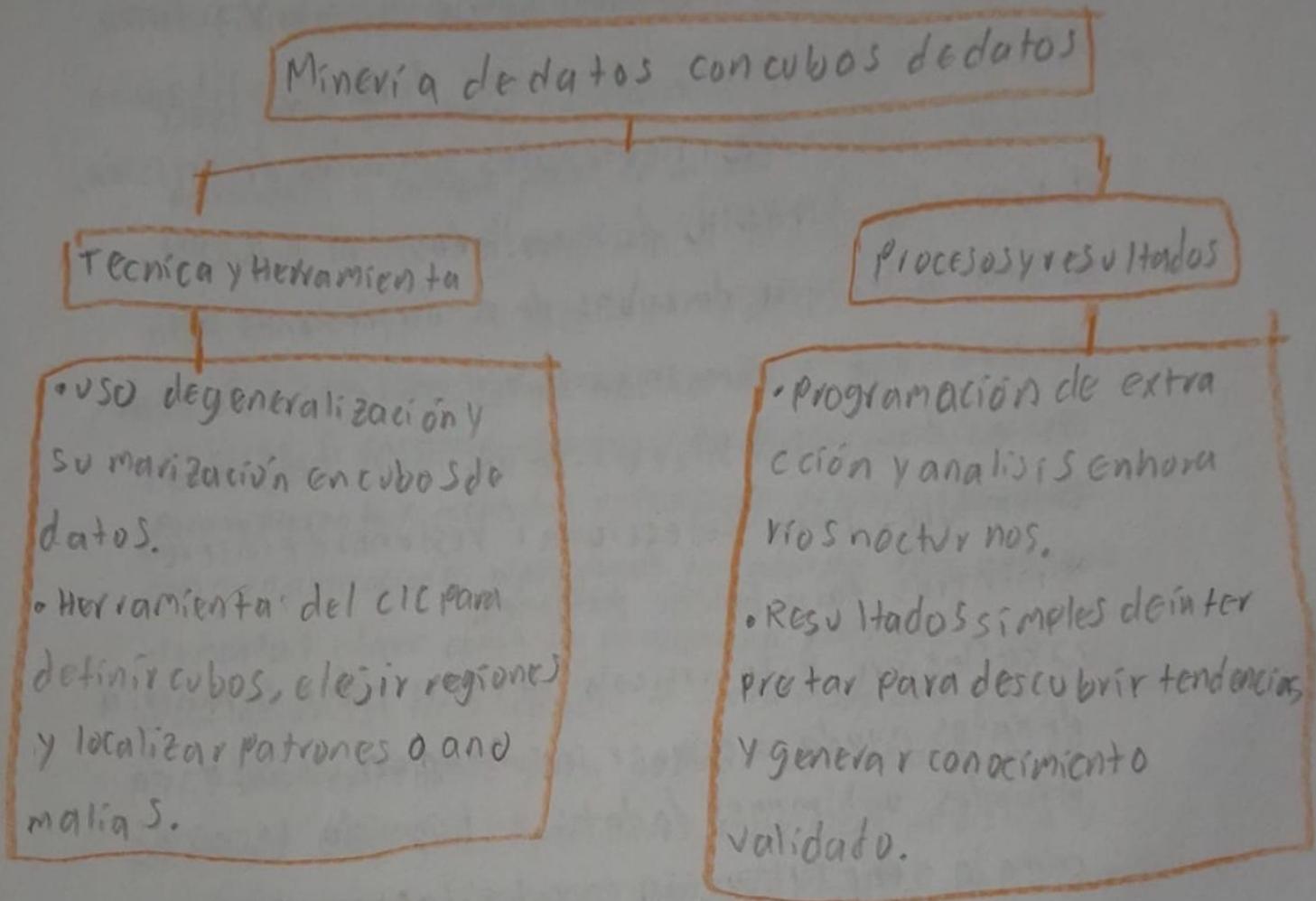
1) Resumen: La minería de datos busca descubrir información valiosa de grandes volúmenes de datos, utilizando técnicas como redes neuronales y árboles de decisión. El proyecto ANASIN, desarrollado en el CICIPN, aplica la técnica de cubos de n-dimensiones para generalizar y sumarizar datos, permitiendo análisis en diferentes niveles. Se utilizan programas extractores para seleccionar regiones de interés y mineros para buscar patrones.

2) Reflexión: Este artículo destaca cómo la minería de datos ayuda a extraer información valiosa de grandes volúmenes de datos, utilizando técnicas como la generalización en cubos. Las herramientas automatizan procesos complejos, optimizando el tiempo y recursos. Esto facilita la toma de decisiones y la identificación de patrones estratégicos. Aunque la tecnología es útil, aún existen desafíos por resolver, lo que ofrece oportunidades para mejorar las herramientas.

3) Bibliografía:

Autor: Gilberto Lorenzo Martínez Luna, Adolfo Guzmán Arenas
 Año: 1994

UD 6 Análisis:



Perfil UML para el desarrollo wap

D) Resumen: Los sistemas informáticos han evolucionado en complejidad, desplazando el enfoque desde la calidad del código hacia el modelado del sistema. UML, como el lenguaje de modelado más utilizado, permite especificar y documentar sistemas en diversos dominios, gracias a su flexibilidad. Sin embargo, su carácter genérico puede ser insuficiente para representar aspectos específicos de ciertos dominios, como en aplicaciones WAP, donde no aborda adecuadamente elementos clave como la navegación y diseño.

2) Reflexión: El texto aborda el meta modelado y el uso de un perfil UML extendido para modelar aplicaciones WAP, adaptando UML a las necesidades específicas de diseño y navegación. Describe las capas WAP, adaptando UML a las necesidades específicas de diseño y navegación. Describe las capas de modelado de OMG CMU a M3 y detalla la creación del perfil en cinco pasos: definir el metamodelo de OMG CMU a M3 y detalla la creación del perfil en cinco pasos: definir el metamodelo, estereotipos y valores etiquetados y restricciones en OCL, finalmente, aplica el perfil a una librería virtual, mostrando su utilidad en diagramas de componentes y clases para modelar páginas WAP dinámicas.

3) Bibliografía:

Autor: Ricardo Soto, Mauricio Camara

Año: 2004

Herramientas

Perfiles UML para el desarrollo de aplicaciones WAP

Evolución de sistemas informáticos
Enfoque en modelado sobre
calidad de código.

UML y aplicaciones WAP:
flexibilidad genérica,
pero insuficiente para
navegación y diseño
específicos.

Aplicación de MDA al desarrollo de aplicaciones web en OOWS

1) Resumen: Este trabajo propone usar la arquitectura dirigida por modelos (MDA) junto con el método OOWS para desarrollar aplicaciones web, explica cómo MDA permite especificar sistemas de manera independiente de las tecnologías, facilitando su integración y mantenimiento. Introduce el metamodelado PIM-OOWS, enfocado en un nivel conceptual, y el PSM-OOWS basado en J2EE, estableciendo correspondencias entre ambos para generar código automáticamente. Se destacan las ventajas de combinar OOWS, que modela aplicaciones web de forma expresiva, con MDA, que asegura flexibilidad y escalabilidad multiplataforma.

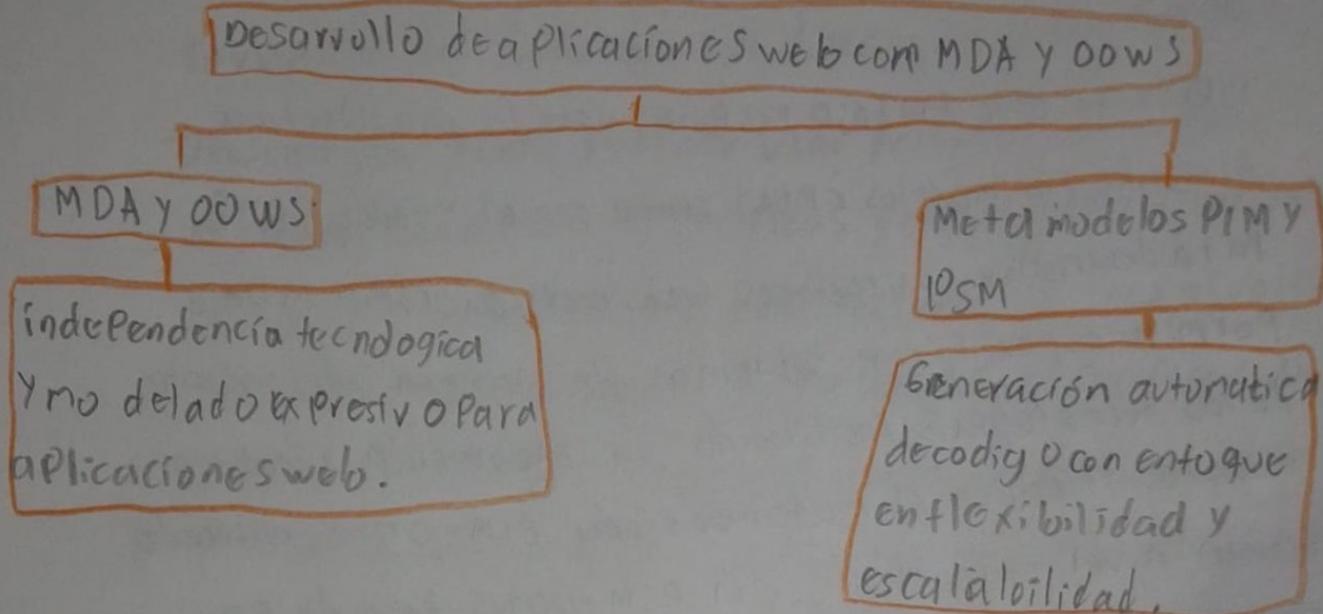
2) Referencias: La propuesta de combinar MDA con OOWS en el desarrollo de aplicaciones web refleja un enfoque innovador y práctico. Este modelo fomenta un diseño más abstracto y sostenible, separado de las limitaciones tecnológicas actuales. La integración de OOWS con su capacidad de modelar sistemas web de forma detallada, asegura resultados precisos y adaptables. Además, la generación automática de código desde modelos sistemas web de forma detallada, asegura resultados precisos y adaptables.

3) Bibliografía:

AUTOR: Ricardo Quintero, Vicente Pelechano, Joan Fons, Oscar Pastor

AÑO: 2003

4D Gráfica



Formalización de patrones de diseño de comportamiento

1) Resumen: El texto propone usar perfiles UML para expresar conceptos específicos y crear especificaciones precisas. Se presentan una arquitectura de tres niveles: General, clasificación y patrones específicos. Un ejemplo es el patrón command donde se definen este tipos y restricciones. Se plantan objetivos futuros, como un modelo para validar patrones y detectar coincidencias en diagramas UML.

2) Reflexión: Los perfiles UML para patrones de comportamiento son herramientas que fortalecen la precisión y utilidad en el diseño de software. Esta propuesta destaca cómo estandarizar y simplificar especificaciones mediante estereotipos y restricciones, facilitando la validación y detección de patrones en diagramas. Esto no solo optimiza procesos actuales, sino que sienta bases sólidas para innovaciones futuras.

3) Bibliografía:

Autor: A. Cortez, A. Garis, D. Riesco

Año: 2011

Extensión de UML

Perfiles para especificar
patrones y crear especificaciones precisas.

Arquitectura y objetivos

Tres niveles (general,
básicos, específicos) y
validación de patrones
como futuro

Diseño de métricas basadas en estresores laborales Para desarrolladores de software

1) Resumen: El estrés laboral en desarrolladores de software se ha incrementado debido a las altas demandas de innovación tecnológica, lo que provoca ansiedad, monotonía y colapso. Este fenómeno afecta tanto la productividad como la salud física y mental de los trabajadores. En México, el estrés. Aunque existen estudios sobre el tema, aun no se han desarrollado métricas precisas para medir este estrés. El objetivo de este trabajo es crear un modelo de métricas que ayude a predecir y monitorear al estrés desarrolladores, mejorando el rendimiento y el de las organizaciones.

2) Reflexión: El estrés laboral en los desarrolladores de software es un desafío creciente que afecta no solo la productividad, sino también la salud emocional y física. A medida que la industria exige más innovación y rapidez, es crucial abordar este problema mediante métricas precisas que ayuden a monitorear y reducir el estrés. Al hacerlo, no solo se mejora el bienestar de los trabajadores, sino también el rendimiento general de las organizaciones, permitiendo que el entorno laboral sea más saludable y eficiente.

3) Bibliografía:

Autor: María Alejandra Zamudio, Ramón R. Palacio, Luis A. Castro
Año: 2017

Demandas tecnologicas
elevadas generan estres,
afectando salud y pro-
ductividad.

Crear metricas
para predecir y
monitorear el estres,
mejorando rendimiento
organizacional.

La clase invertida y el uso de videos de software educativo en la formación inicial del profesorado.

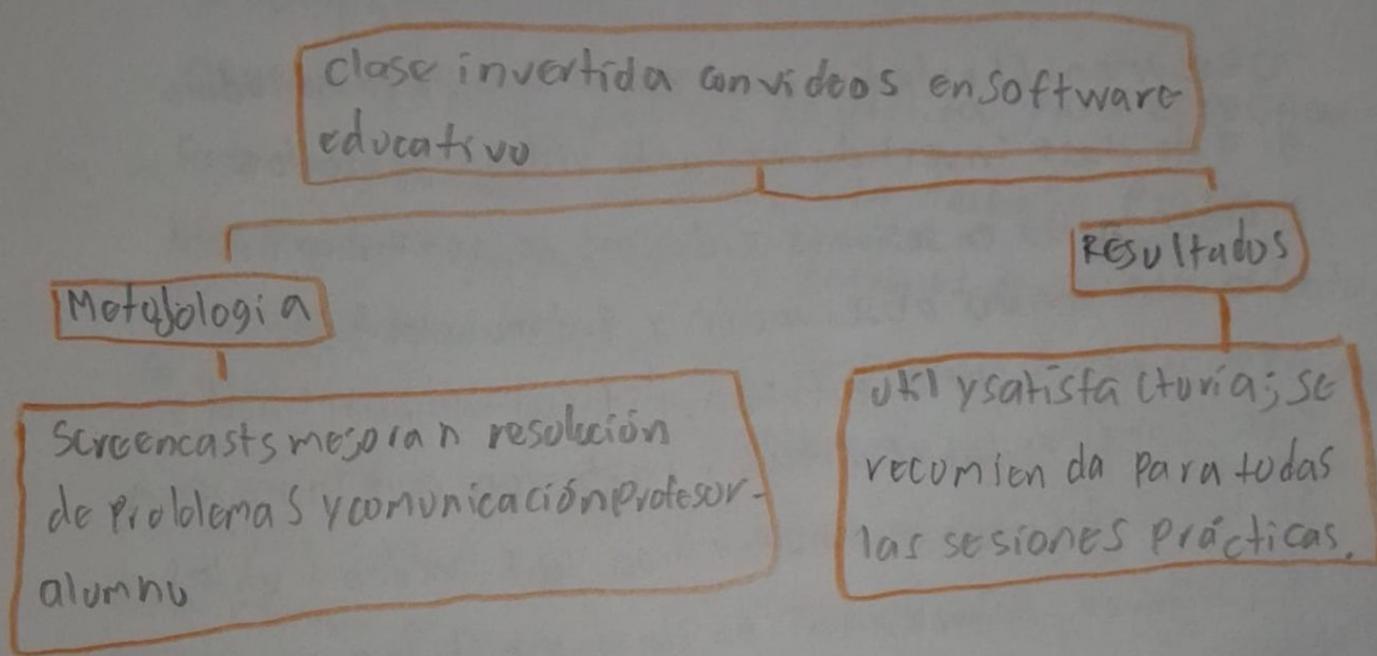
Estudio cualitativo

1) Resumen: El estudio analiza el uso de la metodología de clase invertida mediante videos grabados (screen casts) en sesiones prácticas de software educativo. Utilizando observación y entrevistas, se evalúan áreas como esfuerzo, dificultad, utilidad y satisfacción. Los resultados mostraron que la mayoría de los estudiantes consideraron los videos útiles para resolver problemas en clase y notaron una mejora en la comunicación con los profesores. Se sugirió que esta metodología debería aplicarse en todas las sesiones prácticas.

2) Reflexión: La metodología de clase invertida, especialmente al incorporar videos como recursos, demuestra cómo la tecnología para mejorar la interacción entre estudiantes y profesores. Al permitir que los estudiantes y profesores. Al permitir que los estudiantes resolvieran problemas prácticos durante las sesiones, se potenció su aprendizaje activo y mejoró la comunicación. Sin embargo, es clave que los docentes adapten esta técnica de acuerdo con las necesidades y ritmos de sus estudiantes, asegurándose de que la tecnología no solo sea un recurso, sino una herramienta eficaz para lograr mejores resultados educativos.

3) Bibliografía: Autor: Blasco, A.C., Lorenzo, J., & Sarza, J. Año: 2016

4) Grafía



Perfiles UML para la definición de patrones de diseño de comportamiento.

DREUmen: El lenguaje UML se utiliza para especificar sistemas y patrones de diseño; Este trabajo propone usar perfiles UML para definir patrones de comportamiento, aplicados en modelos mediante diagramas de clases y secuencias. Se implementó una arquitectura de 3 niveles en RSA para validar las estructuras y comportamientos de los modelos. La investigación formalizó patrones de GOF y creó un modelo para detectar elementos en diagramas relacionados con estos patrones, buscando mejorar la especificación y validar las interacciones en los modelos.

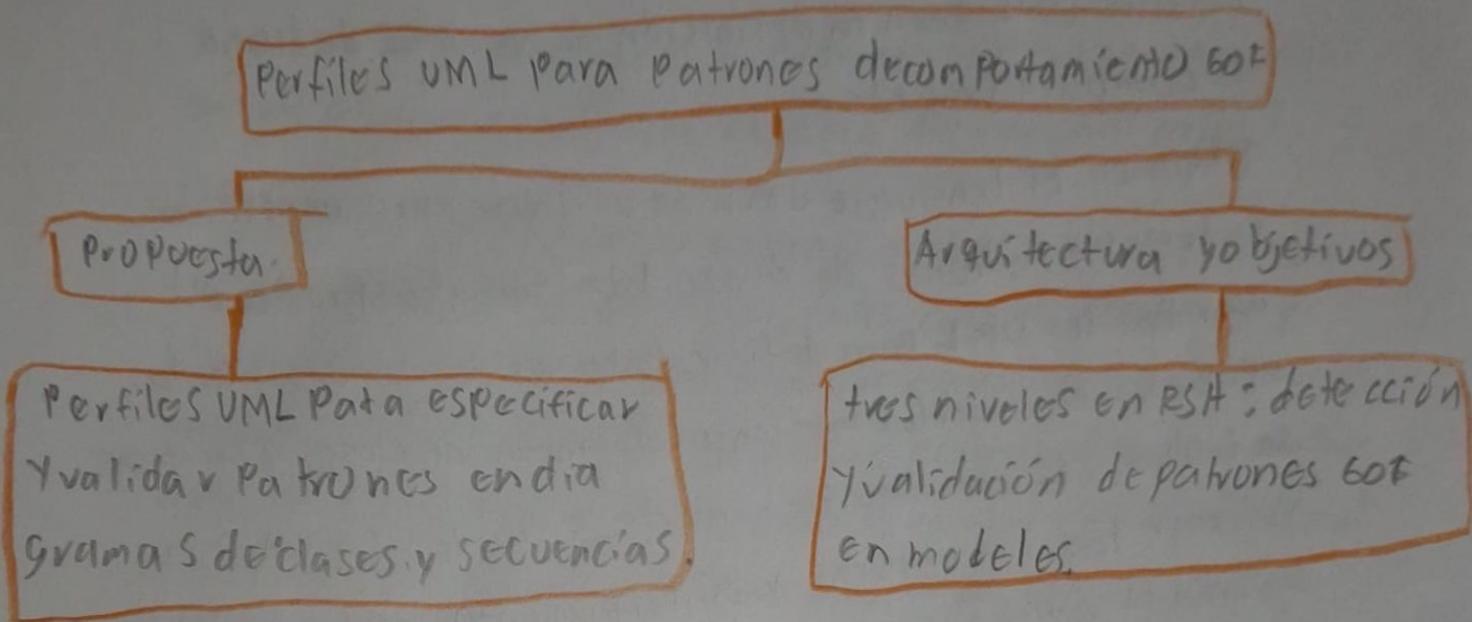
2) Reflexión: El uso de perfiles UML para especificar patrones de diseño es una forma eficiente de formalizar y validar sistemas complejos. Al aplicar estos perfiles a modelos, no solo se documentan las estructuras, sino también los comportamientos, lo que permite una verificación más completa y precisa de los diseños. Este enfoque mejora la reutilización de los patrones y optimiza la validación de interacciones en los modelos, destacando la importancia de metodologías como el UML en el desarrollo de software robusto y bien estructurado.

)) Bibliografía:

Autor: Cortez, A., Riesco, D.E., & Garis, A.G.

Año: 2012

4) GRAFICA



Analisis de interacciones y patrones de comportamiento en un curso virtual con objetos de aprendizaje.

1) Resumen: El EVEA es un entorno educativo que prioriza la interacción virtual en modalidades como el e-learning y blended learning, permitiendo la colaboración activa entre estudiantes generando conocimiento compartido. Las plataformas tecnológicas registran interacciones que ayudan a evaluar la efectividad y motivación del aprendizaje. Analizar estas interacciones mejora el rendimiento académico al identificar patrones de comportamiento. Desde los años noventa, se investiga cómo las TIC favorecen el aprendizaje colaborativo. Un análisis de un curso virtual mostró patrones de comportamiento en diferentes módulos para optimizar los resultados académicos.

2) Reflexión: Los entornos virtuales de enseñanza y aprendizaje (EVEA) promueven su aprendizaje colaborativo, donde los estudiantes construyen conocimientos de manera activa. A través de plataformas virtuales, los docentes pueden analizar las interacciones y ajustar sus enfoques pedagógicos para mejorar los resultados académicos. El análisis de estos patrones permite optimizar la experiencia educativa, haciendo que los entornos virtuales sean cada vez más efectivos para el aprendizaje en línea.

3) Bibliografía:

Autor: Claudia López de munain, Silvina Bramati, Zulema Beatriz Rosanigo, Pedro Bramati, Marcela Torrent

Año: 2014

EVEA: Entorno Virtual Educativo

Características

facilitan interacción,
aprendizaje colaborativo
y análisis de efectividad
en e-learning

Resultados

identificación de
patrones en modulos
para optimizar rendimiento
académico

Entorno virtual para la formación de tecnologías
Ingenieros desistemas en patrones de diseño de
software

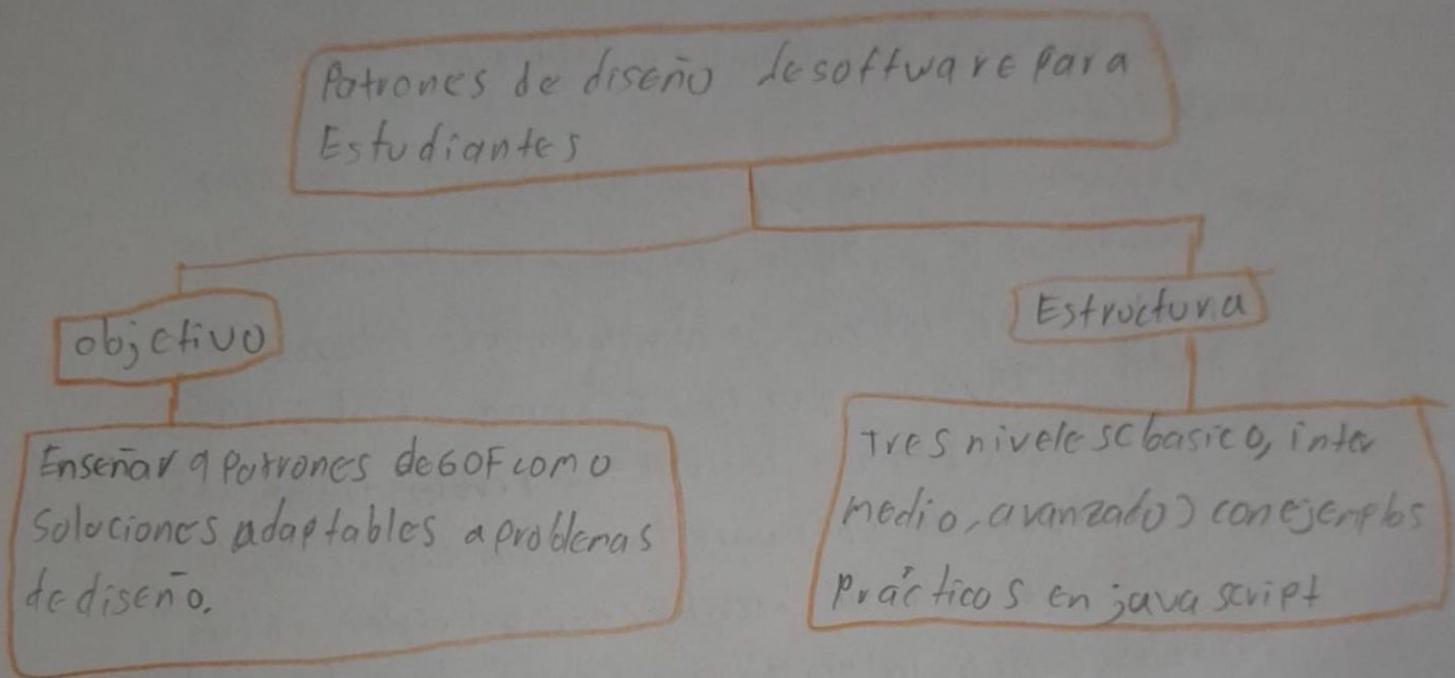
1) Resumen: Este trabajo tiene como objetivo explicar los Patrones de diseño de software a estudiantes de ingeniería en sistemas. Se abordan nueve patrones seleccionados del libro Gang of four (Gamma et al., 1995), que proporcionan soluciones a problemas comunes en el diseño de software. Los patrones no son solo código que se está escribiendo, se dividen en tres niveles: básico, intermedio y avanzado. Cada nivel cubre tres patrones: creacionales, estructurales y de comportamiento, con ejemplos prácticos en JavaScript.

2) Reflexión: La enseñanza de patrones de diseño de software refleja un enfoque práctico y adaptable para resolver problemas comunes en la programación. A través de ejemplos y niveles de dificultad, los estudiantes no solo aprenden a usar patrones, sino a comprender su aplicabilidad según el contexto. La educación virtual, al integrarse con metodologías como el e-learning y el b-learning, facilita la adquisición de estos conocimientos en cualquier momento y lugar, brindando flexibilidad en el aprendizaje.

3) Bibliografía:

Autor: Ramiro Andrés Dedoya Escobar y tonia isadora Mora pedveros
Año: 2027

4) Gráfica:



Describiendo patrones de comportamiento del aire:
Un enfoque de minería de datos.

1) Resumen: La contaminación atmosférica afecta la salud y el medio ambiente, lo que impulsa a gestionar y monitorear los contaminantes del aire en zonas urbanas.

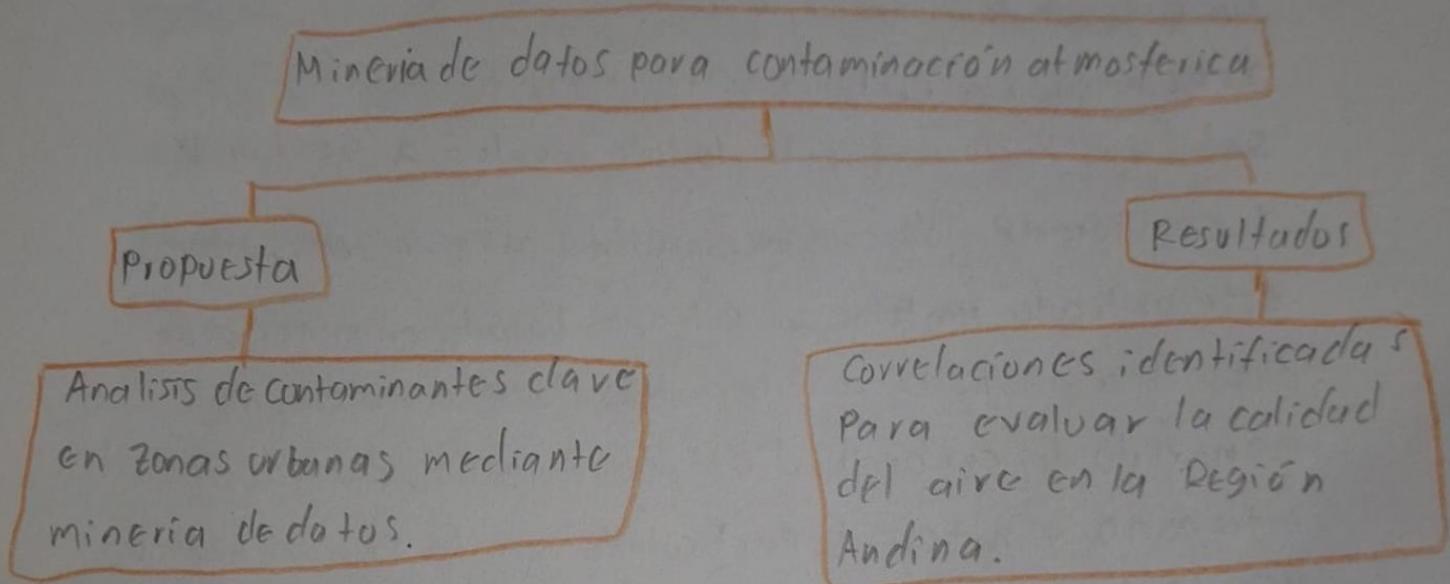
Este artículo propone un enfoque basado en minería de datos para analizar contaminantes clave como Ozono, monóxido de carbono, dióxido de azufre, dióxido de hidrógeno y material particulado en la Región Andina. Usando datos de Quenca, Ecuador, se identificaron correlaciones entre contaminantes y se apoyó la evaluación de calidad del aire.

2) Reflexión: La contaminación atmosférica no solo es un desafío ambiental, sino también una amenaza directa para la salud humana y la sostenibilidad urbana. Este problema requiere un enfoque integral que combine tecnología y ciencia para entender y mitigar sus efectos. El análisis de datos, como se presenta en el artículo, demuestra cómo la información puede ser una herramienta efectiva para tomar decisiones informadas y promover soluciones más efectivas nacidas del conocimiento, la acción coordinada y el compromiso con el bienestar de nuestro entorno.

3) Bibliografía:

Autor: Diana Arce, Fernando Lima Marcos Orellana, John Ortega, Chester Sellers, Patricia Ortega
Año: 2018

4) Graficas:



optimizar cursos online
con guías pedagógicas
que superen problemas
de capacitación docente.

mejora en diseño
docente y usabilidad
estudiantil para una
educación de calidad!

Aplicación de patrones en el modelado de procesos de negocio

1) Resumen: Un modelo de negocio es una representación simplificada de cómo funciona una organización, útil para comunicar, mejorar y diseñar sistemas. facilita la conexión entre objetivos empresariales y software, mejorando calidad y trazabilidad. Los patrones de procesos estructuran operaciones organizacionales, optimizando su coordinación. Estos patrones se dividen en reglas y recursos, metas y procesos, y se integran al software mediante herramientas como SIAR adaptadas para implementar y gestionar estos modelos de manera eficiente.

2) Reflexión: Reflexionar sobre el modelado de negocios invita a reconocer su importancia como puente entre la visión empresarial y la tecnología. Al diseñar y patrónes y procesos claros, no solo optimizamos operaciones, sino que también construimos bases sólidas para la innovación y el crecimiento. Es un recordatorio de que la planificación estratégica y la implementación efectiva son clave para transformar ideas en realidades sostenibles.

3) Bibliografía:

Autor: Marcelo Marciszack, Claudia Castro, Claudia Sánchez, Andrea Delgado

Año, 2016

Conexión entre objetivos
empresariales y software
mediante patrones de pro-
cesos.

optimización organizacional
con SIAR para implementar
y gestionar modelos efici-
entemente.

Técnicas de minería de datos aplicadas al diagnóstico de entidades clínicas

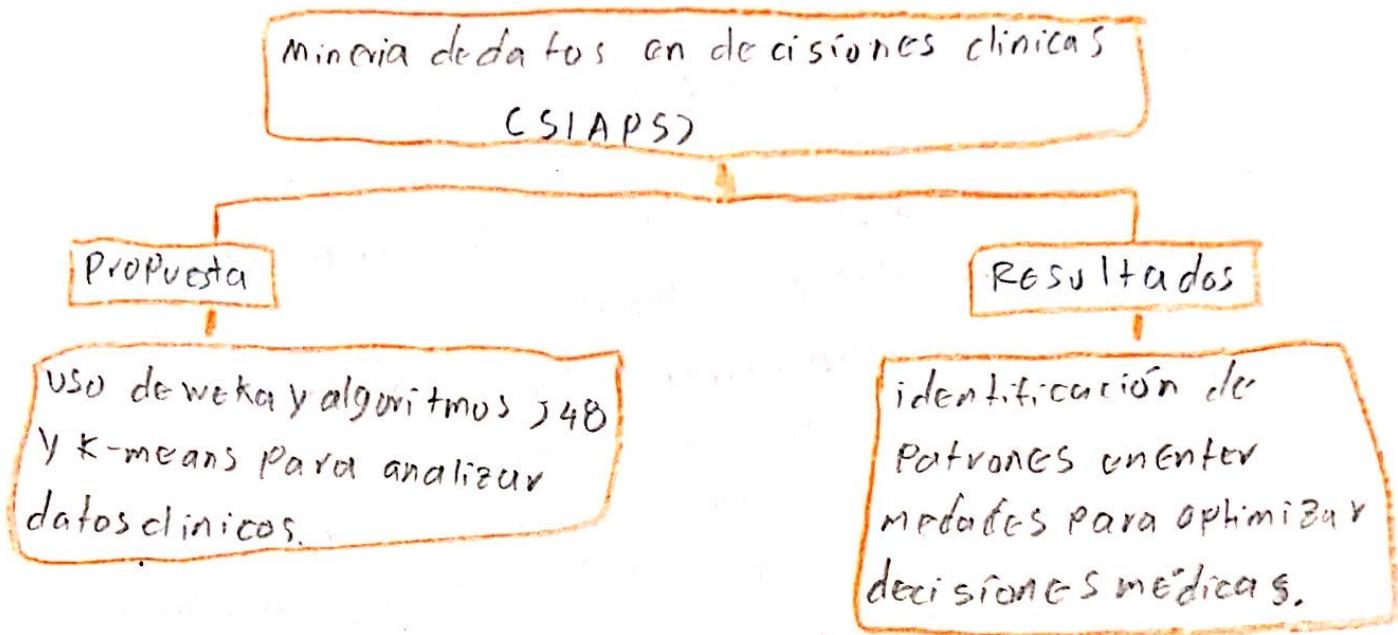
1) Resumen: La Universidad de las Ciencias Informáticas (UCI) desarrolla sistemas como el SIAPS, que aplica minería de datos para apoyar decisiones clínicas. Ante el volumen de datos en historias clínicas y la dificultad de análisis, se utilizan herramientas como WEKA y algoritmos como J48 y k-means para identificar patrones relacionados con enfermedades como la hipertensión arterial. Esto mejora la predicción y análisis, ayudando a comprender factores como edad, género y antecedentes familiares, optimizando la toma de decisiones médicas.

2) Reflexión: La aplicación de minería de datos en la medicina muestra cómo la tecnología puede transformar el análisis clínico, revelando patrones ocultos en grandes volúmenes de datos. Esto no solo mejora la toma de decisiones, sino que también resalta la importancia de la innovación para abordar problemas de salud global, como la hipertensión, permitiendo diagnósticos más precisos y estrategias preventivas más efectivas.

3) Bibliografía:

Autor: Frank Dávila Hernández, Yovanny Sánchez Corales
Año: 2012

4) Gráfica:



Construcción de un framework para la enseñanza de patrones de diseño.

D) Resumen: Los patrones de diseño ayudan a resolver problemas recurrentes en la ingeniería de software. En el Universidad Nacional de Río Cuarto, se usan plantillas genéricas para casos de uso como inserción y eliminación de datos. Se propone un framework que integra patrones como mediator, Data transfer object y registry, para crear sistemas distribuidos, reutilizar interfaces y reducir llamadas remotas. Usando Java y RMI, el framework busca una solución eficiente y flexible para el desarrollo de software.

E) Reflexión: El uso de patrones de diseño en software permite aplicar soluciones probadas a problemas comunes, mejorando la eficiencia y la sostenibilidad de los sistemas. Al integrar patrones y plantillas genéricas, se facilita la creación de proyectos más flexibles y escalables. Este enfoque destaca la importancia de aprender de la experiencia colectiva y adaptarse a diferentes contextos sin reinventar soluciones.

D) Bibliografía:

Año: Marcela Daniel P., Daniel Romero
Autor: 2006

Framework de Patrones de diseño para sistemas distribuidos

Propuesta

integrar patrones Cmediator, DTO, Registry) para sistemas distribuidos con Java y RMI.

Resultados

Reutilización de interfaces y reducción de llamadas remotas para ofrecer más fiabilidad

34

Desarrollo de un software como herramienta para el reconocimiento de patrones faciales caninos

1) Resumen: Este estudio presenta un software para reconocer la identidad de un canino mediante fotografías. Utiliza procesamiento de imágenes e inteligencia artificial para identificar los rasgos faciales de los caninos. El software fue desarrollado en Google Colab y emplea Inception-Resnet v2 para la detección facial. Se usaron las bibliotecas PyTorch, Support Vector Classifier y Directed Acyclic Graph Support Vector Machines para la identificación. El sistema alcanzó una precisión promedio del 93,70% entre dos clases y del 90,30% entre cuatro clases.

2) Reflexión: Este software muestra el poder de la inteligencia artificial para resolver problemas prácticos, como la identificación canina a través de imágenes. Facilita el reconocimiento preciso de animales, lo cual puede ser útil en diversas aplicaciones, desde anuncios hasta control de mascotas. La precisión alcanzada demuestra el potencial de las técnicas avanzadas de procesamiento de imágenes para mejorar la interacción entre tecnología y el mundo animal.

3) Bibliografía:

Autor: Carlos Emilio Burbano López, Marcos Leandro Castro
Año: 2027

4) Gráfica:

