Resúmenes:

La edad de Oro de la arquitectura de Software

Inicialmente, se usaba para describir sistemas de forma general. Hoy en día, es una disciplina fundamental para diseñar y construir software complejo. Gracias a diversos métodos y herramientas, los arquitectos de software pueden crear sistemas más robustos, escalables y mantenibles. En pocas palabras, la arquitectura de software ha pasado de ser una descripción para convertirse en una guía esencial para el desarrollo de software.

Implementación de Arquitectura de Software por el Dominio

Esto habla sobre un enfoque a la ingeniería de software para crear software, metodología para un enfoque al dominio de negocio, como funciona el negocio, como se refleja en el software, resolución de problemas, están vinculadas arquitecturas limpias y hexagonales, la arquitectura limpia busca mantener el código organizado y la hexagonal busca desacoplar el sistema, beneficios, este tiene mayor alineación con el sistema, mantenimiento más sencillo y una sostenibilidad más segura.

Arquitectura de software: fundamentos, teoría y práctica  
Esta representa el conjunto de decisiones, estructurales, despliegues, patrones de cambio, y tiempos de ejecución, en esta se usan patrones reutilizables, destaca tendencias emergentes y futuras, como la web y REST. Es esencial para el desarrollo, lo q cual si no hay algo bien planteado, no hay un buen programa, ya que esta es la base, porque este es el boceto para los desarrolladores, ya que esta puede ser más efectiva

Arquitectura para Herramienta de Costos y Programación:

arquitectura de software educativo para la materia de “Costos, Presupuestos y Programación de Obra” en Ingeniería Civil, en la Universidad Francisco de Paula Santander Este software apoya el aprendizaje autónomo de los estudiantes, gestionando temas como el análisis de precios unitarios (APU), planificación de actividades y optimización de recursos. Utilizando diagramas UML.

Documentación y análisis de los principales frameworks de arquitectura de software en aplicaciones empresariales  
La arquitectura es clave, como herramientas ERP y CRM, la arquitectura en capas, cliente y servidor, igual para sistemas, arquitectura en 3 capas.

Las empresas, cada uno maneja su arquitectura, ERP, gestiona el inventario de una empresa, las cuentas, es la parte financiera CRM, gestión de contactos, análisis de datos, marketing, mayormente, en esos dos grupos se usan diferentes arquitecturas.

Análisis comparativo de patrones de diseño MVC y MVP para el rendimiento de aplicaciones web

En este artículo se evaluó la eficiencia de los patrones de diseño en la realización de un proyecto, enfocándose en elegir el patrón más adecuado. primero fue identificar y validar los patrones existentes, seleccionando el MVC y el MVP. Cada uno tuvo su propio proyecto para realizar la prueba, donde se evaluaron diferentes aspectos, como el tiempo de desarrollo, las líneas de código y el uso de memoria ram, entre otros.

Arquitectura Hexagonal

Arquitectura de puertos, separa responsabilidades, facilita el desarrollo, mantenimiento, dividir el proyecto en una capa central, esta se comunica a través de puertos y adaptadores. Esto mejora la mantenibilidad y facilita las pruebas, aunque puede ser más compleja de implementar en aplicaciones simples.

Modelo y Herramienta software para la gestión de riesgos en el desarrollo de aplicaciones web soportado en el estándar ISO/IEC 27005

Perspectivo conceptual, define 5 fases, creación del proyecto parametrización, evaluación, identificador de riesgos y gestión de riesgos, perspectiva lógica, usa UML para diagramar, esta esta enfocada a proyectos grandes y medianos, ciclo de vida viene siendo en incrementales e iterativos. Estándar de seguridad, lo cual puede identificar los problemas en etapas tempranas  
5 principales fases de desarrollo en esto

Una arquitectura para una Herramienta de patrones de Diseño

La arquitectura q esta escrita, describe una arquitectura para integra una arquitectura orientada a objetos, esta aumenta la reutilización de código, permite al usuario vistas gráficas, permite manipular patrones de modelado básico, optimiza la eficiencia y reutilización de software, usa 3 patrones, composite, permite estructurar elementos en jerarquías más flexibles, comad, facilita la ejecución, observar, sincroniza diferentes componentes, facilita las interfaces gráficas, lo q ve y hace q funciones, es un gran ejemplo de cómo estos puede mejorar la eficiencia y reutilización, esto se vuelve más efectivo

Mapeo de arquitecturas de software

Se refiere a un proceso de análisis y de identificar como esta estructurado internamente un proyecto, servicios, base de datos, documentos, hay situaciones donde la documentación no esta actualizada, problemas, cuando esta mal gestionado y proponen soluciones para el desarrollo de sistemas, si la información recuperada no está bien documentada hay problemas.

Arquitectura de Microservicios para Desarrollo Web

El estudio analiza los desafíos de la arquitectura monolítica en la CGTIC de la Asamblea Nacional del Ecuador, como el mantenimiento y la escalabilidad. Propone adoptar microservicios, un enfoque modular y escalable, para mejorar el desarrollo de aplicaciones web

Patrones de Usabilidad en la Arquitectura de Software

Nos habla como el proyecto status nos mejora nuestro proyecto desde el inicio, nos propone incluir opciones útiles como deshacer y cancelar, esta hace q sea fácil de usar, la usabilidad, nos hace ajustes,

Arquitectura de software para entornos móviles

Se define una arquitectura móvil, porque de esta forma permite ayudar a estandarizar metodologías, esto ha tenido un impacto significativo, para crear aplicaciones de mayor tamaño

Marco de trabajo para Seleccionar un Patrón Arquitectónico en el desarrollo de Software

El artículo destaca la importancia de la arquitectura de software en proyectos tecnológicos, proponiendo un marco para seleccionar patrones como MVC, MVP y Microservicios según las necesidades del software.

Desarrollo de aplicaciones web utilizando el patrón de diseño MVC

Es útil para sistemas de software interactivos, pero su implementación en web es complicada ya que particiona las aplicaciones durante su diseño.

Análisis comparativo de patrones de Diseño de Software

Aborda la importancia de los patrones de diseño como soluciones estandarizadas a problemas comunes en el desarrollo, destaca su capacidad para evitar duplicaciones de código, y facilitar la reutilización. Se analizan 5 patrones clave: Templated Method, MVC, MVP, model front controller y MVVM, evaluando sus componentes, ventajas y desventajas, este habla de cómo no hay un patrón superior, dado a q cada uno cumple un rol especifico, son importantes para la organización, mantenimiento y calidad.

Marco de Trabajo para Seleccionar un Patrón Arquitectónico en el desarrollo de Software

El artículo aborda los desafíos de seguridad en arquitecturas de microservicios, proponiendo el patrón Microservice Security Pattern API Gateway (MSPAG) que usa JWT y H256 para centralizar la autenticación y proteger endpoints.

Análisis comparativo de Patrones de Diseño de Software

Los patrones de diseño son fundamentales para crear software robusto y escalable. Cada uno ofrece soluciones específicas según el contexto y los requisitos del proyecto. Se comparan patrones como Template Method, MVC, MVP, Front Controller y MVVM, considerando factores como lenguaje, complejidad y seguridad.

Buenas prácticas en la construcción del software

Explora arquitecturas y metodologías de desarrollo utilizadas para crear soluciones eficientes flexibles en el sistema. Detalla la arquitectura de solución, que guía el diseño de estructuras integradas.

Introducción a los Patrones de Diseño

ARTICULOS DE LA FICHA DEL FIN DE SEMANA

Johan Calderón Perdomo

Revisión de elementos conceptuales para la representacion de las arquitecturas de referencias de software

la arquitectura de software un sistema de alto nivel definiendo elementos, componentes mediante arquitecturas de referencia atreves de enfoques estructurales y diseños.

Maryury Bonilla Gonzales

Monolitos vs Microservicios en arquitectura de software

"este compara las arquitecturas monolíticas y de micro servicios analizando sus ventajas y desventajas en diferentes escenarios para que pueda ser más eficiente su rendimiento"

Carlos Andrés Pantoja Jaramillo

Artículo 8: Impacto de implementaciones web del patrón MVC en los requisitos de calidad percibidos

"el articulo evalúa el modelo MVC en las aplicaciones web y su impacto de en los atributos de calidad y también utiliza unos filtros y tuberías para atreves de cortinas en Python que concluye una investigación ."

Stefanny Nikoll Hidalgo Urrea

Artículo 16: Desarrollo de una arquitectura de software para el robot móvil Lázaro

La arquitectura de software tiene 3 niveles el primero gestiona los componentes básicos el segundo proporciona librería para crear aplicaciones de control y el tener interfaces para los usuarios.

Juan David Cerquera Salazar

Artículo 7: Marco de trabajo para seleccionar un Patrón Arquitectónico en el desarrollo de software

Se realizo una investigación sobre las arquitecturas más utilizadas que son MVC , MVP, Micro servicios y que cada uno desarrolla su función dependiendo de lo que quiere el Usuario y el desarrollador.

Erick Daniel Peña Cedeño

Artículo 2: Arquitectura de software basada en microservicios para desarrollo de aplicaciones web

Esto explora el desarrollo de una arquitectura de software más flexible y moderna que busca superar las limitaciones por eso los microservicios dan una solucionas más flexible más eficiente para un mantenimiento función del programa.

Marlon Estiven Torres Medina

Artículo 8: Patrones de Diseño (XII): Patrones Estructurales – Flyweight

El patrón Flyweigth, permite reducir el uso de memoria compartiendo tantos datos como sea posible con todos los otros objetos similares. Este se usa más que todo para videojuegos y chats