

ANYI ZUJEY = Un diseño de software permita durabilidad, funcionalidad, como arquitectos de software o programadores debemos velar por que los programas que realicemos sean funcionales a mediano y largo plazo.

AURA = Como funciona el negocio, como resolver algunos problemas, dentro de esto se vincula al arquitectura limpia y hexagonal, limpia busca mantener el código organizado y hexagonal busca desacoplar la lógica de negocio y permite tener adaptabilidad diagonal y vertical y es la base de los microservicios.

CAMILO = La arquitectura de software es esencial para el desarrollo, ya que es la base con la que se realiza el programa, es un boceto que se necesita para desarrollar un software más efectivo y así evitamos confusiones.

CAROLINA = Habla de la optimización de recursos y se apoya de diagramas UML y diagramas de frecuencia, facilitando el impacto financiero.

NARVAEZ = La arquitectura de software es clave en el software de sistemas empresariales, las empresas tienen sectores erp, es encargada de gestionar la parte financiera y crm es la gestión de contacto, márketing y analista de datos. Se utilizan arquitectura en capas, cliente servidor y arquitecturas en tres capas.

BAHAMON = El mvc es mas eficaz que el mvp en el tiempo de desarrollo, líneas de código, uso de la memoria ram y tiempo de respuesta.

PENNA = Se basa en el uso de puertos y adaptadores para promover la separación de responsabilidades. Esto permite un diseño modular, independiente y flexible, facilitando la evolución del sistema.

GASCA = El modelo iso 27005 trata de gestionar los riesgos en el desarrollo de aplicaciones web, se usa para proyectos medianos y grandes.

JULIAN = permite manipular y gestionar patrones como modelado base, Optimiza la eficiencia y usabilidad del software, ayudando a los programadores a desarrollar

ARIZA = El mapeo sirve para encontrar problemas en el código, ayudando a mantener y mejorar programas más fáciles si llegan a ser antiguos o complejos

MAR = La arquitectura monolítica es algo estático, no cambia y fue utilizado en la asamblea nacional de ecuador y quieren pasar el proyecto a la arquitectura de microservicios.

MARIA JOSE = Habla sobre como el proyecto statu mejora la usabilidad, evita trabajos costosos y nos ayuda a crear un sistema funcional. Además, nos ayuda a encontrar los errores desde el inicio del proyecto.

CHARRY = Las aplicaciones móviles son de mayor tamaño y complejidad, o sea, son más estables.

YO = La elección de patrones arquitectónicos tiene un marco de trabajo el cual se basa en la mejora de la calidad, rendimiento, mantenimiento y adaptabilidad del software. Algunos son mvc, mvp, microservicios y arquitectura en la nube.

MAYDY = El mvc es fundamental para la modularidad y la organización de sistemas interactivos.

MAYRA = Aborda la importancia de los patrones de diseño como soluciones estandarizadas a problemas comunes en el desarrollo de software, destacando su capacidad para evitar la duplicación de código y facilitar la reutilización.

PATRICIA = Presenta un marco de trabajo para seleccionar patrones arquitectónicos en el desarrollo de software, abordando problemas de desacoplamiento y falta de conocimiento arquitectónico que afectan la calidad del producto.

WILLIAM = Este artículo habla sobre las diferentes arquitecturas y metodologías de software utilizadas para crear soluciones eficientes y flexibles

YORDY = Resalta la importancia que conlleva el empleo de patrones de diseño, ya que nos muestra lo efectivo que es para abordar los desafíos habituales en la creación de software.

VALEN SILVA = El artículo analiza varios patrones populares como mvc, mvp, front controller, template method y mvvm. Se comparan estos patrones en base a su escalabilidad, facilidad de implementación, complejidad, uso, seguridad y compatibilidad con diferentes lenguajes de programación.

Ficha fin de semana

CALDERON (cap. 4.1) = Organiza un sistema a nivel alto, definiendo elementos y sus relaciones. Facilitando el desarrollo al reutilizar componentes.

MARYURY (cap. 5.5) = Compara las arquitecturas monolíticas y de microservicios, analizando sus ventajas, desventajas y los escenarios en los que cada uno es más eficiente.

PANTOJA (cap. 6.8) = El artículo estudia dos variantes de la implementación del patrón mvc en aplicaciones web y su impacto en los atributos de calidad percibidos, como el tiempo de respuesta y la escalabilidad.

NIKOL (cap. 8.16) = La arquitectura tiene tres niveles, primero gestiona los componentes básicos, el segundo proporciona librerías para crear aplicaciones de robot y el tercer nivel ofrece una interfaz de usuario con un papel de control y un simulador 3d para monitorear y programar el robot.

CERQUERA (cap. 9.7) = Habla de las arquitecturas más utilizadas según desarrolladores y arquitectos de software, como, arquitectura en la nube, microservicios, mvc y mvp.

ERICK (cap. 10.2) = Microservicios explora el desarrollo de una arquitectura de software más flexible y moderna que busca superar las limitaciones de la tradicional arquitectura

monolítica, la arquitectura de microservicios es más adaptable y eficiente para el desarrollo de aplicaciones.

MARLON (cap. 14.8) = El patron Flyweight permite reducir el uso de memoria ya que comparte datos con otros objetos similares. Dando eficiencia en la memoria.