Resumen de la Sesión - 23/11/2024

Angie(yo)

La arquitectura de software era muy general en la década de 1980, pero se ha convertido en un proceso central para que los proyectos de software mantengan la escalabilidad y la disponibilidad.

Aura

En el diseño impulsado por dominio (DDD), los sistemas "impulsados por dominio" se centran en comprender y modelar el negocio o problema para cuya solución está diseñado el software.

Utilice un modelo de dominio que refleje terminología y reglas comerciales. Este enfoque hace que el software sea más fácil de entender, crea un lenguaje común entre desarrolladores y expertos empresariales y permite una fácil adaptación y desarrollo.

Camilo Bautista

La arquitectura de software es fundamental para el desarrollo. Es la base del proyecto, y un análisis equivocado puede llevar al fracaso, lo que supone pérdida de tiempo y dinero.

Carolina

El uso de herramientas como los diagramas UML proporciona visibilidad del sistema o programa que se está desarrollando. Esto permite que tanto técnicos como no técnicos tengan una idea clara del proyecto.

Cristian Narváez

La arquitectura de software es la clave para los sistemas empresariales como ERP y CRM, la optimización y la gestión de procesos. Análisis del marco arquitectónico típico: Arquitectura en capas: una organización jerárquica de responsabilidades que es fácil de mantener pero que puede ser redundante. Cliente-Servidor: Separa los roles de cliente y servidor, ideal para sistemas descentralizados. Arquitectura de tres capas: modular y escalable, dividida en capa de presentación, capa lógica y capa de datos. El uso adecuado garantiza que la aplicación esté alineada con los objetivos comerciales.

Cristian Bahamon

MVC es más rápido y eficiente en términos de rendimiento, más liviano en la máquina y requiere menos tiempo de codificación y desarrollo. MVC es más adecuado para aplicaciones grandes, mientras que MVP es más adecuado para aplicaciones pequeñas.

Sebastián Penna

La arquitectura hexagonal (o "puertos y adaptadores") está diseñada para aislar la lógica empresarial del resto del sistema.

Divide el sistema en núcleo (lógica empresarial) y adaptadores (base de datos, interfaz de usuario, servicios) que interactúan con el mundo exterior. Mejore el mantenimiento y las pruebas del sistema con el desacoplamiento.

Gasca

Modelo de gestión de riesgos para el desarrollo de aplicaciones web basado en la norma ISO/IEC 27005.Perspectiva conceptual: 5 fases (creación del proyecto, determinación de parámetros, evaluación, identificación de riesgos, gestión de riesgos). Una perspectiva lógica: diagramas UML para modelar categorías y relaciones. Perspectiva física: estructura de la base de datos. Este modelo admite ciclos de vida incrementales e iterativos y está diseñado para proyectos medianos y grandes.

Julián

Integrar patrones de diseño en la arquitectura de herramientas orientadas a objetos.Le permite crear, manipular y gestionar modelos como bloques de construcción, optimizando la eficiencia y la reutilización. Utiliza patrones como Composite, Command y Obse**rver** para implementar la interacción entre la interfaz de usuario y la lógica interna. Proporciona vistas gráficas, jerárquicas y de código para mantener la coherencia del sistema.

Valentina Ariza

Mapeo del sistema restaurado en la vista de arquitectura del sistema de software:

Se identifican las investigaciones existentes sobre métodos de recuperación, aspectos de la arquitectura de recuperación y mecanismos de representación. Los resultados indican el tipo, finalidad y métodos de investigación utilizados para mejorar la representación de las vistas arquitectónicas.

Maria José

El proyecto STATUS integra la usabilidad en la fase de diseño con estados como deshacer, cancelar y multilingüe. A diferencia de los métodos tradicionales, STATUS permite ajustes continuos durante el proceso de diseño. Proporciona un sistema intuitivo y accesible desde cero, mejorando la experiencia del usuario sin desperdiciar recursos.

Mariana Charry

La evaluación de los sistemas operativos ha influido en el desarrollo de aplicaciones móviles, permitiendo mayor tamaño y complejidad.

Este artículo define soluciones de arquitectura móvil basadas en principios generales de arquitectura de software. Ayudar a estandarizar y adaptar métodos, técnicas, procesos y enfoques.

Mariana Gonzalez

Este artículo explica la importancia de la arquitectura de software en el diseño de ingeniería. Se presenta un marco para seleccionar patrones arquitectónicos para mejorar la calidad, el rendimiento, la mantenibilidad y la adaptabilidad del software. Este marco ayuda a los desarrolladores y arquitectos a tomar decisiones basadas en las características del proyecto, como el tipo de desarrollo y los requisitos críticos. Describe patrones como MVC, MVP, microservicios y arquitectura de nube y su aplicabilidad en función de los contextos y necesidades del software. Finalmente, el marco se valida mediante un caso de uso en el que el usuario opta por recibir recomendaciones arquitectónicas personalizadas.

Maydy

El patrón Modelo-Vista-Controlador es una herramienta poderosa para desarrollar aplicaciones web y sistemas interactivos, pero la implementación debe adaptarse al tamaño y la naturaleza del proyecto. Al utilizar estrategias como la partición flexible, puede aprovechar al máximo MVC y al mismo tiempo mantener un enfoque de desarrollo ágil y adaptable. Esto hace que el proceso sea más eficiente y escalable y sienta una base sólida para el futuro desarrollo de software.

Mayra

Los patrones de diseño son soluciones estándar a problemas comunes de desarrollo de software que evitan la duplicación de código y fomentan la reutilización.

Se analizan cinco modelos: método de plantilla, MVC, MVP, controlador frontal y MVVM. Cada modelo tiene ciertos componentes, ventajas y desventajas, y ningún modelo es mejor que los demás. El estudio concluyó que los patrones de diseño son esenciales:Mejore la organización, la mantenibilidad y la calidad del código. Promover la comprensión y el desarrollo del sistema. Este análisis orienta a los desarrolladores a elegir el modelo más adecuado en función de las necesidades del proyecto, promoviendo así prácticas más eficientes y sostenibles.

Patricia

Este artículo presenta un marco para la selección, el desacoplamiento y el abordaje de las lagunas de conocimiento arquitectónico que afectan la calidad del producto. Se basa en cinco fases que identifican patrones clave (como MVC, microservicios y arquitectura de nube) y definen reglas para seleccionarlos en función del tipo de desarrollo y la funcionalidad requerida. A través de pruebas de prototipos, el sistema mejora el tiempo, la calidad, la

escalabilidad y la mantenibilidad. Concluyó que la solución es una herramienta práctica que puede ayudar a arquitectos y desarrolladores a elegir modelos adecuados para construir estructuras desde el principio de un proyecto.

Valentina Silva

Los patrones de diseño son una herramienta importante para crear software estable y escalable. Cada modo ofrece estructuras y soluciones específicas, pero la elección correcta depende del contexto y los requisitos del proyecto. En este análisis, comparamos patrones populares como métodos de plantilla, MVC, MVP, controlador front-end y MVVM en términos de complejidad de programación, legibilidad, seguridad y usabilidad.

William

Este artículo analiza diferentes arquitecturas de software y métodos de desarrollo para crear soluciones eficientes y flexibles en sistemas de TI. Primero, describe la arquitectura de la solución (que guía el desarrollo de estructuras de integración para responder a las necesidades actuales y futuras) y la arquitectura del software (incluidos modelos y tecnologías específicos). Luego describe el estilo arquitectónico, por ejemplo:

- Número de capas
- Monolito
- Microservicios (arquitectura basada en eventos)
- cliente-servidor
- Estos estilos organizan la estructura de comunicación y aplicaciones.

Yordy

Este artículo destaca la importancia de los patrones de diseño como una herramienta eficaz para resolver problemas de desarrollo de software. Elegir y utilizar el modelo adecuado para cada situación permite crear un sistema flexible. Esto tiene un impacto positivo en la calidad y escalabilidad del código, que son esenciales para los proyectos modernos. Además, los modelos no limitan la creatividad, sino que fomentan un desarrollo y una resolución de problemas más asertivos.

Johan Calderón

La arquitectura de software organiza sistemas a un alto nivel definiendo elementos y sus relaciones. Simplifique el desarrollo reutilizando componentes utilizando arquitecturas de referencia. Se desarrolla utilizando métodos, modelos y escenarios estructurales para guiar el diseño y la comprensión de sistemas complejos.

Maryury

El artículo compara las arquitecturas **monolíticas** y **de microservicios**, analizando sus ventajas, desventajas y los escenarios en los que cada una es más eficiente. También examina casos de empresas como Amazon y eBay que han migrado de una arquitectura a otra, explorando las dificultades y beneficios en términos de escalabilidad, mantenimiento y desarrollo.

Andrés Pantoja

Este artículo examina dos variaciones del patrón MVC para el desarrollo web y su impacto en el tiempo y la escalabilidad. También analizó el uso de tubos y filtros y concluyó que los tubos son más eficaces, aunque su implementación requiere más investigación.

Nikoll

La arquitectura descrita divide el control del robot en tres niveles. Nivel básico: gestiona componentes como motores y sensores y proporciona acceso al hardware. Nivel medio: proporciona bibliotecas de funciones para simplificar el desarrollo de aplicaciones de control y actúa como puente entre hardware y software. Nivel avanzado: incluye una interfaz de usuario con un panel de control y un simulador 3D para el seguimiento y programación del robot en un entorno virtual. La arquitectura está desarrollada en C# y es modular y extensible. Estas pruebas demuestran una integración efectiva entre diferentes capas, simplifican el desarrollo y reducen el riesgo mediante el uso de simulación antes de la implementación real.

Juan Cerquera

Las arquitecturas más utilizadas se estudiaron junto con arquitectos y desarrolladores de software. Los más importantes son: Arquitectura de la nube: seguridad y flexibilidad. MVC: Mantenimiento, Rendimiento, Velocidad y Memoria. Microservicios: Mantenimiento, Rendimiento, Seguridad y Flexibilidad. MVP: modificabilidad, rendimiento, estabilidad, flexibilidad y modularidad. Principales dispositivos y aplicaciones por arquitectura: Arquitectura de la nube: aplicaciones web. MVC: Aplicaciones móviles, de escritorio y web. Microservicios: Aplicaciones Web. MVP: Aplicaciones móviles y web.

Erick

El artículo Microservicios explora cómo esta arquitectura supera las limitaciones de los sistemas monolíticos tradicionales. Los microservicios optimizan el desarrollo y mantenimiento de aplicaciones al permitir que los componentes funcionen de forma autónoma e independiente, proporcionando soluciones eficientes y escalables.

Marlon

Este artículo describe el modo Flyweight, que reduce el consumo de memoria al compartir datos entre objetos similares. Se encuentran ejemplos de su uso en sistemas donde la eficiencia de la memoria es crítica, como aplicaciones gráficas y videojuegos.