

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

La arquitectura de software es como el mapa que organiza todo un sistema, explicando cómo se conectan los diferentes elementos. Lo chévere de esto es que ayuda a reutilizar cosas ya hechas, ahorrando tiempo. A lo largo del tiempo, ha ido mejorando con nuevos enfoques y patrones, lo que hace que diseñar sistemas complejos sea más sencillo y eficiente.

Monolitos vs. Microservicios en Arquitectura de Software

Este artículo compara dos formas de construir software: los monolitos y los microservicios. Por un lado, los monolitos son como un todo pegado; por otro, los microservicios son partes independientes. Analizan cuándo usar cada uno dependiendo del caso, explicando pros y contras. Hablan de ejemplos como Amazon y eBay, que se cambiaron a microservicios, y mencionan que, aunque es una chimba para escalar y mantener, tiene sus retos.

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

El patrón MVC es esa estructura clásica para hacer apps web que organiza todo en tres partes: modelo, vista y controlador. Este artículo compara la versión clásica de MVC, que funciona por llamadas y respuestas, con una nueva que usa tuberías y filtros en Python. Aunque esta última pinta prometedora, dicen que todavía falta estudiarla más antes de lanzarse de cabeza.

Desarrollo de una arquitectura de software para el robot móvil Lázaró

Aquí cuentan cómo hicieron la arquitectura para manejar un robot llamado Lázaró. Tiene tres niveles: uno para controlar las piezas del robot, otro para crear aplicaciones que lo manejen, y un tercero con una interfaz de usuario para que cualquiera pueda interactuar con él. Todo esto lo programaron en C# y probaron que funciona bien, mostrando cómo estructurar el robot hace más fácil mejorar y reutilizar componentes.

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

El man hizo una encuesta a desarrolladores para ver cuáles patrones arquitectónicos son los más usados y en qué sobresale cada uno:

- **Arquitectura en la nube:** Es flexible y segura, ideal para apps web.
- **MVC:** Muy rápido y fácil de mantener, sirve para móviles, escritorio y web.
- **Microservicios:** Perfectos para web, con buen rendimiento y flexibilidad.
- **MVP:** Genial para móviles, con enfoque en modularidad y estabilidad.
Cada uno se adapta a distintos dispositivos y necesidades, lo que ayuda a los desarrolladores a decidir mejor qué usar.

Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web

Aquí explican cómo los microservicios son la respuesta a los problemas de las arquitecturas monolíticas. En vez de tener todo pegado en un solo bloque, con microservicios se separa cada función para que trabaje sola, lo que hace que las aplicaciones sean más fáciles de mantener, escalar y mejorar. Aunque suena bacano, también mencionan que no todo es perfecto y hay que manejar bien el cambio.

Patrones de Diseño (XII): Patrones Estructurales - Flyweight

El patrón Flyweight es como compartir datos entre objetos para ahorrar memoria. Es súper útil en cosas como gráficos y videojuegos donde cada recurso cuenta. Sin embargo, también advierten que no siempre es la mejor opción y que hay que pensar bien antes de implementarlo.