

## **Resumen de exposiciones sobre arquitectura de software**

**Angi Lizeth Trujillo Gonzáles**

### **Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web**

Los microservicios son identificados como la mejor alternativa frente a las arquitecturas monolíticas, al ser más escalables y adaptables al desarrollo de aplicaciones web.

**Anyi Zuley Gómez Casanova**

### **La edad de oro de la arquitectura de software**

Desde los años 80, la arquitectura de software pasó de ser solo una descripción del sistema a convertirse en una disciplina fundamental para crear software robusto, escalable y mantenible, adaptándose constantemente a los cambios tecnológicos.

**Aura María Fierro Fierro**

### **Implementación de una arquitectura guiada por el dominio**

El diseño guiado por el dominio (DDD) permite alinear los sistemas con las necesidades del negocio, usando arquitecturas limpias y hexagonales para desacoplar y mantener el código eficiente y entendible.

**Camilo Andrés Bautista Cuellar**

### **Arquitectura de software: fundamentos, teoría y práctica**

La arquitectura de software implica tomar decisiones estructurales clave para garantizar sistemas robustos y escalables. Sin una buena base arquitectónica, no se logra un software de calidad.

**Carolina Martínez Cortés**

### **Arquitectura para Herramienta de Costos y Programación**

Se desarrolla una herramienta educativa que utiliza diagramas UML para planificar y medir costos, apoyando el aprendizaje autónomo con una metodología secuencial.

**Cristian Fernando Narváez Sánchez**

### **Documentación y análisis de frameworks arquitectónicos**

Los frameworks más comunes en el ámbito empresarial son la arquitectura en capas, cliente-servidor y en tres capas, utilizados para sistemas ERP y CRM, favoreciendo la escalabilidad y el mantenimiento.

**Cristian JeanPool Bahamón Granado**

### **Análisis comparativo MVC y MVP para el rendimiento de aplicaciones web**

El patrón MVC demostró ser más eficiente que MVP en pruebas de rendimiento, considerando factores como uso de CPU, líneas de código y tiempos de respuesta.

**Jhon Sebastián Penna Arias**

### **Arquitectura hexagonal**

Promueve la separación de responsabilidades mediante una capa central con lógica de

negocio que se comunica con puertos y adaptadores, facilitando el desarrollo y mantenimiento.

**José Gasca**

#### **Modelo y herramientas de software para la gestión de riesgos**

Basado en ISO/IEC 27005, se propone un modelo con cinco fases: creación, parametrización, evaluación, identificación y gestión de riesgos, apoyado en diagramas UML y bases de datos.

**Julian David Fierro**

#### **Una arquitectura para una herramienta de patrones de diseño**

Se destacan patrones como *Composite* para estructurar jerarquías y *Observer* para sincronizar componentes, mejorando la flexibilidad y organización del software.

**Mariana Charry Prada**

#### **Arquitectura de software para aplicaciones móviles**

Se define una arquitectura estándar que permite el desarrollo de aplicaciones móviles más robustas y mantenibles, adaptándose a las necesidades actuales del mercado.

**Mariana González**

#### **Marco de trabajo para seleccionar un patrón arquitectónico**

Analiza patrones como MVC, MVP, microservicios y arquitecturas en la nube, ayudando a elegir la mejor opción según las necesidades del desarrollo de software.

**Maryuri Bonilla González**

#### **Monolito vs Microservicios en arquitectura de software**

Se comparan ambas arquitecturas, destacando los microservicios por su capacidad de escalabilidad y su facilidad para adaptarse a cambios, a diferencia de los monolitos más rígidos.

**Yordy Erik Núñez Pineda**

#### **Introducción a los patrones de diseño**

Se describen los principales patrones: estructurales, comportamentales y creacionales, como herramientas esenciales para organizar y mejorar el desarrollo de software.

**Willian Esteban**

#### **Buenas prácticas en la construcción del software**

Aplicar buenas prácticas arquitectónicas hace los proyectos más flexibles y aptos para resolver necesidades presentes y futuras, garantizando escalabilidad y mantenimiento.