LA IMPORTANCIA DE LA ARQUITECTURA DE SOFTWARE

M.Sc. Ing. Saul Mamani Mamani

Software Architect y Desarrollador de Software

ORBIS COMPLIANCE LLC – USA

luaso_1@yahoo.es

RESUMEN

Se pretende mostrar la importancia que tiene el modelado de una buena arquitectura para garantizar el éxito y la calidad de los proyectos software. Se mostrarán ejemplos de arquitecturas desarrolladas por mi autoría que tuvieron éxito, además de ejemplos con la notación C4Model.

1. INTRODUCCIÓN

La programación y el desarrollo de software son componentes fundamentales en el mundo actual, donde la demanda de aplicaciones eficientes y confiables es creciente. Antes de escribir código, es esencial comprender conceptos básicos como algoritmos, programas y software, así como la importancia de aplicar ingeniería de software para garantizar productos de calidad. Esta sección introduce los conceptos y destaca la necesidad de una planificación adecuada antes de la implementación de cualquier proyecto informático.

- **Algoritmo:** Conjunto ordenado de pasos para resolver un problema, con inicio y fin definidos.
- **Programa:** Secuencia de instrucciones en un lenguaje de programación para ejecutar una tarea específica.

- **Software:** Conjunto de programas, procedimientos, datos y documentación asociados al funcionamiento de un sistema informático.

2. DESARROLLO

El desarrollo de software no se limita a escribir código; requiere aplicar metodologías de ingeniería de software, garantizando que los sistemas sean confiables, seguros y eficientes. Según R. Pressman, la ingeniería de software es el conjunto de métodos, técnicas y herramientas para desarrollar y mantener software de calidad, rentable y funcional en máquinas reales.

Importancia de la arquitectura de software

Antes de codificar, es necesario diseñar una arquitectura sólida, que sirva como plano del sistema:

- Define la estructura de alto nivel del software.
- Permite organizar los proyectos medianos y grandes.
- Involucra roles especializados como ingenieros de software, arquitectos de sistemas y diseñadores de base de datos

Ciclo de vida de la arquitectura

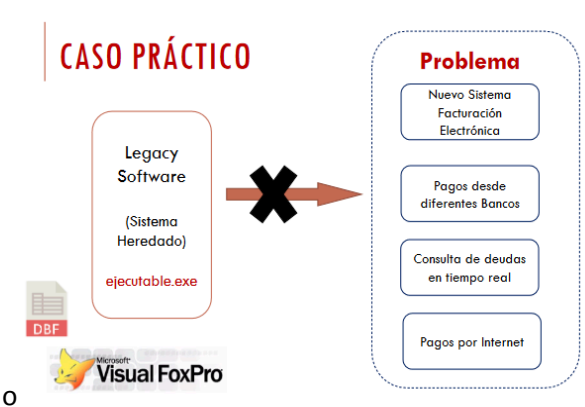
1. **Requerimientos:** Captura y documentación de requerimientos de alto nivel.
2. **Diseño:** Modelado de la estructura mediante patrones de diseño y selección de tecnología.
3. **Documentación:** Registro de decisiones a través de diagramas y documentación técnica.
4. **Evaluación:** Verificación temprana de la arquitectura para reducir riesgos.
5. **Implementación:** Construcción del sistema según la arquitectura definida.

Buenas prácticas y patrones de arquitectura

- **Principios:** ETC (Easy To Change), DRY (Don’t Repeat Yourself), Reversibilidad, Orthogonality, Tracer Bullet.
- **Patrones comunes:** Monolítica, Cliente/Servidor, MVC, SOA/Microservicios.
- **Modelado:** Uso de notaciones como **C4 Model**, que permite representar jerárquicamente el contexto, contenedores, componentes y clases del software.

Caso práctico

Se presenta un ejemplo de aplicación en la empresa **SELA Oruro**, donde la planificación arquitectónica y la definición de requerimientos permitieron diseñar un sistema eficiente, seguro y escalable, siguiendo todas las buenas prácticas de ingeniería de software.



o

Figura 1: Problema del caso práctico

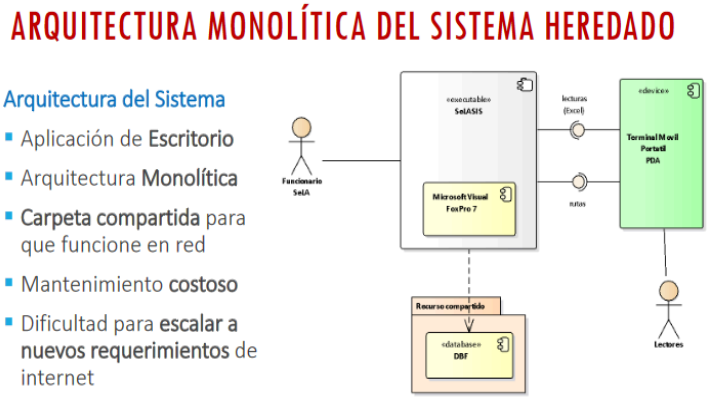


Figura 2: Descripción del problema del caso práctico

Figura 3: Base de datos del sistema heredado

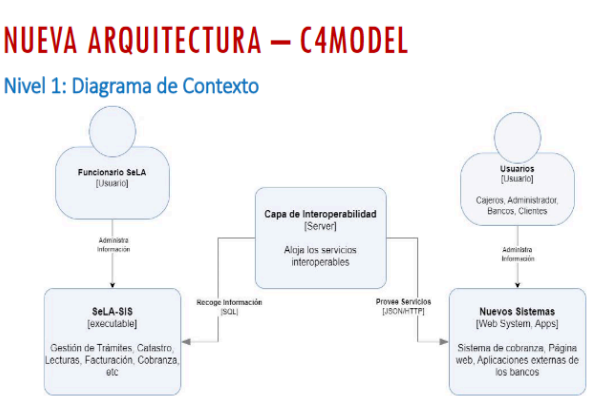


Figura 4: Diagrama de Contexto de la nueva arquitectura

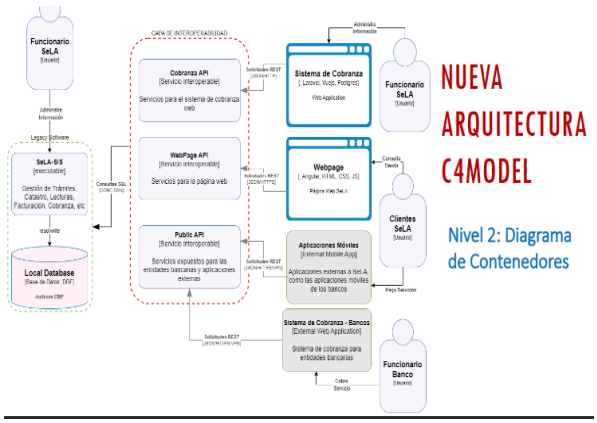


Figura 5: Diagrama de contenedores

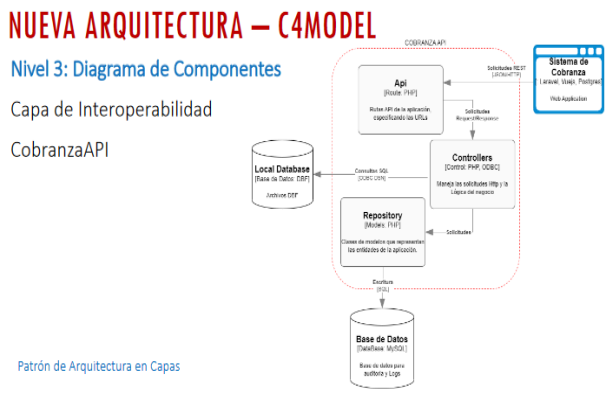


Figura 6: Diagrama de Componentes

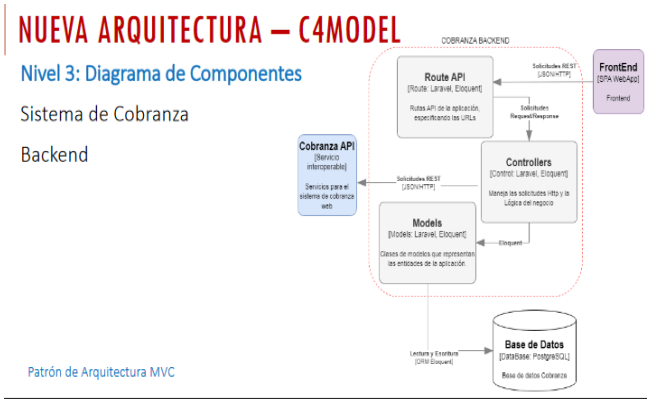


Figura 7: Diagrama de Componentes

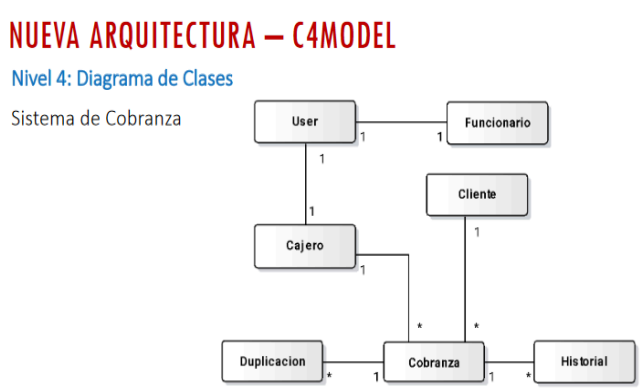


Figura 8: Diagrama de clases

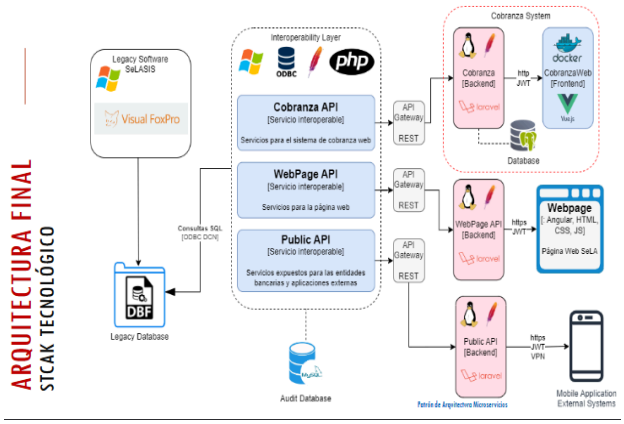


Figura 9: Arquitectura Final

Figura 10: Implementación nuevo sistema de cobranza

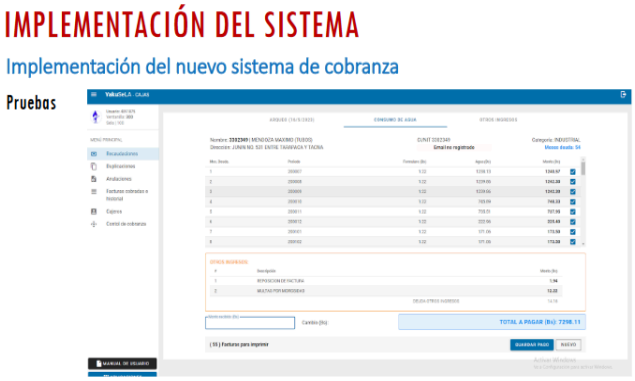


Figura 11: Implementación del sistema móvil

Figura 12: El sistema en actual funcionamiento

3. CONCLUSIONES

La arquitectura de software te resuelve problemas a un alto nivel

SOBRE EL AUTOR

Ingeniero Informático, cuenta con una maestría en Ingeniería de Software, cuenta con los siguientes diplomados: Diplomado en Diseño y Arquitectura de Software, Diplomado Experto en Desarrollo de Aplicaciones Empresariales, Diplomado en Software Libre, Diplomado en Ciencias de la Educación Superior, Diplomado en Desarrollo de Aplicaciones con Software Libre, Diplomado en Software Libre y GNU/Linux.

También cuenta con las siguientes certificaciones internacionales: Scrum Foundation Professional Certificate (SFPC), Certified ScrumMaster (CSM), Certificación Internacional de Experto en LINUX.

Dentro su experiencia profesional fué: Software Architect y Desarrollador de Software en ORBIS COMPLIANCE LLC – USA, Software Engineer y Desarrollador de Software en ASSURESOFT BOLIVIA S.RL., Desarrollador de Software en COLQUECHACA MINING LTDA. Desarrollador de Software en Servicio Local de Acueductos y Alcantarillado de Oruro, entre otros, también es Docente universitario.

Figura 22: Fotografía de presentación de la ponencia