Proyecto de Investigación

TÉCNICAS PARA EL MANTENIMIENTO

Reingeniería e Ingeniería Inversa

Realizado por:

Eric Josue Hernandez Duran Luis Alejandro Sanchez Brizio Diego Guillen Aguallo Ethan Gael Carranco Alfaro

REINGENIERIA

La reingeniería es el examen y alteración de un sistema para reconstruirlo en una nueva forma, generalmente para mejorar su calidad, mantenibilidad o rendimiento.



DEFINICIÓN Y ALCANCE

CALIDAD INTERNA

Estructura, legibilidad y mantenibilidad del código

ESCALABILIDAD

Capacidad para manejar mayor volumen de datos o usuarios

PORTABILIDAD

Adaptación a nuevas plataformas o entornos

INTEROPERABILIDAD

Integración con otros sistemas modernos

RENDIMIENTO

Velocidad y eficiencia de procesamiento



MOTIVACIONES PARA LA REINGENIERÍA

OBSOLESCENCIA TECNOLÓGICA

Cuando las plataformas o lenguajes se vuelven obsoletos y no soportados

EVOLUCIÓN DE REQUISITOS

Cambios fundamentales en las necesidades del negocio

COSTOS DE MANTENIMIENTO CRECIENTES

Cuando el mantenimiento correctivo y adaptativo consume recursos excesivos

PÉRDIDA DE CONOCIMIENTO

Cuando los desarrolladores originales ya no están disponibles



PROBLEMAS DE CALIDAD ACUMULADOS

Deuda técnica severa que impide la evolución del sistema

NECESIDADES DE MODERNIZACIÓN

Adaptación a nuevos paradigmas como cloud computing, microservicios o dispositivos móviles

PROCESO DE REINGENIERÍA

 Análisis de Inventario y Evaluación

Catalogación y evaluación de todos los componentes del sistema para identificar su valor, calidad y problemas actuales.

 Reestructuración de Código

3.

Mejora de la estructura interna del código mediante refactorización para aumentar su mantenibilidad.

 Reestructuración de **Documentación**

2.

Actualización y creación de documentación técnica que refleje con precisión el sistema existente.

 Reestructuración de Datos

Optimización de esquemas y migración de datos para mejorar integridad, rendimiento y escalabilidad.

 Modularización y **Desacoplamiento**

División del sistema en componentes más pequeños con interfaces bien definidas para reducir acoplamiento.

5.

ANÁLISIS DE INVENTARIO Y EVALUACIÓN

Catalogación
 Exhaustiva de Activos

El primer paso implica un mapeo detallado de todos los componentes del sistema:

- Módulos de software
- Bibliotecas y Frameworks
- Bases de datos y Almacenes de datos
- Interfaces
- Documentación
- Procesos Operativos

 Análisis de Valor y Riesgo

Para cada componente identificado, se evalúa:

- Valor para el negocio
- Complejidad Técnica
- Estabilidad
- Adaptabilidad
- Dependencias

 Métricas de Calidad Detalladas

Se emplean herramientas de análisis estático y dinámico para evaluar:

- Complejidad Ciclomática
- Acoplamiento
- Cohesión
- Duplicación de Código
- Cobertura de Pruebas
- Densidad de Defectos
- Deuda Técnica Cuantificada

REESTRUCTURACIÓN DE DOCUMENTACIÓN

Arqueología de Software

Proceso sistemático para recuperar conocimiento perdido sobre el sistema:

- Análisis de código
- Entrevistas con Stakeholders
- Análisis de logs y registros
- Revisión de documentación histórica



Creación de documentación actualizada y completa:

- Diagramas Arquitectónicos
- Mapas de procesos de negocio
- Catálogos de APIs
- Diccionarios de datos
- Documentación Operativa

 Herramientas de Documentación Automatizadas

Implementación de soluciones para mantener la documentación sincronizada:

- Generadores de documentación de código
- Herramientas de documentación de API
- Sistemas de trazabilidad
- Repositorios de Conocimiento









REESTRUCTURACIÓN DE CÓDIGO

Análisis Profundo del Código Fuente

Antes de modificar el código, es crucial comprenderlo a fondo:

- Análisis de flujo de control
- Análisis de flujo de datos
- Detección de anti-patrones
- Identificación de componentes críticos

• Estrategias de Refactorización Sistemática

Implementación de técnicas específicas para mejorar la estructura del código:

- Mejora de nombrado
- Extracción de métodos
- Eliminación de duplicación
- Simplificación de condicionales
- Introducción de patrones de diseño
- Encapsulación Mejorada
- Eliminación de código muerto

Modernización de Arquitectura

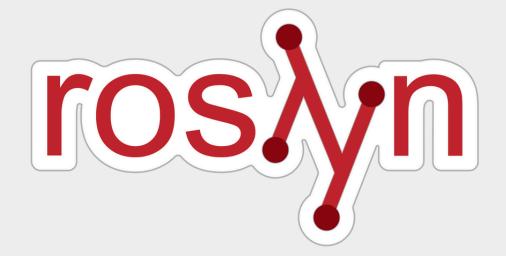
Transformación de la estructura general del sistema:

- Migración entre paradigmas
- Aplicación de arquitecturas en capas
- Implementación de inversión de dependencias
- Introducción de inyección de dependencias
- Migración a patrones arquitectónicos modernos









REESTRUCTURACIÓN DE DATOS

Análisis del Modelo de Datos Existente

Evaluación detallada de las estructuras de datos actuales:

- Análisis de esquemas
- Evaluación de anomalías
- Medición de la calidad de datos
- Análisis de rendimiento

Normalización y Optimización

Mejora de la estructura de datos para mayor eficiencia:

- Normalización de bases de datos
- Desnormalización estratégica
- Optimización de Índices
- Particionamiento
- Implementación de Caché

Migración de Datos

Transferencia de datos a nuevas estructuras o tecnologías:

- Mapeo de Datos
- Procesos ETL
- Estrategias de limpieza de datos
- Validación y Verificación
- Migraciones Incrementales







MODULARIZACIÓN Y DESACOPLAMIENTO

Descomposición Arquitectónica

Fragmentación estratégica de sistemas monolíticos:

- Análisis de acoplamiento
- Identificación de límites de contexto
- Extracción de bibliotecas compartidas
- Implementación de interfaces bien definidas
- Desarrollo de gateways y adaptadores



Estrategias de Modularización

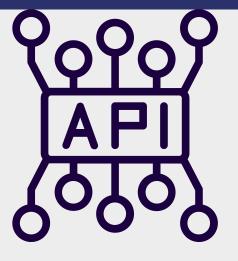
Enfoques para dividir sistemas complejos:

- Modularización por capas horizontales
- Modularización por dominios verticales
- Microkernels
- Descomposición por volumen de cambios

Transición a Microservicios

Procesos para fragmentar monolitos en servicios independientes:

- Identificación de candidatos a microservicios
- Estrategias de extracción
- Diseño de comunicación entre servicios
- Implementación de API Gateways
- Gestión de datos distribuidos
- Orquestación y coreografía



ENFOQUES METODOLÓGICOS PARA LA REINGENIERÍA

01

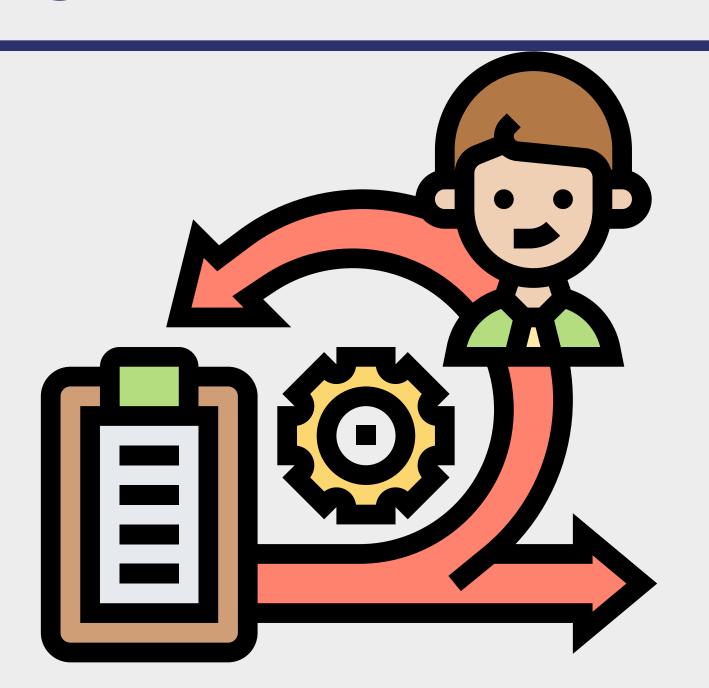
Reingeniería Incremental

02

Reingeniería Big Bang

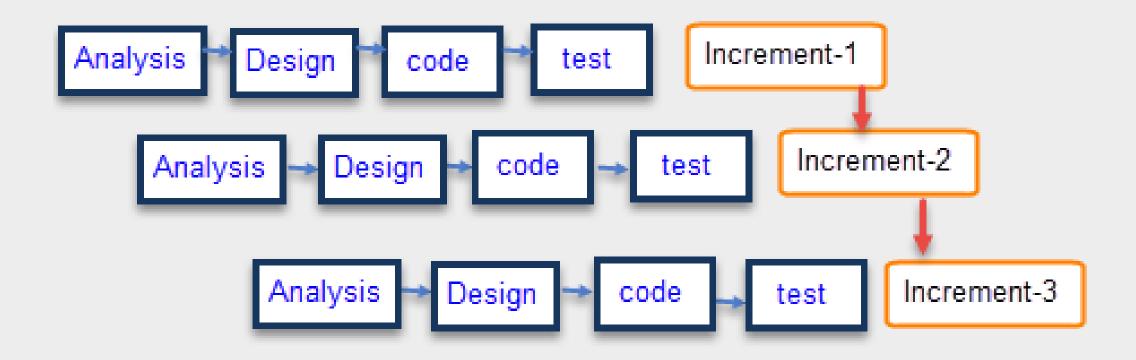


Reingeniería Dirigida por Modelos (MDA)



REINGENIERÍA INCREMENTAL

Enfoque que divide el proceso en etapas manejables, transformando el sistema componente por componente mientras sigue en operación. Ofrece menor riesgo operacional y beneficios visibles más rápido, pero implica mayor complejidad durante la transición y posible inconsistencia arquitectónica.



Incremental Model



•

REINGENIERÍA INCREMENTAL

Ventajas:

- Menor riesgo operacional
- Beneficios visibles antes de completar el proyecto
- Oportunidad de ajustar el enfoque basado en resultados tempranos
- Menor resistencia organizacional

Desafíos:

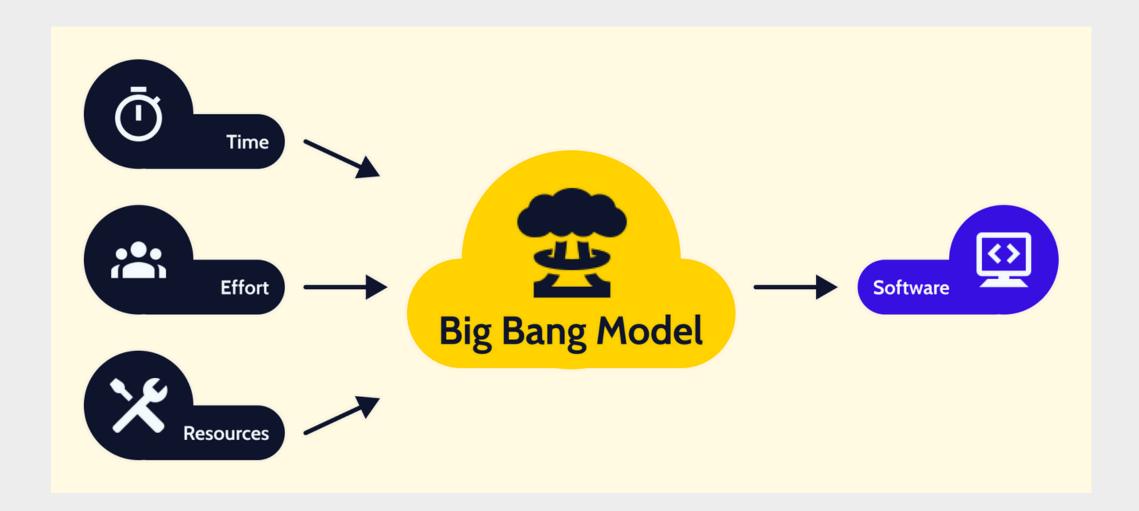
- Mayor complejidad durante la transición
- Posible inconsistencia arquitectónica
- Período extendido de mantenimiento dual





REINGENIERÍA BIG BANG

Transformación completa del sistema en una única operación, desarrollando el nuevo sistema en paralelo mientras el antiguo sigue funcionando. Proporciona diseño más coherente y evita compromisos de compatibilidad, pero conlleva mayor riesgo de fallos durante la transición y un período sin nuevas características.





REINGENIERÍA BIG BANG

Ventajas:

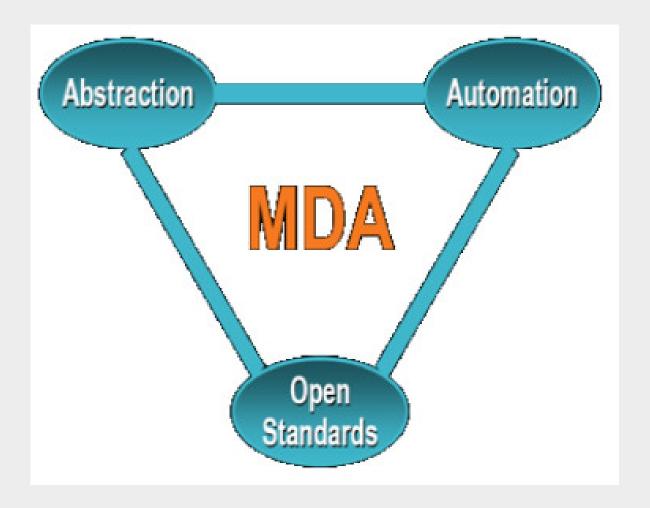
- Diseño arquitectónico más coherente
- Evita compromisos de compatibilidad con código antiguo
- Menor duración total del proyecto
- Oportunidad para cambiosfundamentales

Desafíos:

- Mayor riesgo de fallos catastróficos durante la transición
- Período prolongado sin nuevas características
- Dificultad para demostrar valor antes de la finalización
- Resistencia organizacional significativa

REINGENIERÍA DIRIGIDA POR MODELOS (MDA)

Enfoque basado en abstracciones de alto nivel, donde se extraen modelos del sistema existente, se transforman y luego se genera código automáticamente. Ofrece mayor nivel de abstracción y mejor trazabilidad, aunque enfrenta desafíos en la complejidad de herramientas y la representación completa del sistema.





REINGENIERÍA DIRIGIDA POR MODELOS (MDA)

Ventajas:

- Mayor nivel de abstracción
- Posibilidad de generar múltiples implementaciones
- Mejor trazabilidad entre requisitos e implementación
- Documentación inherente en los modelos

Desafíos:

- Complejidad de las herramientas de modelado
- Dificultad para representar todos los aspectos del sistema
- Posible pérdida de optimizaciones específicas
- Curva de aprendizaje pronunciada





ASPECTOS CRÍTICOS PARA EL ÉXITO DE LA REINGENIERÍA

Gestión de Pruebas y Verificación

Estrategias para asegurar la preservación de la funcionalidad:

- Establecimiento de línea base funcional
- Desarrollo de suites de pruebas de regresión
- Pruebas comparativas
- Pruebas de no regresión
- Testing A/B
- Monitoreo de producción



Gestión del Cambio Organizacional

Consideraciones humanas y de proceso:

- Comunicación transparente
- Capacitación del personal
- Participación de stakeholders
- Gestión de expectativas
- Planificación de la transición operativa
- Documentación de procedimientos

Consideraciones Económicas y ROI

Análisis financiero del proceso:

- Cálculo de costo total
- Valoración de beneficios tangibles
- Evaluación de beneficios intangibles
- Análisis de punto de equilibrio
- Medición continua
- Gestión de contingencias





PLATAFORMAS INTEGRADAS DE REINGENIERÍA

HERRAMIENTAS DE ANÁLISIS

AVANZADO

FRAMEWORKS DE MIGRACIÓN

• IBM Rational Software Architect

Rational software

CAST Application Intelligence Platform



• Blu Age



• Structure101



Lattix Architect



NDepende/JArchitect



JArchitect

Spring Boot Migration



AWS App2Container



Azure Migrate



INCENIERIA INVERSA

La ingeniería inversa es un proceso en el que se analiza un objeto, sistema o software ya terminado para entender su diseño, funcionamiento o estructura interna.

Es el poder de tomar un producto terminado...

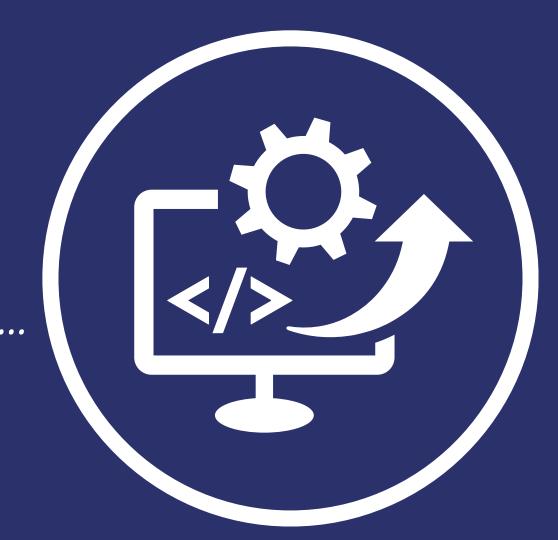
Desarmarlo pieza por pieza...

Y descubrir sus secretos más profundos.

"¿Cómo funciona?"

"¿Cómo lo hicieron?"

"¿Puedo mejorarlo... o destruirlo?"



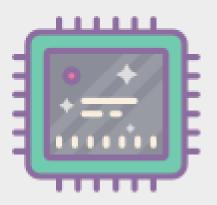
APLICACIONES COMUNES

Software

Analizar malware, encontrar vulnerabilidades, hacer crack de programas, compatibilidad.

Hardware

Réplicas de circuitos, piezas de autos, drones, tecnología militar.



Industria

Copiar productos
de la competencia,
reparar
maquinaria sin
planos originales.



Seguridad informatica

Estudiar exploits,
proteger
sistemas,
ingeniería inversa
de apps móviles
o videojuegos.

Biotecnologia

Clonar moléculas, estudiar medicamentos o vacunas.









¿Que nesesitas?

DESCOMPILADORES

Como Ghidra o IDA Pro





SNIFFERS DE RED

Tipo Wireshark



DEPURADORES

OllyDbg, x64dbg





ESCÁNERES 3D

Para copiar piezas...; al milímetro!



Aspectos legales



¿llegal? DEPENDE.

Cumplimiento Legal: Asegurarse de que el proceso de ingeniería inversa no infrinja patentes ni derechos de autor es esencial para evitar problemas legales y mantener la ética profesional.

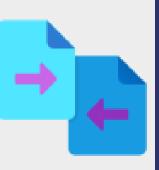


VENTAJAS Y DESVENTAJAS GENERALES

- Ventajas
- Comprender sistemas complejos
- Mejorar o reparar tecnología
- Detectar fallos y vulnerabilidades
- Recuperar información perdida



- Implicaciones legales y éticas
- Costos altos en equipo y tiempo
- Técnicamente complicado
- Puede dañar el producto original



¿Qué tiene que ver la ingeniería inversa con el mantenimiento de software?

Imagina que heredas un sistema viejo...

- ⚠ No hay código fuente
- **⚠** No hay manuales
- **L** El programador original ya ni trabaja ahí...
- Ahí entra la Ingeniería Inversa

 Descompilar, analizar y entender cómo funciona
 el sistema... para poder mantenerlo o mejorarlo.





• • •

"La ingeniería inversa no es solo romper y copiar... es entender, dominar y reescribir las reglas del juego..."

•

MUCHAS GRACIAS

Actividad