

# Modelo de Desarrollo de Software para la Empresa X

(Documento en Formato APA 7)

## Título del trabajo

Modelo Integral de Desarrollo de Software para la Empresa X

## Autor

Nombre de los integrantes

## Docente

Nombre del docente

## Institución

KHIPU

## Fecha

2025

## Resumen

El presente documento describe un modelo integral para el desarrollo de software en la Empresa X, diseñado conforme a estándares internacionales y metodologías ampliamente utilizadas en la industria. El modelo propone un proceso sistemático compuesto por fases claramente definidas: planificación, análisis, diseño, desarrollo, pruebas, implementación, mantenimiento y mejora continua. Su objetivo es garantizar la eficiencia, calidad, seguridad y trazabilidad durante todo el ciclo de vida del software.

**Palabras clave:** desarrollo de software, ciclo de vida, ingeniería de software, modelo empresarial.

## Introducción

El desarrollo de software requiere procesos estandarizados que permitan garantizar la calidad del producto final. La Empresa X necesita un modelo formal que unifique criterios, optimice tiempos, reduzca riesgos y facilite la colaboración entre equipos técnicos y áreas de negocio. El modelo aquí propuesto se basa en proyectos reales ejecutados bajo metodologías como ISO/IEC 12207, Scrum y DevOps, combinando un enfoque híbrido que integra estructura y flexibilidad.

## Modelo Integral de Desarrollo de Software

### 1. Planificación del Proyecto

La fase de planificación tiene como propósito definir el alcance, los recursos y la estrategia general del proyecto.

### **1.1 Definición del alcance**

- Identificación del problema o necesidad.
- Objetivos específicos y generales.
- Límites del sistema.

### **1.2 Viabilidad**

- Viabilidad técnica.
- Viabilidad financiera.
- Viabilidad operacional.

### **1.3 Plan de gestión**

- Cronograma
- Presupuesto preliminar.
- Gestión de riesgos.
- Roles y responsabilidades.

## **2. Análisis de Requisitos**

El análisis formaliza lo que el sistema debe hacer.

### **2.1 Requisitos funcionales**

- Casos de uso.
- Historias de usuario.
- Diagrama de flujo.

### **2.2 Requisitos no funcionales**

- Seguridad.
- Rendimiento.
- Usabilidad.
- Escalabilidad.

### **2.3 Especificación de requisitos**

- Documento SRS (Software Requirements Specification).
- Aprobación del cliente.

## **3. Diseño del Sistema**

En esta fase se define cómo funcionará el software a nivel lógico y técnico.

### **3.1 Arquitectura**

- Arquitectura de referencia (monolítica, microservicios, etc.).
- Diagramas de arquitectura.

### **3.2 Diseño de base de datos**

- Modelo entidad–relación.
- Diccionario de datos.

### **3.3 Diseño de interfaz**

- Prototipos UI/UX.
- Diagramas de navegación.

### **3.4 Diseño técnico detallado**

- Diagramas UML.
- API contracts.
- Gestión de integraciones.

## **4. Desarrollo / Implementación**

Corresponde a la construcción del software.

### **4.1 Programación**

- Estándares de codificación.
- Desarrollo por módulos.
- Integración continua.

### **4.2 Revisiones**

- Code reviews.
- Pair programming.

## **5. Pruebas (Testing)**

Garantiza la calidad y confiabilidad del software.

### **5.1 Tipos de pruebas**

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas funcionales.
- Pruebas de aceptación (UAT).

## **5.2 Documentación**

- Informe de pruebas.
- Registro de defectos.
- Informe final de calidad.

## **6. Conclusiones**

El modelo de desarrollo de software propuesto para la Empresa X ofrece una guía clara, estructurada y completa para la ejecución de proyectos tecnológicos. Al integrar prácticas de metodologías modernas como Scrum y DevOps con marcos tradicionales como ISO/IEC 12207, el modelo asegura un enfoque holístico que incrementa la calidad del producto, reduce riesgos y mejora la eficiencia operativa.

## **Referencias**

(Ejemplo en formato APA, puedes sustituirlas por las reales que utilices)

Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9.<sup>a</sup> ed.). McGraw-Hill.

Sommerville, I. (2016). *Software Engineering* (10.<sup>a</sup> ed.). Pearson.

ISO/IEC/IEEE. (2017). *ISO/IEC/IEEE 12207: Systems and software engineering — Software life cycle processes*.

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org.