

Caso práctico integrador: Sistema de Gestión Académica (SIGA)

Contexto general

Una institución de educación superior requiere desarrollar un **Sistema de Gestión Académica (SIGA)** que apoye los procesos de inscripción, control escolar y seguimiento académico. Actualmente, estos procesos se realizan de forma manual y con sistemas aislados, lo que genera errores, retrasos y falta de información confiable para la toma de decisiones.

El proyecto será desarrollado por un equipo de Ingeniería de Software, aplicando **procesos, metodologías, pruebas y modelos de calidad**, siguiendo buenas prácticas de la disciplina.

Objetivo del caso

Que el alumno **integre los conceptos de Ingeniería de Software** a través del análisis, diseño, planeación, pruebas y evaluación de la calidad de un sistema software.

Actividades por unidad

UNIDAD 1. Fundamentos y gestión del proyecto

Actividad 1.1 – Fundamentos Identifique por qué el desarrollo del SIGA requiere un enfoque de Ingeniería de Software y no solo programación.

Actividad 1.2 – Plan de proyecto Elabore un esquema de plan de proyecto que incluya:

- Objetivo del sistema
- Alcance y exclusiones
- Principales entregables

Actividad 1.3 – Ámbito y factibilidad Determine:

- Ámbito del software (qué incluye y qué no incluye)
- Factibilidad técnica, económica y operativa

Actividad 1.4 – Riesgos Identifique al menos **3 riesgos** del proyecto y una acción de mitigación para cada uno.

Actividad 1.5 – Métricas y planificación

- Estimación

Se estima un tamaño aproximado de desarrollo de 53,000 línea de código fuente, el cliente decide que ya están establecidos algunos requerimientos sin embargo es muy probable que pudieran agregarse y/o cambiar algunos procesos e interfaces y se espera que el sistema tenga un nivel de reusabilidad alto. El equipo nunca ha desarrollado un proyecto relacionado con la educación y desafortunadamente por falta de especificaciones y una pésima coordinación de los Stakeholders no se ha desarrollado un análisis de riesgo adecuado. El equipo de desarrollo lleva 5 años trabajando de manera aceptable, aunque el mes pasado dos de los mejores analistas renunciaron por causas personales; aun así, el equipo cuenta con un nivel de madurez CMM 2, un nivel de experiencia nominal en el desarrollo de software, además, la capacidad del personal es media y los espacios físicos, hardware y software proporcionados por Instituto tienen un nivel muy alto. El cliente nos pide que se use en todos los sentidos una plataforma de alta calidad y se nos solicita que la confiabilidad del producto sea muy alta. Por cuestiones de cambio de gobierno la presión de la agenda es muy alta y nos piden mantener una estrecha relación Stakeholders – Desarrolladores, así como también planear de manera asidua y coordinada entregas para su debida validación.

Considere un salario de \$16,000 y otros gastos de \$100,000.

Mediante Cocomo calcule el factor de escala, los multiplicadores de esfuerzo, el esfuerzo, el tiempo de desarrollo y el costo

DATOS PARA REALIZAR LAS ESTIMACIONES CON EL MODELO COCOMO MEDIANTE MULTIPLICADORES DE ESFUERZO

Fórmulas Generales

Esfuerzo

$$PM_{NS} = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$\text{where } E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

Tiempo Calendario

$$TDEV_{NS} = C \times (PM_{NS})^F$$

$$\begin{aligned} \text{where } F &= D + 0.2 \times 0.01 \times \sum_{j=1}^5 SF_j \\ &= D + 0.2 \times (E - B) \end{aligned}$$

Constantes de Calibración del modelo

$$A = 2.94$$

$$B = 0.91$$

$$C = 3.67$$

$$D = 0.28$$

PM = Esfuerzo en Persona-Mes (152 horas)
Size = Tamaño en Miles de LOCs (KLOCs)
o Puntos Función NO AJUSTADOS
EM = Multiplicadores de esfuerzo
SF = Factores de Escala Exponencial
E = Incremento de esfuerzo para proyectos grandes

Multiplicadores de Esfuerzo Diseño Inicial

Code	Effort modifier	Extra low	Very low	Low	Nominal	High	Very high	Extra high
RCPX	Product reliability and complexity	0.49	0.60	0.83	1.00	1.33	1.91	2.72
RUSE	Required reusability			0.95	1.00	1.07	1.15	1.24
PDIF	Platform difficulty			0.87	1.00	1.29	1.81	2.61
PERS	Personnel capability	2.12	1.62	1.26	1.00	0.83	0.63	0.50
PREX	Personnel experience	1.59	1.33	1.12	1.00	0.87	0.74	0.62
FCIL	Facilities available	1.43	1.30	1.10	1.00	0.87	0.73	0.62
SCED	Schedule pressure		1.43	1.14	1.00	1.00	1.00	

TABLE 5.6 COCOMO II Early design effort multipliers

PMAT Rating	Maturity Level	EPML
Very Low	CMM Level 1 (lower half)	0
Low	CMM Level 1 (upper half)	1
Nominal	CMM Level 2	2
High	CMM Level 3	3
Very High	CMM Level 4	4
Extra High	CMM Level 5	5

- Seleccione una métrica de estimación (LOC o puntos de función)
- Proponga un cronograma general (diagrama de Gantt simplificado)

Actividad 1.6 – Metodología Justifique el uso de **SCRUM con apoyo de Kanban** para el desarrollo del SIGA.

UNIDAD 2. Requerimientos, diseño y modelado

Actividad 2.1 – Ingeniería de requerimientos Defina:

- 5 requerimientos funcionales
- 3 requerimientos no funcionales

Debidamente requisitados en Matriz de requerimientos, en el cual se pueda dar seguimiento puntual a los requerimientos, estatus y nomenclatura.

Actividad 2.2 – Diseño del sistema Proponga una **arquitectura lógica en capas** (presentación, lógica y datos), describiendo brevemente cada una.

Actividad 2.3 – UI y UX

- Describa los principales elementos de la interfaz de usuario.
- Justifique decisiones de UX enfocadas en estudiantes y personal administrativo.

Actividad 2.4 – Modelado Elabore los siguientes diagramas UML:

- Diagrama de casos de uso
- Diagrama de clases
- Diagrama de secuencia (proceso de inscripción)

UNIDAD 3. Pruebas de software

Actividad 3.1 – Tipos de pruebas Clasifique las siguientes pruebas para el SIGA:

- Validación del registro de alumnos
- Medición de tiempo de respuesta en inscripciones
- Prueba de acceso con credenciales incorrectas

Actividad 3.2 – Casos de prueba Diseñe 2 casos de prueba funcionales y 1 no funcional, incluyendo resultado esperado.

Actividad 3.3 – Herramientas Seleccione una herramienta de pruebas funcionales y una de rendimiento, justificando su elección.

Actividad 3.4 – Evaluación Explique cómo evaluaría:

- Cobertura de pruebas
- Severidad de defectos

UNIDAD 4. Calidad del software

Actividad 4.1 – Calidad del proceso Explique cómo se aseguraría la calidad del proceso durante el desarrollo del SIGA.

Actividad 4.2 – Calidad del producto Relacione el SIGA con al menos 4 atributos de calidad del producto software.

Actividad 4.3 – Normas y modelos Asocie:

- ISO/IEC 25010 con calidad del producto
- IEEE con documentación
- CMMI o MoProSoft con mejora del proceso

Actividad 4.4 – Madurez Determine qué modelo de madurez (PSP, TSP, CMMI o MoProSoft) sería más adecuado para el equipo y justifique.

Realice una auditoría al proyecto , tomando como base la PA SQA de CMMI.

Producto integrador esperado

El alumno deberá entregar el día que se presente a su evaluación de ETS la siguiente documentación al correo electrónico **msalinasc@ipn.com** , se recomienda compartir link de la carpeta, asegurándose de dar los permisos “Cualquier persona que tenga el link puede acceder”, esto es con la finalidad de poder visualizar la información. En caso de no tener permisos, no se dará oportunidad de poder entregar y se enviará al correo institucional captura de pantalla, indicando que no se cuenta con los permisos o que no cuenta con información completa.

Se enumerará la información de la siguiente manera:

1. Documento de plan de proyecto
2. Especificación de requerimientos (Matriz de requerimientos)
3. Estimación del sistema
4. Diagramas UML (casos de uso y secuencia)
5. Mapa de navegación
6. Diagrama de arquitectura del SW
7. Maquetado
8. Plan de pruebas
9. Casos de prueba
10. Análisis de calidad del proceso y del producto