

# 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**

- 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 en dispositivo USB (Nombrando los archivos como se indica , así como la numeración y en formato PDF):

- Documento de plan de proyecto
- Especificación de requerimientos (Matriz de requerimientos)
- Estimación del sistema
- Diagramas UML (casos de uso y secuencia)
- Mapa de navegación

- Diagrama de arquitectura del SW
- Maquetado
- Plan de pruebas
- Casos de prueba
- Análisis de calidad del proceso y del producto