

Documento de Especificación de Requisitos del Software (SRS)

Proyecto: Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA-SUMO)

Versión: 1.0.1 (Borrador)

Estado: En elaboración

CAPÍTULO 1 INTRODUCCION

1.1 Introducción

1.1.1 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales y no funcionales para el desarrollo del **Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA-SUMO)**, destinado al Instituto de Preparación Preuniversitaria.

El documento servirá como guía durante el proceso de análisis, diseño, desarrollo, implementación y mantenimiento del sistema, permitiendo establecer de manera clara los requerimientos solicitados por la institución y asegurando que el producto final satisfaga las necesidades operativas y administrativas del instituto.

Asimismo, este documento constituye la referencia principal entre el equipo de desarrollo y la dirección del instituto para validar el alcance del proyecto y controlar su avance.

1.1.2 Alcance

El Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO) permitirá automatizar los procesos académicos, administrativos y financieros del instituto, centralizando la información de las diferentes sucursales en una única plataforma web.

El sistema estará orientado a optimizar el proceso de inscripción de estudiantes, la administración de cursos, el control financiero institucional, la gestión académica y el seguimiento de docentes y estudiantes.

La primera versión del sistema contemplará los siguientes módulos:

- Gestión de usuarios y seguridad.
- Gestión de estudiantes.
- Gestión de docentes.
- Gestión de cursos y áreas académicas.
- Gestión de inscripciones.
- Gestión de pagos.
- Impresión de recibos mediante impresora térmica.

- Control de ingreso con maquina QR.
- Integración con pagos mediante código QR (según disponibilidad de la pasarela de pago).
- Gestión financiera de ingresos y egresos.
- Portal del docente.
- Portal del estudiante.
- Gestión de materiales académicos.
- Gestión de tareas y prácticas.
- Registro y consulta de calificaciones.
- Control de asistencia.
- Reportes administrativos y financieros.
- Dashboard con indicadores de gestión.

1.1.3 Definiciones

Esta sección tiene como finalidad establecer un lenguaje común entre los usuarios, la dirección del instituto y el equipo de desarrollo, definiendo los términos técnicos y administrativos que serán utilizados durante el desarrollo e implementación del Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO).

El objetivo es evitar ambigüedades en la interpretación de los requerimientos y facilitar la comprensión del documento por parte de todos los involucrados en el proyecto.

Término	Definición
SIGA-SUMO	Sistema Integral de Gestión Académica y Administrativa de SUMO
Estudiante	Persona inscrita en uno o varios cursos del instituto
Docente	Profesional encargado de impartir una o varias asignaturas
Área	Facultad o programa de preparación (Medicina, Ingeniería, Derecho, etc.)
Curso	Grupo académico asociado a un área
Inscripción	Registro oficial del estudiante en un curso
Recibo	Documento que acredita un pago realizado
Control QR	Identificación del estudiante control de ingreso
Pago QR	Pago electrónico realizado mediante código QR

Portal del Estudiante	Plataforma donde el estudiante consulta su información académica
Portal del Docente	Plataforma donde el docente administra sus cursos y estudiantes
CRUD	Create, Read, Update and Delete. Operaciones básicas de gestión de información (Crear, Consultar, Modificar y Eliminar).
UI	User Interface. Interfaz gráfica utilizada por los usuarios del sistema.
UX	User Experience. Experiencia del usuario durante la utilización del sistema.
MVC	Model-View-Controller. Patrón arquitectónico utilizado para organizar el desarrollo del software.
API	Application Programming Interface. Conjunto de servicios que permiten la comunicación entre aplicaciones externas y el sistema.

1.2 Referencias

El desarrollo del presente documento y del Sistema Integral de Gestión Académica y Administrativa (SIGA) se basa en estándares, metodologías y buenas prácticas internacionalmente reconocidas para el desarrollo de software, la gestión de proyectos y el aseguramiento de la calidad. Estas referencias proporcionan una estructura metodológica que permitirá desarrollar un sistema robusto, escalable y alineado con las necesidades de la institución.

IEEE 29148:2018 – Systems and Software Engineering – Life Cycle Processes – Requirements Engineering

El estándar IEEE 29148:2018 establece las mejores prácticas para la ingeniería de requisitos en proyectos de software y sistemas. Define la estructura, organización y contenido recomendado para la elaboración de un Documento de Especificación de Requisitos del Software (Software Requirements Specification - SRS).

En el presente proyecto, este estándar servirá como guía para:

- Definir claramente los requisitos funcionales y no funcionales.
- Documentar las reglas del negocio.
- Identificar los actores del sistema.

- Establecer los casos de uso.
- Delimitar el alcance del proyecto.
- Facilitar la comunicación entre los desarrolladores y los responsables

del instituto.

ISO/IEC 25010:2011 – Systems and Software Quality Requirements and Evaluation (SQuaRE)

La norma ISO/IEC 25010 define el modelo internacional de calidad del software, estableciendo las características que debe cumplir un sistema para garantizar un funcionamiento adecuado.

Durante el desarrollo del SIGA se considerarán principalmente las siguientes características de calidad:

- **Adecuación funcional:** El sistema deberá satisfacer los procesos administrativos, académicos y financieros del instituto.
- **Fiabilidad:** El sistema deberá operar correctamente minimizando errores y garantizando la disponibilidad de la información.
- **Usabilidad:** Las interfaces deberán ser intuitivas y fáciles de utilizar por administrativos, docentes y estudiantes.
- **Eficiencia del desempeño:** El sistema deberá responder de manera rápida incluso cuando múltiples usuarios accedan simultáneamente.
- **Seguridad:** Se implementarán mecanismos de autenticación, autorización y protección de la información.
- **Mantenibilidad:** La arquitectura permitirá realizar modificaciones y agregar nuevas funcionalidades con facilidad.
- **Portabilidad:** El sistema podrá ejecutarse en diferentes navegadores modernos y servidores compatibles con la tecnología seleccionada.

PMBOK (Project Management Body of Knowledge)

El PMBOK, desarrollado por el Project Management Institute (PMI), proporciona un conjunto de buenas prácticas para la dirección y gestión de proyectos.

Aunque el proyecto será desarrollado por un único desarrollador, se adoptarán principios del PMBOK para mejorar la organización y el control del proyecto, tales como:

- Planificación del alcance.
- Gestión del cronograma.
- Gestión de riesgos.
- Gestión de interesados.
- Seguimiento del avance.
- Control de cambios.
- Gestión de entregables.

Estas prácticas permitirán mantener una adecuada planificación y facilitarán la supervisión del proyecto por parte de la dirección del instituto.

Scrum Guide

Scrum es un marco de trabajo ágil utilizado para la gestión y desarrollo de proyectos de software mediante entregas incrementales.

Para el desarrollo del SIGA se adoptará una versión adaptada de Scrum, considerando las características particulares del proyecto y el tamaño del equipo de desarrollo.

La metodología permitirá:

- Dividir el proyecto en módulos funcionales.
- Planificar el trabajo semanal mediante pequeñas iteraciones.
- Presentar avances periódicos a la dirección del instituto.
- Incorporar cambios de manera controlada durante el desarrollo.
- Validar continuamente las funcionalidades implementadas.

Las reuniones de seguimiento se realizarán de forma periódica con el objetivo de revisar el progreso del proyecto, identificar posibles dificultades y definir las actividades de la siguiente etapa.

Otras referencias

Además de los estándares anteriormente mencionados, durante el desarrollo del sistema se considerarán las siguientes tecnologías y buenas prácticas:

- Arquitectura MVC (Model–View–Controller) para la organización del código fuente.
- Principios SOLID para el desarrollo orientado a objetos.
- Normalización de bases de datos hasta la Tercera Forma Normal (3FN), con el fin de reducir redundancias y garantizar la integridad de la información.
- Control de versiones mediante Git para gestionar el historial del código fuente y facilitar el mantenimiento del proyecto.
- Buenas prácticas de seguridad para aplicaciones web, incluyendo validación de datos, control de sesiones, gestión de permisos y protección contra vulnerabilidades comunes.

1.3 Visión General

El presente documento describe de manera estructurada los requerimientos necesarios para el desarrollo del **Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA -SUMO)**, estableciendo las bases para el análisis, diseño, implementación, pruebas y puesta en funcionamiento del sistema.

Su contenido ha sido organizado siguiendo una secuencia lógica que permitirá comprender el alcance del proyecto, las necesidades identificadas durante el levantamiento de

información y las funcionalidades que deberán implementarse para satisfacer los procesos administrativos, financieros y académicos del instituto.

El documento se encuentra dividido en los siguientes capítulos:

Capítulo 1. Introducción

Presenta el propósito del documento, el alcance del sistema, las definiciones utilizadas, los estándares de referencia y una visión general del contenido del proyecto.

Capítulo 2. Descripción General del Sistema

Describe el contexto institucional, la problemática actual, los objetivos del sistema, los procesos involucrados, los usuarios que interactuarán con la aplicación y las restricciones identificadas durante el levantamiento de requerimientos.

Capítulo 3. Requerimientos Funcionales

Define de manera detallada todas las funcionalidades que deberá proporcionar el sistema, organizadas por módulos, incluyendo las acciones que podrán realizar cada uno de los tipos de usuarios.

Capítulo 4. Requerimientos No Funcionales

Especifica las características de calidad que deberá cumplir el sistema, tales como seguridad, rendimiento, disponibilidad, usabilidad, mantenibilidad, compatibilidad, escalabilidad y confiabilidad.

Capítulo 5. Reglas del Negocio

Describe las políticas institucionales y restricciones operativas que regularán el comportamiento del sistema, garantizando que los procesos implementados respeten el funcionamiento establecido por el instituto.

Capítulo 6. Casos de Uso

Presenta la interacción entre los diferentes actores y el sistema mediante diagramas y descripciones detalladas de los procesos principales, permitiendo visualizar el comportamiento esperado de cada funcionalidad.

Capítulo 7. Modelado de Datos

Incluye el modelo entidad-relación, el diccionario de datos y la descripción de las estructuras de almacenamiento utilizadas para administrar la información del sistema.

Capítulo 8. Planificación del Proyecto

Describe la metodología de desarrollo, el cronograma de actividades, los entregables, los hitos del proyecto y los mecanismos de seguimiento que permitirán controlar el avance durante la implementación.

Capítulo 9. Anexos

Contendrá información complementaria, diagramas adicionales, prototipos de interfaces, formatos utilizados durante el análisis, documentación técnica y cualquier otro material de apoyo relacionado con el desarrollo del sistema.

1.4 Metodología de Desarrollo del Proyecto

El desarrollo del **Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA-SUMO)** se llevará a cabo mediante una metodología ágil basada en **Scrum**, adaptada a las características y necesidades del proyecto.

Debido a que el proyecto será desarrollado inicialmente por un único desarrollador y supervisado directamente por la dirección del instituto, la metodología Scrum será aplicada de manera simplificada, manteniendo sus principios fundamentales de planificación, desarrollo incremental, revisión continua y mejora constante.

Esta metodología permitirá desarrollar el sistema por módulos funcionales, entregando avances periódicos que podrán ser revisados y validados por la institución, facilitando la incorporación de mejoras y cambios conforme evolucionen las necesidades del proyecto.

El desarrollo se realizará siguiendo un enfoque iterativo e incremental, donde cada módulo será analizado, diseñado, implementado y probado antes de iniciar el siguiente, garantizando un crecimiento ordenado del sistema y reduciendo el riesgo de errores durante las etapas finales del proyecto.

1.4.1 Metodología Scrum Adaptada

El proyecto adoptará los principios fundamentales del marco de trabajo Scrum, realizando adaptaciones acordes al tamaño del equipo de desarrollo.

Los principales componentes de la metodología serán los siguientes:

1.4.2 Planificación del Sprint

Al inicio de cada ciclo de trabajo se definirán las actividades que serán desarrolladas durante la semana, estableciendo objetivos concretos, prioridades y entregables.

Cada planificación tendrá como propósito organizar las tareas del desarrollador y establecer expectativas claras sobre los avances que serán presentados a la dirección del instituto.

1.4.3 Desarrollo Incremental

El sistema será construido por módulos independientes, permitiendo que cada uno pueda ser evaluado antes de continuar con el siguiente.

Entre los módulos principales se encuentran:

- Gestión de Usuarios.
- Gestión de Estudiantes.
- Gestión de Docentes.
- Gestión de Cursos.
- Gestión de Inscripciones.
- Gestión Financiera.
- Portal del Docente.
- Portal del Estudiante.
- Reportes.
- Configuración General.

Esta estrategia facilitará detectar errores tempranamente y disminuir el impacto de posibles modificaciones durante el desarrollo.

1.4.4 Revisión de Avances

Al finalizar cada semana de trabajo se realizará una reunión de seguimiento con la dirección del instituto, donde se presentarán los avances obtenidos durante el periodo correspondiente.

Durante esta revisión se evaluará:

- Actividades completadas.
- Funcionalidades implementadas.
- Problemas encontrados.
- Requerimientos adicionales identificados.
- Actividades planificadas para la siguiente semana.

Estas reuniones permitirán mantener un control permanente sobre el desarrollo del proyecto y asegurar que el sistema evolucione conforme a las necesidades institucionales.

1.4.5 Mejora Continua

Después de cada revisión semanal se analizarán las observaciones realizadas por los usuarios responsables del proyecto.

Las sugerencias aprobadas serán incorporadas a la planificación de los siguientes ciclos de desarrollo, siempre que no afecten significativamente el alcance definido en el presente documento.

En caso de requerimientos completamente nuevos, estos serán registrados como solicitudes de cambio y evaluados para futuras versiones del sistema.

1.5 Organización del Proyecto

El proyecto será desarrollado siguiendo una planificación organizada en fases, permitiendo controlar adecuadamente el progreso y asegurar la calidad de cada entregable.

Las principales fases del proyecto son las siguientes:

Fase I – Análisis de Requerimientos

Durante esta etapa se realizará el levantamiento de información, identificación de procesos institucionales y documentación de los requerimientos funcionales y no funcionales.

Entregables:

- Documento SRS.
- Reglas del negocio.
- Casos de uso.
- Alcance del proyecto.

Fase II – Diseño

En esta fase se definirá la arquitectura del sistema, el diseño de la base de datos, la estructura de navegación y los prototipos de las interfaces gráficas.

Entregables:

- Modelo Entidad–Relación.
- Diccionario de Datos.
- Diagramas UML.
- Prototipos de Interfaces.

Fase III – Desarrollo

Comprende la implementación progresiva de cada módulo del sistema conforme a la planificación establecida.

Cada módulo será desarrollado de manera independiente y validado antes de continuar con el siguiente.

Fase IV – Pruebas

Se realizarán pruebas funcionales, pruebas de integración y validaciones con usuarios finales para verificar el correcto funcionamiento del sistema.

Las incidencias identificadas serán corregidas antes de la puesta en producción.

Fase V – Implementación

Una vez finalizadas las pruebas, el sistema será instalado en el servidor institucional y se realizará la migración de la información necesaria.

Asimismo, se capacitará al personal encargado de utilizar la aplicación.

Fase VI – Mantenimiento

Posteriormente a la implementación se brindará soporte técnico para corregir incidencias, optimizar procesos e incorporar mejoras autorizadas por la dirección del instituto.

1.6 Seguimiento y Control del Proyecto

El seguimiento del proyecto se realizará mediante reuniones periódicas con la dirección del instituto y la presentación de informes de avance.

Cada informe incluirá como mínimo:

- Actividades desarrolladas durante el periodo.
- Funcionalidades implementadas.
- Porcentaje de avance del proyecto.
- Riesgos identificados.
- Dificultades encontradas.
- Actividades planificadas para el siguiente periodo.

Este mecanismo permitirá mantener un control continuo sobre el estado del proyecto y facilitará la toma de decisiones durante el proceso de desarrollo.

1.7 Control de Versiones

Con el propósito de garantizar la integridad del código fuente y mantener un historial de los cambios realizados, el desarrollo del sistema utilizará un sistema de control de versiones basado en **Git**.

Cada modificación importante será registrada mediante confirmaciones (*commits*), permitiendo:

- Mantener un historial de cambios del proyecto.
- Recuperar versiones anteriores en caso de ser necesario.
- Documentar la evolución del sistema.
- Facilitar el mantenimiento y futuras ampliaciones.

La documentación técnica y funcional también será actualizada conforme avance el desarrollo, asegurando la correspondencia entre el sistema implementado y los requisitos definidos.

CAPÍTULO 2 DESCRIPCIÓN GENERAL DEL SISTEMA

2.1 Perspectiva del Producto

El **Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA – SUMO)** será una plataforma web diseñada para apoyar la gestión administrativa, académica y financiera del instituto de preparación preuniversitaria.

El sistema reemplazará gran parte de los procesos manuales y de los sistemas actualmente utilizados, permitiendo centralizar la información institucional en una única plataforma accesible desde cualquiera de las sucursales autorizadas.

SIGA permitirá integrar en un mismo entorno los procesos de inscripción de estudiantes, administración de cursos, control de pagos, gestión financiera, administración de docentes, seguimiento académico y generación de reportes administrativos.

La plataforma estará diseñada bajo una arquitectura modular, permitiendo incorporar nuevas funcionalidades conforme evolucionen las necesidades institucionales sin afectar el funcionamiento de los módulos existentes.

Además, el sistema permitirá operar simultáneamente en las distintas sucursales del instituto, garantizando que la información permanezca sincronizada y disponible en tiempo real.

2.2 Objetivos del Sistema

El objetivo principal del sistema consiste en automatizar los procesos administrativos y académicos del instituto, mejorando la eficiencia operativa y reduciendo el uso de registros manuales.

De manera específica, el sistema buscará:

- Centralizar la información de estudiantes, docentes, cursos e inscripciones.
- Automatizar el proceso de inscripción de nuevos estudiantes.
- Registrar pagos únicos y pagos realizados en cuotas.
- Emitir comprobantes mediante impresoras térmicas.
- Integrar pagos electrónicos mediante códigos QR.
- Administrar los cursos ofrecidos por el instituto.
- Facilitar la gestión académica realizada por los docentes.
- Permitir a los estudiantes acceder a material académico y consultar su información.
- Gestionar los ingresos y egresos financieros de la institución.
- Proporcionar indicadores estadísticos para apoyar la toma de decisiones.

2.3 Descripción de la Institución

El instituto constituye una institución dedicada a la preparación preuniversitaria de estudiantes que buscan ingresar a diferentes carreras universitarias.

Actualmente cuenta con aproximadamente:

Concepto	Cantidad aproximada
Estudiantes	500
Docentes	30 a 45
Áreas Académicas	10 a 15
Sucursales	3

Las sucursales actualmente identificadas son:

- Sucursal Ceja.
- Sucursal UPEA.
- Sucursal Illimani.

La institución ofrece diferentes programas de preparación organizados por áreas académicas, donde cada una de ellas cuenta con uno o varios cursos asignados y un equipo de docentes responsables de impartir las clases correspondientes.

El crecimiento constante del número de estudiantes y la necesidad de mejorar el control administrativo han generado la necesidad de implementar una plataforma tecnológica que permita optimizar los procesos institucionales.

2.4 Situación Actual

En la actualidad, el instituto desarrolla gran parte de sus actividades administrativas mediante procesos manuales y herramientas informáticas que no se encuentran completamente integradas.

El proceso de inscripción requiere registrar manualmente la información del estudiante, asignar cursos, registrar los pagos y emitir comprobantes físicos en las sucursales del Alto La Ceja y Complejo UPEA. Aunque existe un sistema previo para algunas tareas, este no cubre todas las necesidades actuales de la institución y presenta limitaciones para el control administrativo y financiero en la sucursal de la ciudad de la calle Illimani.

Asimismo, el seguimiento de los pagos, la generación de reportes y la administración de la información académica requieren una importante intervención manual, incrementando el tiempo necesario para realizar estas actividades y aumentando la posibilidad de errores.

En el ámbito académico, los docentes registran la asistencia y otras actividades utilizando planillas físicas, lo que dificulta el seguimiento del desempeño de los estudiantes y la consolidación de información.

La administración financiera también se realiza mediante procedimientos que requieren múltiples registros manuales para controlar ingresos, egresos y pagos al personal docente.

2.5 Problemática Identificada

Durante el levantamiento de información realizado con el personal administrativo y directivo del instituto se identificaron las siguientes problemáticas principales:

- Procesos de inscripción parcialmente manuales.
- Información distribuida en diferentes registros y sistemas.
- Limitado control sobre el estado de los pagos realizados por los estudiantes.
- Ausencia de integración entre el proceso de inscripción y la emisión automática de recibos.
- Dificultades para realizar conciliaciones financieras.
- Escasa disponibilidad de indicadores estadísticos para apoyar la toma de decisiones.
- Control manual de la asistencia de los estudiantes.
- Falta de una plataforma para compartir material académico y realizar evaluaciones.
- Inexistencia de un portal institucional para docentes y estudiantes.
- Generación manual de diversos reportes administrativos.

Estas situaciones generan retrasos operativos, duplicidad de información y una mayor carga administrativa para el personal del instituto.

2.6 Solución Propuesta

Como respuesta a las necesidades identificadas, se propone el desarrollo del **Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO)**, una plataforma web que integrará en un único sistema los procesos administrativos, financieros y académicos del instituto.

El sistema permitirá administrar de manera centralizada la información de estudiantes, docentes, cursos, pagos, inscripciones y recursos académicos, facilitando el acceso a la información desde cualquiera de las sucursales autorizadas.

Además, incorporará herramientas para el seguimiento financiero, la impresión de recibos mediante impresoras térmicas, la integración con pagos electrónicos mediante códigos QR y la generación automática de reportes.

En el ámbito académico, la plataforma ofrecerá espacios diferenciados para docentes y estudiantes, permitiendo gestionar la asistencia, compartir materiales, publicar prácticas, registrar calificaciones y realizar el seguimiento del progreso académico.

La implementación del sistema contribuirá a optimizar los tiempos de atención, reducir errores administrativos, fortalecer el control financiero y mejorar la calidad de los servicios ofrecidos por la institución.

2.7 Funciones Generales del Sistema

El Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO) estará conformado por un conjunto de módulos integrados que permitirán administrar los procesos académicos, administrativos y financieros del instituto desde una única plataforma web.

Cada módulo será desarrollado de manera independiente, permitiendo su mantenimiento, actualización y ampliación sin afectar el funcionamiento del resto del sistema.

Las principales funcionalidades del sistema se describen a continuación.

2.7.1 Módulo de Autenticación y Seguridad

Este módulo será el encargado de controlar el acceso al sistema mediante mecanismos de autenticación y autorización.

Permitirá garantizar que únicamente usuarios autorizados puedan acceder a la información institucional, restringiendo las operaciones según el rol asignado.

Las principales funcionalidades serán:

- Inicio de sesión mediante usuario y contraseña.
- Recuperación de contraseña.
- Cambio de contraseña.
- Administración de usuarios.
- Administración de roles.
- Administración de permisos.
- Control de sesiones activas.
- Registro de auditoría (bitácora).
- Bloqueo temporal por múltiples intentos fallidos.

2.7.2 Gestión de Estudiantes

Este módulo permitirá administrar toda la información relacionada con los estudiantes inscritos en el instituto.

Cada estudiante contará con un expediente único que almacenará su información personal, historial académico e historial financiero.

Las principales funciones serán:

- Registrar estudiantes.
- Actualizar información.
- Consultar información.

- Buscar estudiantes.
- Registrar tutores.
- Registrar datos de contacto.
- Registrar observaciones.
- Consultar historial de inscripciones.
- Consultar historial de pagos.
- Cambio de estado (Activo, Retirado, Suspendido, Egresado).

2.7.3 Gestión de Inscripciones

Este constituye uno de los módulos principales del sistema.

Permitirá registrar oficialmente la inscripción de un estudiante a uno o varios cursos ofertados por el instituto.

Durante el proceso de inscripción se registrará:

- Estudiante.
- Curso.
- Área académica.
- Docente asignado.
- Horario.
- Fecha de inicio.
- Fecha de finalización.
- Tipo de pago.
- Descuento (si corresponde).
- Observaciones.

Al concluir la inscripción, el sistema generará automáticamente el registro financiero correspondiente y permitirá emitir el comprobante de pago.

2.7.4 Gestión de Pagos

Este módulo administrará todas las transacciones económicas realizadas por los estudiantes.

Permitirá registrar pagos realizados tanto en efectivo como mediante código QR.

Entre sus principales funcionalidades se encuentran:

- Registro de pagos.
- Pagos parciales.
- Pago total.
- Registro de anticipos.
- Registro de descuentos.
- Control de saldos pendientes.

- Historial de pagos.
- Reimpresión de recibos.
- Anulación de recibos.
- Consulta de deudas.

2.7.5 Gestión Financiera

Este módulo permitirá controlar los movimientos económicos del instituto.

Comprenderá:

- Registro de ingresos.
- Registro de egresos.
- Registro de gastos administrativos.
- Pago de salarios.
- Arqueo de caja.
- Flujo de caja.
- Balance de ingresos.
- Balance de egresos.
- Reportes financieros.

Toda la información financiera quedará centralizada para facilitar la elaboración de informes económicos.

2.7.6 Gestión Académica

Permitirá administrar la estructura académica del instituto.

Comprenderá:

- Áreas académicas.
- Materias.
- Cursos.
- Paralelos.
- Horarios.
- Aulas.
- Periodos académicos.

Cada curso podrá asociarse a uno o varios docentes y contará con un conjunto de estudiantes inscritos.

2.7.7 Portal del Docente

Cada docente dispondrá de un espacio personalizado para administrar las actividades académicas relacionadas con los cursos que tenga asignados.

Desde este módulo podrá:

- Consultar sus cursos.
- Consultar estudiantes inscritos.
- Registrar asistencia.
- Crear prácticas.
- Publicar tareas.
- Subir material académico.
- Compartir documentos PDF.
- Compartir videos o enlaces.
- Registrar calificaciones.
- Consultar historial de actividades.

2.7.8 Portal del Estudiante

El sistema ofrecerá un portal personalizado para cada estudiante.

Desde este espacio podrá:

- Consultar sus cursos.
- Consultar horarios.
- Descargar material académico.
- Descargar prácticas.
- Subir tareas.
- Consultar calificaciones.
- Consultar asistencia.
- Consultar historial de pagos.
- Descargar recibos.
- Actualizar algunos datos personales autorizados.

2.7.9 Gestión de Docentes

Permitirá administrar la información institucional del personal docente.

El módulo incluirá:

- Registro de docentes.
- Actualización de información.
- Asignación de cursos.
- Asignación de horarios.
- Registro de carga horaria.
- Historial laboral.
- Registro de salarios.
- Consulta de pagos.

2.7.10 Reportes

El sistema proporcionará diversos reportes administrativos, académicos y financieros para facilitar la toma de decisiones.

Entre ellos:

Reportes Académicos

- Estudiantes inscritos.
- Estudiantes por área.
- Estudiantes por curso.
- Cursos activos.
- Docentes asignados.
- Asistencia.
- Calificaciones.

Reportes Financieros

- Ingresos diarios.
- Ingresos mensuales.
- Pagos pendientes.
- Pagos realizados.
- Recibos anulados.
- Egresos.
- Balance financiero.

Reportes Administrativos

- Usuarios registrados.
- Docentes activos.
- Estudiantes activos.
- Inscripciones por periodo.
- Estadísticas por sucursal.

2.7.11 Dashboard Ejecutivo

El sistema contará con un panel principal que mostrará indicadores en tiempo real para facilitar la supervisión institucional.

El Dashboard incluirá información como:

- Número de estudiantes inscritos.
- Número de estudiantes activos.
- Ingresos diarios.

- Ingresos mensuales.
- Pagos pendientes.
- Cursos activos.
- Docentes activos.
- Distribución de estudiantes por área.
- Indicadores por sucursal.
- Gráficos comparativos de ingresos.

2.7.12 Configuración General

Permitirá administrar todos los parámetros del sistema sin necesidad de modificar el código fuente.

Entre ellos:

- Gestión de sucursales.
- Gestión de periodos académicos.
- Gestión de parámetros institucionales.
- Configuración de recibos.
- Configuración de impresoras térmicas.
- Configuración de métodos de pago.
- Configuración de QR.
- Configuración de respaldos.
- Configuración de notificaciones.

2.8 Características de los Usuarios

El Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA – SUMO) será utilizado por diferentes tipos de usuarios, cada uno con responsabilidades y niveles de acceso específicos. Con el propósito de garantizar la seguridad de la información y la correcta ejecución de los procesos institucionales, el sistema implementará un modelo de control de acceso basado en roles y permisos.

Cada usuario dispondrá de credenciales personales de autenticación y únicamente podrá acceder a los módulos y funcionalidades autorizadas según las funciones que desempeñe dentro de la institución.

Los principales tipos de usuarios identificados durante el levantamiento de requerimientos son los siguientes:

2.8.1 Director General

El director general constituye el usuario con mayor nivel de supervisión dentro del sistema. Tendrá acceso a la información administrativa, académica y financiera con fines de control, planificación y toma de decisiones.

Entre sus principales responsabilidades se encuentran:

- Consultar indicadores generales del instituto.
- Supervisar el estado financiero de la institución.
- Revisar reportes de ingresos y egresos.
- Consultar estadísticas de estudiantes inscritos.
- Supervisar el desempeño general de las sucursales.
- Autorizar determinados procesos administrativos, cuando corresponda.
- Acceder a reportes consolidados para la toma de decisiones estratégicas.

El Director tendrá acceso principalmente a información de consulta y supervisión, sin intervenir necesariamente en los procesos operativos diarios.

2.8.2 Administrador del Sistema

El Administrador del Sistema será responsable de la configuración técnica y funcional de la plataforma.

Entre sus funciones principales estarán:

- Crear y administrar cuentas de usuario.
- Gestionar roles y permisos.
- Configurar parámetros generales del sistema.
- Administrar las sucursales.
- Configurar periodos académicos.
- Configurar métodos de pago.
- Configurar impresoras térmicas.
- Gestionar respaldos de información.
- Supervisar el funcionamiento general del sistema.
- Atender incidencias técnicas.

Este usuario contará con acceso completo a todas las funcionalidades del sistema.

2.8.3 Personal Administrativo

El personal administrativo será responsable de ejecutar la mayor parte de los procesos operativos relacionados con la atención a los estudiantes y la gestión institucional.

Sus principales actividades serán:

- Registrar estudiantes.
- Actualizar información.
- Gestionar inscripciones.
- Registrar pagos.
- Emitir recibos.

- Reimprimir comprobantes.
- Anular recibos autorizados.
- Consultar historial de pagos.
- Registrar ingresos y egresos administrativos.
- Elaborar reportes operativos.

Este perfil será uno de los usuarios con mayor interacción diaria con el sistema.

2.8.4 Responsable de Caja

El responsable de caja administrará exclusivamente las operaciones financieras relacionadas con los estudiantes y la institución.

Entre sus funciones estarán:

- Registrar pagos.
- Registrar pagos parciales.
- Registrar pagos mediante código QR.
- Confirmar pagos.
- Emitir recibos.
- Reimprimir recibos.
- Anular comprobantes autorizados.
- Realizar arqueos diarios.
- Consultar movimientos de caja.
- Generar reportes financieros diarios.

El acceso del responsable de caja estará restringido únicamente a los módulos financieros.

2.8.5 Docente

Cada docente dispondrá de un acceso personalizado que le permitirá administrar únicamente los cursos que le hayan sido asignados.

Desde su portal podrá:

- Consultar sus cursos.
- Consultar la lista de estudiantes.
- Registrar asistencia.
- Publicar material académico.
- Crear tareas y prácticas.
- Calificar actividades.
- Consultar el historial de evaluaciones.
- Descargar listados de estudiantes.

Los docentes no tendrán acceso a la información financiera ni administrativa de la institución.

2.8.6 Estudiante

Cada estudiante contará con un acceso individual al sistema.

Desde su portal podrá:

- Consultar sus cursos inscritos.
- Descargar material académico.
- Descargar prácticas.
- Subir tareas.
- Consultar sus calificaciones.
- Consultar su asistencia.
- Consultar el historial de pagos.
- Descargar comprobantes.
- Actualizar algunos datos personales autorizados.

El estudiante únicamente tendrá acceso a su propia información.

2.8.7 Matriz de Roles y Permisos

Con el fin de garantizar la seguridad y el correcto funcionamiento del sistema, se implementará una matriz de permisos basada en roles. La siguiente tabla presenta un resumen de los accesos previstos para cada tipo de usuario:

Funcionalidad	Director	Administrador	Administrativo	Caja	Docente	Estudiante
Dashboard	✓	✓	✓	✓	✓	✓
Gestión de Usuarios	Consultar	✓	✗	✗	✗	✗
Gestión de Estudiantes	Consultar	✓	✓	Consultar	Consultar	Solo su información
Inscripciones	Consultar	✓	✓	Consultar	✗	✗
Gestión de Cursos	Consultar	✓	✓	✗	Consultar	Consultar
Gestión de Docentes	Consultar	✓	✓	✗	Solo su perfil	✗
Registro de Asistencia	Consultar	✓	✗	✗	✓	Consultar
Material Académico	Consultar	✓	✗	✗	✓	Descargar
Prácticas y Evaluaciones	Consultar	✓	✗	✗	✓	Entregar y consultar
Pagos	Consultar	✓	✓	✓	✗	Consultar
Recibos	Consultar	✓	✓	✓	✗	Descargar
Gestión Financiera	✓	✓	Consultar	✓	✗	✗
Reportes	✓	✓	Según permisos	Financieros	Académicos	Personales
Configuración del Sistema	✗	✓	✗	✗	✗	✗

Nota: Los permisos específicos podrán ampliarse o modificarse durante el desarrollo del sistema, de acuerdo con las necesidades operativas de la institución y las políticas definidas por la dirección.

2.9 Restricciones del Sistema

El desarrollo e implementación del **Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO)** estará sujeto a una serie de restricciones técnicas, operativas, organizacionales y de seguridad, las cuales deberán ser consideradas durante todas las fases del proyecto.

Estas restricciones representan las condiciones bajo las cuales el sistema deberá operar y constituyen criterios fundamentales para el diseño de la arquitectura, la implementación de las funcionalidades y la planificación del proyecto.

2.9.1 Restricciones Tecnológicas

El sistema será desarrollado como una aplicación web, permitiendo el acceso desde cualquier equipo autorizado con conexión a Internet y un navegador web compatible.

La arquitectura del sistema deberá facilitar su mantenimiento, escalabilidad y futuras ampliaciones, considerando el crecimiento de la institución.

El desarrollo contemplará las siguientes restricciones tecnológicas:

- El sistema deberá funcionar en los navegadores modernos (Google Chrome, Microsoft Edge y Mozilla Firefox).
- La información será almacenada en una base de datos centralizada.
- La plataforma deberá permitir el acceso simultáneo desde las diferentes sucursales del instituto.
- El sistema deberá ser compatible con impresoras térmicas utilizadas para la emisión de recibos.
- Se deberá contemplar la integración con servicios externos para pagos mediante código QR, dependiendo de la entidad financiera seleccionada por la institución.
- La arquitectura deberá permitir la incorporación de nuevos módulos sin afectar el funcionamiento de los existentes.

2.9.2 Restricciones Operativas

El funcionamiento del sistema dependerá de los procedimientos establecidos por la institución.

Durante el levantamiento de información se identificaron las siguientes restricciones operativas:

- Toda inscripción deberá ser registrada en el sistema antes de iniciar el curso correspondiente.
- Ningún pago podrá registrarse sin una inscripción asociada.
- Cada recibo emitido deberá quedar almacenado en el historial financiero del estudiante.
- La anulación de recibos únicamente podrá ser realizada por usuarios autorizados y deberá quedar registrada en la bitácora del sistema.
- Los docentes únicamente podrán administrar los cursos que les hayan sido asignados.
- Los estudiantes únicamente podrán consultar y modificar la información que les corresponda según los permisos establecidos.
- Cada movimiento financiero deberá generar un registro que permita realizar auditorías posteriores.

2.9.3 Restricciones de Seguridad

Debido a que el sistema administrará información académica, administrativa y financiera, será indispensable implementar mecanismos que garanticen la confidencialidad, integridad y disponibilidad de los datos.

Entre las principales restricciones de seguridad se encuentran:

- Todos los usuarios deberán autenticarse mediante credenciales personales.
- El acceso a los módulos estará controlado mediante roles y permisos.
- Las contraseñas deberán almacenarse utilizando mecanismos seguros de cifrado.
- El sistema registrará las operaciones críticas realizadas por los usuarios mediante una bitácora de auditoría.
- Se restringirá el acceso a información financiera únicamente al personal autorizado.
- Se deberán implementar mecanismos para prevenir accesos no autorizados y proteger la información institucional.

2.9.4 Restricciones de Infraestructura

Para el correcto funcionamiento del sistema, la institución deberá disponer de una infraestructura tecnológica adecuada.

Se consideran las siguientes condiciones mínimas:

- Conectividad estable a Internet en las sucursales.
- Equipos de cómputo con navegadores web actualizados.
- Servidor donde se alojará la aplicación y la base de datos.
- Impresoras térmicas para la emisión de recibos.
- Dispositivos compatibles con los métodos de pago mediante código QR, cuando estos sean implementados.

2.9.5 Restricciones Legales y Administrativas

El sistema deberá respetar las políticas internas del instituto y las disposiciones legales aplicables al manejo de información y registros administrativos.

En particular, se considerarán las siguientes restricciones:

- La información registrada deberá mantenerse confidencial y ser utilizada únicamente para fines institucionales.
- Los registros financieros deberán conservarse para fines de control y auditoría.
- Las anulaciones de recibos deberán conservar un historial que permita identificar el usuario responsable, la fecha y el motivo de la anulación.
- El acceso a la información deberá estar sujeto a las funciones y responsabilidades asignadas a cada usuario.

2.9.6 Restricciones de Rendimiento

El sistema deberá ofrecer un desempeño adecuado para garantizar una experiencia de uso eficiente.

Se establecen las siguientes consideraciones:

- Soportar aproximadamente 500 estudiantes registrados y alrededor de 50 docentes, con posibilidad de crecimiento futuro.
- Permitir el acceso concurrente de múltiples usuarios sin afectar significativamente el rendimiento.
- Generar reportes en tiempos razonables.
- Emitir recibos de manera inmediata después del registro de un pago.
- Mantener tiempos de respuesta adecuados durante las operaciones de consulta y registro.

2.9.7 Restricciones de Mantenimiento y Escalabilidad

El sistema será diseñado considerando futuras ampliaciones funcionales.

Por ello:

- La arquitectura deberá facilitar la incorporación de nuevos módulos.
- El código fuente deberá organizarse siguiendo buenas prácticas de desarrollo.
- La base de datos deberá diseñarse de forma normalizada para reducir redundancias y facilitar su mantenimiento.
- La documentación técnica deberá mantenerse actualizada durante el desarrollo del proyecto.

2.10 Suposiciones y Dependencias

El desarrollo del Sistema Integral de Gestión Académica y Administrativa (SIGA) parte de una serie de supuestos que permitirán planificar y ejecutar el proyecto de manera adecuada. Asimismo, existen dependencias externas cuyo cumplimiento será necesario para garantizar el correcto funcionamiento de la solución.

2.10.1 Suposiciones

Para el desarrollo del proyecto se asume que:

- La dirección del instituto proporcionará oportunamente la información necesaria para el análisis y validación de los requerimientos.
- Los responsables de cada área participarán en la validación de las funcionalidades correspondientes.

- Los procesos administrativos y académicos descritos durante el levantamiento de información representan el funcionamiento actual del instituto.
- El personal utilizará el sistema conforme a los procedimientos definidos y recibirá la capacitación necesaria para su operación.
- La institución mantendrá una infraestructura tecnológica adecuada para el funcionamiento de la plataforma.

2.10.2 Dependencias

El proyecto dependerá de diversos factores externos, entre los que destacan:

- Disponibilidad de un servidor para el despliegue de la aplicación.
- Acceso a una conexión estable a Internet en las sucursales.
- Disponibilidad y compatibilidad de las impresoras térmicas utilizadas por la institución.
- Definición de la entidad financiera o pasarela de pagos que permitirá la integración de los pagos mediante código QR.
- Disponibilidad de los usuarios finales para realizar pruebas y validar los módulos desarrollados.
- Entrega oportuna de logotipos, formatos de recibos y demás recursos institucionales necesarios para la personalización del sistema.

2.11 Alcance Funcional

El Sistema Integral de Gestión Académica y Administrativa (SIGA) tiene como finalidad proporcionar una solución tecnológica integral para la administración de los procesos académicos, administrativos y financieros del instituto, permitiendo centralizar la información institucional y optimizar las actividades realizadas por el personal administrativo, docentes, estudiantes y directivos.

El alcance funcional definido en este documento comprende el desarrollo de los módulos necesarios para cubrir los procesos principales identificados durante el levantamiento de requerimientos. Cada módulo será implementado de forma integrada, garantizando la consistencia de la información y facilitando la interacción entre las distintas áreas de la institución.

La primera versión del sistema incluirá las funcionalidades descritas a continuación.

2.11.1 Gestión de Usuarios y Seguridad

Este módulo permitirá administrar el acceso al sistema mediante autenticación y autorización basada en roles.

Comprenderá las siguientes funcionalidades:

- Inicio y cierre de sesión.

- Administración de usuarios.
- Gestión de roles y permisos.
- Recuperación y cambio de contraseña.
- Administración de sesiones.
- Registro de auditoría de actividades (bitácora).

2.11.2 Gestión de Estudiantes

Permitirá registrar y administrar la información académica y administrativa de los estudiantes.

Las principales funciones serán:

- Registro de estudiantes.
- Actualización de datos personales.
- Registro de tutores y contactos.
- Consulta del historial de inscripciones.
- Consulta del historial de pagos.
- Búsqueda y filtrado de estudiantes.
- Administración del estado académico del estudiante.

2.11.3 Gestión Académica

Permitirá administrar la estructura académica del instituto.

Incluye:

- Registro de áreas académicas.
- Administración de cursos.
- Administración de materias.
- Gestión de horarios.
- Asignación de docentes.
- Gestión de periodos académicos.
- Administración de aulas y sucursales.

2.11.4 Gestión de Inscripciones

Permitirá registrar la inscripción de estudiantes en los diferentes cursos ofertados por la institución.

El proceso comprenderá:

- Selección del estudiante.
- Selección del curso.
- Asignación de horario.

- Registro de fechas de inicio y finalización.
- Registro del tipo de pago.
- Cálculo automático del periodo de inscripción.
- Generación del registro académico correspondiente.

2.11.5 Gestión de Pagos

Permitirá administrar los pagos realizados por los estudiantes.

El módulo incluirá:

- Registro de pagos únicos.
- Registro de pagos parciales.
- Registro de pagos en cuotas.
- Consulta del estado de cuenta.
- Control de saldos pendientes.
- Historial de pagos.
- Registro de descuentos.
- Control de deudas.

2.11.6 Emisión de Recibos

El sistema permitirá emitir comprobantes de pago inmediatamente después del registro de una transacción.

Las principales funciones serán:

- Generación automática del recibo.
- Impresión mediante impresora térmica.
- Reimpresión de recibos.
- Consulta del historial de recibos.
- Anulación de recibos con registro del motivo y del usuario responsable.

2.11.7 Integración con Pagos mediante Código QR

El sistema permitirá registrar pagos electrónicos realizados mediante códigos QR, siempre que la institución disponga de una pasarela de pagos o servicio bancario compatible.

Las funcionalidades contempladas incluyen:

- Generación del código QR para el pago.
- Registro del comprobante de pago.
- Confirmación de la transacción.
- Asociación automática del pago con la inscripción correspondiente.
- Registro del historial de pagos electrónicos.

2.11.8 Portal del Docente

Cada docente dispondrá de un espacio personalizado para administrar la información de los cursos asignados.

Las funcionalidades principales serán:

- Consulta de cursos asignados.
- Registro de asistencia.
- Publicación de material académico.
- Creación de prácticas y tareas.
- Registro de calificaciones.
- Consulta del listado de estudiantes.
- Descarga de reportes académicos.

2.11.9 Portal del Estudiante

Cada estudiante contará con un portal académico donde podrá consultar su información personal y académica.

Entre las funcionalidades se incluyen:

- Consulta de cursos inscritos.
- Descarga de material académico.
- Entrega de prácticas y tareas.
- Consulta de calificaciones.
- Consulta de asistencia.
- Consulta del estado de cuenta.
- Descarga de comprobantes de pago.

2.11.10 Gestión Financiera

Este módulo permitirá controlar los movimientos económicos del instituto.

Comprenderá:

- Registro de ingresos.
- Registro de egresos.
- Control de caja.
- Pago de salarios.
- Arqueo de caja.
- Flujo de efectivo.
- Reportes financieros.
- Historial de movimientos económicos.

2.11.11 Reportes e Indicadores

El sistema generará reportes administrativos, académicos y financieros que apoyarán la toma de decisiones institucionales.

Entre ellos:

- Estudiantes inscritos.
- Inscripciones por periodo.
- Estudiantes por curso.
- Estudiantes por sucursal.
- Pagos realizados.
- Pagos pendientes.
- Ingresos diarios.
- Ingresos mensuales.
- Egresos.
- Salarios.
- Asistencia.
- Calificaciones.
- Reportes estadísticos.

2.11.12 Dashboard Ejecutivo

El sistema incorporará un panel de control con indicadores en tiempo real que facilitarán el seguimiento de las actividades institucionales.

El Dashboard presentará información como:

- Total de estudiantes inscritos.
- Estudiantes activos.
- Cursos activos.
- Docentes activos.
- Ingresos diarios.
- Ingresos mensuales.
- Pagos pendientes.
- Distribución de estudiantes por área.
- Indicadores por sucursal.
- Estadísticas financieras.

2.11.13 Funcionalidades Previstas para Versiones Futuras

Con el propósito de mantener una adecuada planificación del proyecto, algunas funcionalidades podrán ser incorporadas en futuras versiones del sistema, de acuerdo con las necesidades de la institución y la disponibilidad de recursos.

Entre las funcionalidades previstas se consideran:

- Aplicación móvil para estudiantes y docentes.
- Integración con plataformas de videoconferencia.
- Envío de notificaciones mediante correo electrónico y mensajería instantánea.
- Integración con servicios de facturación electrónica, si la normativa lo requiere.
- Módulo de biblioteca digital.
- Módulo de estadísticas avanzadas mediante inteligencia de negocios (Business Intelligence).
- Integración con sistemas universitarios o plataformas externas.

2.12 Conclusión del Capítulo 2

Con la información presentada en este capítulo se establece el contexto general del proyecto, describiendo la situación actual del instituto, la problemática identificada, los objetivos del sistema, los perfiles de usuarios, las restricciones y el alcance funcional que servirá como base para el diseño e implementación del Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA – SUMO).

La definición del alcance permitirá controlar el desarrollo del proyecto, establecer prioridades de implementación y facilitar la validación de las funcionalidades solicitadas por la institución.

CAPÍTULO 3 REQUERIMIENTOS FUNCIONALES

3.1 Introducción

Los requerimientos funcionales describen las funcionalidades que deberá proporcionar el Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO), especificando el comportamiento esperado del sistema en respuesta a las acciones realizadas por los diferentes tipos de usuarios.

Cada requerimiento funcional ha sido identificado mediante un código único que permitirá facilitar su seguimiento durante las etapas de análisis, diseño, implementación, pruebas y mantenimiento del proyecto.

Los requerimientos se encuentran organizados por módulos funcionales, permitiendo una mejor administración del desarrollo del sistema y facilitando futuras ampliaciones.

Los niveles de prioridad utilizados en este documento son los siguientes:

Prioridad	Descripción
Alta	Funcionalidad indispensable para el funcionamiento del sistema.
Media	Funcionalidad importante que puede desarrollarse después de las funciones principales.

Baja	Funcionalidad complementaria que mejora la experiencia del usuario o amplía las capacidades del sistema.
------	--

3.2 Organización de los Requerimientos Funcionales

Los requerimientos funcionales se encuentran agrupados según los módulos que conforman el sistema.

Módulo	Código
Gestión de Usuarios	RF-001 – RF-020
Gestión de Roles y Permisos	RF-021 – RF-035
Gestión de Estudiantes	RF-036 – RF-070
Gestión Académica	RF-071 – RF-095
Gestión de Docentes	RF-096 – RF-120
Gestión de Inscripciones	RF-121 – RF-155
Gestión de Pagos	RF-156 – RF-185
Gestión Financiera	RF-186 – RF-210
Portal del Docente	RF-211 – RF-240
Portal del Estudiante	RF-241 – RF-265
Reportes	RF-266 – RF-290
Dashboard	RF-291 – RF-300

Nota: La numeración presentada constituye una planificación inicial y podrá ampliarse conforme se identifiquen nuevos requerimientos durante el desarrollo del proyecto.

3.3 Requerimientos Funcionales del Módulo de Gestión de Usuarios

RF-001 – Inicio de Sesión

Código: RF-001

Nombre: Inicio de Sesión

Descripción:

El sistema deberá permitir que los usuarios autorizados accedan a la plataforma mediante el ingreso de sus credenciales de autenticación.

Actores

- Administrador
- Director
- Administrativo
- Caja
- Docente
- Estudiante

Prioridad

Alta

Precondiciones

- El usuario deberá estar registrado.
- La cuenta deberá encontrarse activa.

Flujo General

1. El usuario ingresa al sistema.
2. Introduce su usuario o correo electrónico.
3. Introduce su contraseña.
4. El sistema valida las credenciales.
5. El sistema identifica el rol correspondiente.
6. El sistema redirige al Dashboard.

Postcondiciones

- Se crea una sesión autenticada.
- Se registra el acceso en la bitácora.

Criterios de aceptación

- Solo usuarios con credenciales válidas podrán acceder.
- El sistema deberá mostrar mensajes claros en caso de error.
- Después de varios intentos fallidos consecutivos, la cuenta podrá bloquearse temporalmente según la política de seguridad definida.

RF-002 – Cierre de Sesión

Descripción

El sistema deberá permitir que cualquier usuario autenticado cierre su sesión de forma segura.

Actores

Todos los usuarios.

Prioridad

Alta.

Precondiciones

El usuario debe encontrarse autenticado.

Flujo

1. Seleccionar "Cerrar sesión".
2. El sistema finaliza la sesión.
3. Redirecciona al Login.

Postcondiciones

La sesión queda invalidada.

Criterios de aceptación

No será posible acceder nuevamente a páginas protegidas utilizando la sesión finalizada.

RF-003 – Recuperación de Contraseña

Descripción

El sistema permitirá recuperar el acceso a la cuenta mediante un proceso seguro de restablecimiento de contraseña.

Actores

Todos los usuarios.

Prioridad

Alta.

Flujo General

1. Solicitar recuperación.
2. Verificar identidad del usuario.
3. Generar un enlace o código temporal.
4. Permitir establecer una nueva contraseña.

Criterios de aceptación

El enlace o código de recuperación deberá tener una vigencia limitada y solo podrá utilizarse una vez.

RF-004 – Cambio de Contraseña

Descripción

El sistema permitirá que el usuario cambie su contraseña desde su perfil personal.

Actores

Todos los usuarios autenticados.

Prioridad

Alta.

Precondiciones

El usuario debe haber iniciado sesión y proporcionar su contraseña actual.

Flujo General

1. Acceder a la opción de cambio de contraseña.
2. Ingresar la contraseña actual.
3. Ingresar y confirmar la nueva contraseña.
4. El sistema valida la información.
5. Se actualiza la contraseña y se notifica al usuario.

Postcondiciones

La nueva contraseña queda registrada y la anterior deja de ser válida.

Criterios de aceptación

- La nueva contraseña deberá cumplir la política de seguridad establecida.
- El sistema deberá confirmar que la actualización fue realizada correctamente.

RF-005 – Gestión de Usuarios

Descripción

El sistema permitirá al Administrador registrar, modificar, consultar, activar, desactivar y eliminar usuarios autorizados, asignándoles el rol correspondiente para el acceso al sistema.

Actores

- Administrador del Sistema.

Prioridad

Alta.

Precondiciones

El Administrador deberá haber iniciado sesión con permisos de gestión de usuarios.

Flujo General

1. Acceder al módulo de usuarios.
2. Registrar un nuevo usuario o seleccionar uno existente.
3. Completar o modificar la información requerida.
4. Asignar el rol correspondiente.
5. Guardar los cambios.

Postcondiciones

El usuario queda registrado o actualizado y podrá acceder al sistema de acuerdo con los permisos asignados.

Criterios de aceptación

- No se permitirán usuarios duplicados con el mismo correo electrónico o nombre de usuario.
- El sistema deberá registrar en la bitácora todas las operaciones realizadas sobre los usuarios.

RF-006 – Consulta de Usuarios

Descripción:

El sistema permitirá al Administrador consultar la lista de usuarios registrados, mostrando información relevante como nombre, rol, estado de la cuenta, correo electrónico y fecha de creación.

Actores

- Administrador del Sistema

Prioridad

Alta

Precondiciones

- El Administrador deberá haber iniciado sesión.
- Deberán existir usuarios registrados en el sistema.

Flujo General

1. Acceder al módulo de Gestión de Usuarios.
2. Seleccionar la opción "Consultar Usuarios".
3. El sistema mostrará la lista paginada de usuarios.
4. El Administrador podrá visualizar el detalle de cualquier usuario.

Postcondiciones

No se modifica la información del sistema.

Criterios de aceptación

- La lista deberá mostrar únicamente usuarios registrados.
- La información deberá presentarse de forma ordenada y paginada.

RF-007 – Búsqueda de Usuarios

Descripción

El sistema permitirá localizar usuarios mediante diferentes criterios de búsqueda.

Actores

- Administrador

Prioridad

Alta

Precondiciones

Existencia de usuarios registrados.

Flujo General

1. Ingresar un criterio de búsqueda.
2. Seleccionar el filtro correspondiente.
3. El sistema mostrará los resultados encontrados.

Filtros disponibles

- Nombre
- Apellidos
- Usuario
- Correo electrónico
- Rol
- Estado

Postcondiciones

No modifica información.

Criterios de aceptación

La búsqueda deberá ejecutarse en pocos segundos y permitir coincidencias parciales.

RF-008 – Modificación de Usuarios

Descripción

El sistema permitirá actualizar la información de cualquier usuario registrado.

Actores

- Administrador

Prioridad

Alta

Precondiciones

- Usuario existente.
- Permisos suficientes.

Flujo General

1. Buscar usuario.
2. Abrir formulario de edición.
3. Modificar información.
4. Guardar cambios.
5. El sistema valida los datos.
6. Actualizar la información.

Postcondiciones

Los nuevos datos quedan almacenados.

Criterios de aceptación

Toda modificación deberá registrarse en la bitácora.

RF-009 – Desactivar Usuario

Descripción

El sistema permitirá suspender temporalmente una cuenta sin eliminar su información.

Actores

- Administrador

Prioridad

Alta

Flujo General

1. Seleccionar usuario.
2. Elegir "Desactivar".
3. Confirmar operación.
4. Actualizar estado.

Postcondiciones

El usuario no podrá iniciar sesión.

Criterios de aceptación

Toda desactivación deberá quedar registrada con:

- Usuario responsable.
- Fecha.
- Hora.
- Motivo.

RF-010 – Reactivar Usuario

Descripción

Permitirá habilitar nuevamente una cuenta previamente suspendida.

Actores

Administrador.

Prioridad

Alta.

Flujo

1. Buscar usuario.
2. Seleccionar "Reactivar".
3. Confirmar.

Postcondiciones

El usuario recupera el acceso al sistema.

RF-011 – Eliminación Lógica de Usuario

Descripción

El sistema permitirá marcar un usuario como eliminado sin borrar físicamente su información de la base de datos, preservando la integridad histórica de los registros.

Actores

Administrador.

Prioridad

Media.

Precondiciones

El usuario no deberá tener procesos críticos pendientes.

Postcondiciones

La cuenta dejará de estar disponible para su utilización, pero su historial permanecerá almacenado.

RF-012 – Asignación de Rol

Descripción

El Administrador podrá asignar un rol institucional a cada usuario.

Roles iniciales

- Administrador
- Director

- Administrativo
- Caja
- Docente
- Estudiante

Prioridad

Alta.

Criterios de aceptación

Cada usuario deberá poseer al menos un rol activo.

RF-013 – Cambio de Rol**Descripción**

Permitirá modificar el rol asignado a un usuario cuando cambie su función dentro del instituto.

RF-014 – Visualización del Perfil**Descripción**

Cada usuario podrá consultar su información personal.

Podrá visualizar:

- Nombre
- Correo
- Teléfono
- Rol
- Fecha de creación

RF-015 – Actualización del Perfil**Descripción**

Cada usuario podrá actualizar determinados datos personales autorizados.

No podrá modificar:

- Rol
- Estado
- Permisos

RF-016 – Gestión de Fotografía

Descripción

El sistema permitirá subir, actualizar o eliminar la fotografía del perfil del usuario.

Formatos permitidos:

- JPG
- PNG

RF-017 – Registro de Auditoría

Descripción

Toda operación realizada por un usuario será registrada automáticamente.

La auditoría almacenará:

- Usuario
- Fecha
- Hora
- Dirección IP
- Acción realizada
- Módulo
- Resultado

RF-018 – Consulta de Bitácora

Descripción

El Administrador podrá consultar el historial completo de actividades registradas.

Filtros:

- Usuario
- Fecha
- Acción
- Módulo

RF-019 – Bloqueo Automático

Descripción

Después de varios intentos fallidos consecutivos, el sistema bloqueará temporalmente la cuenta del usuario.

El bloqueo podrá configurarse desde los parámetros del sistema.

RF-020 – Notificaciones de Seguridad

Descripción

El sistema notificará al usuario sobre eventos importantes relacionados con la seguridad de su cuenta.

Ejemplos:

- Inicio de sesión desde un nuevo dispositivo.
- Cambio de contraseña.
- Restablecimiento de contraseña.
- Bloqueo temporal.
- Reactivación de la cuenta.

3.4 Requerimientos Funcionales del Módulo de Roles y Permisos

Este módulo permitirá administrar los perfiles de acceso y las autorizaciones dentro del sistema, garantizando que cada usuario pueda ejecutar únicamente las operaciones correspondientes a sus responsabilidades institucionales.

Los requerimientos de este módulo estarán identificados desde **RF-021** hasta **RF-035** y abarcarán la creación de roles, asignación de permisos, control de accesos y políticas de seguridad.

RF-021 – Registrar Rol

Descripción

El sistema permitirá al Administrador crear nuevos roles para adaptar la plataforma a futuras necesidades organizacionales.

Actores

- Administrador del Sistema.

Prioridad

Alta.

Precondiciones

- El Administrador debe haber iniciado sesión.
- Debe contar con permisos para administrar roles.

Flujo General

1. Acceder al módulo de Roles y Permisos.
2. Seleccionar la opción "Nuevo Rol".
3. Registrar el nombre y descripción del rol.
4. Asignar los permisos iniciales.
5. Guardar la información.

Postcondiciones

El nuevo rol quedará disponible para ser asignado a usuarios.

Criterios de aceptación

- No se permitirá registrar dos roles con el mismo nombre.
- Todo nuevo rol deberá quedar registrado en la bitácora del sistema.

RF-022 – Modificar Rol

Descripción

El sistema permitirá al Administrador modificar la información de un rol existente, incluyendo su nombre, descripción y los permisos asociados.

Actores

- Administrador del Sistema.

Prioridad

Alta.

Precondiciones

- El rol debe existir.
- El Administrador debe contar con permisos para gestionar roles.

Flujo General

1. Acceder al módulo de Roles.
2. Seleccionar el rol a modificar.
3. Editar la información necesaria.
4. Guardar los cambios.

5. El sistema validará la información y actualizará el registro.

Postcondiciones

El rol quedará actualizado y los cambios se reflejarán en los usuarios asociados.

Criterios de aceptación

- No se permitirá duplicar nombres de roles.
- La modificación quedará registrada en la bitácora.

RF-023 – Eliminar Rol**Descripción**

El sistema permitirá eliminar un rol únicamente cuando no esté asignado a ningún usuario.

Actores

- Administrador del Sistema.

Prioridad

Media.

Precondiciones

- El rol no debe estar asignado a usuarios activos.

Flujo General

1. Seleccionar el rol.
2. Solicitar eliminación.
3. Confirmar la operación.
4. El sistema verifica si existen usuarios asociados.
5. Si no existen, elimina el rol.

Postcondiciones

El rol deja de estar disponible.

Criterios de aceptación

- No se permitirá eliminar el rol Administrador.
- El sistema deberá informar si existen usuarios asociados.

RF-024 – Asignar Permisos a un Rol

Descripción

El sistema permitirá asignar permisos específicos a cada rol, determinando las acciones que podrán realizar los usuarios.

Actores

- Administrador.

Prioridad

Alta.

Permisos

- Crear
- Consultar
- Modificar
- Eliminar
- Aprobar
- Imprimir
- Exportar
- Configurar

Flujo General

1. Seleccionar un rol.
2. Elegir el módulo.
3. Marcar los permisos.
4. Guardar la configuración.

Postcondiciones

Los usuarios con ese rol heredarán los permisos configurados.

RF-025 – Revocar Permisos

Descripción

Permitirá retirar permisos previamente asignados a un rol.

Actores

- Administrador.

Prioridad

Alta.

Criterios de aceptación

Los cambios deberán aplicarse inmediatamente después de ser guardados.

RF-026 – Consultar Roles**Descripción**

Permitirá visualizar todos los roles existentes junto con sus permisos asociados.

Actores

Administrador.

Prioridad

Alta.

RF-027 – Buscar Roles**Descripción**

Permitirá localizar roles mediante nombre o descripción.

Filtros

- Nombre
- Estado
- Fecha de creación

RF-028 – Activar Rol**Descripción**

Permitirá habilitar un rol previamente deshabilitado.

RF-029 – Desactivar Rol**Descripción**

Permitirá suspender temporalmente un rol sin eliminarlo.

Los usuarios que tengan asignado dicho rol perderán temporalmente el acceso correspondiente.

RF-030 – Duplicar Rol

Descripción

Permitirá crear un nuevo rol copiando la configuración y los permisos de otro rol existente.

Objetivo

Facilitar la creación de perfiles similares.

RF-031 – Consulta de Permisos

Descripción

Permitirá visualizar todos los permisos asignados a un determinado rol.

RF-032 – Validación de Permisos

Descripción

Antes de ejecutar cualquier operación, el sistema verificará que el usuario posea el permiso correspondiente.

Criterios de aceptación

Si el usuario no cuenta con autorización, el sistema deberá:

- Denegar la operación.
- Mostrar un mensaje informativo.
- Registrar el intento en la bitácora.

RF-033 – Herencia de Permisos

Descripción

Los permisos asignados a un rol serán heredados automáticamente por todos los usuarios que pertenezcan a dicho rol.

Cuando un permiso sea modificado, el cambio deberá reflejarse inmediatamente.

RF-034 – Auditoría de Roles y Permisos

Descripción

Toda modificación relacionada con roles o permisos deberá registrarse automáticamente.

La auditoría almacenará:

- Usuario responsable.
- Fecha.
- Hora.
- Acción realizada.
- Rol afectado.
- Permisos modificados.

RF-035 – Configuración de Políticas de Seguridad

Descripción

El sistema permitirá definir políticas generales relacionadas con la autenticación y el acceso.

Entre ellas:

- Longitud mínima de contraseña.
- Complejidad de contraseña.
- Tiempo máximo de sesión.
- Número máximo de intentos fallidos.
- Tiempo de bloqueo.
- Vigencia de recuperación de contraseña.

Prioridad

Alta.

Criterios de aceptación

Las políticas configuradas deberán aplicarse automáticamente a todos los usuarios del sistema.

3.5 Requerimientos Funcionales del Módulo de Gestión de Estudiantes

Con este apartado comienza uno de los módulos más importantes del sistema, ya que concentra toda la información académica y administrativa de los estudiantes.

Los procesos documentados incluirán:

- Registro de estudiantes.
- Gestión de datos personales.
- Información de tutores.

- Historial de inscripciones.
- Historial de pagos.
- Búsquedas y filtros.
- Estados del estudiante.
- Consulta del expediente.
- Gestión documental.

RF-036 – Registrar Estudiante

Descripción

El sistema permitirá registrar un nuevo estudiante en la base de datos institucional, almacenando toda la información personal, de contacto y administrativa necesaria para su posterior inscripción en los cursos ofertados por el instituto.

Actores

- Personal Administrativo.
- Administrador del Sistema.

Prioridad

Alta.

Precondiciones

- El usuario debe haber iniciado sesión.
- Debe contar con permisos para registrar estudiantes.

Información mínima requerida

Datos personales

- Apellido paterno.
- Apellido materno.
- Nombres.
- Número de Cédula de Identidad.
- Correo electrónico.
- Teléfono celular.

Información de contacto

- Nombre del Tutor 1.
- Celular del Tutor 1.
- Nombre del Tutor 2 (opcional).

- Celular del Tutor 2 (opcional).

Flujo General

1. Acceder al módulo "Gestión de Estudiantes".
2. Seleccionar la opción "Nuevo Estudiante".
3. Completar el formulario.
4. Validar la información ingresada.
5. Guardar el registro.
6. El sistema genera el expediente del estudiante.

Postcondiciones

El estudiante quedará registrado y disponible para realizar procesos de inscripción.

Criterios de aceptación

- No se permitirá registrar dos estudiantes con el mismo número de Cédula de Identidad.
- Todos los campos obligatorios deberán validarse antes de guardar la información.
- El sistema asignará un identificador único al expediente del estudiante.

RF-037 – Consultar Estudiante

Descripción

El sistema deberá permitir consultar la información completa de un estudiante registrado, incluyendo sus datos personales, información de contacto, tutores, historial de inscripciones, estado académico e historial financiero.

Actores

- Administrador del Sistema
- Personal Administrativo
- Director (solo consulta)

Prioridad

Alta.

Precondiciones

- El usuario debe haber iniciado sesión.
- El estudiante debe estar registrado.

Flujo General

1. Acceder al módulo Gestión de Estudiantes.
2. Buscar el estudiante.
3. Seleccionar "Ver Información".
4. El sistema muestra el expediente completo.

Postcondiciones

No modifica información.

Criterios de aceptación

- La información deberá presentarse organizada por secciones.
- Solo usuarios autorizados podrán acceder al expediente.

RF-038 – Buscar Estudiante

Descripción

El sistema permitirá localizar estudiantes mediante diferentes criterios de búsqueda.

Actores

- Administrador
- Administrativo
- Director

Prioridad

Alta.

Filtros

- Código del estudiante.
- CI.
- Nombre.
- Apellido paterno.
- Apellido materno.
- Curso.
- Área.
- Estado.
- Tutor.

Flujo General

1. Ingresar criterio.

2. Seleccionar filtro.
3. Mostrar coincidencias.

Postcondiciones

No modifica información.

Criterios de aceptación

- Debe permitir búsquedas parciales.
- Debe ordenar resultados.
- Debe paginar resultados.

RF-039 – Modificar Información del Estudiante**Descripción**

Permitirá actualizar la información personal y de contacto del estudiante.

Actores

- Administrativo
- Administrador

Prioridad

Alta.

Precondiciones

El estudiante debe existir.

Información modificable

- Correo.
- Celular.
- Dirección (si se incorpora).
- Tutor.
- Celular del tutor.

Información restringida

- Código interno.
- Historial de pagos.
- Historial de inscripciones.

Flujo

1. Buscar estudiante.
2. Abrir formulario.
3. Editar datos.
4. Guardar cambios.

Postcondiciones

La información queda actualizada.

Criterios de aceptación

Toda modificación deberá registrarse en la bitácora.

RF-040 – Cambiar Estado del Estudiante

Descripción

El sistema permitirá modificar el estado del estudiante de acuerdo con su situación académica o administrativa.

Estados posibles

- Activo.
- Inactivo.
- Retirado.
- Suspendido.
- Egresado.

Actores

Administrador.

Administrativo.

Prioridad

Alta.

Criterios de aceptación

Todo cambio deberá registrar:

- Usuario.
- Fecha.

- Hora.
- Motivo.

RF-041 – Registrar Tutor

Descripción

Permitirá registrar la información del tutor principal del estudiante.

Datos

- Nombre completo.
- Celular.
- Correo electrónico (opcional).
- Relación con el estudiante.

RF-042 – Registrar Segundo Tutor

Descripción

Permitirá registrar un segundo tutor cuando corresponda.

Este registro será opcional.

RF-043 – Actualizar Información del Tutor

Descripción

Permitirá modificar la información registrada de los tutores.

RF-044 – Historial Académico

Descripción

El sistema almacenará automáticamente todo el historial académico del estudiante.

Incluye:

- Cursos inscritos.
- Fechas.
- Docentes.
- Asistencia.
- Calificaciones.

El historial nunca podrá eliminarse.

RF-045 – Historial Financiero

Descripción

El sistema registrará todos los movimientos económicos relacionados con el estudiante.

Incluye:

- Inscripciones.
- Pagos.
- Cuotas.
- Descuentos.
- Deudas.
- Recibos.

RF-046 – Consulta del Estado de Cuenta

Descripción

Permitirá visualizar el saldo pendiente del estudiante.

Mostrará:

- Total.
- Pagado.
- Pendiente.

RF-047 – Visualización del Expediente Completo

Descripción

El sistema mostrará un expediente único del estudiante.

El expediente incluirá:

Información personal

- Fotografía.
- Nombre.
- CI.
- Contacto.

Información académica

- Cursos.

- Asistencia.
- Calificaciones.

Información financiera

- Pagos.
- Recibos.
- Estado de cuenta.

RF-048 – Adjuntar Documentación

Descripción

Permitirá almacenar documentos digitales asociados al estudiante.

Ejemplos:

- CI.
- Certificados.
- Comprobantes.

Formatos:

- PDF.
- JPG.
- PNG.

RF-049 – Validación de Duplicados

Descripción

Antes de registrar un estudiante el sistema verificará automáticamente que no exista otro con el mismo CI.

En caso de duplicidad:

- Cancelará el registro.
- Informará el motivo.

RF-050 – Eliminación Lógica del Estudiante

Descripción

El sistema permitirá realizar únicamente una eliminación lógica del estudiante.

Nunca eliminará información física debido a:

- Historial académico.
- Historial financiero.
- Auditorías.

El estudiante quedará marcado como:

INACTIVO

RF-051 – Registro de Observaciones

Descripción

El sistema permitirá registrar observaciones administrativas o académicas relacionadas con un estudiante, las cuales podrán ser consultadas posteriormente por los usuarios autorizados.

Actores

- Administrador del Sistema.
- Personal Administrativo.
- Director (solo consulta).

Prioridad

Media.

Precondiciones

- El estudiante debe estar registrado.
- El usuario debe contar con permisos para registrar observaciones.

Flujo General

1. Buscar el estudiante.
2. Acceder a la sección de observaciones.
3. Registrar el comentario.
4. Guardar la información.

Postcondiciones

La observación quedará asociada al expediente del estudiante.

Criterios de aceptación

- Cada observación deberá almacenar la fecha, hora y usuario responsable.

- Las observaciones no podrán eliminarse, solo marcarse como cerradas o archivadas.

RF-052 – Consulta de Observaciones

Descripción

El sistema permitirá consultar el historial completo de observaciones registradas para un estudiante.

Actores

- Administrador.
- Personal Administrativo.
- Director.

Prioridad

Media.

Precondiciones

- El estudiante debe tener observaciones registradas.

Flujo General

1. Buscar el estudiante.
2. Acceder a la sección de observaciones.
3. Visualizar el listado cronológico.

Postcondiciones

No modifica información.

Criterios de aceptación

- Las observaciones deberán mostrarse ordenadas por fecha.
- Se visualizará el usuario que realizó cada registro y el estado de la observación.

RF-053 – Gestión de Fotografía del Estudiante

Descripción: El sistema deberá permitir registrar y actualizar la fotografía del estudiante para facilitar su identificación dentro del expediente institucional.

Actores: Administrativo, Administrador.

Prioridad: Media.

Precondiciones:

- El estudiante debe estar registrado.
- El usuario debe tener permisos suficientes.

Flujo general:

1. Buscar al estudiante.
2. Acceder a su expediente.
3. Seleccionar la opción para agregar o modificar fotografía.
4. Seleccionar el archivo.
5. El sistema validará el formato y tamaño.
6. Guardar la fotografía.

Criterios de aceptación:

- Se permitirán formatos de imagen configurados por el administrador.
- La fotografía deberá asociarse únicamente al estudiante seleccionado.
- La modificación deberá registrarse en la bitácora.

RF-054 – Cambio de Área Académica

Descripción: El sistema deberá permitir modificar el área académica asociada a un estudiante cuando exista una solicitud autorizada de cambio.

Actores: Administrador, Administrativo.

Prioridad: Media.

Precondiciones:

- El estudiante debe estar registrado.
- Debe existir autorización para realizar el cambio.

Flujo general:

1. Buscar al estudiante.
2. Seleccionar la opción de cambio de área.
3. Seleccionar la nueva área.
4. Registrar el motivo.
5. Confirmar la operación.
6. Guardar el cambio.

Criterios de aceptación:

- El sistema conservará el historial del área anterior.

- El cambio no deberá eliminar las inscripciones ni pagos anteriores.
- La operación deberá quedar registrada en la bitácora.

RF-055 – Cambio de Curso

Descripción: El sistema deberá permitir transferir a un estudiante de un curso a otro cuando la institución lo autorice.

Actores: Administrador, Administrativo.

Prioridad: Media.

Precondiciones:

- El estudiante debe contar con una inscripción activa.
- El nuevo curso debe encontrarse habilitado.

Flujo general:

1. Buscar al estudiante.
2. Consultar sus inscripciones.
3. Seleccionar la inscripción correspondiente.
4. Seleccionar el nuevo curso.
5. Registrar el motivo.
6. Confirmar el cambio.

Criterios de aceptación:

- La inscripción anterior deberá conservarse como parte del historial.
- El nuevo curso deberá quedar asociado al estudiante.
- El sistema deberá registrar quién realizó el cambio y cuándo.

RF-056 – Transferencia de Sucursal

Descripción: El sistema deberá permitir transferir la atención académica o administrativa de un estudiante de una sucursal a otra.

Actores: Administrador, Administrativo.

Prioridad: Media.

Precondiciones:

- Ambas sucursales deberán encontrarse activas.
- El usuario deberá tener autorización.

Flujo general:

1. Seleccionar estudiante.
2. Seleccionar sucursal actual.
3. Seleccionar nueva sucursal.
4. Registrar motivo.
5. Confirmar transferencia.

Criterios de aceptación:

- Se conservará el historial de la sucursal anterior.
- Los pagos e inscripciones existentes no deberán perderse.

RF-057 – Registro de Contactos de Emergencia

Descripción: El sistema deberá permitir registrar información de contacto adicional para situaciones de emergencia relacionadas con el estudiante.

Actores: Administrativo, Administrador.

Prioridad: Baja.

Datos mínimos:

- Nombre completo.
- Relación con el estudiante.
- Número de teléfono.
- Observación.

RF-058 – Actualización de Contactos de Emergencia

Descripción: El sistema deberá permitir modificar o actualizar la información de los contactos de emergencia registrados.

Prioridad: Baja.

Criterios de aceptación:

Toda modificación deberá conservar la fecha y el usuario responsable de la operación.

RF-059 – Registro de Baja del Estudiante

Descripción: El sistema deberá permitir registrar la baja de un estudiante cuando este deje de participar en las actividades del instituto.

Actores: Administrador, Administrativo.

Prioridad: Alta.

Datos asociados:

- Fecha de baja.
- Motivo.
- Observaciones.
- Usuario responsable.

Criterios de aceptación:

- La información histórica deberá conservarse.
- El estudiante no deberá ser eliminado físicamente de la base de datos.

RF-060 – Registro de Reingreso

Descripción: El sistema deberá permitir registrar nuevamente a un estudiante que anteriormente haya sido dado de baja.

Actores: Administrativo, Administrador.

Prioridad: Media.

Flujo general:

1. Buscar estudiante inactivo.
2. Seleccionar "Registrar reingreso".
3. Actualizar la información necesaria.
4. Registrar el nuevo periodo.
5. Activar al estudiante.

Criterios de aceptación:

El historial anterior deberá mantenerse y el nuevo proceso deberá registrarse como un evento independiente.

RF-061 – Importación Masiva de Estudiantes

Descripción: El sistema deberá permitir importar múltiples registros de estudiantes mediante un archivo estructurado, reduciendo el tiempo necesario para migrar información existente.

Actores: Administrador.

Prioridad: Media.

Formato inicial propuesto:

- XLSX.
- CSV.

Flujo general:

1. Seleccionar archivo.
2. Validar estructura.
3. Validar datos obligatorios.
4. Detectar duplicados.
5. Mostrar resumen de registros válidos y rechazados.
6. Confirmar importación.
7. Registrar los estudiantes válidos.

Criterios de aceptación:

El sistema no deberá importar registros que presenten errores críticos o duplicidad de CI.

RF-062 – Exportación de Listado de Estudiantes

Descripción: El sistema deberá permitir exportar listados de estudiantes según los filtros seleccionados.

Actores: Administrador, Administrativo, Director.

Prioridad: Media.

Formatos:

- Excel.
- CSV.
- PDF.

Criterios de aceptación:

La información exportada deberá respetar los permisos del usuario que solicita la operación.

RF-063 – Impresión del Expediente

Descripción: El sistema deberá permitir generar una versión imprimible del expediente de un estudiante.

Actores: Administrador, Administrativo, Director.

Prioridad: Media.

El documento podrá incluir:

- Datos personales.
- Tutores.
- Inscripciones.
- Estado académico.
- Información financiera autorizada.
- Observaciones.

RF-064 – Consulta de Historial de Cambios

Descripción: El sistema deberá permitir consultar las modificaciones relevantes realizadas sobre la información del estudiante.

Actores: Administrador, Director.

Prioridad: Media.

El historial deberá indicar:

- Campo modificado.
- Valor anterior.
- Valor nuevo.
- Usuario.
- Fecha.
- Hora.

RF-065 – Alertas de Pagos Pendientes

Descripción: El sistema deberá identificar estudiantes que mantengan saldos pendientes asociados a sus inscripciones.

Actores: Administrador, Administrativo, Caja.

Prioridad: Alta.

Criterios de aceptación:

- El sistema deberá mostrar el saldo pendiente.
- Deberá identificar la inscripción relacionada.
- La información deberá actualizarse después de registrar un nuevo pago.

RF-066 – Alertas de Inasistencia

Descripción: El sistema podrá generar alertas cuando un estudiante alcance un número determinado de inasistencias.

Actores: Administrador, Docente.

Prioridad: Media.

Nota: El número de inasistencias requerido para generar una alerta deberá ser configurable.

RF-067 – Consulta de Estado Académico

Descripción: El sistema deberá permitir consultar el estado académico actual del estudiante.

Podrá incluir:

- Cursos activos.
- Asistencia.
- Calificaciones.
- Actividades pendientes.
- Periodo académico.

RF-068 – Consulta de Estado Administrativo

Descripción: El sistema deberá permitir consultar el estado administrativo del estudiante.

La información podrá incluir:

- Inscripción vigente.
- Estado de pagos.
- Saldo pendiente.
- Recibos emitidos.
- Recibos anulados.

RF-069 – Validación de Información Obligatoria

Descripción: El sistema deberá validar los datos ingresados durante el registro y modificación de estudiantes.

Las validaciones incluirán:

- Campos obligatorios.
- Formato del CI.
- Formato de correo electrónico.
- Formato de números telefónicos.
- Duplicidad de CI.
- Longitud máxima de campos.

Criterios de aceptación:

El sistema deberá informar claramente al usuario qué campo presenta un error antes de permitir guardar la información.

RF-070 – Auditoría del Expediente del Estudiante

Descripción: El sistema deberá mantener un registro de las operaciones relevantes realizadas sobre el expediente del estudiante.

Se registrarán, entre otras:

- Creación del estudiante.
- Modificaciones.
- Cambios de estado.
- Cambios de curso.
- Cambios de área.
- Transferencias de sucursal.
- Bajas.
- Reingresos.

Actores: Sistema.

Prioridad: Alta.

Criterios de aceptación:

La información de auditoría no deberá poder ser modificada por usuarios operativos.

3.6 Requerimientos Funcionales del Módulo de Gestión Académica

Con el **RF-070** queda completado el módulo de Gestión de Estudiantes.

Ahora pasamos al **Módulo de Gestión Académica**, que será fundamental porque aquí vamos a estructurar cómo funciona el instituto: **sucursales** → **áreas** → **cursos** → **materias** → **horarios** → **docentes** → **estudiantes**.

RF-071 – Registrar Sucursal

Descripción: El sistema deberá permitir registrar las diferentes sucursales pertenecientes al instituto.

Actores: Administrador.

Prioridad: Alta.

Datos mínimos:

- Nombre de la sucursal.
- Dirección.
- Teléfono.
- Responsable.
- Estado.

Criterios de aceptación:

No se permitirá registrar dos sucursales con el mismo nombre.

RF-072 – Modificar Sucursal

Descripción: El sistema permitirá actualizar la información de una sucursal registrada.

Actores: Administrador.

Prioridad: Alta.

RF-073 – Activar o Desactivar Sucursal

Descripción: El sistema permitirá activar o desactivar una sucursal.

Una sucursal desactivada no podrá utilizarse para nuevas inscripciones, pero su información histórica deberá mantenerse.

Prioridad: Alta.

RF-074 – Registrar Área Académica

Descripción: El sistema deberá permitir registrar las áreas académicas correspondientes a las diferentes preparaciones ofrecidas por el instituto.

Actores: Administrador, Administrativo autorizado.

Prioridad: Alta.

Datos:

- Nombre del área.
- Descripción.
- Sucursal.
- Estado.

RF-075 – Modificar Área Académica

Descripción: Permitirá actualizar la información de un área académica existente.

Criterios de aceptación:

Las modificaciones no deberán afectar el historial académico de los estudiantes.

RF-076 – Registrar Curso

Descripción: El sistema deberá permitir registrar los cursos que serán ofertados por el instituto.

Datos principales:

- Nombre del curso.
- Área académica.
- Periodo académico.
- Sucursal.
- Descripción.
- Capacidad máxima.
- Estado.

Prioridad: Alta.

RF-077 – Modificar Curso

Descripción: Permitirá modificar la información de un curso existente.

Criterios de aceptación:

Los cambios deberán conservar la información histórica de las inscripciones relacionadas.

RF-078 – Gestionar Capacidad del Curso

Descripción: El sistema deberá permitir establecer la cantidad máxima de estudiantes que pueden ser inscritos en un curso.

Prioridad: Alta.

Regla de negocio:

El sistema no deberá permitir nuevas inscripciones cuando se haya alcanzado la capacidad máxima, salvo que un usuario autorizado habilite una excepción.

RF-079 – Registrar Materia

Descripción: El sistema deberá permitir registrar las materias que forman parte de los diferentes programas académicos.

Datos:

- Nombre.
- Código.
- Área.
- Descripción.
- Estado.

Prioridad: Alta.

RF-080 – Asociar Materia a Curso

Descripción: El sistema deberá permitir definir las materias que serán impartidas dentro de cada curso.

Prioridad: Alta.

Criterios de aceptación:

Una materia podrá formar parte de diferentes cursos cuando corresponda.

RF-081 – Registrar Periodo Académico

Descripción: El sistema deberá permitir registrar los periodos académicos en los cuales se desarrollarán los cursos.

Actores: Administrador.

Prioridad: Alta.

Datos:

- Nombre del periodo.
- Fecha de inicio.
- Fecha de finalización.
- Estado.

Criterios de aceptación: No se permitirá registrar periodos con información obligatoria incompleta.

RF-082 – Activar o Desactivar Periodo Académico

El sistema deberá permitir activar o desactivar un periodo académico.

Un periodo desactivado no podrá utilizarse para nuevas inscripciones, pero deberá conservar su información histórica.

Prioridad: Alta.

RF-083 – Registrar Horario

El sistema deberá permitir registrar los horarios en los que se desarrollarán las actividades académicas.

Datos:

- Día.
- Hora de inicio.
- Hora de finalización.
- Curso.
- Materia.
- Docente.
- Aula.

Prioridad: Alta.

RF-084 – Modificar Horario

El sistema deberá permitir modificar los horarios registrados cuando exista autorización administrativa.

Los cambios deberán conservarse en la bitácora.

Prioridad: Alta.

RF-085 – Validar Conflicto de Horarios

El sistema deberá verificar automáticamente que un docente no sea asignado a dos actividades académicas simultáneas.

También deberá detectar conflictos de horario relacionados con aulas y cursos.

Prioridad: Alta.

RF-086 – Registrar Aula

El sistema deberá permitir registrar las aulas disponibles en cada sucursal.

Datos:

- Nombre o número de aula.
- Sucursal.
- Capacidad.
- Estado.

Prioridad: Media.

RF-087 – Asignar Aula

El sistema deberá permitir asignar un aula a una actividad académica.

El sistema deberá validar que el aula no se encuentre ocupada durante el horario seleccionado.

Prioridad: Media.

RF-088 – Registrar Paralelo

El sistema deberá permitir crear diferentes paralelos para un mismo curso cuando la cantidad de estudiantes lo requiera.

Ejemplo:

- Informática – Paralelo A.
- Informática – Paralelo B.
- Informática – Paralelo C.

Prioridad: Alta.

RF-089 – Asignar Docente a Curso

El sistema deberá permitir asignar uno o varios docentes a los cursos y materias correspondientes.

Prioridad: Alta.

RF-090 – Modificar Asignación Docente

El sistema deberá permitir modificar la asignación de un docente cuando sea necesario.

El cambio deberá conservar el historial de asignaciones anteriores.

RF-091 – Consultar Cursos

El sistema deberá permitir consultar los cursos disponibles utilizando filtros como:

- Área.
- Sucursal.
- Periodo.
- Materia.
- Docente.
- Estado.

Prioridad: Alta.

RF-092 – Consultar Estudiantes por Curso

El sistema deberá permitir consultar el listado de estudiantes inscritos en cada curso o paralelo.

La información podrá incluir:

- Nombre.
- CI.
- Correo.
- Estado.
- Asistencia.
- Situación de pago.

Prioridad: Alta.

RF-093 – Control de Cupos

El sistema deberá controlar automáticamente la cantidad de estudiantes inscritos en cada curso.

Cuando se alcance el límite establecido, el sistema deberá impedir nuevas inscripciones, salvo autorización de un usuario con permisos especiales.

Prioridad: Alta.

RF-094 – Consultar Horarios

El sistema deberá permitir consultar los horarios académicos por:

- Curso.
- Docente.
- Materia.
- Sucursal.
- Aula.
- Día.

Prioridad: Alta.

RF-095 – Calendario Académico

El sistema deberá proporcionar una vista de calendario que permita visualizar las actividades académicas programadas.

Prioridad: Media.

3.7 Requerimientos Funcionales del Módulo de Gestión de Docentes

Ahora comenzamos con el módulo de docentes.

RF-096 – Registrar Docente

El sistema deberá permitir registrar docentes y almacenar su información personal, de contacto y laboral.

Datos principales:

- Nombres.
- Apellidos.
- CI.
- Teléfono.
- Correo electrónico.
- Especialidad.
- Estado.

Prioridad: Alta.

RF-097 – Modificar Docente

Permitirá actualizar la información de los docentes registrados.

Prioridad: Alta.

RF-098 – Consultar Docente

Permitirá consultar el expediente laboral y académico del docente.

Podrá mostrar:

- Datos personales.
- Cursos asignados.
- Materias.
- Horarios.
- Carga horaria.
- Historial de asignaciones.

RF-099 – Buscar Docente

Permitirá buscar docentes mediante:

- CI.
- Nombre.
- Apellido.
- Materia.
- Curso.
- Sucursal.
- Estado.

RF-100 – Activar o Desactivar Docente

Permitirá modificar el estado del docente.

Un docente desactivado no podrá recibir nuevas asignaciones académicas.

RF-101 – Asignar Materia al Docente

Permitirá asociar un docente con una o varias materias de acuerdo con su área de trabajo.

RF-102 – Asignar Curso al Docente

Permitirá asignar docentes a cursos o paralelos específicos.

RF-103 – Registrar Carga Horaria

El sistema deberá registrar la cantidad de horas asignadas a cada docente.

La información podrá incluir:

- Materia.
- Curso.
- Horario.
- Horas semanales.
- Sucursal.
- Periodo.

RF-104 – Consultar Carga Horaria

Permitirá consultar la carga horaria individual de cada docente.

También permitirá consultar la carga horaria total de un periodo.

RF-105 – Validar Conflictos del Docente

El sistema deberá detectar automáticamente cuando un docente tenga dos asignaciones en el mismo horario.

Prioridad: Alta.

RF-106 – Consultar Cursos del Docente

Cada docente podrá consultar desde su cuenta los cursos y materias que tenga asignados.

RF-107 – Consultar Lista de Estudiantes

El docente podrá visualizar los estudiantes pertenecientes a sus cursos.

RF-108 – Registrar Estado Laboral

El sistema permitirá registrar el estado del docente.

Estados propuestos:

- Activo.
- Inactivo.
- Suspendido.
- Retirado.

RF-109 – Registrar Información Salarial

El sistema permitirá registrar la información necesaria para calcular y controlar los pagos correspondientes al personal docente.

Datos posibles:

- Tipo de pago.
- Monto.
- Horas trabajadas.
- Periodo.
- Observaciones.

La información salarial deberá estar restringida al personal autorizado.

RF-110 – Consultar Historial del Docente

El sistema deberá conservar el historial de:

- Cursos impartidos.
- Materias.

- Horarios.
- Carga horaria.
- Periodos.
- Pagos registrados.

RF-111 – Registrar Observaciones del Docente

Los usuarios autorizados podrán registrar observaciones administrativas relacionadas con un docente.

RF-112 – Importación de Docentes

El Administrador podrá importar información de múltiples docentes mediante archivos CSV o Excel.

RF-113 – Exportación de Docentes

El sistema permitirá exportar listados de docentes en formatos como PDF, Excel o CSV.

RF-114 – Reporte de Docentes

El sistema deberá generar reportes de docentes filtrados por:

- Sucursal.
- Área.
- Materia.
- Curso.
- Estado.
- Periodo.

RF-115 – Auditoría del Docente

Las modificaciones importantes realizadas sobre la información del docente deberán quedar registradas en la bitácora.

3.8 Requerimientos Funcionales del Módulo de Inscripciones

Este módulo conectará directamente:

Estudiante → Curso → Periodo → Pago → Recibo

Y posteriormente permitirá alimentar el módulo académico.

RF-116 – Nueva Inscripción

El sistema deberá permitir registrar la inscripción de un estudiante a un curso.

Datos principales:

- Estudiante.
- Área.
- Curso.
- Paralelo.
- Sucursal.
- Horario.
- Fecha de inicio.
- Fecha de finalización.
- Monto.
- Modalidad de pago.
- Descuento, si corresponde.
- Observaciones.

Prioridad: Crítica.

RF-117 – Validar Estudiante

Antes de realizar una inscripción, el sistema deberá verificar que el estudiante exista.

Si no existe, el usuario deberá registrarlo previamente.

RF-118 – Validar Curso

El sistema deberá verificar que:

- El curso esté activo.
- El periodo esté vigente.
- Exista disponibilidad de cupos.
- El horario esté habilitado.

RF-119 – Validar Inscripción Duplicada

El sistema deberá impedir que un estudiante sea inscrito dos veces en el mismo curso y periodo, salvo que exista una autorización especial.

RF-120 – Registrar Fecha de Inicio

El sistema deberá permitir registrar la fecha en que comienza la inscripción del estudiante.

RF-121 – Calcular Fecha de Finalización

El sistema deberá calcular automáticamente la fecha de finalización considerando el periodo de inscripción establecido.

Por ejemplo:

Fecha de inicio: 15/08/2026

Duración: 1 mes

Fecha de finalización: 15/09/2026

La regla exacta para el cálculo deberá ser definida por la administración.

RF-122 – Registrar Modalidad de Pago

El sistema deberá permitir seleccionar la modalidad de pago:

- Pago único.
- Pago en cuotas.
- Pago parcial.

RF-123 – Generar Plan de Pagos

Cuando una inscripción sea realizada mediante cuotas, el sistema deberá generar el calendario correspondiente.

Cada cuota deberá contener:

- Número.
- Fecha límite.
- Monto.
- Estado.
- Fecha de pago.

RF-124 – Aplicar Descuento

El sistema deberá permitir aplicar descuentos autorizados al monto de inscripción.

El descuento podrá expresarse como:

- Monto fijo.
- Porcentaje.

RF-125 – Calcular Total de Inscripción

El sistema deberá calcular automáticamente:

Monto original – Descuento = Total a pagar

El resultado deberá mostrarse antes de confirmar la inscripción.

RF-126 – Registrar Inscripción

Una vez validados todos los datos, el sistema deberá guardar la inscripción y generar un identificador único.

RF-127 – Generar Cuenta por Cobrar

Al registrar una inscripción con saldo pendiente, el sistema deberá generar automáticamente la cuenta por cobrar correspondiente al estudiante.

RF-128 – Generar Recibo de Inscripción

Cuando corresponda, el sistema deberá generar automáticamente el comprobante relacionado con el pago realizado durante la inscripción.

RF-129 – Imprimir Recibo en Impresora Térmica

El sistema deberá permitir imprimir el recibo mediante la impresora térmica configurada.

El recibo deberá contener como mínimo:

- Nombre del instituto.
- Número de recibo.
- Fecha.
- Estudiante.
- CI.
- Curso.
- Concepto.
- Monto.
- Método de pago.
- Usuario que realizó la operación.

RF-130 – Consultar Inscripción

Los usuarios autorizados podrán consultar una inscripción y visualizar toda la información relacionada.

RF-131 – Modificar Inscripción

El sistema permitirá modificar determinados datos de una inscripción cuando el usuario posea autorización.

Los cambios críticos deberán quedar registrados en la auditoría.

RF-132 – Anular Inscripción

El sistema permitirá anular una inscripción cuando exista autorización.

La anulación no deberá eliminar físicamente la información.

RF-133 – Anular Recibo

El sistema deberá permitir anular un recibo previamente emitido.

La operación deberá solicitar:

- Motivo.
- Usuario responsable.
- Confirmación.

El recibo deberá conservarse en el historial con estado **ANULADO**.

RF-134 – Historial de Inscripciones

El sistema deberá conservar todas las inscripciones realizadas por cada estudiante.

RF-135 – Consultar Inscripciones por Periodo

Permitirá consultar las inscripciones correspondientes a un periodo determinado.

RF-136 – Consultar Inscripciones por Curso

Permitirá visualizar todos los estudiantes inscritos en un curso.

RF-137 – Consultar Inscripciones por Sucursal

Permitirá consultar las inscripciones realizadas en cada sucursal.

RF-138 – Consultar Inscripciones por Área

Permitirá obtener información de estudiantes inscritos según el área académica.

RF-139 – Exportar Inscripciones

El sistema permitirá exportar listados de inscripciones en formatos PDF, Excel y CSV.

RF-140 – Auditoría de Inscripciones

Toda operación importante relacionada con una inscripción deberá registrarse en la bitácora.

RF-141 – Estado de la Inscripción

El sistema deberá permitir identificar el estado actual de una inscripción.

Estados:

- Pendiente.
- Activa.
- Finalizada.
- Cancelada.
- Anulada.

Prioridad: Alta.

RF-142 – Cambio de Estado de Inscripción

El sistema deberá actualizar automáticamente el estado de una inscripción de acuerdo con las operaciones realizadas.

Por ejemplo, una inscripción pendiente podrá pasar a activa después de registrarse el pago correspondiente.

RF-143 – Consulta de Estado de Inscripción

Los usuarios autorizados podrán consultar el estado actual y el historial de cambios de una inscripción.

RF-144 – Reimpresión de Recibo

El sistema permitirá reimprimir un recibo previamente generado.

La reimpresión deberá conservar el mismo número de recibo y quedar registrada en la auditoría.

Prioridad: Alta.

RF-145 – Consulta de Recibo

El sistema permitirá buscar y consultar recibos mediante:

- Número de recibo.
- CI del estudiante.
- Nombre.
- Fecha.
- Concepto.
- Usuario.
- Estado.

RF-146 – Historial de Anulación de Recibos

El sistema deberá conservar el historial de recibos anulados.

Deberá mostrar:

- Número de recibo.
- Motivo.
- Usuario.
- Fecha.
- Hora.
- Monto.

RF-147 – Control de Inscripciones por Usuario

El sistema deberá registrar qué usuario realizó cada inscripción.

Esto permitirá determinar quién realizó cada operación administrativa.

RF-148 – Control de Inscripciones por Sucursal

El sistema deberá asociar cada inscripción con la sucursal donde fue realizada.

RF-149 – Control de Inscripciones por Horario

El sistema deberá asociar la inscripción con el horario correspondiente.

Esto permitirá posteriormente generar listas de estudiantes por horario.

RF-150 – Generación de Lista de Inscritos

El sistema deberá generar automáticamente la lista de estudiantes inscritos en cada:

- Curso.
- Materia.
- Paralelo.
- Horario.
- Sucursal.
- Periodo.

RF-151 – Control de Capacidad

El sistema deberá actualizar automáticamente el número de cupos disponibles después de cada inscripción.

Regla:

Cupos disponibles = Capacidad máxima – Estudiantes inscritos activos

RF-152 – Cancelación de Inscripción

El sistema permitirá cancelar una inscripción mediante autorización.

La cancelación deberá registrar el motivo y usuario responsable.

RF-153 – Reversión de Cancelación

Los usuarios autorizados podrán revertir una cancelación cuando corresponda.

La operación deberá quedar registrada en la auditoría.

RF-154 – Reporte de Inscripciones

El sistema deberá generar reportes de las inscripciones realizadas.

Los reportes podrán filtrarse por:

- Fecha.
- Periodo.
- Curso.
- Área.
- Sucursal.
- Horario.
- Estado.

RF-155 – Cierre del Proceso de Inscripción

El sistema deberá permitir cerrar el proceso de inscripción de un periodo determinado.

Una vez cerrado, no podrán realizarse nuevas inscripciones sin autorización administrativa.

3.9 Requerimientos Funcionales del Módulo de Pagos

Este será uno de los módulos **más importantes**, porque el instituto necesita controlar los pagos de las inscripciones y posteriormente los ingresos y egresos.

RF-156 – Registrar Pago

El sistema deberá permitir registrar los pagos realizados por los estudiantes.

Datos principales:

- Estudiante.
- Inscripción.
- Concepto.
- Monto.

- Fecha.
- Método de pago.
- Usuario responsable.

Prioridad: Crítica.

RF-157 – Métodos de Pago

El sistema deberá permitir registrar diferentes métodos de pago.

Inicialmente:

- Efectivo.
- QR.
- Transferencia bancaria.
- Otros métodos autorizados.

RF-158 – Pago Total

El sistema deberá permitir cancelar completamente el monto pendiente de una inscripción mediante un único pago.

RF-159 – Pago Parcial

El sistema deberá permitir registrar pagos parciales.

El sistema deberá actualizar automáticamente el saldo pendiente.

Ejemplo:

Monto: Bs 300

Pago: Bs 100

Saldo: Bs 200

RF-160 – Pago en Cuotas

El sistema deberá permitir registrar pagos correspondientes a un plan de cuotas.

Cada cuota deberá tener su propio estado:

- Pendiente.
- Pagada.
- Vencida.
- Anulada.

RF-161 – Cálculo Automático de Saldo

Después de cada pago, el sistema deberá calcular:

Saldo = Total de la obligación – Total pagado

El saldo deberá actualizarse inmediatamente.

RF-162 – Validación de Monto

El sistema deberá impedir que un pago supere el saldo pendiente, salvo que exista una autorización administrativa para registrar excedentes.

RF-163 – Generar Comprobante de Pago

Después de registrar correctamente un pago, el sistema deberá generar un comprobante.

El comprobante deberá incluir:

- Número.
- Fecha.
- Estudiante.
- CI.
- Concepto.
- Monto.
- Método de pago.
- Usuario.
- Inscripción relacionada.

RF-164 – Imprimir Comprobante

El sistema deberá permitir imprimir el comprobante mediante la impresora térmica configurada.

RF-165 – Reimprimir Comprobante

El sistema deberá permitir reimprimir comprobantes existentes.

La reimpresión no deberá generar un nuevo pago ni un nuevo número de comprobante.

RF-166 – Anular Pago

El sistema permitirá anular un pago únicamente a usuarios autorizados.

Deberá solicitar:

- Motivo.

- Confirmación.
- Usuario responsable.

RF-167 – Reversión de Pago

El sistema permitirá revertir una operación de pago anulada cuando corresponda.

La operación deberá quedar registrada en la auditoría.

RF-168 – Historial de Pagos del Estudiante

El sistema deberá mostrar todos los pagos realizados por un estudiante.

RF-169 – Estado de Cuenta

El sistema deberá generar el estado de cuenta del estudiante mostrando:

- Total.
- Pagado.
- Pendiente.
- Cuotas.
- Vencimientos.

RF-170 – Pagos Vencidos

El sistema deberá identificar automáticamente las cuotas que hayan superado su fecha límite de pago.

RF-171 – Consulta de Pagos por Fecha

Los usuarios autorizados podrán consultar pagos realizados durante un periodo determinado.

RF-172 – Consulta de Pagos por Método

El sistema permitirá filtrar pagos por método:

- Efectivo.
- QR.
- Transferencia.
- Otros.

RF-173 – Consulta de Pagos por Usuario

El sistema permitirá consultar cuánto dinero fue registrado por cada usuario durante un periodo.

Esto será especialmente útil para el control de caja.

RF-174 – Consulta de Pagos por Sucursal

El sistema deberá permitir consultar los pagos realizados en cada sucursal.

3.10 Integración de Pagos mediante QR

Esta sección será especialmente importante porque tu jefe pidió específicamente **pagos mediante QR con confirmación**.

RF-175 – Generar Solicitud de Pago QR

El sistema deberá permitir generar una solicitud de pago mediante QR asociada a una obligación específica del estudiante.

Prioridad: Crítica.

RF-176 – Identificación del Pago QR

Cada solicitud de pago deberá contar con un identificador único que permita relacionarla con:

- Estudiante.
- Inscripción.
- Monto.
- Fecha.
- Concepto.

RF-177 – Verificación del Pago QR

El sistema deberá permitir verificar si el pago realizado mediante QR fue confirmado.

Estados:

- Pendiente.
- Confirmado.
- Rechazado.
- Expirado.

RF-178 – Confirmación Automática del Pago QR

Cuando la integración bancaria o del proveedor lo permita, el sistema deberá actualizar automáticamente el estado del pago después de recibir la confirmación correspondiente.

RF-179 – Confirmación Manual del Pago QR

En caso de que la integración automática no esté disponible, un usuario autorizado podrá verificar y confirmar manualmente el pago.

La confirmación deberá quedar registrada.

RF-180 – Evitar Duplicidad de Pagos QR

El sistema deberá impedir que una misma transacción QR sea registrada dos veces.

Cada transacción deberá poseer un identificador único.

RF-181 – Expiración de Solicitud QR

El sistema deberá permitir establecer un tiempo de vigencia para las solicitudes de pago mediante QR.

Una solicitud vencida no podrá utilizarse para registrar un nuevo pago.

RF-182 – Historial de Transacciones QR

El sistema deberá conservar el historial de las transacciones realizadas mediante QR.

RF-183 – Consulta de Transacciones QR

Los usuarios autorizados podrán consultar las transacciones QR mediante:

- Fecha.
- Estudiante.
- Monto.
- Estado.
- Número de transacción.

RF-184 – Conciliación de Pagos QR

El sistema deberá permitir comparar los pagos registrados internamente con las confirmaciones recibidas del proveedor de pagos.

Prioridad: Alta.

Esto ayudará a detectar:

- Pagos faltantes.
- Pagos duplicados.
- Pagos rechazados.
- Diferencias de montos.

RF-185 – Auditoría de Pagos QR

Toda operación relacionada con una transacción QR deberá quedar registrada en la bitácora.

3.11 Requerimientos Funcionales del Módulo Financiero

Ahora entramos a otro módulo fundamental: **las finanzas generales del instituto.**

Aquí ya no hablamos únicamente del dinero que paga un estudiante. Vamos a controlar:

Ingresos + Egresos = Movimiento financiero del instituto

RF-186 – Registrar Ingreso

El sistema deberá permitir registrar ingresos económicos de la institución.

Los ingresos podrán provenir de:

- Inscripciones.
- Cuotas.
- Otros conceptos autorizados.

Prioridad: Crítica.

RF-187 – Registrar Egreso

El sistema deberá permitir registrar los gastos realizados por la institución.

Ejemplos:

- Servicios.
- Materiales.
- Alquiler.
- Mantenimiento.
- Salarios.
- Otros gastos.

Prioridad: Crítica.

RF-188 – Categorías Financieras

El sistema deberá permitir administrar categorías para clasificar ingresos y egresos.

Ejemplos:

Ingresos

- Inscripción.
- Mensualidad.
- Curso.

Egresos

- Salarios.
- Servicios.
- Material.
- Mantenimiento.

RF-189 – Registrar Movimiento Financiero

Todo ingreso o egreso deberá generar un movimiento financiero con:

- Tipo.
- Categoría.
- Monto.
- Fecha.
- Concepto.
- Usuario.
- Sucursal.
- Observaciones.

RF-190 – Consultar Movimientos

Los usuarios autorizados podrán consultar todos los movimientos financieros.

Filtros:

- Fecha.
- Tipo.
- Categoría.
- Sucursal.
- Usuario.

RF-191 – Anular Movimiento Financiero

Permitirá anular movimientos financieros mediante autorización.

La información original deberá mantenerse en el historial.

RF-192 – Reporte de Ingresos

El sistema deberá generar reportes de ingresos por:

- Día.
- Semana.

- Mes.
- Periodo.
- Sucursal.
- Categoría.

RF-193 – Reporte de Egresos

El sistema deberá generar reportes de egresos utilizando los mismos criterios de filtrado.

RF-194 – Balance Financiero

El sistema deberá calcular el balance financiero:

$$\text{Balance} = \text{Ingresos} - \text{Egresos}$$

El resultado deberá poder consultarse por periodo y sucursal.

RF-195 – Cierre de Caja

El sistema deberá permitir realizar el cierre de caja de cada jornada.

El cierre deberá mostrar:

- Efectivo registrado.
- Pagos QR.
- Transferencias.
- Total de ingresos.
- Anulaciones.
- Diferencias.

RF-196 – Arqueo de Caja

El sistema deberá permitir registrar el monto físico disponible en caja y compararlo con el monto esperado por el sistema.

RF-197 – Registro de Diferencias de Caja

Cuando exista una diferencia entre el monto esperado y el monto real, el sistema deberá registrar:

- Diferencia.
- Motivo.
- Usuario.
- Fecha.
- Observaciones.

RF-198 – Reporte Financiero General

El sistema deberá generar un reporte consolidado de la situación financiera de la institución.

RF-199 – Reporte Financiero por Sucursal

Permitirá consultar los ingresos, egresos y balance de cada sucursal.

RF-200 – Auditoría Financiera

Toda operación financiera deberá ser registrada en la bitácora para garantizar trazabilidad y control administrativo.

RF-201 – Registrar Pago a Docente

Descripción: El sistema deberá permitir registrar los pagos realizados a los docentes por los servicios prestados.

Actores: Administrador, Responsable Financiero.

Prioridad: Alta.

Datos principales:

- Docente.
- Periodo.
- Concepto.
- Horas correspondientes.
- Monto.
- Fecha de pago.
- Método de pago.
- Observaciones.

Criterios de aceptación: Todo pago deberá quedar asociado al docente y generar un movimiento financiero de egreso.

RF-202 – Calcular Pago de Docente

El sistema deberá permitir calcular el monto correspondiente al pago de un docente utilizando la modalidad de remuneración definida por la institución.

El cálculo podrá considerar:

- Horas asignadas.
- Horas efectivamente trabajadas.
- Tarifa por hora.

- Monto fijo acordado.
- Bonificaciones o ajustes autorizados.

Prioridad: Alta.

La fórmula definitiva deberá configurarse de acuerdo con la política salarial utilizada por el instituto.

RF-203 – Registrar Tarifa del Docente

El sistema deberá permitir registrar la tarifa o modalidad de pago correspondiente a cada docente.

La información salarial únicamente será visible para usuarios autorizados.

RF-204 – Consultar Pagos a Docentes

El sistema permitirá consultar el historial de pagos realizados a cada docente.

Se podrá filtrar por:

- Docente.
- Fecha.
- Periodo.
- Sucursal.
- Estado.

RF-205 – Generar Comprobante de Pago a Docente

El sistema deberá permitir generar un comprobante interno correspondiente al pago realizado a un docente.

RF-206 – Estado de Pago del Docente

Cada obligación salarial podrá presentar los siguientes estados:

- Pendiente.
- Calculada.
- Pagada.
- Anulada.

RF-207 – Reporte de Salarios

El sistema deberá generar reportes de pagos realizados a docentes por periodo, sucursal y área académica.

RF-208 – Anulación de Pago a Docente

Los usuarios autorizados podrán anular un registro de pago realizado incorrectamente.

La operación deberá requerir un motivo y quedar registrada en auditoría.

RF-209 – Consolidación Financiera

El sistema deberá consolidar automáticamente los ingresos y egresos registrados por todas las sucursales.

Esto permitirá obtener una visión financiera general de la institución.

RF-210 – Cierre Financiero por Periodo

El sistema deberá permitir cerrar un periodo financiero para evitar modificaciones posteriores no autorizadas.

Las modificaciones excepcionales requerirán permisos especiales y deberán quedar registradas.

3.12 Requerimientos Funcionales del Portal del Docente

El Portal del Docente proporcionará a cada docente un espacio personalizado para administrar las actividades académicas correspondientes exclusivamente a los cursos y materias que le hayan sido asignados.

RF-211 – Acceso al Portal Docente

El sistema deberá proporcionar una interfaz específica para los usuarios que tengan asignado el rol Docente.

Prioridad: Alta.

RF-212 – Dashboard del Docente

El docente deberá visualizar un panel principal con información resumida de:

- Cursos asignados.
- Materias.
- Horarios.
- Estudiantes.
- Actividades pendientes.
- Próximas clases.

RF-213 – Consultar Cursos Asignados

El docente podrá consultar únicamente los cursos que tenga asignados durante el periodo académico correspondiente.

RF-214 – Consultar Materias Asignadas

El sistema deberá mostrar las materias que el docente tenga asignadas.

RF-215 – Consultar Horario del Docente

El docente podrá visualizar su horario organizado por:

- Día.
- Hora.
- Materia.
- Curso.
- Paralelo.
- Sucursal.
- Aula.

RF-216 – Consultar Lista de Estudiantes

El docente podrá consultar los estudiantes inscritos en cada uno de sus cursos.

Prioridad: Alta.

3.13 Requerimientos Funcionales de Asistencia

RF-217 – Registrar Asistencia

El docente deberá poder registrar la asistencia de los estudiantes correspondientes a una sesión de clase.

Estados iniciales:

- Presente.
- Ausente.
- Retraso.
- Justificado.

Prioridad: Crítica.

RF-218 – Crear Registro de Sesión de Clase

Antes de registrar asistencia, el sistema deberá identificar la sesión correspondiente indicando:

- Fecha.
- Curso.
- Materia.
- Docente.
- Horario.

RF-219 – Registro Masivo de Asistencia

El sistema deberá mostrar la lista completa de estudiantes para permitir registrar la asistencia de todo el curso desde una misma interfaz.

RF-220 – Modificar Asistencia

El docente podrá corregir un registro de asistencia dentro del periodo permitido por la institución.

Después de dicho periodo, la modificación requerirá autorización.

RF-221 – Justificar Inasistencia

El sistema deberá permitir registrar una justificación para una ausencia.

Podrá incluir:

- Motivo.
- Observación.
- Documento de respaldo, cuando corresponda.

RF-222 – Consultar Asistencia por Estudiante

El docente podrá consultar el historial de asistencia de un estudiante dentro de sus cursos.

RF-223 – Consultar Asistencia por Curso

El sistema permitirá visualizar la asistencia completa de un curso por fecha.

RF-224 – Calcular Porcentaje de Asistencia

El sistema deberá calcular automáticamente el porcentaje de asistencia del estudiante.

Porcentaje de asistencia = (asistencias válidas / sesiones computables) × 100

La forma de considerar retrasos y justificaciones deberá configurarse según las políticas académicas del instituto.

RF-225 – Alertas por Inasistencia

El sistema deberá identificar estudiantes que superen el límite de inasistencias configurado.

RF-226 – Exportar Asistencia

El docente podrá exportar listas de asistencia autorizadas en PDF o Excel.

3.14 Requerimientos Funcionales de Material Académico

RF-227 – Publicar Material Académico

El docente podrá publicar material correspondiente a sus cursos.

Prioridad: Alta.

RF-228 – Adjuntar Archivos

El sistema permitirá adjuntar archivos académicos.

Podrán admitirse, según configuración:

- PDF.
- Documentos.
- Presentaciones.
- Hojas de cálculo.
- Imágenes.

RF-229 – Publicar Enlaces

El docente podrá compartir enlaces externos relacionados con el contenido académico.

RF-230 – Asociar Material a Materia

Todo material deberá asociarse como mínimo a:

- Curso.
- Materia.
- Docente.
- Periodo.

RF-231 – Editar Material

El docente podrá modificar el título, descripción y archivos del material que haya publicado.

RF-232 – Retirar Material

El docente podrá retirar un material publicado sin eliminar necesariamente su registro histórico.

RF-233 – Consultar Material Publicado

El docente podrá consultar todos los materiales que haya publicado en sus cursos.

RF-234 – Controlar Visibilidad del Material

El docente podrá definir cuándo estará disponible un material para los estudiantes.

Podrá establecer:

- Publicación inmediata.
- Fecha programada.
- Fecha de cierre, cuando corresponda.

3.15 Requerimientos Funcionales de Prácticas y Tareas

RF-235 – Crear Práctica

El docente deberá poder crear una práctica o tarea para los estudiantes.

Datos principales:

- Título.
- Descripción.
- Materia.
- Curso.
- Fecha de publicación.
- Fecha límite.
- Puntaje.
- Archivos adjuntos.

Prioridad: Alta.

RF-236 – Modificar Práctica

El docente podrá modificar una práctica mientras las condiciones académicas lo permitan.

Los cambios importantes realizados después de existir entregas deberán quedar registrados.

RF-237 – Publicar Práctica

El docente podrá publicar una práctica inmediatamente o programar su publicación.

RF-238 – Establecer Fecha Límite

Toda práctica podrá contar con una fecha y hora límite para su entrega.

RF-239 – Consultar Entregas

El docente podrá visualizar los estudiantes que:

- Entregaron.
- No entregaron.
- Entregaron fuera de plazo.

RF-240 – Descargar Entrega

El docente podrá descargar los archivos enviados por los estudiantes para realizar su revisión.

3.16 Requerimientos Funcionales de Evaluación y Calificaciones

RF-241 – Calificar Práctica

El docente podrá asignar una calificación a cada entrega.

Prioridad: Alta.

RF-242 – Registrar Retroalimentación

El docente podrá agregar comentarios o retroalimentación relacionada con la actividad evaluada.

RF-243 – Modificar Calificación

El docente podrá modificar una calificación dentro del periodo habilitado.

Las modificaciones posteriores al cierre deberán requerir autorización.

RF-244 – Registrar Calificación Manual

El sistema permitirá registrar calificaciones correspondientes a actividades realizadas fuera de la plataforma.

Por ejemplo:

- Exámenes presenciales.
- Participación.
- Exposiciones.

- Prácticas presenciales.

RF-245 – Crear Tipo de Evaluación

El sistema permitirá definir tipos de evaluación, tales como:

- Práctica.
- Examen.
- Tarea.
- Participación.
- Proyecto.
- Evaluación final.

RF-246 – Definir Puntaje Máximo

Cada evaluación deberá contar con un puntaje máximo definido.

RF-247 – Definir Ponderación

Cuando corresponda, el sistema permitirá establecer el porcentaje que representa cada tipo de evaluación dentro de la nota final.

RF-248 – Calcular Nota Final

El sistema podrá calcular automáticamente la nota final utilizando las ponderaciones definidas.

Prioridad: Alta.

La fórmula utilizada deberá ser configurable de acuerdo con las reglas académicas del curso.

RF-249 – Consultar Libro de Calificaciones

El docente podrá consultar una vista consolidada de las calificaciones de todos los estudiantes de un curso.

RF-250 – Publicar Calificaciones

El docente deberá poder determinar cuándo las calificaciones estarán disponibles para consulta por parte de los estudiantes.

RF-251 – Cerrar Calificaciones

El sistema permitirá cerrar las calificaciones correspondientes a un curso o periodo.

Después del cierre, las modificaciones requerirán autorización especial.

RF-252 – Historial de Cambios de Calificaciones

Toda modificación de una calificación deberá registrar:

- Calificación anterior.
- Nueva calificación.
- Docente o usuario responsable.
- Fecha.
- Hora.
- Motivo, cuando corresponda.

Prioridad: Alta.

RF-253 – Reporte de Calificaciones

El docente podrá generar reportes de calificaciones por:

- Curso.
- Materia.
- Evaluación.
- Periodo.

RF-254 – Estadísticas Académicas

El sistema podrá calcular indicadores básicos como:

- Promedio del curso.
- Nota máxima.
- Nota mínima.
- Número de estudiantes evaluados.
- Número de estudiantes sin calificación.

RF-255 – Auditoría de Actividades Docentes

El sistema deberá registrar las operaciones académicas relevantes realizadas por los docentes, incluyendo:

- Registro y modificación de asistencia.
- Publicación de materiales.
- Creación y modificación de prácticas.
- Registro y modificación de calificaciones.

Prioridad: Alta.

3.17 Requerimientos Funcionales del Portal del Estudiante

Ahora conectamos todo lo anterior con el estudiante.

RF-256 – Acceso al Portal del Estudiante

Cada estudiante con una cuenta activa deberá poder acceder a un espacio personalizado.

Prioridad: Alta.

RF-257 – Dashboard del Estudiante

El estudiante visualizará información resumida sobre:

- Cursos activos.
- Próximas actividades.
- Material reciente.
- Tareas pendientes.
- Calificaciones recientes.
- Estado de asistencia.
- Estado de pagos.

RF-258 – Consultar Cursos Inscritos

El estudiante podrá visualizar exclusivamente los cursos en los que se encuentre inscrito.

RF-259 – Consultar Horario

El estudiante podrá consultar su horario académico indicando:

- Materia.
- Docente.
- Día.
- Hora.
- Aula.
- Sucursal.

RF-260 – Consultar Material Académico

El estudiante podrá visualizar y descargar el material publicado por sus docentes.

RF-261 – Consultar Prácticas

El estudiante podrá consultar las prácticas asignadas mostrando:

- Materia.
- Descripción.
- Fecha de publicación.
- Fecha límite.

- Puntaje.

RF-262 – Entregar Práctica

El estudiante deberá poder enviar archivos correspondientes a una práctica.

Prioridad: Alta.

El sistema deberá registrar:

- Estudiante.
- Fecha.
- Hora.
- Archivo.
- Estado de entrega.

RF-263 – Identificar Entrega Fuera de Plazo

Cuando una actividad sea enviada después de la fecha límite y la configuración permita entregas tardías, deberá marcarse automáticamente como **ENTREGA TARDÍA**.

RF-264 – Reemplazar Entrega

El estudiante podrá reemplazar una entrega antes de la fecha límite, siempre que el docente permita múltiples envíos.

RF-265 – Consultar Calificaciones

El estudiante podrá consultar únicamente sus propias calificaciones publicadas.

Prioridad: Alta.

RF-266 – Consultar Retroalimentación

El estudiante podrá visualizar los comentarios realizados por el docente sobre sus actividades.

RF-267 – Consultar Asistencia

El estudiante podrá consultar su historial y porcentaje de asistencia.

RF-268 – Consultar Estado de Cuenta

El estudiante podrá consultar:

- Monto total.

- Pagos realizados.
- Saldo pendiente.
- Próximas cuotas.
- Cuotas vencidas.

RF-269 – Consultar Recibos

El estudiante podrá visualizar los comprobantes asociados a sus pagos.

RF-270 – Descargar Comprobante

El estudiante podrá obtener una representación digital de sus comprobantes cuando esta funcionalidad se encuentre habilitada.

RF-271 – Consultar Historial de Inscripciones

El estudiante podrá consultar sus inscripciones actuales y anteriores.

RF-272 – Notificaciones Académicas

El sistema deberá mostrar notificaciones relacionadas con:

- Nuevos materiales.
- Nuevas prácticas.
- Próximos vencimientos.
- Calificaciones publicadas.

RF-273 – Notificaciones Financieras

El sistema podrá mostrar alertas relacionadas con:

- Cuotas próximas.
- Pagos pendientes.
- Cuotas vencidas.
- Pagos registrados.

RF-274 – Actualización de Datos Personales

El estudiante podrá modificar únicamente los datos personales que la institución autorice.

No podrá modificar directamente información crítica como:

- CI.
- Inscripciones.
- Pagos.
- Calificaciones.
- Estado académico.

RF-275 – Seguridad de Información del Estudiante

El sistema deberá garantizar que cada estudiante pueda acceder exclusivamente a su propia información académica y financiera.

Prioridad: Crítica.

3.18 Requerimientos Funcionales del Módulo de Reportes

Este módulo permitirá generar información consolidada de las áreas académica, administrativa y financiera para apoyar el control institucional y la toma de decisiones.

RF-276 – Reporte General de Estudiantes

El sistema deberá generar un reporte general de estudiantes registrados.

Filtros:

- Sucursal.
- Área.
- Curso.
- Periodo.
- Estado.
- Fecha de inscripción.

Prioridad: Alta.

RF-277 – Reporte de Estudiantes por Curso

El sistema deberá generar un listado de estudiantes inscritos en un curso o paralelo determinado.

RF-278 – Reporte de Estudiantes por Área

El sistema deberá generar reportes de estudiantes agrupados por área académica.

RF-279 – Reporte por Sucursal

El sistema deberá permitir comparar la cantidad de estudiantes registrados e inscritos en cada sucursal.

RF-280 – Reporte de Inscripciones

El sistema deberá generar reportes de inscripciones realizadas durante un periodo determinado.

Filtros:

- Fecha.
- Curso.
- Área.
- Horario.
- Sucursal.
- Estado.
- Usuario que realizó la inscripción.

RF-281 – Reporte de Cursos

El sistema deberá generar información relacionada con los cursos ofertados.

Podrá mostrar:

- Curso.
- Área.
- Paralelo.
- Docente.
- Horario.
- Capacidad.
- Cantidad de inscritos.
- Cupos disponibles.

RF-282 – Reporte de Docentes

El sistema deberá generar reportes sobre los docentes registrados.

Filtros:

- Área.
- Materia.
- Curso.
- Sucursal.
- Estado.
- Periodo.

RF-283 – Reporte de Carga Horaria

El sistema deberá generar reportes sobre la carga horaria asignada a los docentes.

Deberá permitir visualizar horas por docente, materia, curso, sucursal y periodo.

RF-284 – Reporte de Asistencia

El sistema deberá generar reportes de asistencia por:

- Estudiante.
- Curso.
- Materia.
- Docente.
- Fecha.
- Periodo.

RF-285 – Reporte de Rendimiento Académico

El sistema deberá generar reportes basados en las calificaciones registradas.

Podrá incluir:

- Promedio por estudiante.
- Promedio por curso.
- Promedio por materia.
- Evaluaciones pendientes.
- Actividades entregadas y no entregadas.

3.19 Reportes Financieros

RF-286 – Reporte de Pagos

El sistema deberá generar reportes de pagos realizados por los estudiantes.

Filtros:

- Fecha.
- Estudiante.
- Curso.
- Sucursal.
- Método de pago.
- Usuario responsable.

RF-287 – Reporte de Deudas

El sistema deberá generar un listado de estudiantes con saldos pendientes.

Podrá mostrar:

- Estudiante.
- Curso.
- Total.
- Pagado.
- Saldo.
- Próximo vencimiento.
- Cuotas vencidas.

Prioridad: Alta.

RF-288 – Reporte de Pagos QR

El sistema deberá generar un reporte de las operaciones realizadas mediante QR.

Se podrán identificar transacciones:

- Confirmadas.
- Pendientes.
- Rechazadas.
- Expiradas.

RF-289 – Reporte de Recibos

El sistema deberá generar reportes de los recibos emitidos.

Filtros:

- Número.
- Fecha.
- Estudiante.
- Usuario.
- Sucursal.
- Estado.

RF-290 – Reporte de Recibos Anulados

El sistema deberá generar un reporte específico de recibos anulados indicando:

- Número de recibo.
- Monto.
- Motivo.
- Usuario responsable.
- Fecha y hora de anulación.

Prioridad: Alta.

RF-291 – Reporte de Ingresos

El sistema deberá generar reportes de ingresos por:

- Día.
- Semana.
- Mes.
- Periodo personalizado.
- Sucursal.
- Categoría.

RF-292 – Reporte de Egresos

El sistema deberá generar reportes de egresos utilizando filtros equivalentes.

RF-293 – Reporte de Balance

El sistema deberá generar un resumen financiero basado en:

$$\text{Balance} = \text{Ingresos} - \text{Egresos}$$

Podrá consultarse de manera general o por sucursal.

RF-294 – Reporte de Caja

El sistema deberá generar reportes correspondientes a la operación de caja.

Podrá mostrar:

- Saldo inicial.
- Ingresos.
- Egresos autorizados.
- Efectivo.
- QR.
- Transferencias.
- Anulaciones.
- Saldo esperado.
- Saldo declarado.
- Diferencia.

RF-295 – Reporte de Pagos a Docentes

El sistema deberá generar reportes de pagos realizados a docentes.

RF-296 – Exportación de Reportes

Los reportes autorizados podrán exportarse, según su naturaleza, en formatos como:

- PDF.
- XLSX.
- CSV.

Prioridad: Alta.

RF-297 – Impresión de Reportes

El sistema deberá proporcionar versiones imprimibles de los reportes autorizados.

3.20 Requerimientos Funcionales del Dashboard

RF-298 – Dashboard General

El sistema deberá proporcionar un panel de indicadores para los usuarios con permisos de supervisión.

Prioridad: Alta.

RF-299 – Indicador de Estudiantes

El Dashboard deberá mostrar como mínimo:

- Estudiantes registrados.
- Estudiantes activos.
- Nuevas inscripciones.
- Distribución por área.

RF-300 – Indicador de Inscripciones

El sistema deberá mostrar la cantidad de inscripciones realizadas durante diferentes periodos.

RF-301 – Indicadores Financieros

El Dashboard deberá presentar información resumida de:

- Ingresos del día.
- Ingresos del mes.
- Egresos.
- Saldo.
- Cuentas pendientes.

La información mostrada dependerá de los permisos del usuario.

RF-302 – Indicadores por Sucursal

El sistema deberá permitir comparar indicadores entre sucursales.

Por ejemplo:

Sucursal	Inscritos	Ingresos	Cursos activos
Ceja	—	—	—
UPEA	—	—	—
Illimani	—	—	—

RF-303 – Indicadores Académicos

El Dashboard podrá presentar:

- Asistencia promedio.
- Actividades pendientes.
- Cursos activos.
- Cantidad de docentes.
- Estudiantes por curso.

RF-304 – Gráficos Estadísticos

El sistema deberá representar determinados indicadores mediante gráficos adecuados.

Los gráficos podrán mostrar:

- Evolución de inscripciones.
- Evolución de ingresos.
- Distribución de estudiantes.
- Comparaciones entre sucursales.

RF-305 – Filtrado del Dashboard

Los usuarios autorizados podrán filtrar los indicadores por:

- Fecha.
- Periodo.
- Sucursal.
- Área.

3.21 Requerimientos Funcionales de Notificaciones

RF-306 – Centro de Notificaciones

El sistema deberá proporcionar un espacio donde cada usuario pueda consultar las notificaciones relacionadas con su cuenta.

RF-307 – Notificaciones Académicas

El sistema deberá generar notificaciones ante eventos académicos relevantes, como:

- Nueva práctica.
- Nuevo material.
- Publicación de calificación.
- Modificación importante de una actividad.

RF-308 – Notificaciones Financieras

El sistema podrá generar avisos relacionados con:

- Cuotas próximas.
- Pagos vencidos.
- Pagos confirmados.
- Anulaciones autorizadas.

RF-309 – Marcar Notificación como Leída

El usuario podrá identificar las notificaciones que ya haya revisado.

3.22 Requerimientos Funcionales de Configuración

RF-310 – Configuración Institucional

El Administrador podrá configurar información general del instituto, como:

- Nombre.
- Logotipo.
- Datos de contacto.
- Información utilizada en comprobantes.
- Parámetros institucionales.

RF-311 – Configuración de Sucursales

El Administrador podrá establecer parámetros particulares correspondientes a cada sucursal.

RF-312 – Configuración del Recibo Térmico

El sistema deberá permitir configurar los datos y formato lógico del comprobante utilizado para la impresión térmica.

Podrá incluir:

- Logotipo, cuando el equipo lo permita.
- Nombre del instituto.
- Datos institucionales.
- Pie del recibo.
- Información de contacto.

Prioridad: Alta.

Las dimensiones y características definitivas deberán validarse con el modelo real de impresora térmica utilizada por el instituto.

RF-313 – Configuración de Métodos de Pago

El Administrador podrá habilitar o deshabilitar métodos de pago disponibles en el sistema.

RF-314 – Configuración de Parámetros Académicos

El sistema permitirá configurar determinados valores institucionales, tales como:

- Estados de asistencia.
- Límites de inasistencia.
- Escala de calificación.
- Periodos.
- Reglas de entrega.

3.23 Requerimientos Funcionales de Respaldo y Recuperación

RF-315 – Generación de Respaldos

El sistema deberá permitir generar respaldos de la información institucional de acuerdo con la estrategia de infraestructura definida.

Prioridad: Crítica.

RF-316 – Programación de Respaldos

Cuando la infraestructura lo permita, el sistema deberá permitir establecer una política automatizada de respaldos.

La periodicidad deberá ser configurable.

RF-317 – Registro de Respaldos

El sistema deberá conservar información sobre los respaldos ejecutados, incluyendo:

- Fecha.
- Hora.
- Estado.
- Resultado.
- Responsable o proceso ejecutor.

RF-318 – Recuperación de Información

Los responsables técnicos autorizados deberán disponer de un procedimiento para restaurar información a partir de un respaldo válido.

Prioridad: Crítica.

La restauración deberá ser una operación restringida y auditada.

3.24 Auditoría General

RF-319 – Bitácora General del Sistema

El sistema deberá mantener una bitácora central de operaciones críticas.

Podrá registrar:

- Usuario.
- Fecha.
- Hora.
- Acción.
- Módulo.
- Registro afectado.
- Resultado de la operación.
- Dirección IP cuando técnicamente corresponda.

Prioridad: Crítica.

RF-320 – Consulta de Auditoría

Los usuarios expresamente autorizados podrán consultar la bitácora mediante filtros por:

- Usuario.
- Fecha.
- Módulo.
- Acción.
- Tipo de operación.

La información de auditoría no deberá poder ser modificada por usuarios operativos.

Prioridad: Alta.

3.25 Resumen de Requerimientos Funcionales

Con el **RF-320** podemos cerrar formalmente la primera versión del Capítulo 3.

Código	Módulo
RF-001 – RF-020	Usuarios y autenticación
RF-021 – RF-035	Roles y permisos
RF-036 – RF-070	Estudiantes
RF-071 – RF-095	Gestión académica
RF-096 – RF-115	Docentes
RF-116 – RF-155	Inscripciones y recibos
RF-156 – RF-185	Pagos y QR
RF-186 – RF-210	Finanzas y salarios

RF-211 – RF-226	Portal docente y asistencia
RF-227 – RF-234	Material académico
RF-235 – RF-240	Prácticas
RF-241 – RF-255	Calificaciones
RF-256 – RF-275	Portal estudiante
RF-276 – RF-297	Reportes
RF-298 – RF-305	Dashboard
RF-306 – RF-309	Notificaciones
RF-310 – RF-314	Configuración
RF-315 – RF-318	Respaldos
RF-319 – RF-320	Auditoría

Total inicial: 320 requerimientos funcionales.

Una aclaración importante: que tengamos 320 RF **no significa que sean 320 pantallas ni 320 tareas de programación independientes**. Muchos pertenecen a una misma funcionalidad y posteriormente los agruparemos en historias de usuario/sprints.

Cierre del Capítulo 3

Los requerimientos funcionales definidos en el presente capítulo establecen las capacidades que deberá proporcionar el Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO) para satisfacer las necesidades identificadas durante el levantamiento de información realizado en la institución. Estos requerimientos abarcan los procesos de gestión de usuarios, estudiantes, docentes, estructura académica, inscripciones, pagos, emisión y anulación de recibos, integración con pagos mediante código QR, gestión financiera, asistencia, materiales académicos, prácticas, calificaciones, reportes, indicadores, configuración, respaldos y auditoría. La identificación individual mediante códigos RF permitirá mantener la trazabilidad entre los requerimientos, el diseño, las actividades de desarrollo y las pruebas posteriores. Asimismo, los requerimientos establecidos constituyen una línea base inicial y podrán ser refinados mediante un proceso controlado de cambios cuando se identifiquen nuevas necesidades o modificaciones durante las revisiones con la institución.

CAPÍTULO 4 REQUERIMIENTOS NO FUNCIONALES

4.1 Introducción

Los requerimientos no funcionales establecen las características de calidad, restricciones técnicas y condiciones operativas que deberá cumplir el Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO).

A diferencia de los requerimientos funcionales, que especifican **qué deberá hacer el sistema**, los requerimientos no funcionales determinan **cómo deberá comportarse el sistema** durante su funcionamiento.

Estos requerimientos consideran aspectos relacionados con seguridad, rendimiento, disponibilidad, usabilidad, compatibilidad, confiabilidad, mantenibilidad, escalabilidad, integridad de la información, respaldo y recuperación.

Su definición permitirá establecer criterios objetivos para evaluar la calidad del software durante las etapas de desarrollo, pruebas, implementación y mantenimiento.

Para facilitar su seguimiento y trazabilidad, cada requerimiento será identificado mediante el prefijo **RNF**, seguido de una numeración correlativa.

4.2 Clasificación de los Requerimientos No Funcionales

Los requerimientos no funcionales del SIGA se organizarán en las siguientes categorías:

Categoría	Descripción
Seguridad	Protección de cuentas, permisos, datos y operaciones
Rendimiento	Velocidad de respuesta y capacidad de procesamiento
Disponibilidad	Acceso y continuidad del servicio
Usabilidad	Facilidad de aprendizaje y utilización
Compatibilidad	Funcionamiento en navegadores, dispositivos y periféricos
Confiabilidad	Funcionamiento consistente y manejo de fallos
Integridad de datos	Protección contra pérdida o inconsistencia de información
Mantenibilidad	Facilidad para corregir y ampliar el software
Escalabilidad	Capacidad para soportar crecimiento
Respaldo y recuperación	Protección y restauración de información
Auditoría	Trazabilidad de operaciones críticas
Calidad del software	Características generales esperadas del producto

4.3 Requerimientos de Seguridad

La seguridad constituye un aspecto crítico debido a que SIGA administrará información personal, académica y financiera de estudiantes, docentes y personal institucional.

RNF-001 – Autenticación Obligatoria

El sistema deberá requerir autenticación para acceder a cualquier funcionalidad que contenga información institucional no pública.

Prioridad: Crítica.

Criterio de verificación: Un usuario no autenticado no deberá poder acceder directamente a rutas o recursos protegidos.

RNF-002 – Almacenamiento Seguro de Contraseñas

Las contraseñas no deberán almacenarse en texto plano ni mediante cifrado reversible.

El sistema deberá utilizar funciones criptográficas adecuadas para almacenamiento de contraseñas, utilizando mecanismos de *hashing* y *salt*.

Prioridad: Crítica.

RNF-003 – Control de Acceso

El sistema deberá implementar un mecanismo de autorización basado en roles y permisos.

Cada solicitud a una funcionalidad protegida deberá validar que el usuario tenga autorización suficiente.

Prioridad: Crítica.

RNF-004 – Principio de Mínimo Privilegio

Los usuarios deberán disponer únicamente de los permisos necesarios para desempeñar las funciones correspondientes a su rol.

Por ejemplo, un docente no deberá poder acceder a información financiera institucional y un estudiante no deberá acceder a expedientes de otros estudiantes.

RNF-005 – Protección de Información Financiera

La información financiera deberá ser accesible únicamente para usuarios expresamente autorizados.

Las operaciones críticas como anulaciones, cierres de caja, modificación de pagos y movimientos financieros deberán requerir permisos específicos.

Prioridad: Crítica.

RNF-006 – Protección de Sesiones

Las sesiones de usuario deberán administrarse mediante mecanismos seguros.

El sistema deberá impedir la reutilización de sesiones invalidadas después de cerrar sesión.

RNF-007 – Expiración de Sesión

El sistema deberá cerrar o invalidar las sesiones después de un periodo configurable de inactividad.

Antes de finalizar una sesión activa, podrá notificarse al usuario cuando técnicamente corresponda.

RNF-008 – Protección contra Intentos de Acceso

El sistema deberá implementar controles frente a múltiples intentos fallidos consecutivos de autenticación.

El número máximo de intentos y el tiempo de bloqueo deberán ser configurables.

RNF-009 – Validación de Datos de Entrada

Todos los datos enviados al sistema deberán ser validados antes de su procesamiento.

La validación deberá realizarse tanto en la interfaz cuando corresponda como obligatoriamente en el servidor.

Prioridad: Crítica.

RNF-010 – Protección contra Inyección

Las operaciones relacionadas con la base de datos deberán implementarse utilizando mecanismos que reduzcan el riesgo de ataques de inyección, evitando la construcción insegura de consultas mediante datos introducidos directamente por los usuarios.

RNF-011 – Protección contra Vulnerabilidades Web

Durante el desarrollo deberán considerarse controles frente a vulnerabilidades comunes de aplicaciones web, incluyendo, según corresponda:

- Inyección.
- Cross-Site Scripting (XSS).
- Cross-Site Request Forgery (CSRF).
- Control de acceso incorrecto.
- Exposición indebida de información.
- Carga insegura de archivos.

RNF-012 – Comunicación Segura

En el entorno de producción, las comunicaciones entre los usuarios y el servidor deberán realizarse utilizando conexiones cifradas mediante **HTTPS/TLS**.

Prioridad: Crítica.

RNF-013 – Auditoría de Operaciones Críticas

Las operaciones consideradas críticas deberán generar registros de auditoría.

Entre ellas:

- Inicio de sesión.
- Cambios de permisos.
- Inscripciones.
- Pagos.
- Anulaciones.
- Modificaciones financieras.
- Cambios de calificaciones.
- Cierres de caja.

RNF-014 – Inmutabilidad de Auditoría

Los usuarios operativos no deberán poder modificar o eliminar los registros de auditoría generados por el sistema.

RNF-015 – Seguridad de Archivos

Los archivos cargados por docentes, estudiantes o administrativos deberán ser validados antes de almacenarse.

El sistema deberá controlar como mínimo:

- Tipo de archivo permitido.
- Tamaño máximo.
- Nombre seguro del archivo.
- Asociación con el usuario y recurso correspondiente.

4.4 Requerimientos de Rendimiento

RNF-016 – Tiempo de Respuesta

En condiciones normales de operación, las operaciones habituales como consultas, búsquedas, navegación y registros simples deberán procurar tiempos de respuesta inferiores a **3 segundos**, excluyendo factores externos como velocidad de Internet, servicios de terceros o generación de reportes complejos.

Prioridad: Alta.

RNF-017 – Operaciones Críticas

Las operaciones críticas, como el registro de una inscripción o un pago, deberán procesarse evitando duplicidades incluso cuando el usuario intente ejecutar accidentalmente la misma operación más de una vez.

RNF-018 – Búsqueda de Información

Las consultas frecuentes de estudiantes, pagos, recibos, docentes e inscripciones deberán optimizarse mediante índices y mecanismos apropiados de acceso a datos.

RNF-019 – Generación de Reportes

Los reportes de volumen habitual deberán generarse en un tiempo razonable. Como objetivo inicial, los reportes comunes deberán intentar completarse en menos de **10 segundos** bajo condiciones normales.

Los reportes de gran volumen podrán procesarse mediante mecanismos diferentes cuando sea necesario.

RNF-020 – Usuarios Concurrentes

La primera versión del sistema deberá diseñarse para soportar, como objetivo inicial, al menos **50 usuarios concurrentes** realizando operaciones habituales sin una degradación significativa del servicio.

Esta capacidad deberá poder ampliarse posteriormente mediante mejoras de infraestructura.

4.5 Requerimientos de Disponibilidad y Confiabilidad

RNF-021 – Disponibilidad del Sistema

El sistema deberá estar disponible durante los horarios operativos definidos por la institución, exceptuando periodos programados de mantenimiento o interrupciones atribuibles a servicios externos.

RNF-022 – Manejo de Errores

Los errores internos no deberán provocar la pérdida de información previamente confirmada.

El sistema deberá presentar mensajes comprensibles al usuario sin revelar información técnica sensible.

RNF-023 – Transacciones Críticas

Las operaciones financieras deberán ejecutarse de manera transaccional.

Una operación deberá completarse correctamente en su totalidad o revertirse cuando se produzca un error.

Ejemplo:

Al registrar un pago:

Pago → movimiento financiero → recibo

Si una parte crítica falla, el sistema no deberá dejar registros financieros inconsistentes.

Prioridad: Crítica.

RNF-024 – Integridad Referencial

La base de datos deberá aplicar mecanismos que garanticen la consistencia de las relaciones entre estudiantes, inscripciones, pagos, cursos, docentes y demás entidades relacionadas.

RNF-025 – Conservación del Historial

La eliminación de información relacionada con operaciones académicas o financieras no deberá provocar la pérdida de información histórica necesaria para auditoría.

Se utilizarán mecanismos como estados, anulaciones o eliminación lógica cuando corresponda.

4.6 Requerimientos de Usabilidad

RNF-026 – Interfaz Intuitiva

La interfaz deberá diseñarse de manera que las operaciones principales puedan ser comprendidas por usuarios con conocimientos básicos de informática.

RNF-027 – Consistencia Visual

Los módulos deberán mantener criterios visuales consistentes respecto a:

- Menús.
- Botones.
- Formularios.
- Tablas.
- Mensajes.
- Iconografía.
- Navegación.

RNF-028 – Mensajes Comprensibles

Los mensajes de error, advertencia y confirmación deberán utilizar lenguaje comprensible para el usuario final.

En lugar de presentar errores técnicos internos, el sistema deberá indicar qué ocurrió y, cuando sea posible, cómo corregirlo.

RNF-029 – Confirmación de Operaciones Críticas

El sistema deberá solicitar confirmación antes de ejecutar acciones sensibles como:

- Anular recibos.

- Anular pagos.
- Cancelar inscripciones.
- Cerrar caja.
- Modificar permisos.
- Registrar determinadas operaciones financieras.

RNF-030 – Diseño Adaptable

La interfaz deberá adaptarse adecuadamente a diferentes resoluciones de pantalla, priorizando computadoras y considerando el acceso mediante dispositivos móviles para los portales de estudiantes y docentes.

4.7 Requerimientos de Compatibilidad

RNF-031 – Compatibilidad con Navegadores

El sistema deberá ser compatible con versiones modernas y soportadas de los principales navegadores utilizados por la institución, principalmente:

- Google Chrome.
- Microsoft Edge.
- Mozilla Firefox.

RNF-032 – Compatibilidad con Impresora Térmica

La emisión de recibos deberá adaptarse al modelo y ancho de papel de la impresora térmica seleccionada por la institución.

La compatibilidad definitiva será validada mediante pruebas con el dispositivo físico.

Prioridad: Crítica.

RNF-033 – Integración con Servicios QR

La arquitectura deberá permitir integrar servicios externos para generación y/o confirmación de pagos QR mediante mecanismos proporcionados por la entidad financiera o proveedor seleccionado.

La disponibilidad de confirmación automática dependerá de las capacidades técnicas proporcionadas por dicho proveedor.

4.8 Requerimientos de Escalabilidad

RNF-034 – Crecimiento de Estudiantes

La arquitectura y la base de datos deberán permitir un crecimiento considerable respecto al volumen inicial de estudiantes sin requerir rediseñar completamente el sistema.

RNF-035 – Crecimiento de Sucursales

El sistema deberá permitir incorporar nuevas sucursales mediante configuración y datos, evitando cambios estructurales importantes en el software.

RNF-036 – Crecimiento Académico

El sistema deberá permitir incorporar nuevas:

- Áreas.
- Materias.
- Cursos.
- Paralelos.
- Docentes.
- Periodos académicos.

RNF-037 – Arquitectura Modular

El sistema deberá organizarse de forma modular para permitir desarrollar, modificar o ampliar funcionalidades reduciendo el impacto sobre otros componentes.

4.9 Requerimientos de Mantenibilidad

RNF-038 – Organización del Código

El código fuente deberá mantener una estructura clara y coherente con la arquitectura tecnológica definida para el proyecto.

RNF-039 – Control de Versiones

El código fuente deberá mantenerse bajo un sistema de control de versiones basado en Git.

Las modificaciones relevantes deberán registrarse mediante *commits* identificables.

RNF-040 – Separación de Configuración

Las configuraciones sensibles o dependientes del entorno no deberán mantenerse directamente incrustadas en el código fuente cuando puedan gestionarse mediante variables de entorno u otros mecanismos apropiados.

RNF-041 – Documentación Técnica

Los componentes críticos del sistema deberán contar con documentación suficiente para facilitar su mantenimiento y futura ampliación.

RNF-042 – Migraciones de Base de Datos

Las modificaciones estructurales de la base de datos deberán realizarse mediante mecanismos controlados y reproducibles, evitando cambios manuales no documentados en producción.

4.10 Requerimientos de Respaldo y Recuperación

RNF-043 – Respaldo de Base de Datos

La información institucional deberá contar con una estrategia periódica de respaldo.

Prioridad: Crítica.

RNF-044 – Automatización de Respaldos

Cuando la infraestructura seleccionada lo permita, los respaldos deberán ejecutarse automáticamente según una frecuencia definida por la administración técnica.

RNF-045 – Conservación de Respaldos

Se deberá establecer una política de retención que permita conservar múltiples versiones recientes de los respaldos y evitar depender de una única copia.

RNF-046 – Verificación de Respaldos

Los respaldos deberán verificarse periódicamente para confirmar que pueden utilizarse durante un proceso de recuperación.

Un respaldo generado pero nunca probado no deberá considerarse suficiente como mecanismo de recuperación.

RNF-047 – Recuperación ante Fallos

La infraestructura deberá disponer de un procedimiento documentado para recuperar la información institucional después de una pérdida o fallo grave.

4.11 Requerimientos de Integridad de Datos

RNF-048 – Identificadores Únicos

Las entidades principales deberán disponer de identificadores únicos internos que permitan mantener relaciones consistentes independientemente de cambios en sus datos descriptivos.

RNF-049 – Operaciones Financieras Consistentes

No deberán existir pagos sin una relación válida con el concepto u obligación correspondiente, excepto los tipos de ingresos independientes expresamente definidos por la institución.

RNF-050 – Numeración de Recibos

Cada recibo deberá poseer un identificador único y trazable.

La anulación o reimpresión no deberá generar pérdida de la referencia histórica del comprobante original.

RNF-051 – Fechas y Horas

El sistema deberá registrar fechas y horas de operaciones críticas de forma consistente para permitir reconstruir cronológicamente los acontecimientos registrados.

RNF-052 – Precisión Monetaria

Los valores monetarios deberán almacenarse utilizando tipos de datos de precisión decimal apropiados para operaciones financieras, evitando mecanismos que puedan producir errores de redondeo acumulativo.

4.12 Requerimientos de Calidad

RNF-053 – Adecuación Funcional

Las funcionalidades implementadas deberán corresponder con los requerimientos aprobados en el presente SRS y sus modificaciones posteriores autorizadas.

RNF-054 – Pruebas

Los módulos críticos deberán ser sometidos a pruebas antes de ser desplegados en el entorno de producción.

Se priorizarán especialmente:

- Inscripciones.
- Pagos.
- Recibos.
- Anulaciones.
- QR.
- Finanzas.
- Roles y permisos.
- Calificaciones.

RNF-055 – Ambiente de Pruebas

Siempre que la infraestructura disponible lo permita, las nuevas funcionalidades deberán validarse en un entorno diferente al utilizado en producción antes de su liberación.

RNF-056 – Registro de Incidencias

Los errores identificados durante el desarrollo y operación deberán poder documentarse para facilitar su seguimiento, corrección y validación posterior.

RNF-057 – Trazabilidad

Los requerimientos funcionales y no funcionales deberán poder relacionarse con las funcionalidades desarrolladas y los casos de prueba correspondientes.

Esto permitirá verificar posteriormente si cada requisito fue correctamente implementado.

4.13 Cierre del Capítulo

Los requerimientos no funcionales definidos en este capítulo establecen los criterios de calidad y las condiciones técnicas que deberá satisfacer el Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO). Su cumplimiento permitirá que las funcionalidades especificadas en el Capítulo 3 sean implementadas bajo condiciones adecuadas de seguridad, rendimiento, confiabilidad, usabilidad, compatibilidad, mantenibilidad y escalabilidad. Asimismo, estos requisitos servirán como criterios de referencia durante las etapas de diseño, desarrollo, pruebas y puesta en producción, permitiendo evaluar no solamente si el sistema realiza las operaciones solicitadas, sino también si las ejecuta con el nivel de calidad requerido por la institución.

Con esto tenemos una **primera versión del Capítulo 4 con 57 RNF**. Y ahora sí tenemos algo muy importante para tu reunión y para tu trabajo: **320 RF + 57 RNF = 377 requisitos identificados**. No significa que debas desarrollar 377 cosas separadas; posteriormente los agruparemos en módulos, historias de usuario y sprints.

CAPÍTULO 5 REGLAS DE NEGOCIO

5.1 Introducción

Las reglas de negocio establecen las condiciones, restricciones, políticas y criterios que regulan el funcionamiento del Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA – SUMO). Estas reglas representan la forma en que la institución desarrolla sus procesos académicos, administrativos y financieros, y determinan cómo deberá responder el sistema ante determinadas operaciones.

A diferencia de los requerimientos funcionales, que describen las funcionalidades que deberá proporcionar el sistema, las reglas de negocio determinan **las condiciones bajo las cuales dichas funcionalidades podrán ejecutarse**.

Para facilitar su identificación, trazabilidad y posterior implementación, cada regla de negocio será identificada mediante el código **RN-XXX**.

Las reglas establecidas en este capítulo constituyen una línea base inicial y podrán ser modificadas o ampliadas mediante la validación de los responsables institucionales.

5.2 Reglas Generales

RN-001 – Identificación Única del Estudiante

Cada estudiante deberá contar con un identificador interno único generado por el sistema.

El número de Cédula de Identidad será utilizado como uno de los principales datos para identificar al estudiante y evitar registros duplicados.

RN-002 – Unicidad de Cédula de Identidad

No deberá existir más de un estudiante activo registrado con el mismo número de Cédula de Identidad.

Cuando se intente registrar un CI existente, el sistema deberá mostrar el registro encontrado antes de permitir continuar.

RN-003 – Conservación del Historial

Los registros relacionados con operaciones académicas, administrativas o financieras no deberán eliminarse físicamente cuando formen parte del historial institucional.

Cuando corresponda, deberán utilizarse estados como:

- Activo.
- Inactivo.
- Cancelado.
- Anulado.
- Finalizado.

RN-004 – Registro de Operaciones

Toda operación crítica deberá registrar como mínimo:

- Usuario responsable.
- Fecha.

- Hora.
- Tipo de operación.
- Registro afectado.

RN-005 – Acceso según Rol

Cada usuario únicamente podrá realizar las operaciones correspondientes a los roles y permisos que tenga asignados.

5.3 Reglas de Inscripción

RN-006 – Estudiante Registrado

Para realizar una inscripción, el estudiante deberá encontrarse previamente registrado en el sistema.

Si el estudiante no existe, deberá completarse primero su registro.

RN-007 – Curso Activo

Únicamente podrán realizarse nuevas inscripciones en cursos que se encuentren habilitados para recibir estudiantes.

RN-008 – Selección de Sucursal

Toda inscripción deberá estar asociada a la sucursal correspondiente.

Esto permitirá controlar estudiantes, pagos, cursos e ingresos por cada sede institucional.

RN-009 – Selección de Área

Toda inscripción deberá estar asociada al área académica correspondiente al curso seleccionado.

RN-010 – Asignación de Horario

Toda inscripción deberá contar con un horario válido y habilitado.

RN-011 – Control de Cupos

El sistema no deberá permitir nuevas inscripciones cuando un curso haya alcanzado su capacidad máxima.

Los usuarios con permisos especiales podrán autorizar una excepción cuando la institución así lo determine.

RN-012 – Inscripción Duplicada

Un estudiante no podrá mantener dos inscripciones activas idénticas para el mismo curso, paralelo y periodo, salvo autorización administrativa.

RN-013 – Fecha de Inicio

Toda inscripción deberá registrar una fecha de inicio.

Esta fecha será utilizada como referencia para determinar la vigencia correspondiente.

RN-014 – Fecha de Finalización

Cuando la duración contratada sea de un mes, el sistema deberá calcular automáticamente la fecha de finalización tomando como referencia la fecha de inicio y agregando **un mes calendario**.

Por ejemplo:

Inicio:	18/08/2026
Finalización:	18/09/2026

En casos donde el día equivalente no exista en el mes de destino, deberá aplicarse una regla previamente definida y validada por la institución.

RN-015 – Modificación de Inscripción

Una inscripción podrá modificarse únicamente por usuarios autorizados.

Los cambios relevantes deberán quedar registrados en la bitácora.

RN-016 – Cancelación de Inscripción

Una inscripción cancelada no deberá eliminarse de la base de datos.

Deberá cambiar su estado a **CANCELADA** y conservar:

- Fecha de cancelación.
- Motivo.
- Usuario responsable.

RN-017 – Historial de Inscripciones

Todas las inscripciones realizadas por un estudiante deberán conservarse como parte de su historial.

5.4 Reglas de Pagos

RN-018 – Obligación de Pago

Toda inscripción que implique un costo deberá generar una obligación financiera asociada al estudiante.

RN-019 – Modalidades de Pago

La institución podrá manejar diferentes modalidades de pago:

- Pago único.
- Pago parcial.
- Pago mediante cuotas.

La modalidad deberá establecerse durante la inscripción.

RN-020 – Monto Total

Toda inscripción deberá registrar el monto acordado que el estudiante deberá pagar.

El monto final deberá considerar los descuentos autorizados cuando corresponda.

RN-021 – Pagos Parciales

Cuando se permita realizar pagos parciales, cada pago deberá reducir automáticamente el saldo pendiente.

La relación básica será:

Saldo pendiente = Total a pagar – Total de pagos válidos

RN-022 – Pago Completo

Cuando el saldo pendiente llegue a cero, la obligación financiera deberá pasar automáticamente al estado **PAGADA**.

RN-023 – Saldo Pendiente

Mientras exista un monto pendiente, la obligación deberá mantenerse en estado pendiente o parcialmente pagada, según corresponda.

RN-024 – Pagos en Cuotas

Cuando se seleccione pago en cuotas, el sistema deberá registrar como mínimo:

- Número de cuota.
- Monto.
- Fecha límite.
- Estado.

RN-025 – Cuota Vencida

Una cuota deberá considerarse vencida cuando su fecha límite haya transcurrido y mantenga un saldo pendiente.

RN-026 – Métodos de Pago

Cada pago deberá identificar el método utilizado.

Inicialmente se contemplan:

- Efectivo.
- QR.
- Transferencia.
- Otros métodos autorizados.

RN-027 – Pago Mayor al Saldo

De manera predeterminada, el sistema no deberá permitir registrar un pago superior al saldo pendiente.

Cualquier tratamiento de pagos excedentes deberá definirse expresamente como una política institucional.

5.5 Reglas de Recibos

RN-028 – Generación de Recibo

Todo pago confirmado deberá permitir generar el comprobante correspondiente.

RN-029 – Numeración Única

Cada recibo deberá poseer un número único que permita identificarlo dentro del sistema.

Un número de recibo no podrá reutilizarse para otro comprobante.

RN-030 – Asociación del Recibo

Todo recibo deberá encontrarse asociado al pago que originó su emisión.

RN-031 – Reimpresión

La reimpresión de un recibo no deberá generar:

- Un nuevo pago.
- Un nuevo ingreso.
- Un nuevo número de recibo.

La operación deberá utilizar el comprobante original.

RN-032 – Anulación de Recibo

Un recibo anulado deberá permanecer registrado con estado **ANULADO**.

No deberá eliminarse físicamente.

RN-033 – Motivo de Anulación

Para anular un recibo será obligatorio registrar:

- Motivo.
- Usuario responsable.
- Fecha.
- Hora.

RN-034 – Permisos de Anulación

Únicamente usuarios expresamente autorizados podrán anular recibos.

RN-035 – Consecuencia Financiera de la Anulación

Cuando la anulación de un recibo implique también la anulación del pago relacionado, el sistema deberá revertir los efectos financieros correspondientes mediante una operación trazable.

No se deberán eliminar silenciosamente los movimientos originales.

5.6 Reglas para Pagos mediante QR

RN-036 – Asociación de QR

Toda solicitud de pago mediante QR deberá estar relacionada con una obligación financiera determinada.

RN-037 – Monto del QR

El monto solicitado mediante QR deberá corresponder al monto que se desea cancelar.

RN-038 – Identificador de Transacción

Cada operación QR deberá contar con un identificador que permita evitar duplicidades.

RN-039 – Estados de Transacción QR

Las transacciones podrán manejar, como mínimo, los estados:

- Pendiente.
- Confirmada.
- Rechazada.
- Expirada.

RN-040 – Confirmación de Pago QR

Un pago QR únicamente deberá considerarse efectivamente pagado cuando exista confirmación válida del proveedor o mediante el procedimiento manual autorizado por la institución.

RN-041 – Duplicidad de Confirmación

Una misma transacción QR no podrá utilizarse para confirmar más de un pago.

RN-042 – QR Expirado

Una solicitud QR expirada no deberá poder confirmarse como una nueva transacción sin realizar previamente el procedimiento correspondiente definido por el proveedor.

5.7 Reglas Académicas

RN-043 – Asignación Docente

Un docente únicamente podrá gestionar académicamente cursos y materias que tenga asignados.

RN-044 – Conflicto de Horarios

Un docente no deberá ser asignado a dos clases cuyos horarios se superpongan.

RN-045 – Lista de Estudiantes

La lista académica de un curso deberá obtenerse a partir de las inscripciones válidas correspondientes.

RN-046 – Estudiantes Cancelados

Los estudiantes cuya inscripción haya sido cancelada no deberán considerarse como estudiantes activos del curso para nuevos registros académicos.

Su información histórica deberá conservarse.

RN-047 – Registro de Asistencia

La asistencia deberá registrarse por sesión de clase, estudiante, curso y materia.

RN-048 – Estados de Asistencia

Inicialmente se consideran:

- Presente.
- Ausente.
- Retraso.
- Justificado.

Los estados podrán configurarse según las políticas institucionales.

RN-049 – Modificación de Asistencia

Los docentes podrán modificar registros de asistencia dentro del periodo permitido.

Después del cierre correspondiente, será necesaria una autorización especial.

5.8 Reglas de Materiales, Prácticas y Entregas

RN-050 – Publicación de Material

Un docente únicamente podrá publicar material en cursos y materias que tenga asignados.

RN-051 – Acceso al Material

Solo los estudiantes con inscripción válida en el curso correspondiente podrán acceder al material publicado.

RN-052 – Creación de Prácticas

Una práctica deberá encontrarse asociada como mínimo a:

- Curso.
- Materia.
- Docente.
- Fecha de publicación.
- Fecha límite, cuando corresponda.

RN-053 – Entrega de Prácticas

Solo los estudiantes inscritos en el curso podrán realizar entregas correspondientes a sus actividades.

RN-054 – Fecha de Entrega

Toda entrega deberá registrar automáticamente la fecha y hora en que fue realizada.

RN-055 – Entregas Tardías

Cuando se permitan entregas después de la fecha límite, el sistema deberá identificarlas como **TARDÍAS**.

RN-056 – Reemplazo de Entrega

El estudiante podrá reemplazar una entrega únicamente cuando la configuración de la práctica lo permita.

El sistema deberá conservar la trazabilidad necesaria cuando corresponda.

5.9 Reglas de Calificaciones

RN-057 – Registro de Calificación

Solo el docente responsable o un usuario expresamente autorizado podrá registrar calificaciones.

RN-058 – Rango de Calificación

Las calificaciones deberán encontrarse dentro del rango definido para la evaluación.

Por ejemplo, si una práctica tiene un máximo de 20 puntos, no podrá registrarse una calificación superior a 20.

RN-059 – Publicación de Calificaciones

Una calificación podrá permanecer registrada internamente antes de ser publicada.

El estudiante únicamente visualizará las calificaciones que hayan sido habilitadas para consulta.

RN-060 – Modificación de Calificación

Toda modificación posterior de una calificación deberá conservar información suficiente para identificar:

- Valor anterior.
- Valor nuevo.
- Usuario responsable.
- Fecha.
- Hora.

RN-061 – Cierre de Calificaciones

Después del cierre académico, las calificaciones no podrán modificarse mediante el procedimiento habitual.

Las modificaciones excepcionales requerirán autorización.

RN-062 – Cálculo de Nota Final

Cuando se utilicen ponderaciones, la nota final deberá calcularse utilizando la fórmula académica configurada para el curso.

5.10 Reglas Financieras

RN-063 – Registro de Ingresos

Todo ingreso institucional deberá poseer un concepto, monto, fecha, categoría y usuario responsable.

RN-064 – Pagos de Estudiantes como Ingresos

Los pagos confirmados de estudiantes deberán reflejarse en los registros financieros correspondientes sin generar duplicidades.

RN-065 – Registro de Egresos

Todo egreso deberá registrar:

- Categoría.
- Concepto.
- Monto.
- Fecha.
- Usuario responsable.
- Sucursal cuando corresponda.

RN-066 – Salarios como Egreso

Todo pago salarial confirmado deberá reflejarse como un egreso institucional.

RN-067 – Anulación Financiera

Los movimientos financieros anulados deberán conservarse en el historial con su respectivo estado.

RN-068 – Cierre de Caja

El cierre de caja deberá considerar todos los movimientos válidos registrados durante la jornada o periodo correspondiente.

RN-069 – Diferencias de Caja

Toda diferencia entre el monto esperado y el monto declarado deberá registrarse y requerir una justificación.

RN-070 – Cierre Financiero

Después del cierre de un periodo financiero, las operaciones correspondientes no deberán modificarse mediante los procedimientos normales.

Cualquier corrección posterior deberá ser autorizada y auditada.

5.11 Reglas de Docentes y Salarios

RN-071 – Docente Activo

Solo docentes con estado activo podrán recibir nuevas asignaciones académicas.

RN-072 – Carga Horaria

Toda asignación docente deberá indicar la carga horaria correspondiente.

RN-073 – Horas Trabajadas

Cuando el pago dependa de horas trabajadas, el sistema deberá diferenciar entre horas planificadas y horas efectivamente reconocidas para pago.

RN-074 – Modalidad de Remuneración

La institución podrá definir diferentes modalidades de remuneración, como:

- Pago por hora.
- Pago fijo.
- Otra modalidad autorizada.

RN-075 – Confidencialidad Salarial

La información salarial solo podrá ser consultada por usuarios con permisos financieros específicos.

RN-076 – Historial de Pagos

Los pagos realizados a docentes deberán conservarse permanentemente como parte del historial financiero institucional.

5.12 Reglas de Auditoría y Seguridad

RN-077 – Auditoría Obligatoria

Las operaciones críticas deberán quedar registradas automáticamente en la bitácora.

RN-078 – Usuario Responsable

Las operaciones realizadas dentro del sistema deberán asociarse al usuario autenticado que las ejecutó.

No deberán utilizarse cuentas compartidas para operaciones administrativas críticas.

RN-079 – Restricción de Auditoría

Los registros de auditoría no podrán ser modificados mediante las interfaces operativas normales del sistema.

RN-080 – Protección de Información

Cada usuario deberá visualizar únicamente la información permitida por sus roles y permisos.

RN-081 – Datos Financieros

La información financiera general no deberá estar disponible para docentes o estudiantes, excepto la información individual que corresponda a cada usuario.

RN-082 – Datos Académicos del Estudiante

Un estudiante únicamente podrá consultar sus propias calificaciones, asistencia, prácticas e información académica personal.

5.13 Reglas de Reportes

RN-083 – Información Actualizada

Los reportes deberán obtener la información desde los registros válidos disponibles al momento de su generación.

RN-084 – Registros Anulados

Los registros anulados no deberán contabilizarse como operaciones válidas en los totales financieros, salvo cuando el reporte solicite expresamente información histórica o de anulaciones.

RN-085 – Permisos de Reportes

El acceso a cada reporte dependerá del rol y los permisos del usuario.

RN-086 – Reportes Financieros

Los reportes financieros deberán considerar únicamente movimientos válidos dentro del periodo seleccionado.

5.14 Reglas de Respaldo y Conservación

RN-087 – Respaldo de Información

La información institucional deberá formar parte de los procedimientos de respaldo definidos para el sistema.

RN-088 – Restauración

La restauración de un respaldo deberá ser ejecutada únicamente por personal técnico autorizado.

RN-089 – Conservación Histórica

La información financiera, académica y de auditoría deberá conservarse de acuerdo con las políticas institucionales y obligaciones aplicables.

RN-090 – Trazabilidad

Las operaciones críticas deberán disponer de información suficiente para reconstruir posteriormente:

quién realizó la operación, qué hizo, cuándo lo hizo y sobre qué registro actuó.

5.15 Reglas Pendientes de Validación Institucional

Durante el análisis se han identificado algunas reglas cuya definición definitiva deberá realizarse junto con los responsables del instituto antes de implementar los módulos correspondientes.

Entre las principales se encuentran:

- Política exacta de devolución de dinero ante cancelaciones.
- Tratamiento de pagos realizados en exceso.
- Procedimiento para cambio de curso después de realizar un pago.
- Número máximo de cuotas permitidas.
- Fechas y condiciones de vencimiento de cuotas.
- Política institucional de descuentos.
- Usuarios autorizados para aplicar descuentos especiales.
- Tiempo permitido para modificar asistencia.
- Escala oficial de calificaciones.
- Fórmula utilizada para obtener la nota final.
- Tratamiento de actividades entregadas fuera de plazo.
- Método definitivo para calcular salarios docentes.
- Procedimiento de autorización de egresos.
- Responsables del cierre y arqueo de caja.
- Procedimiento definitivo para anulación de pagos y recibos.
- Entidad financiera o proveedor que será utilizado para los pagos QR.
- Condiciones para confirmación automática o manual de pagos QR.
- Política de retención de respaldos.

Estas definiciones deberán documentarse conforme sean validadas, actualizando las reglas de negocio y los requerimientos relacionados.

5.16 Cierre del Capítulo

Las reglas de negocio definidas en este capítulo establecen las condiciones que deberán cumplirse durante la ejecución de los principales procesos académicos, administrativos y

financieros del Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA – SUMO). Su definición permite complementar los requerimientos funcionales y no funcionales previamente establecidos, proporcionando criterios claros para el diseño de la base de datos, la implementación de las funcionalidades y la elaboración de casos de prueba. Asimismo, las reglas identificadas como pendientes de validación deberán ser revisadas con los responsables institucionales antes de implementar las funcionalidades directamente relacionadas, permitiendo reducir ambigüedades y controlar los cambios durante el desarrollo.

CAPÍTULO 6 ARQUITECTURA Y DISEÑO GENERAL DEL SISTEMA

6.1 Introducción

El presente capítulo describe la arquitectura tecnológica y el diseño general propuesto para el desarrollo del Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA – SUMO).

La arquitectura del sistema constituye uno de los componentes fundamentales del proyecto, debido a que define la forma en que los diferentes módulos interactuarán entre sí, cómo será procesada la información, cómo se garantizará la seguridad de los datos y de qué manera podrá ampliarse el sistema en futuras versiones.

La solución será desarrollada bajo un enfoque de arquitectura web cliente-servidor, separando claramente la interfaz de usuario, la lógica de negocio, los servicios de aplicación y la persistencia de datos.

Esta separación permitirá reducir el acoplamiento entre los diferentes componentes del sistema y facilitará actividades posteriores como mantenimiento, corrección de errores, actualización de funcionalidades, incorporación de nuevos módulos y escalamiento de infraestructura.

La arquitectura propuesta deberá responder a los requerimientos funcionales y no funcionales establecidos en capítulos anteriores, principalmente aquellos relacionados con:

- Gestión de estudiantes.
- Inscripciones.
- Pagos.
- Recibos.
- Pagos mediante código QR.
- Gestión financiera.
- Gestión académica.
- Control de docentes.
- Asistencia.
- Material académico.
- Prácticas.
- Calificaciones.

- Reportes.
- Auditoría.
- Seguridad.
- Disponibilidad.
- Rendimiento.
- Mantenibilidad.
- Escalabilidad.

Las principales tecnologías seleccionadas inicialmente para la implementación del sistema son:

- **Frontend:** React.
- **Lenguaje del frontend:** TypeScript.
- **Backend:** Python.
- **Framework backend:** Django.
- **Servicios web:** Django REST Framework.
- **Base de datos:** MariaDB.
- **Arquitectura de comunicación:** API REST.
- **Formato principal de intercambio de información:** JSON.
- **Control de versiones:** Git.
- **Servidor de producción:** Linux.
- **Proxy inverso y servidor web:** Nginx.
- **Comunicación segura:** HTTPS mediante TLS.
- **Gestión de archivos:** almacenamiento externo a las tablas principales de la base de datos.
- **Autenticación:** mecanismo seguro basado en sesiones o tokens, según la configuración final de despliegue.

La selección definitiva de versiones y herramientas complementarias deberá realizarse antes de iniciar la etapa de desarrollo e instalación del entorno de trabajo.

6.2 Principios Arquitectónicos

La arquitectura del sistema se diseñará considerando los siguientes principios generales.

6.2.1 Separación de Responsabilidades

Cada componente del sistema deberá cumplir funciones claramente definidas.

La interfaz será responsable de presentar información y recibir las acciones realizadas por el usuario.

El backend será responsable de ejecutar las reglas de negocio, validar operaciones, controlar permisos y coordinar el almacenamiento de información.

La base de datos será responsable de mantener la información persistente y garantizar la integridad de las relaciones.

Esta separación permitirá evitar que la lógica crítica del sistema quede dispersa entre diferentes componentes.

6.2.2 Modularidad

SIGA-SUMO será organizado mediante módulos funcionales independientes relacionados con los procesos principales de la institución.

Entre los principales módulos se consideran:

- Usuarios.
- Estudiantes.
- Docentes.
- Gestión académica.
- Inscripciones.
- Pagos.
- Finanzas.
- Asistencia.
- Material académico.
- Prácticas.
- Calificaciones.
- Reportes.
- Configuración.
- Auditoría.

Cada módulo deberá mantener responsabilidades específicas y comunicarse con los demás mediante mecanismos controlados.

6.2.3 Bajo Acoplamiento

Los módulos deberán depender lo menos posible de implementaciones internas de otros módulos.

Por ejemplo, el módulo de pagos no deberá modificar directamente las tablas internas del módulo académico sin pasar por los servicios definidos para dicha operación.

Este principio facilitará realizar modificaciones sin afectar innecesariamente otras partes del sistema.

6.2.4 Alta Cohesión

Cada módulo deberá concentrar funcionalidades relacionadas con un mismo dominio.

Por ejemplo, el módulo financiero concentrará las operaciones de ingresos, egresos, caja y movimientos económicos, mientras que el módulo académico administrará cursos, materias, periodos y horarios.

6.2.5 Seguridad desde el Diseño

La seguridad no será incorporada únicamente al finalizar el desarrollo.

Desde el diseño deberán considerarse:

- Autenticación.
- Autorización.
- Roles.
- Permisos.
- Auditoría.
- Protección de datos.
- Validación de entradas.
- Gestión segura de archivos.
- Control de acceso.
- Protección de información financiera.

6.2.6 Escalabilidad

La arquitectura deberá permitir que el sistema continúe funcionando cuando aumente la cantidad de estudiantes, docentes, cursos, usuarios o sucursales.

El crecimiento no deberá exigir un rediseño completo de la plataforma.

6.3 Arquitectura General

SIGA – SUMO utilizará una arquitectura distribuida en diferentes niveles lógicos.

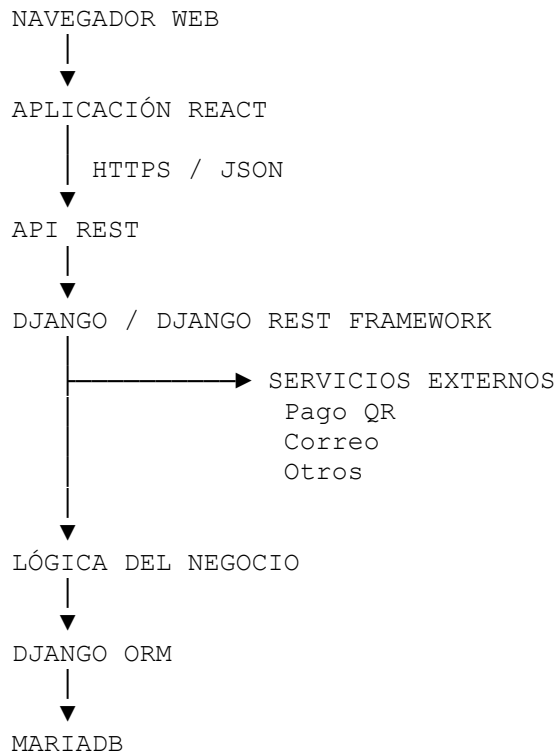
Los principales componentes serán:

1. Cliente o navegador.
2. Aplicación frontend.
3. API backend.
4. Servicios de negocio.
5. Base de datos.
6. Almacenamiento de archivos.
7. Servicios externos.

La comunicación seguirá principalmente el siguiente flujo:

USUARIO





Los usuarios no deberán acceder directamente a MariaDB.

Todas las operaciones deberán pasar por el backend.

Esto permitirá aplicar consistentemente:

- Validaciones.
- Reglas de negocio.
- Permisos.
- Seguridad.
- Auditoría.

6.4 Arquitectura Cliente-Servidor

SIGA será una aplicación web basada en el modelo cliente-servidor.

El cliente estará compuesto por la interfaz desarrollada con React y ejecutada desde el navegador web.

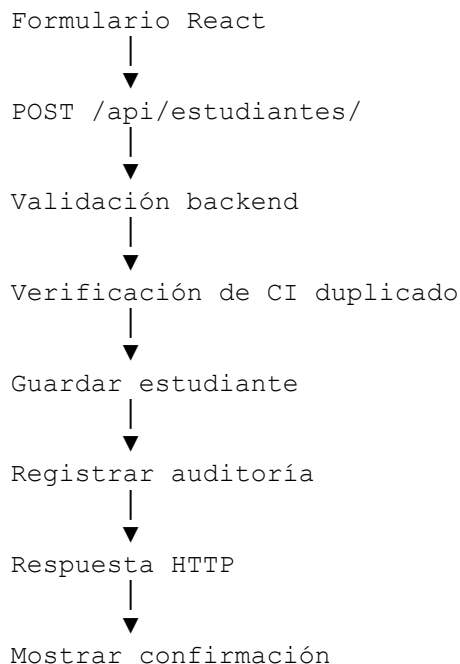
El servidor estará compuesto principalmente por Django, Django REST Framework, MariaDB y los servicios necesarios para ejecutar la lógica del negocio.

Funcionamiento general

Cuando un usuario realice una acción:

1. El usuario interactuará con una pantalla desarrollada en React.
2. React recopilará los datos necesarios.
3. La aplicación enviará una solicitud HTTPS al backend.
4. Django REST Framework recibirá la solicitud.
5. El backend verificará autenticación y permisos.
6. Se validarán los datos recibidos.
7. Se ejecutarán las reglas de negocio correspondientes.
8. Django realizará las operaciones necesarias mediante su ORM.
9. MariaDB procesará las consultas.
10. El backend generará una respuesta.
11. React actualizará la interfaz.
12. El usuario visualizará el resultado.

Un ejemplo podría ser el registro de un estudiante:



6.5 Frontend

El frontend constituirá la capa visible del sistema.

Será desarrollado utilizando React y se recomienda utilizar TypeScript para aumentar la seguridad y mantenibilidad del código.

React permitirá implementar una interfaz basada en componentes reutilizables.

Por ejemplo, elementos como:

- Tablas.

- Formularios.
- Modales.
- Menús.
- Botones.
- Indicadores.
- Tarjetas.
- Alertas.
- Selectores.
- Filtros.

podrán desarrollarse como componentes reutilizables.

6.5.1 Responsabilidades

El frontend será responsable de:

- Mostrar información.
- Presentar formularios.
- Recibir entradas.
- Validar información básica.
- Consumir APIs.
- Mostrar errores.
- Mostrar confirmaciones.
- Controlar navegación.
- Mostrar menús según el rol.
- Renderizar reportes.
- Gestionar interacción con el usuario.

No deberá contener como única fuente reglas críticas del negocio.

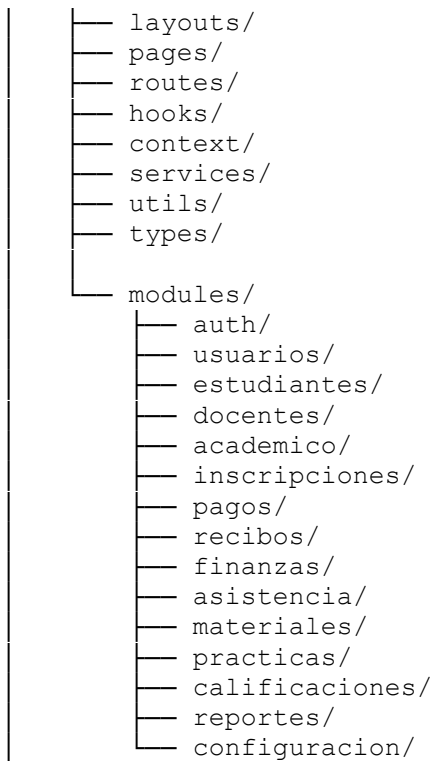
Por ejemplo, React podrá impedir temporalmente que el usuario introduzca un monto negativo, pero Django deberá validar nuevamente ese monto antes de registrarlo.

6.6 Organización del Frontend

El frontend será organizado de manera modular.

Una estructura inicial podría ser:

```
frontend/
├── public/
├── src/
│   ├── api/
│   ├── assets/
│   └── components/
```



Cada módulo podrá contener sus propios:

- Componentes.
- Servicios.
- Tipos.
- Páginas.
- Hooks.
- Validaciones.

Esta organización facilitará el crecimiento del proyecto.

6.7 Gestión del Estado del Frontend

Debido a que SIGA – SUMO contará con una cantidad considerable de módulos, será necesario definir una estrategia clara para administrar los datos disponibles en la aplicación.

Existirán principalmente dos tipos de estado:

Estado local

Información utilizada únicamente dentro de un componente.

Ejemplo:

- Contenido de un formulario.

- Modal abierto.
- Filtro seleccionado.

Estado global

Información que necesita ser compartida entre diferentes partes de la aplicación.

Ejemplo:

- Usuario autenticado.
- Rol.
- Permisos.
- Sucursal seleccionada.
- Notificaciones.

Se procurará evitar almacenar globalmente información que pueda mantenerse local.

6.8 Navegación y Rutas

La aplicación utilizará un sistema de rutas para navegar entre las diferentes páginas.

Ejemplo:

```
/login  
  
/dashboard  
  
/estudiantes  
/estudiantes/nuevo  
/estudiantes/:id  
  
/docentes  
  
/inscripciones  
  
/pagos  
  
/finanzas  
  
/reportes
```

Las rutas deberán protegerse según los permisos del usuario.

Por ejemplo:

Un estudiante no deberá acceder a:

```
/finanzas
```

aunque intente escribir manualmente la dirección en el navegador.

La protección visual del frontend deberá complementarse obligatoriamente con validación del backend.

6.9 Backend

El backend constituye el núcleo lógico del sistema.

Será desarrollado con Python utilizando Django.

Django proporcionará una estructura estable para:

- Modelos.
- Seguridad.
- Usuarios.
- Migraciones.
- ORM.
- Validaciones.
- Administración.
- Gestión de archivos.
- Configuración.

Django REST Framework se utilizará para construir los servicios REST que serán consumidos por React.

6.10 Organización del Backend

El backend deberá organizarse siguiendo los dominios del sistema.

Ejemplo:

```
backend/
├── config/
│   ├── settings/
│   ├── urls.py
│   └── wsgi.py
├── apps/
│   ├── usuarios/
│   ├── roles/
│   ├── estudiantes/
│   ├── docentes/
│   ├── sucursales/
│   ├── academico/
│   ├── inscripciones/
│   ├── pagos/
│   ├── recibos/
│   ├── qr/
│   └── finanzas/
```



```
|
|  └─ asistencia/
|  └─ materiales/
|  └─ practicas/
|  └─ calificaciones/
|  └─ reportes/
|  └─ notificaciones/
|  └─ auditoria/
|
└─ media/
   static/
   requirements/
   manage.py
```

Cada aplicación Django podrá contener:

```
models.py
serializers.py
views.py
urls.py
permissions.py
services.py
tests.py
admin.py
```

Cuando un módulo sea demasiado grande, podrá dividirse en archivos o carpetas internas.

6.11 Capa de Servicios

Las reglas de negocio complejas no deberán concentrarse completamente en las vistas o controladores.

Se recomienda utilizar una capa de servicios.

Por ejemplo:

```
apps/pagos/services/
```

podría contener procesos como:

- registrar_pago()
- anular_pago()
- confirmar_pago_qr()
- calcular_saldo()
- generar_recibo()

Esto permitirá separar las reglas de negocio del mecanismo utilizado para recibir solicitudes HTTP.

6.12 API REST

La API REST constituirá el mecanismo principal de comunicación entre React y Django.

Las operaciones utilizarán métodos HTTP apropiados.

Ejemplo:

```
GET /api/estudiantes/
```

Consultar estudiantes.

```
POST /api/estudiantes/
```

Crear estudiante.

```
GET /api/estudiantes/10/
```

Consultar estudiante específico.

```
PATCH /api/estudiantes/10/
```

Modificar parcialmente.

Cada endpoint deberá verificar:

- Autenticación.
- Autorización.
- Datos.
- Reglas del negocio.

6.13 Estructura de Respuestas de la API

Las respuestas deberán mantener una estructura consistente.

Ejemplo exitoso:

```
{
  "success": true,
  "data": {},
  "message": "Estudiante registrado correctamente"
}
```

Ejemplo de error:

```
{
  "success": false,
  "errors": {
    "ci": [
      "Ya existe un estudiante registrado con este CI"
    ]
  }
}
```

```
}  
}
```

Mantener respuestas consistentes facilitará el trabajo del frontend.

6.14 Manejo de Errores

El backend deberá utilizar un mecanismo centralizado para manejar errores.

Los usuarios no deberán visualizar detalles técnicos como:

- Consultas SQL.
- Rutas internas.
- Variables.
- Stack traces.

Estos detalles podrán almacenarse en los registros técnicos del servidor.

El frontend recibirá mensajes comprensibles.

Ejemplo:

```
No fue posible registrar el pago. Verifique la información e intente  
nuevamente.
```

6.15 Base de Datos

MariaDB será utilizado como gestor de base de datos relacional.

Su función será almacenar de forma persistente la información del sistema.

La estructura deberá considerar:

- Claves primarias.
- Claves foráneas.
- Índices.
- Restricciones.
- Valores obligatorios.
- Transacciones.
- Integridad referencial.

Django gestionará la comunicación mediante su ORM.

6.16 Normalización de Datos

La base de datos será diseñada siguiendo principios de normalización.

Inicialmente se buscará alcanzar al menos la Tercera Forma Normal cuando sea apropiado.

Esto permitirá reducir:

- Duplicidad.
- Redundancia.
- Inconsistencias.

Por ejemplo, el nombre de una sucursal no deberá almacenarse repetidamente como texto en cada inscripción.

En su lugar:

```
sucursal
-----
id
nombre
```

y la inscripción mantendrá una referencia a dicha sucursal.

6.17 Índices

Los índices deberán utilizarse para optimizar consultas frecuentes.

Por ejemplo:

- CI del estudiante.
- Correo electrónico.
- Número de recibo.
- Fecha de pago.
- Estado.
- Número de transacción QR.

Los índices deberán definirse cuidadosamente para evitar degradar operaciones de escritura innecesariamente.

6.18 Transacciones de Base de Datos

Las operaciones financieras deberán utilizar transacciones.

Un ejemplo crítico será el registro de pago:

```
INICIO TRANSACCIÓN
```

```
Registrar pago
Registrar movimiento financiero
Actualizar saldo
Generar recibo
```

Registrar auditoría

COMMIT

Si ocurre un error crítico:

ROLLBACK

De esta forma se evitarán registros inconsistentes.

6.19 Principales Entidades

La primera versión contempla entidades relacionadas con los siguientes dominios.

Seguridad

- Usuario.
- Rol.
- Permiso.
- UsuarioRol.
- RolPermiso.
- Auditoría.

Institución

- Sucursal.
- Configuración.
- Aula.

Académico

- Área.
- Curso.
- Materia.
- Paralelo.
- Periodo.
- Horario.
- AsignaciónDocente.

Estudiantes

- Estudiante.
- Tutor.
- EstudianteTutor.
- Observación.

Inscripciones

- Inscripción.
- DetalleInscripción.
- PlanPago.
- Cuota.

Finanzas

- Pago.
- MétodoPago.
- Recibo.
- TransacciónQR.
- MovimientoFinanciero.
- CategoríaFinanciera.
- Caja.
- CierreCaja.
- PagoDocente.

Académico docente

- SesiónClase.
- Asistencia.
- Material.
- Práctica.
- Entrega.
- Evaluación.
- Calificación.

Sistema

- Notificación.
- Archivo.
- Auditoría.
- Respaldo.

6.20 Autenticación

Todos los usuarios que requieran acceso al sistema deberán autenticarse.

El sistema deberá identificar:

- Usuario.
- Estado.
- Rol.
- Permisos.

La autenticación deberá impedir que usuarios desactivados accedan incluso cuando sus credenciales sean correctas.

6.21 Autorización

Después de la autenticación, el backend verificará si el usuario tiene permisos suficientes para ejecutar la operación solicitada.

Por ejemplo:

```
Docente
  ↓
Puede registrar asistencia
  ↓
Sí
```

pero:

```
Docente
  ↓
Intentar anular recibo
  ↓
Acceso denegado
```

La autorización deberá ejecutarse en el backend.

6.22 Roles y Permisos

SIGA utilizará un modelo de permisos basado en roles.

Roles iniciales:

- Administrador.
- Director.
- Administrativo.
- Caja.
- Docente.
- Estudiante.

Los permisos podrán relacionarse con acciones como:

- Ver.
- Crear.
- Editar.
- Anular.
- Exportar.
- Imprimir.
- Administrar.

Esto permitirá evitar codificar reglas rígidas para cada usuario.

6.23 Auditoría

El sistema deberá registrar operaciones relevantes.

Ejemplo de registro:

```
Usuario: admin01
Acción: ANULAR_RECIBO
Registro: REC-000213
Fecha: 18/08/2026
Hora: 11:20
Motivo: Registro incorrecto
```

La auditoría será especialmente importante en:

- Pagos.
- Recibos.
- Finanzas.
- Calificaciones.
- Roles.
- Permisos.
- Inscripciones.

6.24 Gestión de Archivos

Los archivos académicos y administrativos no deberán almacenarse directamente como grandes objetos binarios dentro de las tablas principales.

Se utilizará almacenamiento de archivos y MariaDB guardará principalmente:

- Nombre.
- Ruta.
- Tipo.
- Tamaño.
- Usuario.
- Fecha.
- Relación con el recurso.

Los archivos podrán pertenecer a:

- Material académico.
- Prácticas.
- Entregas.
- Fotografías.
- Documentos.
- Comprobantes.

6.25 Validación de Archivos

Antes de aceptar un archivo, el backend deberá comprobar:

- Tipo.
- Extensión.
- Tamaño.
- Usuario.
- Recurso relacionado.

El nombre final utilizado en almacenamiento podrá diferir del nombre original para evitar conflictos.

6.26 Gestión de Impresora Térmica

La impresión térmica constituye un requisito especial debido a las restricciones de los navegadores.

Antes de implementar definitivamente este módulo se deberá identificar:

- Fabricante.
- Modelo.
- Tipo de conexión.
- Driver.
- Ancho de papel.
- Sistema operativo.
- Compatibilidad con impresión desde navegador.
- Compatibilidad con ESC/POS, si corresponde.

El flujo conceptual será:

```
Pago confirmado
↓
Django genera datos
↓
React recibe comprobante
↓
Generar formato imprimible
↓
Enviar impresión
↓
Impresora térmica
```

Dependiendo del hardware, podrá ser necesario utilizar un servicio local de impresión.

6.27 Generación del Recibo

El recibo deberá generarse a partir de información almacenada en el sistema.

Podrá incluir:

- Nombre del instituto.
- Número de recibo.
- Sucursal.
- Fecha.
- Estudiante.
- CI.
- Concepto.
- Curso.
- Método de pago.
- Monto.
- Saldo restante.
- Usuario responsable.

La estructura exacta será validada utilizando una impresión real.

6.28 Integración con Pagos QR

La integración con pagos QR se diseñará como un componente independiente.

El módulo principal de pagos no deberá depender directamente de un proveedor concreto.

Se recomienda definir una interfaz lógica:

```
ProveedorQR
```

```
generar_qr()  
consultar_estado()  
confirmar_pago()  
cancelar_solicitud()
```

Después podrá implementarse:

```
BancoAProvider  
BancoBProvider  
ProveedorExternoProvider
```

Esta estrategia facilitará cambiar de entidad financiera.

6.29 Flujo de Pago QR

El proceso general será:

```
Usuario genera obligación  
↓  
Solicita pago QR  
↓
```

```
Backend genera solicitud
↓
Proveedor financiero
↓
Código QR
↓
Estudiante paga
↓
Proveedor confirma
↓
Backend valida
↓
Registrar pago
↓
Actualizar saldo
↓
Registrar ingreso
↓
Generar recibo
```

El pago no deberá marcarse como confirmado antes de recibir una validación válida.

6.30 Servicios Externos

La arquitectura deberá permitir integrar servicios externos.

Entre ellos:

- Proveedor QR.
- Correo electrónico.
- Mensajería.
- Almacenamiento.
- Futuras aplicaciones móviles.

Estas integraciones deberán implementarse mediante servicios desacoplados.

6.31 Reportes

Los reportes podrán obtener información directamente de los servicios del backend.

El backend será responsable de aplicar:

- Permisos.
- Filtros.
- Cálculos.
- Validaciones.

React será responsable de:

- Mostrar tablas.

- Mostrar gráficos.
- Solicitar filtros.
- Descargar archivos.

Los reportes complejos podrán generarse como PDF o Excel.

6.32 Dashboard

El Dashboard será construido a partir de consultas agregadas proporcionadas por el backend.

Ejemplo:

```
GET /api/dashboard/resumen/
```

Respuesta:

```
{
  "estudiantes_activos": 500,
  "inscripciones_mes": 120,
  "ingresos_mes": 35000,
  "cuentas_pendientes": 75
}
```

Los indicadores mostrados dependerán del rol.

6.33 Notificaciones

El sistema podrá implementar un servicio central de notificaciones.

Las notificaciones podrán clasificarse como:

- Académicas.
- Administrativas.
- Financieras.
- Seguridad.

Inicialmente podrán mostrarse dentro del sistema.

Posteriormente podrá incorporarse:

- Correo.
- WhatsApp.
- Push.

6.34 Arquitectura Modular Completa

La arquitectura lógica puede representarse de la siguiente manera:

SIGA

AUTENTICACIÓN

- Usuarios
- Roles
- Permisos

INSTITUCIÓN

- Sucursales
- Aulas
- Configuración

ESTUDIANTES

- Datos personales
- Tutores
- Expediente
- Historial

ACADÉMICO

- Áreas
- Cursos
- Materias
- Paralelos
- Periodos
- Horarios

DOCENTES

- Expediente
- Asignaciones
- Horarios
- Carga horaria

INSCRIPCIONES

- Inscripción
- Descuentos
- Plan de pagos
- Historial

PAGOS

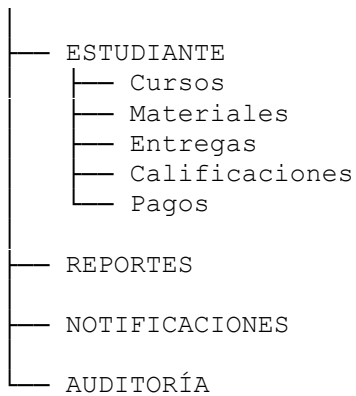
- Efectivo
- QR
- Transferencia
- Cuotas
- Recibos

FINANZAS

- Ingresos
- Egresos
- Caja
- Arqueo
- Salarios

DOCENTE

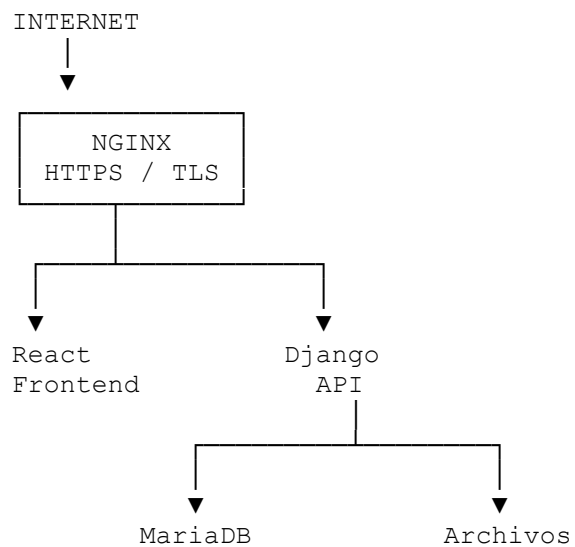
- Asistencia
- Material
- Prácticas
- Calificaciones



6.35 Arquitectura de Despliegue

En producción, la arquitectura podrá utilizar inicialmente un único servidor debido al volumen inicial de usuarios.

Ejemplo:



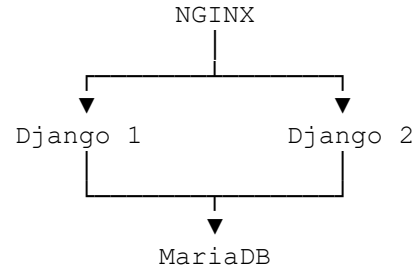
Nginx será responsable de:

- Recibir solicitudes.
- Gestionar HTTPS.
- Servir archivos estáticos.
- Redirigir solicitudes hacia Django.
- Aplicar configuraciones básicas de seguridad.

6.36 Escalamiento Futuro

Si el número de usuarios aumenta considerablemente, la arquitectura podrá evolucionar.

Por ejemplo:



También podrán separarse:

- Archivos.
- Base de datos.
- Backend.
- Frontend.
- Caché.
- Colas de tareas.

No será necesario implementar esta complejidad desde el inicio.

6.37 Entorno de Desarrollo

El desarrollador contará con un entorno local para desarrollar y probar funcionalidades.

Podrá incluir:

- Node.js.
- npm.
- Python.
- Virtual environment.
- Django.
- MariaDB.
- Git.
- VS Code.
- Navegador.

Se recomienda que el entorno pueda reproducirse fácilmente.

6.38 Entorno de Pruebas

Cuando sea posible, se dispondrá de un entorno independiente para validar cambios antes de llevarlos a producción.

El entorno deberá contener datos de prueba.

No se recomienda utilizar información financiera real para realizar pruebas de desarrollo.

6.39 Entorno de Producción

El entorno de producción contendrá los datos reales de la institución.

El acceso administrativo deberá estar restringido.

Las modificaciones directas de la base de datos deberán evitarse salvo situaciones técnicas justificadas.

6.40 Control de Versiones

Todo el código fuente será gestionado mediante Git.

Se recomienda mantener un repositorio privado.

Una posible estrategia inicial será:

```
main
develop
feature/*
fix/*
```

main representará código estable.

develop contendrá la versión actual de desarrollo.

Las ramas **feature/** podrán utilizarse para funcionalidades específicas.

Ejemplo:

```
feature/inscripciones
feature/pagos-qr
feature/asistencia
```

6.41 Estructura del Repositorio

Debido a que el proyecto será desarrollado inicialmente por un equipo pequeño, se recomienda un repositorio único:

```
sigas/
├── frontend/
├── backend/
├── docs/
├── scripts/
├── .gitignore
└── README.md
```


└─ docker/

La estructura podrá evolucionar posteriormente.

6.42 Gestión de Configuración

Las configuraciones relacionadas con el entorno no deberán incluirse directamente en el código fuente.

Ejemplos:

```
DATABASE_HOST
DATABASE_NAME
DATABASE_USER
DATABASE_PASSWORD
SECRET_KEY
QR_API_KEY
EMAIL_PASSWORD
```

Estos valores deberán almacenarse mediante variables de entorno.

Los archivos que contengan secretos no deberán incluirse en el repositorio.

6.43 Registro de Eventos Técnicos

Además de la auditoría funcional, el backend deberá mantener registros técnicos.

Se podrán registrar:

- Errores.
- Excepciones.
- Advertencias.
- Inicio del servidor.
- Problemas de integración.
- Fallos de QR.

Esto ayudará durante mantenimiento y soporte.

6.44 Rendimiento

La arquitectura deberá buscar minimizar consultas innecesarias.

Algunas técnicas que podrán utilizarse incluyen:

- Índices.
- Paginación.
- Consultas optimizadas.
- Select related.

- Prefetch related.
- Caché cuando sea necesario.

Las listas grandes no deberán cargarse completas cuando puedan paginarse.

6.45 Concurrencia

El sistema deberá considerar que diferentes administrativos pueden trabajar simultáneamente.

Por ejemplo:

Dos administrativos podrían intentar inscribir al último estudiante disponible para un curso.

El backend deberá aplicar validaciones suficientes para evitar inconsistencias.

6.46 Integridad Financiera

Las operaciones financieras deberán priorizar la integridad de la información.

Una vez confirmado un movimiento, cualquier modificación deberá realizarse mediante procesos controlados.

Se evitará modificar directamente:

- Montos.
- Recibos.
- Pagos.
- Cierres.

Cuando sea necesario corregir una operación deberá utilizarse:

- Anulación.
- Reversión.
- Ajuste.

6.47 Respaldo

MariaDB deberá formar parte de una política de respaldo periódico.

Los respaldos deberán almacenarse fuera de la ubicación principal de la base de datos cuando la infraestructura lo permita.

También deberán contemplarse los archivos cargados por los usuarios.

6.48 Recuperación

La arquitectura deberá permitir restaurar la información utilizando respaldos válidos.

El procedimiento de recuperación deberá documentarse.

Las pruebas de restauración deberán realizarse periódicamente cuando el sistema entre en producción.

6.49 Tecnologías Seleccionadas

Componente	Tecnología propuesta
Frontend	React
Lenguaje frontend	TypeScript
Gestión de paquetes	npm
Backend	Python
Framework	Django
API	Django REST Framework
Base de datos	MariaDB
ORM	Django ORM
Formato API	JSON
Control de versiones	Git
Servidor	Linux
Proxy inverso	Nginx
Protocolo	HTTPS
Archivos	Sistema de archivos / almacenamiento externo
Documentación API	OpenAPI / Swagger
Pruebas backend	Django Test / Pytest
Pruebas frontend	Herramientas compatibles con React

6.50 Justificación de React

React fue seleccionado debido a que permite construir interfaces modernas basadas en componentes reutilizables.

Su utilización resulta adecuada para SIGA debido a que la aplicación contará con múltiples interfaces dinámicas.

Entre sus ventajas se encuentran:

- Componentización.
- Reutilización.
- Amplio ecosistema.
- Buena integración con APIs REST.
- Facilidad para construir dashboards.
- Capacidad para aplicaciones responsivas.
- Separación clara respecto al backend.

6.51 Justificación de TypeScript

Se recomienda utilizar TypeScript debido al tamaño del proyecto.

TypeScript permitirá definir estructuras de datos claras.

Ejemplo:

```
Estudiante
Pago
Inscripcion
Curso
Usuario
```

Esto reducirá errores relacionados con propiedades incorrectas o datos inesperados.

Además, mejorará:

- Autocompletado.
- Refactorización.
- Mantenimiento.
- Legibilidad.

6.52 Justificación de Python

Python se utilizará como lenguaje principal del backend debido a:

- Legibilidad.
- Productividad.
- Amplio ecosistema.
- Disponibilidad de librerías.
- Facilidad para pruebas.
- Integración con múltiples servicios.

También permitirá ampliar el sistema en el futuro hacia procesamiento de datos o análisis estadístico.

6.53 Justificación de Django

Django resulta especialmente conveniente debido al tamaño y características del sistema.

Proporciona:

- ORM.
- Migraciones.
- Autenticación.
- Seguridad.

- Validación.
- Administración.
- Manejo de archivos.
- Middleware.
- Sistema de permisos.

Esto permitirá reducir considerablemente la cantidad de funcionalidades básicas que tendrían que desarrollarse desde cero.

6.54 Justificación de Django REST Framework

DRF permitirá desarrollar una API REST estructurada.

Facilita:

- Serialización.
- Validación.
- Autenticación.
- Permisos.
- Paginación.
- Filtros.
- Endpoints.
- Manejo de errores.

Esto permitirá concentrar el esfuerzo en las reglas propias del instituto.

6.55 Justificación de MariaDB

MariaDB fue seleccionada debido a que SIGA manejará principalmente información relacional.

Ejemplos:

```

Estudiante
  ↓
Inscripción
  ↓
Pago
  ↓
Recibo

```

y:

```

Docente
  ↓
Curso
  ↓
Materia
  ↓

```

Asistencia

MariaDB proporciona:

- Transacciones.
- Claves foráneas.
- Índices.
- Integridad referencial.
- Consultas SQL.
- Buen rendimiento.
- Herramientas de respaldo.
- Compatibilidad con Django.

6.56 Documentación Técnica

La arquitectura deberá acompañarse de documentación técnica.

Como mínimo se deberá mantener:

- README.
- Configuración del entorno.
- Dependencias.
- Estructura del proyecto.
- Modelo de datos.
- Endpoints.
- Procedimiento de despliegue.
- Procedimiento de respaldo.
- Procedimiento de recuperación.

6.57 Documentación de API

Los endpoints desarrollados deberán documentarse.

Cuando sea posible se utilizará OpenAPI/Swagger.

La documentación deberá indicar:

- Método.
- URL.
- Parámetros.
- Autenticación.
- Permisos.
- Ejemplo de solicitud.
- Ejemplo de respuesta.
- Posibles errores.

6.58 Pruebas Arquitectónicas

Antes de considerar finalizada la arquitectura deberán probarse como mínimo:

- Comunicación React-Django.
- Conexión Django-MariaDB.
- Autenticación.
- Roles.
- Transacciones financieras.
- Carga de archivos.
- Impresión térmica.
- Integración QR.
- Respalos.
- Recuperación.

6.59 Criterios de Evolución

La arquitectura deberá permitir futuras ampliaciones como:

- Aplicación móvil.
- Facturación electrónica.
- WhatsApp.
- Correos automatizados.
- BI.
- Nuevas sucursales.
- Nuevos métodos de pago.
- Integraciones universitarias.

Estas funcionalidades no deberán condicionar excesivamente la primera versión, pero la arquitectura deberá evitar bloqueos innecesarios para incorporarlas posteriormente.

6.60 Cierre del Capítulo

La arquitectura definida para el Sistema Integral de Gestión Académica y Administrativa establece una estructura tecnológica basada en React, TypeScript, Python, Django, Django REST Framework y MariaDB, utilizando una separación clara entre presentación, lógica de negocio, persistencia e integraciones externas.

La utilización de una arquitectura modular permitirá desarrollar progresivamente los diferentes componentes del sistema, reduciendo el impacto de futuros cambios y facilitando el mantenimiento.

La combinación de Django y MariaDB proporcionará una base sólida para implementar los procesos administrativos, académicos y financieros, mientras que React permitirá ofrecer interfaces dinámicas orientadas a administrativos, docentes, estudiantes y directivos.

La separación mediante una API REST permitirá además que el backend pueda ser reutilizado posteriormente por otras interfaces, como aplicaciones móviles o integraciones externas.

Las decisiones arquitectónicas establecidas en el presente capítulo servirán como referencia durante las fases de diseño de base de datos, implementación, pruebas, despliegue y mantenimiento, y podrán ser refinadas cuando se disponga de mayor información sobre infraestructura, servicios financieros, impresoras térmicas y necesidades específicas de producción.

CAPÍTULO 7 DISEÑO Y MODELADO DE LA BASE DE DATOS

7.1 Introducción

El presente capítulo describe el diseño conceptual, lógico y preliminarmente físico de la base de datos correspondiente al Sistema Integral de Gestión Académica y Administrativa de SUMO (SIGA - SUMO).

La base de datos representa uno de los componentes centrales de la arquitectura del sistema, ya que será responsable de almacenar, relacionar, proteger y proporcionar la información utilizada por los diferentes módulos de la plataforma.

Debido a que SIGA – SUMO administrará procesos académicos, administrativos y financieros, el diseño de la base de datos deberá garantizar especialmente la consistencia, trazabilidad e integridad de la información.

Una estructura incorrecta de datos podría generar problemas como duplicidad de estudiantes, pagos inconsistentes, pérdida del historial académico, recibos sin relación con transacciones financieras, dificultades para generar reportes o inconsistencias entre las diferentes sucursales.

Por esta razón, el diseño se realizará antes de implementar los principales módulos del sistema.

La base de datos será implementada utilizando **MariaDB**, un sistema gestor de bases de datos relacional que proporciona soporte para transacciones, claves foráneas, índices, restricciones y mecanismos de integridad referencial.

El backend desarrollado con Django se comunicará con MariaDB utilizando el **Django ORM**, permitiendo representar las tablas mediante modelos orientados a objetos.

La base de datos deberá permitir almacenar información relacionada con:

- Usuarios.
- Roles.
- Permisos.

- Sucursales.
- Estudiantes.
- Tutores.
- Docentes.
- Áreas académicas.
- Cursos.
- Materias.
- Paralelos.
- Aulas.
- Horarios.
- Periodos académicos.
- Inscripciones.
- Planes de pago.
- Cuotas.
- Pagos.
- Recibos.
- Pagos QR.
- Ingresos.
- Egresos.
- Caja.
- Salarios docentes.
- Asistencia.
- Material académico.
- Prácticas.
- Entregas.
- Evaluaciones.
- Calificaciones.
- Archivos.
- Notificaciones.
- Auditoría.

El diseño presentado en este capítulo constituye una base inicial que podrá ajustarse durante la implementación cuando se detecten nuevas necesidades o reglas de negocio previamente no identificadas.

7.2 Objetivos del Diseño de Base de Datos

El objetivo principal del diseño de la base de datos consiste en proporcionar una estructura organizada, segura, consistente y escalable para almacenar toda la información utilizada por SIGA-SUMO.

Los objetivos específicos son los siguientes.

Centralización de Información

Toda la información generada por las diferentes sucursales deberá almacenarse dentro de una misma estructura centralizada.

Esto permitirá obtener una visión global de la institución y generar reportes consolidados.

Reducción de Duplicidad

La estructura deberá evitar almacenar innecesariamente información repetida.

Por ejemplo, una sucursal será registrada una sola vez y posteriormente relacionada con cursos, cajas e inscripciones.

Integridad de Información

Las relaciones entre las entidades deberán garantizar que los registros sean consistentes.

Por ejemplo, no deberá existir una inscripción correspondiente a un estudiante que no exista.

Trazabilidad

Los procesos importantes deberán permitir reconstruir posteriormente lo ocurrido.

Especialmente:

- Inscripciones.
- Pagos.
- Recibos.
- Anulaciones.
- Cierres de caja.
- Calificaciones.
- Cambios de estado.

Escalabilidad

La estructura deberá permitir incorporar nuevas sucursales, estudiantes, cursos, materias y periodos sin rediseñar completamente el sistema.

Soporte para Reportes

El modelo deberá permitir generar eficientemente consultas administrativas, académicas y financieras.

Protección de Historial

La información histórica no deberá desaparecer debido a modificaciones posteriores.

7.3 Niveles del Diseño de Datos

El diseño de la base de datos se dividirá en tres niveles principales.

7.3.1 Modelo Conceptual

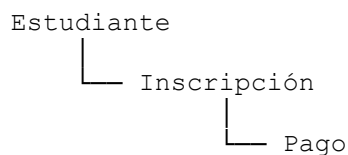
Representará las entidades principales del negocio y sus relaciones.

No incluirá inicialmente detalles técnicos como tipos de datos específicos.

Su objetivo será responder preguntas como:

- ¿Qué objetos existen dentro de la institución?
- ¿Qué relaciones tienen entre ellos?
- ¿Qué procesos generan información?

Ejemplo:



7.3.2 Modelo Lógico

En esta etapa se definirán:

- Entidades.
- Atributos.
- Claves primarias.
- Claves foráneas.
- Relaciones.
- Cardinalidades.
- Restricciones.

Ejemplo:

ESTUDIANTE

```
id_estudiante PK
ci UNIQUE
nombres
apellido_paterno
apellido_materno
```

7.3.3 Modelo Físico

Representará la implementación definitiva en MariaDB.

Aquí se establecerán:

- VARCHAR.
- BIGINT.
- DECIMAL.
- DATE.
- DATETIME.
- Índices.
- Restricciones UNIQUE.
- Claves foráneas.
- Reglas ON DELETE.
- Reglas ON UPDATE.

El modelo físico deberá ser generado principalmente mediante las migraciones de Django.

7.4 Principios de Diseño de Datos

7.4.1 Normalización

La base de datos será diseñada siguiendo los principios de normalización con el objetivo de reducir redundancia y evitar inconsistencias.

Como referencia general se buscará alcanzar hasta la **Tercera Forma Normal (3FN)** cuando resulte apropiado.

La normalización permitirá separar información que conceptualmente pertenece a diferentes entidades.

Ejemplo incorrecto:

INSCRIPCION

```
estudiante
curso
nombre_sucursal
direccion_sucursal
telefono_sucursal
```

En este diseño se repetiría la información de la sucursal en múltiples inscripciones.

Una mejor solución será:

SUCURSAL
id_sucursal
nombre
direccion
telefono

y:

INSCRIPCION
id_inscripcion
id_sucursal

7.4.2 Integridad Referencial

Las claves foráneas serán utilizadas para garantizar la existencia de las entidades relacionadas.

Por ejemplo:

ESTUDIANTE
|
1
|
N
INSCRIPCION

No se permitirá crear una inscripción cuyo estudiante no exista.

7.4.3 Identificadores Internos

Las tablas principales utilizarán identificadores numéricos independientes de los datos del negocio.

Ejemplo:

id_estudiante = 1458

Aunque posteriormente cambie:

- Nombre.
- Correo.
- Teléfono.

el identificador interno permanecerá constante.

7.4.4 No Utilizar CI como Clave Primaria

Aunque el CI permitirá identificar personas, no deberá utilizarse como clave primaria.

La razón es que podría:

- Corregirse.
- Modificarse.
- Tener extensiones.
- Presentar errores de registro.

Se utilizará:

`id_estudiante`

como identificador interno y:

`ci`

como dato único de negocio cuando corresponda.

7.5 Eliminación Física y Eliminación Lógica

Uno de los principios fundamentales del sistema será evitar la eliminación física de información histórica crítica.

Por ejemplo, si un estudiante deja de asistir al instituto, no deberá eliminarse de MariaDB.

Se cambiará su estado.

Ejemplo:

ACTIVO
INACTIVO
RETIRADO

Esto permitirá conservar:

- Inscripciones.
- Pagos.
- Calificaciones.
- Asistencia.
- Recibos.

La eliminación física podrá reservarse únicamente para registros que nunca hayan generado relaciones o que hayan sido creados por error antes de formar parte de procesos institucionales.

7.6 Auditoría de Datos

Las modificaciones críticas deberán permitir conocer el origen de los cambios.

Las principales entidades deberán considerar campos como:

created_at
updated_at
created_by
updated_by

Adicionalmente, existirá una tabla de auditoría independiente para operaciones críticas.

La auditoría permitirá responder preguntas como:

- ¿Quién anuló un recibo?
- ¿Quién modificó una calificación?
- ¿Quién cambió un pago?
- ¿Quién canceló una inscripción?

7.7 Organización de la Base de Datos por Dominios

Con el propósito de facilitar la comprensión del modelo, las entidades serán agrupadas por dominios funcionales.

SEGURIDAD

- Usuario
- Rol
- Permiso
- Auditoría

INSTITUCIÓN

- Sucursal
- Aula
- Configuración

PERSONAS

- Estudiante
- Tutor
- Docente

ACADÉMICO

- Área
- Curso
- Materia
- Paralelo
- Periodo
- Horario
- Asignación Docente

INSCRIPCIONES

- Inscripción
- Plan Pago
- Cuota

FINANZAS

- Pago
- Método Pago
- Recibo
- QR
- Movimiento Financiero
- Caja
- Pago Docente

SEGUIMIENTO ACADÉMICO

- Sesión Clase
- Asistencia
- Material
- Práctica
- Entrega
- Evaluación
- Calificación

Esta organización facilitará también la creación de aplicaciones internas dentro de Django.

7.8 Entidades de Seguridad

7.8.1 Tabla USUARIO

La tabla Usuario será responsable de almacenar las cuentas utilizadas para acceder al sistema.

El registro de una persona y el registro de una cuenta de usuario serán conceptos diferentes.

Por ejemplo, un estudiante podrá existir administrativamente antes de disponer de acceso al portal del estudiante.

De la misma forma, un docente podrá ser registrado antes de habilitar sus credenciales.

Campos propuestos

Campo	Tipo	Restricción	Descripción
id_usuario	BIGINT	PK	Identificador interno
username	VARCHAR(100)	UNIQUE	Usuario
email	VARCHAR(150)	UNIQUE	Correo de acceso
password	VARCHAR(255)	NOT NULL	Hash de contraseña
estado	VARCHAR(20)	NOT NULL	Estado de cuenta
ultimo_acceso	DATETIME	NULL	Último acceso
created_at	DATETIME	NOT NULL	Creación
updated_at	DATETIME	NOT NULL	Modificación

Django podrá utilizar un modelo personalizado basado en `AbstractUser` para evitar implementar desde cero los mecanismos de autenticación.

7.9 Roles y Permisos

7.9.1 Tabla ROL

Un rol representará un conjunto general de responsabilidades.

Roles iniciales:

- Administrador.
- Director.
- Administrativo.
- Caja.
- Docente.
- Estudiante.

No deberán codificarse rígidamente dentro del sistema.

Esto permitirá incorporar posteriormente roles como:

- Contador.
- Coordinador académico.
- Responsable de sucursal.

Campos

Campo	Tipo
id_rol	BIGINT
nombre	VARCHAR(100)
descripcion	VARCHAR(255)
estado	BOOLEAN
created_at	DATETIME
updated_at	DATETIME

7.9.2 Tabla PERMISO

Los permisos permitirán controlar acciones específicas.

Ejemplos:

```
ESTUDIANTE_VIEW  
ESTUDIANTE_CREATE  
ESTUDIANTE_UPDATE
```

```
PAGO_CREATE  
PAGO_CANCEL
```

```
RECIBO_PRINT  
RECIBO_CANCEL
```

```
CALIFICACION_CREATE  
CALIFICACION_UPDATE
```

Esto proporcionará mayor flexibilidad que trabajar únicamente con roles.

7.9.3 Tabla USUARIO_ROL

Permitirá que un usuario pueda disponer de más de un rol cuando sea necesario.

Por ejemplo:

Un usuario podría ser:

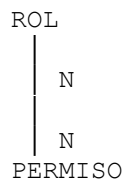
ADMINISTRATIVO
+
CAJA

sin necesidad de duplicar su cuenta.

7.9.4 Tabla ROL_PERMISO

Establecerá la relación entre roles y permisos.

Conceptualmente:



La existencia de esta relación permitirá modificar permisos sin modificar código.

7.10 Gestión de Sucursales

7.10.1 Tabla SUCURSAL

La tabla Sucursal representará las diferentes sedes físicas donde opera la institución.

Cada sucursal podrá disponer de:

- Cursos.
- Aulas.
- Inscripciones.
- Usuarios.
- Cajas.
- Movimientos financieros.

Campos propuestos

Campo	Tipo
id_sucursal	BIGINT
codigo	VARCHAR(30)
nombre	VARCHAR(150)
direccion	VARCHAR(255)
telefono	VARCHAR(30)
responsable	VARCHAR(150)
estado	BOOLEAN
created_at	DATETIME
updated_at	DATETIME

El código permitirá utilizar identificadores abreviados en reportes.

Ejemplo:

CEJA
UPEA
ILLIMANI

7.11 Gestión de Aulas

7.11.1 Tabla AULA

Cada aula pertenecerá a una sucursal.

Relación:

SUCURSAL 1 ————— N AULA

Campos

Campo	Tipo
id_aula	BIGINT
id_sucursal	BIGINT
nombre	VARCHAR(100)
capacidad	INT
descripcion	TEXT
estado	BOOLEAN

La capacidad permitirá controlar posteriormente asignaciones académicas.

7.12 Gestión de Estudiantes

7.12.1 Tabla ESTUDIANTE

La tabla Estudiante constituirá una de las principales entidades del sistema.

Deberá almacenar exclusivamente información propia del estudiante.

No deberá contener directamente:

- Curso actual.
- Monto pagado.
- Nota.
- Sucursal actual.

Estos datos deberán obtenerse mediante relaciones.

Campos propuestos

Campo	Tipo	Restricción
id_estudiante	BIGINT	PK
id_usuario	BIGINT	FK, NULL
apellido_paterno	VARCHAR(100)	NOT NULL
apellido_materno	VARCHAR(100)	NULL
nombres	VARCHAR(150)	NOT NULL
ci	VARCHAR(30)	UNIQUE
email	VARCHAR(150)	NULL
telefono	VARCHAR(30)	NULL
estado	VARCHAR(30)	NOT NULL
created_at	DATETIME	NOT NULL
updated_at	DATETIME	NOT NULL
created_by	BIGINT	FK

`id_usuario` podrá ser NULL cuando el estudiante todavía no tenga acceso al portal.

7.13 Gestión de Tutores

7.13.1 Tabla TUTOR

Inicialmente se solicitarán hasta dos tutores, pero el diseño no deberá limitar técnicamente al estudiante a exactamente dos.

Por ello se utilizará una tabla independiente.

Campos

Campo	Tipo
id_tutor	BIGINT
nombre_completo	VARCHAR(200)
telefono	VARCHAR(30)
email	VARCHAR(150)
parentesco	VARCHAR(100)
estado	BOOLEAN

7.13.2 Tabla ESTUDIANTE_TUTOR

Esta tabla permitirá implementar una relación muchos a muchos.

Un tutor podría incluso estar relacionado con más de un estudiante.

Ejemplo:

Tutor: María Pérez

├── Estudiante A
└── Estudiante B

Campos

Campo	Tipo
id_estudiante_tutor	BIGINT
id_estudiante	BIGINT
id_tutor	BIGINT
tipo	VARCHAR(30)
principal	BOOLEAN
estado	BOOLEAN

7.14 Gestión de Docentes

7.14.1 Tabla DOCENTE

La tabla Docente almacenará información personal y administrativa del personal académico.

Campos

Campo	Tipo
id_docente	BIGINT
id_usuario	BIGINT NULL
apellido_paterno	VARCHAR(100)
apellido_materno	VARCHAR(100)
nombres	VARCHAR(150)
ci	VARCHAR(30)
telefono	VARCHAR(30)
email	VARCHAR(150)
especialidad	VARCHAR(150)
estado	VARCHAR(30)
created_at	DATETIME
updated_at	DATETIME

La información salarial no deberá almacenarse directamente en esta tabla.

7.15 Estructura Académica

7.15.1 Tabla AREA

La tabla Área representará los programas generales de preparación ofrecidos por la institución.

Ejemplos:

Informática
Medicina
Ingeniería
Derecho

Campos

Campo	Tipo
id_area	BIGINT
codigo	VARCHAR(30)
nombre	VARCHAR(150)
descripcion	TEXT
estado	BOOLEAN

7.15.2 Tabla PERIODO_ACADEMICO

Permitirá organizar los cursos temporalmente.

Aunque actualmente algunos cursos funcionen por mensualidades, mantener periodos permitirá generar reportes históricos y organizar futuras gestiones académicas.

Campos

Campo	Tipo
id_periodo	BIGINT
nombre	VARCHAR(150)
fecha_inicio	DATE
fecha_fin	DATE
estado	VARCHAR(30)

7.16 Tabla CURSO

La tabla Curso representará un programa académico ofertado dentro de un determinado contexto.

Un curso estará relacionado con:

- Área.
- Sucursal.
- Periodo.

Campos

Campo	Tipo
id_curso	BIGINT
id_area	BIGINT
id_sucursal	BIGINT
id_periodo	BIGINT
codigo	VARCHAR(50)
nombre	VARCHAR(150)
descripcion	TEXT
capacidad	INT
precio_referencia	DECIMAL(12,2)
estado	VARCHAR(30)

El precio será únicamente una referencia, debido a que durante la inscripción podrá existir un monto diferente por descuentos o promociones.

7.17 Paralelos

7.17.1 Tabla PARALELO

Los paralelos permitirán dividir un curso cuando existan múltiples grupos.

Ejemplo:

Curso PSA Informática

```

├─ Paralelo A
├─ Paralelo B
└─ Paralelo C

```

Cada paralelo podrá tener:

- Capacidad.
- Horario.
- Docentes.

Campos

Campo	Tipo
id_paralelo	BIGINT
id_curso	BIGINT
codigo	VARCHAR(30)
nombre	VARCHAR(100)
capacidad	INT
estado	BOOLEAN

7.18 Gestión de Materias

7.18.1 Tabla MATERIA

Las materias no deberán depender directamente de un único curso.

Por ejemplo, Matemática podría formar parte de varios programas.

Campos

Campo	Tipo
id_materia	BIGINT
codigo	VARCHAR(50)
nombre	VARCHAR(150)
descripcion	TEXT
estado	BOOLEAN

7.18.2 Tabla CURSO_MATERIA

Permitirá relacionar múltiples cursos con múltiples materias.

Relación conceptual:

CURSO N ————— N MATERIA

Campos

Campo	Tipo
id_curso_materia	BIGINT
id_curso	BIGINT
id_materia	BIGINT
estado	BOOLEAN

Deberá existir una restricción para evitar registrar dos veces la misma combinación curso-materia.

7.19 Asignación Docente

7.19.1 Tabla ASIGNACION_DOCENTE

Esta tabla será fundamental para el Portal Docente.

Relacionará:

DOCENTE
+
CURSO
+
MATERIA
+
PARALELO

+

PERIODO

Campos

Campo	Tipo
id_asignacion	BIGINT
id_docente	BIGINT
id_curso_materia	BIGINT
id_paralelo	BIGINT
fecha_inicio	DATE
fecha_fin	DATE NULL
estado	BOOLEAN

El Portal Docente deberá consultar esta tabla para determinar qué información puede administrar cada profesor.

7.20 Gestión de Horarios

7.20.1 Tabla HORARIO

Representará los horarios correspondientes a las asignaciones docentes.

Campos

Campo	Tipo
id_horario	BIGINT
id_asignacion	BIGINT
id_aula	BIGINT
dia_semana	SMALLINT
hora_inicio	TIME
hora_fin	TIME
estado	BOOLEAN

El día de semana podría almacenarse numéricamente:

1 = lunes
2 = martes
3 = miércoles
...

Esto facilita consultas y ordenamiento.

Antes de crear un horario se deberán comprobar conflictos relacionados con:

- Docente.
- Aula.
- Paralelo.

7.21 Gestión de Inscripciones

7.21.1 Tabla INSCRIPCION

Esta será una de las tablas más importantes de toda la base de datos.

Representará el vínculo entre un estudiante y el servicio académico contratado.

Campos

Campo	Tipo
id_inscripcion	BIGINT
id_estudiante	BIGINT
id_paralelo	BIGINT
id_sucursal	BIGINT
fecha_inscripcion	DATETIME
fecha_inicio	DATE
fecha_fin	DATE
monto_base	DECIMAL(12,2)
descuento	DECIMAL(12,2)
monto_total	DECIMAL(12,2)
modalidad_pago	VARCHAR(30)
detalle	TEXT
estado	VARCHAR(30)
created_by	BIGINT
created_at	DATETIME
updated_at	DATETIME

Relación principal

ESTUDIANTE 1 ————— N INSCRIPCION

Esto permitirá que un estudiante tenga múltiples inscripciones históricas.

7.22 Cálculo de Fecha de Finalización

Aunque la fecha final será almacenada en la inscripción, deberá calcularse mediante lógica del backend.

Por ejemplo:

```
fecha_inicio = 18/08/2026
duracion = 1 mes
fecha_fin = 18/09/2026
```

No deberá calcularse directamente mediante una suma fija de 30 días, debido a que los meses tienen diferente duración.

Se utilizará lógica basada en meses calendario.

7.23 Gestión de Descuentos

Conviene considerar desde el diseño una estructura que permita ampliar descuentos.

Inicialmente podría almacenarse directamente en la inscripción.

Sin embargo, si la institución empieza a utilizar múltiples promociones, podrá implementarse posteriormente:

DESCUENTO

con:

- Tipo.
- Porcentaje.
- Monto.
- Motivo.
- Usuario autorizador.

Los descuentos aplicados deberán conservar el valor utilizado en el momento de la inscripción.

7.24 Plan de Pago

7.24.1 Tabla PLAN_PAGO

Representará la estructura financiera correspondiente a una inscripción.

Campos

Campo	Tipo
id_plan_pago	BIGINT
id_inscripcion	BIGINT
monto_total	DECIMAL(12,2)
cantidad_cuotas	INT
fecha_creacion	DATETIME
estado	VARCHAR(30)

Relación

INSCRIPCION 1 — 1 PLAN_PAGO

salvo que en el futuro se permita renegociar planes, en cuyo caso la relación podría pasar a ser histórica.

7.25 Tabla CUOTA

Las cuotas permitirán dividir el monto pendiente de una inscripción.

Campos

Campo	Tipo
id_cuota	BIGINT
id_plan_pago	BIGINT
numero	INT
fecha_vencimiento	DATE
monto	DECIMAL(12,2)
saldo	DECIMAL(12,2)
estado	VARCHAR(30)
created_at	DATETIME
updated_at	DATETIME

Ejemplo:

Inscripción Bs 900

Cuota 1 = Bs 300

Cuota 2 = Bs 300

Cuota 3 = Bs 300

Una cuota podrá recibir uno o más pagos parciales.

7.26 Gestión de Pagos

7.26.1 Tabla METODO_PAGO

Los métodos de pago deberán almacenarse de forma parametrizable.

Ejemplos:

- Efectivo.
- QR.
- Transferencia.

Esto permitirá agregar nuevos métodos sin modificar la estructura principal.

7.26.2 Tabla PAGO

Representará cada ingreso realizado por un estudiante.

Campos

Campo	Tipo
id_pago	BIGINT
id_inscripcion	BIGINT
id_cuota	BIGINT NULL
id_metodo_pago	BIGINT
monto	DECIMAL(12,2)
fecha_pago	DATETIME
referencia	VARCHAR(200) NULL
estado	VARCHAR(30)
usuario_registro	BIGINT
observacion	TEXT
created_at	DATETIME

Estados:

PENDIENTE
CONFIRMADO
ANULADO

Los pagos anulados deberán conservarse.

7.27 Relación Pago-Cuota

La relación entre pagos y cuotas merece especial atención.

Un pago podría:

- Cancelar una cuota completa.
- Cubrir parcialmente una cuota.
- Cubrir potencialmente varias cuotas.

Para la primera versión se podrá restringir un pago a una cuota específica.

Sin embargo, si la institución requiere distribuir un único pago entre varias cuotas, será recomendable implementar una tabla:

PAGO_CUOTA

con:

id_pago
id_cuota
monto_aplicado

Esto proporcionará mayor flexibilidad financiera.

7.28 Gestión de Recibos

7.28.1 Tabla RECIBO

El recibo representa el documento generado después de un pago confirmado.

Campos

Campo	Tipo
id_recibo	BIGINT
id_pago	BIGINT
numero_recibo	VARCHAR(50)
fecha_emision	DATETIME
estado	VARCHAR(30)
usuario_emision	BIGINT
fecha_anulacion	DATETIME NULL
usuario_anulacion	BIGINT NULL
motivo_anulacion	TEXT NULL

La relación inicial será:

PAGO 1 ——— 1 RECIBO

La reimpresión no generará un nuevo registro.

7.29 Numeración de Recibos

La numeración deberá ser única y controlada.

Se podrá utilizar una estructura como:

CEJA-0000125
UPEA-0000541
ILL-0000318

o una numeración institucional global.

La política definitiva deberá definirse con administración.

La generación deberá ejecutarse de forma segura para evitar duplicados cuando existan múltiples cajeros trabajando simultáneamente.

7.30 Anulación de Recibos

Un recibo anulado deberá conservar:

- Número original.
- Pago original.
- Fecha original.
- Motivo de anulación.

- Usuario responsable.
- Fecha de anulación.

No se deberá eliminar físicamente.

Cuando la anulación implique reversión del pago, esta operación deberá realizarse mediante una transacción controlada.

7.31 Transacciones QR

7.31.1 Tabla TRANSACCION_QR

Será responsable de almacenar la interacción con proveedores externos de pagos.

Campos

Campo	Tipo
id_transaccion_qr	BIGINT
id_inscripcion	BIGINT
id_pago	BIGINT NULL
identificador_externo	VARCHAR(200)
monto	DECIMAL(12,2)
qr_data	TEXT
fecha_generacion	DATETIME
fecha_expiracion	DATETIME NULL
fecha_confirmacion	DATETIME NULL
estado	VARCHAR(30)
proveedor	VARCHAR(100)
respuesta_proveedor	JSON/TEXT

El identificador externo deberá tener una restricción de unicidad cuando el proveedor lo garantice.

7.32 Flujo de Persistencia QR

Conceptualmente:

```

Crear solicitud QR
↓
TRANSACCION_QR = PENDIENTE
↓
Proveedor confirma
↓
TRANSACCION_QR = CONFIRMADA
↓
Crear PAGO
↓
Crear MOVIMIENTO_FINANCIERO
↓

```

Crear RECIBO

Estas operaciones deberán ejecutarse de forma controlada para evitar registrar dos pagos cuando el proveedor envíe más de una confirmación.

7.33 Gestión Financiera

7.33.1 Tabla CATEGORIA_FINANCIERA

Permitirá clasificar los movimientos.

Ejemplos de ingresos:

- Inscripciones.
- Mensualidades.
- Otros.

Ejemplos de egresos:

- Salarios.
- Alquiler.
- Servicios.
- Material.

Campos

Campo	Tipo
id_categoria	BIGINT
nombre	VARCHAR(150)
tipo	VARCHAR(20)
descripcion	TEXT
estado	BOOLEAN

7.34 Tabla MOVIMIENTO_FINANCIERO

Será la entidad contable-operativa central para representar ingresos y egresos.

Campos

Campo	Tipo
id_movimiento	BIGINT
id_categoria	BIGINT
id_sucursal	BIGINT
tipo	VARCHAR(20)
monto	DECIMAL(12,2)
fecha	DATETIME
concepto	VARCHAR(255)
origen_tipo	VARCHAR(50)

origen_id	BIGINT NULL
estado	VARCHAR(30)
usuario_registro	BIGINT
observacion	TEXT

Ejemplo:

TIPO: INGRESO
ORIGEN: PAGO_ESTUDIANTE
ORIGEN_ID: 1523
MONTO: Bs 300

Esto permitirá consolidar información financiera sin duplicar procesos.

7.35 Integración Pago-Movimiento Financiero

Cuando un pago sea confirmado:

PAGO
↓
MOVIMIENTO_FINANCIERO

No deberá registrarse manualmente el mismo ingreso nuevamente.

Esto evitará duplicidad financiera.

7.36 Gestión de Caja

7.36.1 Tabla CAJA

La caja representará un punto operativo donde se registran cobros.

Podrá existir una o varias cajas por sucursal.

Campos

Campo	Tipo
id_caja	BIGINT
id_sucursal	BIGINT
codigo	VARCHAR(30)
nombre	VARCHAR(100)
estado	BOOLEAN

7.36.2 Tabla SESION_CAJA

Cada apertura y cierre deberá generar una sesión independiente.

Campos

Campo	Tipo
id_sesion_caja	BIGINT
id_caja	BIGINT
usuario_apertura	BIGINT
fecha_apertura	DATETIME
monto_inicial	DECIMAL(12,2)
fecha_cierre	DATETIME NULL
usuario_cierre	BIGINT NULL
monto_esperado	DECIMAL(12,2) NULL
monto_declarado	DECIMAL(12,2) NULL
diferencia	DECIMAL(12,2) NULL
observacion_cierre	TEXT NULL
estado	VARCHAR(30)

Estados:

ABIERTA
CERRADA
ANULADA

7.37 Relación Pago-Caja

Los pagos presenciales deberán asociarse a una sesión de caja abierta.

Podría agregarse a PAGO:

id_sesion_caja

Esto permitirá saber exactamente qué caja y qué usuario procesaron una operación.

7.38 Gestión Salarial

7.38.1 Tabla TARIFA_DOCENTE

La remuneración no deberá almacenarse directamente en DOCENTE porque puede cambiar con el tiempo. Una tabla independiente permitirá conservar historial.

Campos

Campo	Tipo
id_tarifa	BIGINT
id_docente	BIGINT
tipo_pago	VARCHAR(50)
monto	DECIMAL(12,2)
fecha_inicio	DATE
fecha_fin	DATE NULL
estado	BOOLEAN

7.39 Tabla PAGO_DOCENTE

Representará los pagos realizados al docente.

Campos

Campo	Tipo
id_pago_docente	BIGINT
id_docente	BIGINT
fecha_desde	DATE
fecha_hasta	DATE
horas_planificadas	DECIMAL(8,2)
horas_reconocidas	DECIMAL(8,2)
monto	DECIMAL(12,2)
fecha_pago	DATETIME
estado	VARCHAR(30)
usuario_registro	BIGINT
observacion	TEXT

Al confirmarse:

PAGO_DOCENTE
↓
MOVIMIENTO_FINANCIERO (EGRESO)

7.40 Sesiones de Clase

7.40.1 Tabla SESION_CLASE

Una sesión representa una clase concreta impartida en determinada fecha.

No debe confundirse con HORARIO.

HORARIO representa:

Lunes 08:00 - 10:00

SESION_CLASE representa:

Lunes 24/08/2026
08:00 - 10:00

Campos

Campo	Tipo
id_sesion_clase	BIGINT
id_asignacion	BIGINT
fecha	DATE
hora_inicio	TIME
hora_fin	TIME

tema	VARCHAR(255) NULL
estado	VARCHAR(30)

7.41 Asistencia

7.41.1 Tabla ASISTENCIA

Cada estudiante podrá tener un único registro de asistencia por sesión.

Campos

Campo	Tipo
id_asistencia	BIGINT
id_sesion_clase	BIGINT
id_estudiante	BIGINT
estado	VARCHAR(30)
observacion	TEXT
usuario_registro	BIGINT
created_at	DATETIME
updated_at	DATETIME

Deberá existir una restricción:

```

UNIQUE (
    id_sesion_clase,
    id_estudiante
)

```

Esto evitará registrar dos asistencias para la misma clase.

7.42 Material Académico

7.42.1 Tabla MATERIAL

Representará contenido publicado por docentes.

Campos

Campo	Tipo
id_material	BIGINT
id_asignacion	BIGINT
titulo	VARCHAR(200)
descripcion	TEXT
fecha_publicacion	DATETIME
fecha_disponible	DATETIME
fecha_fin_disponibilidad	DATETIME NULL
estado	VARCHAR(30)
created_by	BIGINT

La relación con archivos deberá manejarse independientemente.

7.43 Gestión Centralizada de Archivos

7.43.1 Tabla ARCHIVO

Los archivos deberán almacenarse físicamente fuera de MariaDB.

La tabla únicamente contendrá metadatos.

Campos

Campo	Tipo
id_archivo	BIGINT
nombre_original	VARCHAR(255)
nombre_sistema	VARCHAR(255)
ruta	VARCHAR(500)
mime_type	VARCHAR(100)
extension	VARCHAR(20)
tamano_bytes	BIGINT
usuario_carga	BIGINT
created_at	DATETIME

Posteriormente pueden utilizarse tablas intermedias como:

```
MATERIAL_ARCHIVO  
PRACTICA_ARCHIVO  
ENTREGA_ARCHIVO
```

Esto resulta más normalizado que agregar múltiples campos de archivos directamente a cada tabla.

7.44 Prácticas

7.44.1 Tabla PRACTICA

Representará actividades asignadas a estudiantes.

Campos

Campo	Tipo
id_practica	BIGINT
id_asignacion	BIGINT
titulo	VARCHAR(200)
descripcion	TEXT
fecha_publicacion	DATETIME
fecha_limite	DATETIME
puntaje_maximo	DECIMAL(8,2)

permite_tardias	BOOLEAN
permite_reemplazo	BOOLEAN
estado	VARCHAR(30)
created_by	BIGINT

7.45 Entregas

7.45.1 Tabla ENTREGA

Representará una entrega realizada por un estudiante.

Campos

Campo	Tipo
id_entrega	BIGINT
id_practica	BIGINT
id_estudiante	BIGINT
numero_intento	INT
fecha_entrega	DATETIME
estado	VARCHAR(30)
comentario_estudiante	TEXT
created_at	DATETIME

Si se permiten múltiples intentos, cada entrega puede conservarse en lugar de sobrescribir la anterior.

Esto proporciona mayor trazabilidad.

7.46 Evaluaciones

7.46.1 Tabla TIPO_EVALUACION

Permitirá parametrizar:

- Examen.
- Práctica.
- Participación.
- Proyecto.

7.46.2 Tabla EVALUACION

Representará una actividad evaluable.

Campos

Campo	Tipo
id_evaluacion	BIGINT
id_asignacion	BIGINT

id_tipo_evaluacion	BIGINT
titulo	VARCHAR(200)
descripcion	TEXT
puntaje_maximo	DECIMAL(8,2)
ponderacion	DECIMAL(5,2)
fecha	DATE
publicada	BOOLEAN
estado	VARCHAR(30)

La suma de ponderaciones deberá validarse según la regla académica definida.

7.47 Calificaciones

7.47.1 Tabla CALIFICACION

Relacionará estudiante y evaluación.

Campos

Campo	Tipo
id_calificacion	BIGINT
id_evaluacion	BIGINT
id_estudiante	BIGINT
nota	DECIMAL(8,2)
retroalimentacion	TEXT
publicada	BOOLEAN
usuario_registro	BIGINT
created_at	DATETIME
updated_at	DATETIME

Deberá establecerse:

```

UNIQUE (
    id_evaluacion,
    id_estudiante
)

```

salvo que posteriormente se requiera guardar múltiples versiones mediante una tabla histórica.

7.48 Historial de Calificaciones

Debido a que las calificaciones pueden ser modificadas, se recomienda crear:

HISTORIAL_CALIFICACION

Campos

Campo	Tipo
id_historial	BIGINT

id_calificacion	BIGINT
nota_anterior	DECIMAL(8,2)
nota_nueva	DECIMAL(8,2)
motivo	TEXT
usuario_modificacion	BIGINT
fecha_modificacion	DATETIME

Esto permitirá realizar auditorías académicas.

7.49 Notificaciones

7.49.1 Tabla NOTIFICACION

Permitirá gestionar avisos internos.

Campos

Campo	Tipo
id_notificacion	BIGINT
id_usuario	BIGINT
tipo	VARCHAR(50)
titulo	VARCHAR(200)
mensaje	TEXT
url_destino	VARCHAR(500) NULL
leida	BOOLEAN
created_at	DATETIME
fecha_lectura	DATETIME NULL

7.50 Auditoría

7.50.1 Tabla AUDITORIA

La auditoría será uno de los componentes más críticos del sistema.

Campos

Campo	Tipo
id_auditoria	BIGINT
id_usuario	BIGINT NULL
modulo	VARCHAR(100)
accion	VARCHAR(100)
entidad	VARCHAR(100)
id_entidad	BIGINT NULL
datos_anteriores	JSON/TEXT NULL
datos_nuevos	JSON/TEXT NULL
direccion_ip	VARCHAR(50)
user_agent	TEXT NULL
descripcion	TEXT
created_at	DATETIME

La tabla no deberá ser modificable por usuarios operativos.

7.51 Índices

Los índices son esenciales para mantener un buen rendimiento cuando la base de datos crezca.

Se establecerán índices en campos utilizados frecuentemente para:

- Búsquedas.
- Ordenamientos.
- Filtros.
- Relaciones.

Ejemplos:

```
estudiante(ci)
```

```
inscripcion(fecha_inscripcion)
```

```
inscripcion(id_estudiante, estado)
```

```
pago(fecha_pago)
```

```
pago(id_inscripcion)
```

```
recibo(numero_recibo)
```

```
transaccion_qr(identificador_externo)
```

```
movimiento_financiero(fecha, tipo)
```

```
asistencia(id_estudiante)
```

```
calificacion(id_estudiante)
```

No deberán agregarse índices innecesariamente, ya que también tienen un costo durante inserciones y actualizaciones.

7.52 Restricciones UNIQUE

Las restricciones únicas serán utilizadas para garantizar reglas estructurales.

Ejemplos:

```
usuario.username
```

```
usuario.email
```

```
estudiante.ci
```

`docente.ci`

`recibo.numero_recibo`

`transaccion_gr.identificador_externo`

También existirán restricciones compuestas.

Ejemplo:

```
UNIQUE (  
    curso,  
    materia  
)
```

dentro de CURSO_MATERIA.

7.53 Tipos Monetarios

Todos los montos financieros deberán almacenarse mediante:

`DECIMAL`

Nunca mediante:

`FLOAT`

debido a problemas de precisión.

Un tamaño inicial adecuado podría ser:

`DECIMAL(12,2)`

permitiendo valores de hasta:

`9,999,999,999.99`

si la configuración del gestor y el diseño concreto lo requieren.

7.54 Estados

Las entidades principales utilizarán estados para representar su ciclo de vida.

Ejemplo:

Inscripción

PENDIENTE
ACTIVA
FINALIZADA

CANCELADA
ANULADA

Pago

PENDIENTE
CONFIRMADO
ANULADO

QR

PENDIENTE
CONFIRMADO
RECHAZADO
EXPIRADO

Recibo

EMITIDO
ANULADO

Se recomienda representar los estados mediante enumeraciones en Django.

7.55 Reglas ON DELETE

Las relaciones deberán configurarse cuidadosamente.

No todas las claves foráneas deberán utilizar:

CASCADE

Por ejemplo, eliminar un estudiante no deberá provocar automáticamente la eliminación de todos sus pagos.

Para información histórica podrían utilizarse estrategias como:

- PROTECT.
- RESTRICT.
- SET NULL.
- Eliminación lógica.

La estrategia deberá definirse por relación.

7.56 Migraciones de Django

La estructura física deberá gestionarse mediante migraciones.

Flujo:

```
Modificar models.py
↓
python manage.py makemigrations
↓
Revisión de migración
↓
```

```
python manage.py migrate
↓
MariaDB
```

Las migraciones deberán incluirse en Git.

No deberán realizarse modificaciones manuales no documentadas en producción salvo situaciones excepcionales.

7.57 Convenciones de Nombres

Se establecerá una convención uniforme.

Ejemplo de nombres de tablas:

```
estudiante
inscripcion
pago
recibo
```

Django podrá gestionar nombres automáticamente, pero se deberán mantener convenciones coherentes.

Campos:

```
created_at
updated_at
fecha_inicio
fecha_fin
```

Las abreviaturas innecesarias deberán evitarse.

7.58 Integridad Transaccional

Los procesos críticos utilizarán transacciones.

Ejemplo:

```
BEGIN

INSERT pago

UPDATE cuota

UPDATE plan_pago

INSERT movimiento_financiero

INSERT recibo

INSERT auditoria
```

COMMIT

En caso de error:

ROLLBACK

Esto será obligatorio especialmente para:

- Pagos.
- Anulaciones.
- QR.
- Cierres.
- Salarios.

7.59 Concurrencia

El sistema deberá considerar múltiples usuarios trabajando simultáneamente.

Ejemplo:

Dos administrativos podrían intentar registrar una inscripción en el último cupo disponible.

El backend deberá revalidar la capacidad del curso durante la transacción.

Otro ejemplo:

Dos cajeros no deberán generar el mismo número de recibo.

La numeración deberá administrarse de forma atómica.

7.60 Seguridad de la Base de Datos

MariaDB deberá operar dentro de una red controlada.

No deberá exponerse públicamente a Internet salvo una necesidad técnica excepcional y debidamente protegida.

Se utilizará un usuario de aplicación con permisos únicamente sobre la base de datos de SIGA.

Las credenciales deberán mantenerse fuera del repositorio.

7.61 Copias de Seguridad

Los respaldos deberán incluir:

- Base de datos.
- Archivos académicos.
- Documentación necesaria.

Se establecerá una política de respaldo.

Por ejemplo:

Diario:
copia de base de datos

Semanal:
respaldo completo

Mensual:
copia histórica

La política definitiva dependerá de la infraestructura seleccionada.

7.62 Restauración

Un respaldo únicamente tiene valor si puede restaurarse.

Por esta razón deberán realizarse pruebas periódicas de restauración.

El procedimiento deberá documentar:

1. Detener operaciones críticas.
2. Seleccionar respaldo.
3. Restaurar base de datos.
4. Restaurar archivos.
5. Verificar integridad.
6. Verificar acceso.
7. Reactivar sistema.

7.63 Retención de Información

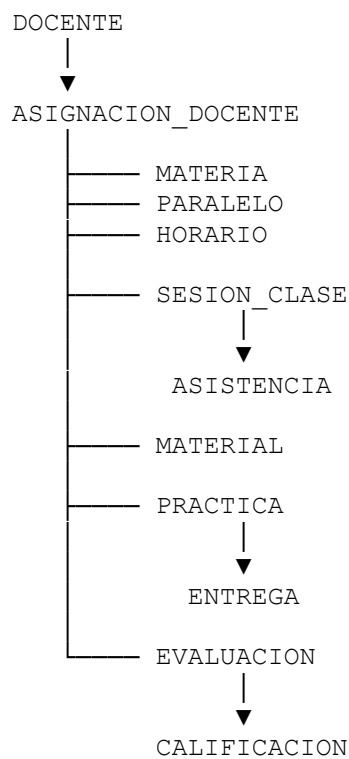
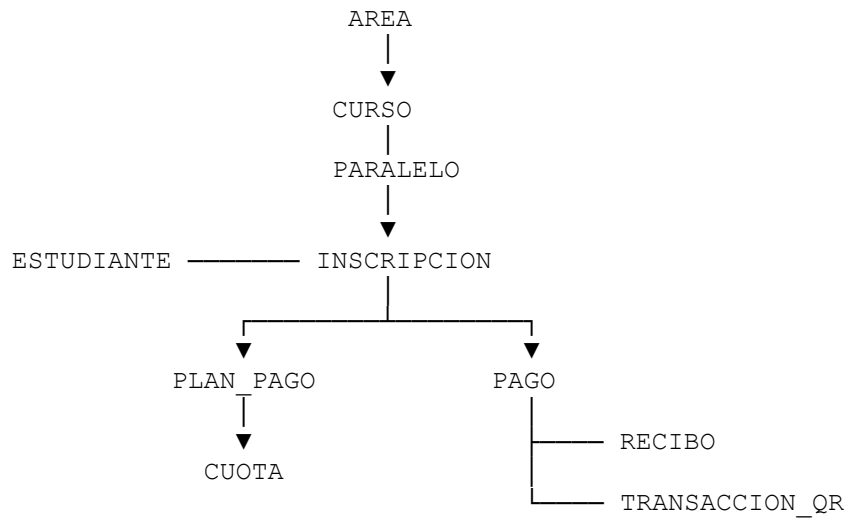
Deberá definirse cuánto tiempo se conservarán determinados registros.

La información histórica académica y financiera no deberá eliminarse automáticamente sin una política institucional explícita.

La auditoría podrá requerir una política específica de conservación.

7.64 Modelo Entidad-Relación General

El núcleo principal puede representarse conceptualmente así:



7.65 Escenarios de Validación del Modelo

Antes de implementar masivamente el sistema, el modelo deberá ser probado mediante escenarios reales.

Escenario A – Inscripción con pago total

Registrar estudiante
↓
Registrar inscripción
↓

Registrar pago completo
↓
Actualizar saldo
↓
Generar movimiento financiero
↓
Generar recibo

Escenario B – Pago en cuotas

Inscripción Bs 1.200

↓
Plan 3 cuotas
↓
 $400 + 400 + 400$

↓
Registrar primer pago
↓
Actualizar saldo

Escenario C – Anulación

Pago confirmado

↓
Recibo emitido
↓
Solicitud de anulación
↓
Registrar motivo
↓
Anular recibo

↓
Revertir pago
↓
Revertir movimiento
↓
Actualizar saldo

Escenario D – Docente

Asignar docente

↓
Registrar horario
↓
Crear sesión
↓
Pasar asistencia
↓
Publicar práctica
↓
Estudiante entrega
↓
Docente califica

Escenario E – QR

Generar QR

↓
Registrar solicitud pendiente
↓
Proveedor confirma
↓
Crear pago

↓
Crear ingreso
↓
Crear recibo

7.66 Diccionario de Datos

Después de validar el modelo deberá elaborarse un diccionario detallado.

Por cada campo se registrará:

Elemento	Descripción
Tabla	Entidad correspondiente
Campo	Nombre físico
Descripción	Significado
Tipo	Tipo de dato
Longitud	Tamaño
NULL	Si acepta vacío
PK	Clave primaria
FK	Clave foránea
UNIQUE	Restricción de unicidad
Default	Valor predeterminado
Validación	Regla aplicable
Observaciones	Información adicional

El diccionario servirá como referencia para desarrollo, pruebas y mantenimiento.

7.67 Consideraciones de Rendimiento

Aunque inicialmente SIGA manejará aproximadamente cientos de estudiantes, el diseño deberá prever un crecimiento considerable.

Las principales técnicas serán:

- Índices.
- Paginación.
- Consultas optimizadas.
- Evitar SELECT innecesarios.
- Utilizar relaciones adecuadamente.
- Optimizar consultas del ORM.
- Utilizar `select_related`.
- Utilizar `prefetch_related`.

Los reportes complejos deberán analizarse mediante consultas optimizadas.

7.68 Datos Derivados

No toda información deberá almacenarse.

Por ejemplo:

`saldo pendiente`

podría calcularse desde:

`monto_total - pagos_confirmados`

Sin embargo, en determinados casos podría mantenerse un campo de saldo para optimización.

Cuando se utilicen datos derivados almacenados deberá garantizarse su actualización transaccional para evitar inconsistencias.

7.69 Catálogos y Parámetros

Los valores configurables deberán evitar almacenarse rígidamente dentro del código.

Se podrán utilizar tablas para:

- Métodos de pago.
- Categorías financieras.
- Tipos de evaluación.
- Estados configurables.
- Configuraciones institucionales.

Esto permitirá modificar comportamientos sin desplegar una nueva versión en determinados casos.

7.70 Información Sensible

La base de datos contendrá información que deberá tratarse con especial cuidado.

Entre ella:

- Datos personales.
- Información financiera.
- Salarios.
- Calificaciones.
- Información de contacto.

El acceso deberá estar controlado por la aplicación.

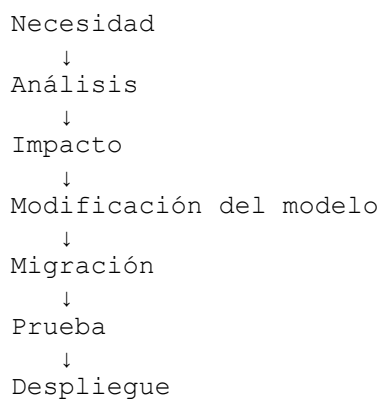
Los respaldos también deberán considerarse información sensible.

7.71 Evolución del Modelo

El modelo presentado no deberá considerarse completamente inmutable.

Conforme avance el desarrollo podrán identificarse nuevas entidades.

Toda modificación deberá pasar por:



Esto evitará modificaciones improvisadas.

7.72 Cierre del Capítulo

El diseño de base de datos presentado establece la estructura fundamental que permitirá implementar los procesos académicos, administrativos y financieros del Sistema Integral de Gestión Académica y Administrativa de Sumo.

El modelo se ha diseñado considerando normalización, integridad referencial, trazabilidad, seguridad, escalabilidad y rendimiento, con especial atención a los procesos críticos de inscripción, pagos, recibos, transacciones QR, caja, salarios, asistencia y calificaciones.

La separación de la información en diferentes dominios permitirá mantener una arquitectura organizada y facilitará la correspondencia entre los modelos de Django y los módulos funcionales definidos en el sistema.

Asimismo, la incorporación de mecanismos como claves foráneas, restricciones únicas, índices, transacciones, eliminación lógica y auditoría permitirá reducir inconsistencias y mantener información histórica necesaria para la operación institucional.

Antes de implementar completamente la base de datos en MariaDB, el modelo deberá ser validado utilizando escenarios reales del instituto, especialmente aquellos relacionados con pagos parciales, cuotas, anulaciones, cambios de curso, cierres de caja y pagos QR.

Una vez validado, este modelo servirá como base para la creación de los modelos de Django, las migraciones, la API REST, los servicios del backend y las interfaces desarrolladas en React.

El modelo deberá mantenerse actualizado durante todo el desarrollo, asegurando que cualquier modificación estructural sea documentada, versionada y aplicada mediante mecanismos controlados.

CAPÍTULO 8 CASOS DE USO Y FLUJOS DEL SISTEMA

8.1 Introducción

El presente capítulo describe los casos de uso principales del Sistema Integral de Gestión Académica y Administrativa de Sumo (SIGA – SUMO), identificando los actores que interactúan con la plataforma, los objetivos de cada proceso, las condiciones necesarias para su ejecución y los resultados esperados.

Los casos de uso permiten representar de manera estructurada la interacción entre los usuarios y el sistema, facilitando la comprensión de los procesos funcionales y sirviendo como base para el diseño de interfaces, desarrollo de funcionalidades y elaboración de pruebas.

Cada caso de uso será identificado mediante el código **CU-XXX**.

Los principales actores identificados son:

- Administrador del Sistema.
- Director.
- Personal Administrativo.
- Responsable de Caja.
- Responsable Financiero.
- Docente.
- Estudiante.
- Servicios externos de pago QR.

8.2 Estructura de los Casos de Uso

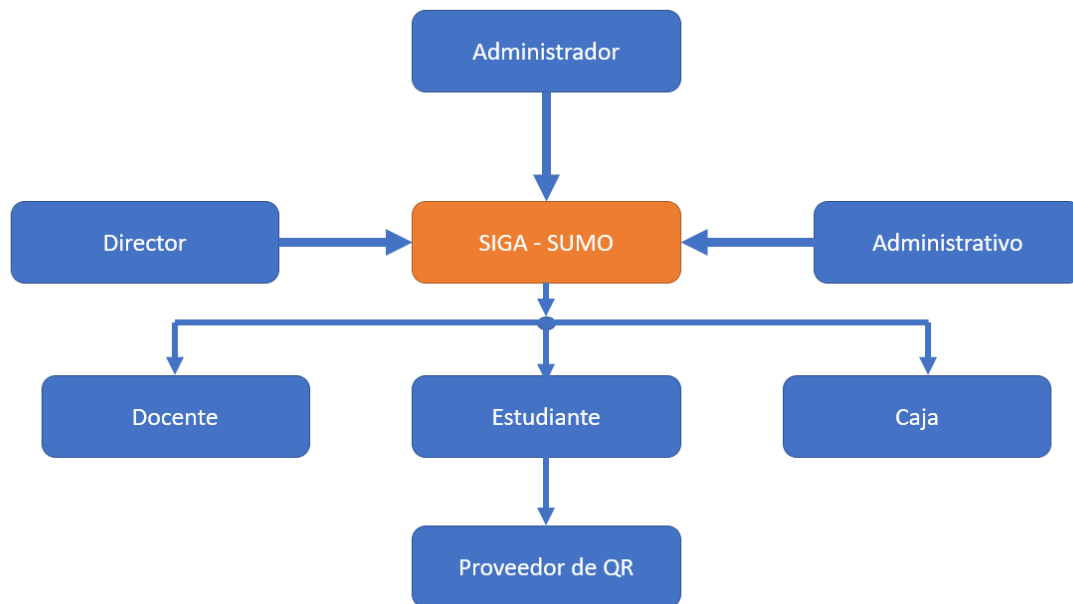
Cada caso de uso será documentado considerando los siguientes elementos:

- Código.
- Nombre.
- Objetivo.
- Actor principal.
- Actores secundarios.
- Precondiciones.
- Disparador.
- Flujo principal.
- Flujos alternativos.
- Excepciones.
- Postcondiciones.
- Reglas de negocio asociadas.
- Requerimientos funcionales relacionados.
- Prioridad.

Esta estructura permitirá mantener trazabilidad entre el análisis, el desarrollo y las pruebas.

8.3 Diagrama General de Actores

Conceptualmente, los actores interactúan con el sistema de la siguiente manera:



8.4 CU-001 – Iniciar Sesión

Objetivo: Permitir que un usuario autorizado acceda al sistema.

Actor principal: Usuario.

Precondiciones:

- El usuario debe estar registrado.
- La cuenta debe encontrarse activa.

Disparador: El usuario ingresa al portal SIGA.

Flujo principal:

1. El usuario accede a la pantalla de inicio de sesión.
2. Ingresa usuario o correo electrónico.
3. Ingresa contraseña.
4. El sistema valida las credenciales.
5. El sistema verifica el estado de la cuenta.
6. El sistema identifica roles y permisos.
7. Se crea la sesión.
8. El usuario es dirigido al Dashboard correspondiente.

Flujos alternativos:

- Credenciales incorrectas.
- Cuenta inactiva.
- Cuenta bloqueada.

Postcondiciones: El usuario accede a las funcionalidades permitidas.

Prioridad: Crítica.

8.5 CU-002 – Registrar Estudiante

Objetivo: Registrar un nuevo estudiante en la institución.

Actor principal: Personal Administrativo.

Precondiciones:

- El usuario debe estar autenticado.
- Debe contar con permiso de registro.

Flujo principal:

1. El administrativo ingresa al módulo de estudiantes.
2. Selecciona "Nuevo estudiante".
3. Registra apellido paterno.
4. Registra apellido materno.
5. Registra nombres.

6. Registra CI.
7. Registra correo electrónico.
8. Registra teléfono.
9. Registra información del Tutor 1.
10. Registra información del Tutor 2, si corresponde.
11. El sistema valida campos obligatorios.
12. El sistema verifica que el CI no se encuentre registrado.
13. El usuario confirma.
14. El sistema almacena el estudiante.
15. Se registra la operación en auditoría.

Flujo alternativo – Estudiante existente:

1. El sistema detecta el CI.
2. Muestra el expediente existente.
3. No genera un segundo registro.

Postcondiciones: El estudiante queda disponible para inscripciones.

Reglas relacionadas: RN-001, RN-002.

8.6 CU-003 – Modificar Estudiante

Objetivo: Actualizar información personal o de contacto.

Actor principal: Administrativo.

Flujo principal:

1. Buscar estudiante.
2. Abrir expediente.
3. Seleccionar editar.
4. Modificar información autorizada.
5. Validar.
6. Guardar cambios.
7. Registrar auditoría.

Excepción: Los datos históricos de pagos, inscripciones y calificaciones no podrán modificarse desde esta operación.

8.7 CU-004 – Registrar Inscripción

Objetivo: Inscribir oficialmente a un estudiante en un curso.

Actor principal: Personal Administrativo.

Prioridad: Crítica.

Precondiciones:

- Estudiante registrado.
- Curso activo.
- Horario disponible.
- Cupo disponible.

Flujo principal:

1. El usuario busca al estudiante.
2. Selecciona la opción "Nueva inscripción".
3. Selecciona sucursal.
4. Selecciona área académica.
5. Selecciona curso.
6. Selecciona paralelo.
7. Selecciona horario.
8. Registra fecha de inicio.
9. El sistema calcula la fecha de finalización.
10. Se registra el monto del curso.
11. Se define descuento, cuando corresponda.
12. Se selecciona modalidad de pago.
13. El sistema calcula el total.
14. Se verifica disponibilidad de cupo.
15. El usuario confirma la inscripción.
16. El sistema registra la inscripción.
17. Se genera la obligación financiera.
18. Si existe pago inicial, se inicia el proceso de pago.
19. Se registra auditoría.

Flujos alternativos:

- Curso sin cupo.
- Inscripción duplicada.
- Curso inactivo.
- Información obligatoria incompleta.

Postcondiciones: Se crea una inscripción activa o pendiente.

8.8 CU-005 – Calcular Fecha Final de Inscripción

Objetivo: Determinar automáticamente la vigencia contratada.

Actor principal: Sistema.

Flujo principal:

1. Recibir fecha inicial.
2. Determinar duración.
3. Añadir el número de meses calendario correspondientes.
4. Resolver fechas límite en meses con distinta cantidad de días.
5. Devolver fecha final.

Ejemplo:

Inicio: 31/01/2026

Duración: 1 mes

Resultado esperado según regla definida: último día válido de febrero

La regla definitiva deberá ser validada institucionalmente.

8.9 CU-006 – Registrar Pago en Efectivo

Objetivo: Registrar un pago realizado por el estudiante.

Actor principal: Responsable de Caja.

Precondiciones:

- Inscripción válida.
- Saldo pendiente.
- Sesión de caja abierta.

Flujo principal:

1. Buscar estudiante o inscripción.
2. Consultar saldo.
3. Seleccionar concepto.
4. Introducir monto.
5. Seleccionar método Efectivo.
6. Validar que el monto sea permitido.
7. Confirmar.
8. Registrar pago.
9. Aplicar monto a la obligación correspondiente.
10. Actualizar saldo.
11. Registrar movimiento financiero.
12. Generar recibo.
13. Enviar recibo a impresión térmica.
14. Registrar auditoría.

Postcondiciones:

- Pago confirmado.
- Saldo actualizado.
- Ingreso registrado.
- Recibo emitido.

8.10 CU-007 – Registrar Pago Parcial

Objetivo: Permitir que un estudiante pague solo una parte de su obligación.

Flujo principal:

1. Consultar obligación.
2. Mostrar saldo pendiente.
3. Registrar monto parcial.
4. Validar que sea mayor a cero.
5. Confirmar pago.
6. Aplicar pago.
7. Actualizar saldo.
8. Generar recibo.

Postcondición: La obligación queda en estado parcialmente pagado si aún existe saldo.

8.11 CU-008 – Registrar Pago en Cuotas

Objetivo: Administrar una inscripción dividida en varias cuotas.

Actor principal: Administrativo / Caja.

Flujo principal:

1. Seleccionar pago en cuotas.
2. Registrar cantidad de cuotas.
3. Definir montos.
4. Definir fechas de vencimiento.
5. Crear plan de pago.
6. Crear cuotas.
7. Registrar posteriormente los pagos de cada cuota.
8. Actualizar el estado del plan.

8.12 CU-009 – Generar Pago mediante QR

Objetivo: Permitir que un estudiante realice el pago mediante código QR.

Actores:

- Caja.

- Estudiante.
- Proveedor financiero.

Prioridad: Crítica.

Flujo principal:

1. Caja selecciona la obligación.
2. Selecciona método QR.
3. El backend solicita al proveedor generar una transacción.
4. El proveedor devuelve el QR y su identificador.
5. SIGA registra la transacción como pendiente.
6. El QR se muestra al estudiante.
7. El estudiante realiza el pago.
8. El proveedor procesa la operación.
9. Se recibe confirmación.
10. SIGA valida el identificador y monto.
11. La transacción pasa a CONFIRMADA.
12. Se registra el pago.
13. Se actualiza saldo.
14. Se registra ingreso.
15. Se genera recibo.

Flujos alternativos:

- QR expirado.
- Pago rechazado.
- Monto incorrecto.
- Falta de respuesta del proveedor.
- Confirmación duplicada.

8.13 CU-010 – Confirmar Manualmente un Pago QR

Objetivo: Permitir confirmar un pago cuando no exista integración automática.

Actor principal: Usuario autorizado.

Flujo principal:

1. Buscar solicitud QR.
2. Consultar estado.
3. Revisar comprobante o evidencia.
4. Confirmar manualmente.
5. Registrar usuario responsable.
6. Registrar pago.
7. Generar ingreso.

8. Generar recibo.
9. Registrar auditoría.

8.14 CU-011 – Emitir Recibo

Objetivo: Generar el comprobante correspondiente a un pago válido.

Actor principal: Sistema.

Flujo principal:

1. Recibir información del pago.
2. Generar número único.
3. Asociar recibo al pago.
4. Registrar fecha.
5. Almacenar comprobante.
6. Preparar formato de impresión.

8.15 CU-012 – Imprimir Recibo Térmico

Objetivo: Imprimir el comprobante mediante la impresora configurada.

Actor principal: Caja.

Precondiciones:

- Recibo emitido.
- Impresora disponible.

Flujo principal:

1. Visualizar recibo.
2. Seleccionar imprimir.
3. El sistema prepara el formato.
4. Se envía el trabajo de impresión.
5. Se registra la impresión.

Excepciones:

- Impresora desconectada.
- Driver incorrecto.
- Error del dispositivo.

Un fallo de impresión no deberá duplicar el pago ni el recibo.

8.16 CU-013 – Reimprimir Recibo

Objetivo: Obtener una nueva impresión de un comprobante existente.

Flujo principal:

1. Buscar recibo.
2. Verificar estado.
3. Seleccionar reimprimir.
4. Imprimir el mismo número.
5. Registrar auditoría de reimpresión.

8.17 CU-014 – Anular Recibo

Objetivo: Invalidar un comprobante emitido incorrectamente.

Actor principal: Usuario autorizado.

Prioridad: Crítica.

Precondiciones:

- Recibo existente.
- Usuario autorizado.

Flujo principal:

1. Buscar recibo.
2. Seleccionar anular.
3. Registrar motivo obligatorio.
4. Solicitar confirmación.
5. Validar permisos.
6. Cambiar estado a ANULADO.
7. Ejecutar la reversión financiera definida.
8. Actualizar saldo, si corresponde.
9. Registrar auditoría.

Postcondición: La operación original permanece en el historial.

8.18 CU-015 – Abrir Caja

Objetivo: Iniciar las operaciones financieras de una caja.

Actor principal: Cajero.

Flujo principal:

1. Seleccionar caja.

2. Registrar monto inicial.
3. Confirmar apertura.
4. Crear sesión de caja.
5. Registrar usuario y fecha.

8.19 CU-016 – Cerrar Caja

Objetivo: Finalizar una jornada o sesión de caja.

Flujo principal:

1. Solicitar cierre.
2. Calcular movimientos.
3. Mostrar efectivo esperado.
4. Mostrar pagos electrónicos.
5. Cajero registra efectivo contado.
6. Calcular diferencia.
7. Solicitar justificación cuando exista diferencia.
8. Confirmar cierre.
9. Bloquear nuevos movimientos sobre la sesión.

8.20 CU-017 – Registrar Egreso

Objetivo: Registrar un gasto institucional.

Actor principal: Responsable Financiero.

Flujo principal:

1. Ingresar al módulo financiero.
2. Seleccionar nuevo egreso.
3. Seleccionar categoría.
4. Registrar concepto.
5. Registrar monto.
6. Seleccionar sucursal.
7. Agregar observaciones.
8. Confirmar.
9. Crear movimiento financiero.
10. Registrar auditoría.

8.21 CU-018 – Registrar Pago a Docente

Objetivo: Registrar la remuneración correspondiente a un docente.

Actor principal: Responsable Financiero.

Flujo principal:

1. Buscar docente.
2. Seleccionar periodo.
3. Consultar modalidad de remuneración.
4. Registrar o calcular horas reconocidas.
5. Calcular monto.
6. Verificar información.
7. Confirmar pago.
8. Registrar pago docente.
9. Crear movimiento de egreso.
10. Generar comprobante interno.
11. Registrar auditoría.

8.22 CU-019 – Crear Curso

Objetivo: Registrar una nueva oferta académica.

Actor principal: Administrador / Académico.

Flujo principal:

1. Seleccionar área.
2. Seleccionar sucursal.
3. Seleccionar periodo.
4. Registrar nombre.
5. Definir capacidad.
6. Registrar precio de referencia.
7. Asociar materias.
8. Crear paralelos, cuando corresponda.
9. Guardar.

8.23 CU-020 – Asignar Docente

Objetivo: Asignar un profesor a un curso y materia.

Actor principal: Administrador académico.

Flujo principal:

1. Seleccionar curso.
2. Seleccionar materia.
3. Seleccionar paralelo.
4. Seleccionar docente.
5. Definir vigencia.
6. Verificar conflicto.

7. Confirmar asignación.

8.24 CU-021 – Configurar Horario

Objetivo: Establecer el horario académico.

Flujo principal:

1. Seleccionar asignación.
2. Seleccionar día.
3. Definir hora inicial y final.
4. Seleccionar aula.
5. Verificar conflictos de docente.
6. Verificar conflictos de aula.
7. Verificar conflictos de paralelo.
8. Guardar.

8.25 CU-022 – Registrar Asistencia

Objetivo: Registrar la asistencia de estudiantes a una clase.

Actor principal: Docente.

Prioridad: Alta.

Flujo principal:

1. Docente inicia sesión.
2. Ingresa a su curso.
3. Selecciona sesión o fecha.
4. El sistema muestra estudiantes inscritos.
5. Selecciona estado por estudiante.
6. Guarda asistencia.
7. El sistema valida duplicidades.
8. Registra la información.

8.26 CU-023 – Justificar Inasistencia

Objetivo: Registrar una justificación sobre una ausencia.

Actor principal: Docente / Usuario autorizado.

Flujo principal:

1. Consultar asistencia.
2. Seleccionar estudiante.

3. Seleccionar inasistencia.
4. Registrar motivo.
5. Adjuntar documentación, cuando corresponda.
6. Cambiar estado a justificado.

8.27 CU-024 – Publicar Material

Objetivo: Permitir que el docente comparta contenido académico.

Actor principal: Docente.

Flujo principal:

1. Seleccionar curso y materia.
2. Seleccionar nuevo material.
3. Registrar título.
4. Registrar descripción.
5. Adjuntar archivo o enlace.
6. Definir disponibilidad.
7. Publicar.
8. Generar notificación para estudiantes.

8.28 CU-025 – Crear Práctica

Objetivo: Crear una actividad académica para los estudiantes.

Actor principal: Docente.

Flujo principal:

1. Seleccionar curso.
2. Seleccionar materia.
3. Crear práctica.
4. Registrar instrucciones.
5. Definir puntaje.
6. Definir fecha límite.
7. Definir si acepta tardías.
8. Definir si permite reemplazo.
9. Adjuntar material.
10. Publicar.

8.29 CU-026 – Entregar Práctica

Objetivo: Permitir que el estudiante entregue una actividad.

Actor principal: Estudiante.

Flujo principal:

1. Iniciar sesión.
2. Consultar prácticas.
3. Abrir actividad.
4. Adjuntar archivo.
5. Agregar comentario, si corresponde.
6. Confirmar envío.
7. Registrar fecha y hora.
8. Determinar si la entrega es tardía.
9. Notificar recepción.

8.30 CU-027 – Calificar Práctica

Objetivo: Evaluar una actividad entregada.

Actor principal: Docente.

Flujo principal:

1. Abrir actividad.
2. Consultar entregas.
3. Seleccionar estudiante.
4. Revisar archivo.
5. Registrar calificación.
6. Registrar retroalimentación.
7. Guardar.
8. Publicar inmediatamente o mantener oculta.

8.31 CU-028 – Modificar Calificación

Objetivo: Corregir una calificación previamente registrada.

Flujo principal:

1. Buscar evaluación.
2. Seleccionar estudiante.
3. Modificar nota.
4. Registrar motivo, cuando corresponda.
5. Guardar valor anterior.
6. Registrar nuevo valor.
7. Registrar usuario y fecha.

8.32 CU-029 – Consultar Portal del Estudiante

Objetivo: Proporcionar al estudiante acceso a su información.

Actor principal: Estudiante.

El estudiante podrá consultar:

- Cursos.
- Horarios.
- Material.
- Prácticas.
- Entregas.
- Calificaciones.
- Asistencia.
- Pagos.
- Recibos.

El sistema deberá asegurar que únicamente pueda consultar su propia información.

8.33 CU-030 – Consultar Estado de Cuenta

Objetivo: Mostrar la situación financiera individual del estudiante.

Flujo principal:

1. Acceder al portal.
2. Seleccionar Pagos.
3. Consultar inscripciones.
4. Mostrar total contratado.
5. Mostrar pagos válidos.
6. Mostrar saldo pendiente.
7. Mostrar cuotas y vencimientos.
8. Permitir consultar comprobantes.

8.34 CU-031 – Generar Reporte de Inscripciones

Actor principal: Director / Administrativo.

Flujo principal:

1. Acceder a reportes.
2. Seleccionar reporte de inscripciones.
3. Definir filtros.
4. Ejecutar consulta.
5. Mostrar resultados.
6. Permitir exportación autorizada.

Filtros:

- Fecha.
- Área.
- Curso.
- Sucursal.
- Estado.
- Usuario.

8.35 CU-032 – Generar Reporte Financiero

Actor principal: Director / Responsable Financiero.

Flujo principal:

1. Seleccionar periodo.
2. Seleccionar sucursal o consolidado.
3. Consultar ingresos.
4. Consultar egresos.
5. Calcular balance.
6. Mostrar resultados.
7. Exportar, cuando corresponda.

8.36 CU-033 – Consultar Dashboard Ejecutivo

Actor principal: Director.

El Dashboard deberá consolidar información relevante como:

- Estudiantes activos.
- Inscripciones.
- Ingresos.
- Egresos.
- Saldos pendientes.
- Cursos.
- Docentes.
- Comparación por sucursal.

8.37 CU-034 – Administrar Usuarios

Actor principal: Administrador del Sistema.

Flujo principal:

1. Consultar usuarios.
2. Crear o seleccionar usuario.
3. Asignar roles.
4. Asignar permisos cuando corresponda.

5. Activar o desactivar cuenta.
6. Guardar.
7. Registrar auditoría.

8.38 CU-035 – Consultar Auditoría

Actor principal: Administrador / Director autorizado.

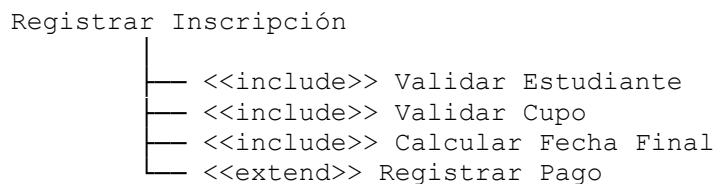
Flujo principal:

1. Ingresar a auditoría.
2. Definir filtros.
3. Consultar operaciones.
4. Visualizar usuario, fecha y acción.
5. Consultar cambios anteriores y posteriores cuando existan.

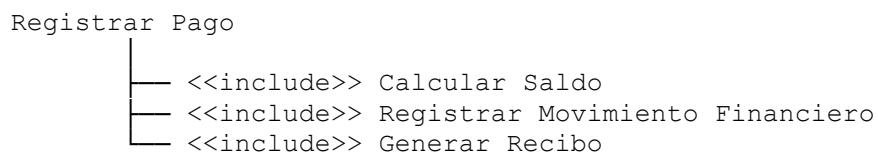
8.39 Relaciones Include y Extend

Los casos de uso podrán relacionarse entre sí mediante comportamientos reutilizables.

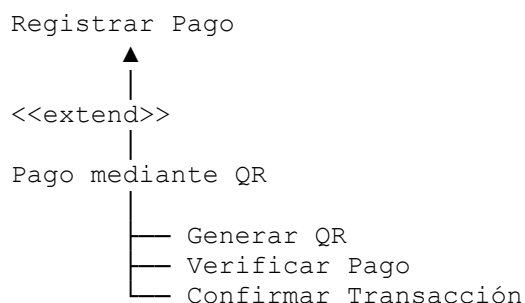
Ejemplo:



Otro ejemplo:



Para QR:



Estas relaciones permitirán reducir duplicación en los diagramas y representar correctamente procesos compartidos.

8.40 Casos de Uso Críticos

Debido al impacto que determinadas operaciones tienen sobre la institución, se consideran casos de uso críticos:

- CU-004 Registrar Inscripción.
- CU-006 Registrar Pago.
- CU-007 Registrar Pago Parcial.
- CU-008 Pago en Cuotas.
- CU-009 Pago QR.
- CU-011 Emitir Recibo.
- CU-014 Anular Recibo.
- CU-016 Cerrar Caja.
- CU-017 Registrar Egreso.
- CU-018 Pago Docente.
- CU-022 Registrar Asistencia.
- CU-027 Calificar Práctica.
- CU-028 Modificar Calificación.

Estos procesos deberán recibir prioridad durante las pruebas funcionales y de integración.

8.41 Matriz de Actor y Casos de Uso

Actor	Casos principales
Administrador	Usuarios, roles, configuración, auditoría
Director	Dashboard, reportes, supervisión
Administrativo	Estudiantes, inscripciones, cursos
Caja	Pagos, QR, recibos, caja
Financiero	Ingresos, egresos, salarios, reportes
Docente	Asistencia, material, prácticas, calificaciones
Estudiante	Material, entregas, notas, pagos
Proveedor QR	Generación y confirmación de pagos

8.42 Trazabilidad con Requerimientos

Cada caso de uso deberá relacionarse con uno o más requerimientos funcionales.

Ejemplo:

Caso de Uso	Requerimientos
CU-002 Registrar estudiante	RF-036, RF-049, RF-069
CU-004 Registrar inscripción	RF-116 a RF-128
CU-006 Registrar pago	RF-156 a RF-169

CU-009 Pago QR	RF-175 a RF-185
CU-014 Anular recibo	RF-133, RF-146
CU-022 Asistencia	RF-217 a RF-226
CU-025 Crear práctica	RF-235 a RF-240
CU-027 Calificar práctica	RF-241 a RF-255

La matriz completa podrá desarrollarse como anexo.

8.43 Relación con Reglas de Negocio

Los casos de uso deberán respetar las reglas de negocio definidas en el Capítulo 5.

Por ejemplo:

CU-004 Registrar inscripción

se relacionará con:

- RN-006 Estudiante registrado.
- RN-007 Curso activo.
- RN-011 Control de cupos.
- RN-012 Inscripción duplicada.
- RN-014 Fecha de finalización.

Esto permitirá validar correctamente cada proceso.

8.44 Uso de los Casos de Uso en Desarrollo

Los casos de uso serán utilizados posteriormente para:

- Diseñar interfaces.
- Construir endpoints.
- Definir servicios de backend.
- Diseñar pruebas.
- Definir historias de usuario.
- Elaborar el cronograma.
- Validar funcionalidades con los responsables institucionales.

Por ejemplo:

```
CU-004 Registrar Inscripción
    ↓
Pantalla React
    ↓
Endpoint Django
    ↓
Servicio de inscripción
    ↓
```

8.45 Cierre del Capítulo

Los casos de uso presentados en este capítulo describen los principales procesos que serán ejecutados por los diferentes actores del Sistema Integral de Gestión Académica y Administrativa de Sumo (SIGA – SUMO).

Su definición permite representar de manera estructurada la interacción entre usuarios y sistema, complementando los requerimientos funcionales, reglas de negocio, arquitectura y modelo de datos desarrollados previamente.

Los casos de uso servirán como una referencia fundamental durante las etapas de diseño de interfaces, implementación, pruebas y validación con los usuarios finales.

Asimismo, la relación establecida entre casos de uso, requerimientos funcionales y reglas de negocio permitirá mantener trazabilidad durante todo el ciclo de desarrollo y facilitará la identificación de cambios en el alcance.

Los casos de uso críticos relacionados con inscripción, pagos, recibos, QR, caja y operaciones financieras deberán ser validados prioritariamente antes de su implementación definitiva, debido al impacto directo que tienen sobre la operación institucional.

Una vez aprobados los principales flujos de negocio, será posible utilizarlos como base para construir el cronograma detallado de desarrollo, las historias de usuario, los prototipos de interfaz y el plan de pruebas del sistema.

CAPÍTULO 9 PLANIFICACIÓN Y CRONOGRAMA DEL PROYECTO

9.1 Introducción

El presente capítulo describe la planificación general para el desarrollo del Sistema Integral de Gestión Académica y Administrativa de Sumo (SIGA – SUMO), estableciendo las fases del proyecto, actividades, tiempos estimados, entregables, hitos y mecanismos de control que permitirán realizar el seguimiento de su avance.

La planificación del proyecto considera una jornada de trabajo de **4 horas diarias, de lunes a viernes**, equivalente a aproximadamente **20 horas semanales** y alrededor de **80 horas mensuales**.

Debido al tamaño y complejidad del sistema, se plantea un desarrollo incremental, priorizando inicialmente los módulos críticos para la operación institucional, especialmente aquellos relacionados con:

- Estudiantes.

- Inscripciones.
- Pagos.
- Recibos.
- Impresión térmica.
- Pagos mediante QR.
- Gestión académica.
- Docentes.
- Finanzas.
- Asistencia.
- Materiales.
- Prácticas.
- Calificaciones.
- Reportes.

El proyecto se organizará mediante fases y ciclos de trabajo cortos que permitan mostrar avances periódicos y realizar validaciones con los responsables de la institución.

9.2 Objetivo de la Planificación

El objetivo principal de la planificación consiste en organizar las actividades necesarias para desarrollar SIGA de manera progresiva, permitiendo controlar el tiempo empleado, identificar retrasos y garantizar que los módulos de mayor prioridad sean implementados primero.

De forma específica, la planificación permitirá:

- Establecer actividades concretas.
- Definir prioridades.
- Estimar tiempos.
- Distribuir la carga de trabajo.
- Identificar dependencias.
- Definir entregables.
- Establecer hitos.
- Facilitar reuniones de seguimiento.
- Controlar cambios.
- Medir el porcentaje de avance.
- Mantener informada a la dirección del instituto.

9.3 Jornada de Trabajo

La jornada de desarrollo considerada será:

Concepto	Tiempo
Horas diarias	4 horas
Días por semana	5 días
Horas semanales	20 horas

Horas mensuales aproximadas	80 horas
-----------------------------	----------

La planificación deberá respetar esta disponibilidad y evitar estimaciones realizadas como si existiera una jornada de ocho horas.

9.4 Distribución Recomendada de las 4 Horas Diarias

La jornada diaria podrá organizarse de la siguiente manera:

Actividad	Tiempo aproximado
Revisión y planificación	20 minutos
Desarrollo principal	2 horas 40 minutos
Pruebas	30 minutos
Documentación y control de avance	30 minutos

Esta distribución podrá variar según la etapa del proyecto.

Durante análisis y diseño, por ejemplo, una mayor proporción del tiempo será destinada a documentación.

Durante desarrollo, la mayor parte será utilizada para implementación y pruebas.

9.5 Metodología de Seguimiento

El proyecto utilizará una metodología ágil adaptada, mediante ciclos de trabajo cortos.

Se propone trabajar con **sprints semanales**, debido a que la dirección necesita visualizar avances frecuentes.

Cada sprint tendrá una duración aproximada de:

5 días × 4 horas = 20 horas.

Cada semana deberá comenzar con objetivos claramente definidos y finalizar con un entregable verificable.

9.6 Ciclo Semanal de Trabajo

Una semana típica se organizará de la siguiente manera.

Lunes

- Revisar objetivos.
- Priorizar tareas.
- Preparar entorno.
- Comenzar implementación.

Martes

- Desarrollo de funcionalidades principales.
- Validación parcial.

Miércoles

- Continuación de desarrollo.
- Integración con componentes relacionados.

Jueves

- Pruebas funcionales.
- Corrección de errores.
- Ajustes.

Viernes

- Finalización del entregable.
- Documentación.
- Actualización del cronograma.
- Preparación del informe semanal.
- Presentación de avance.

9.7 Fases Generales del Proyecto

El desarrollo será dividido en las siguientes fases:

1. Análisis y levantamiento de requerimientos.
2. Diseño funcional y técnico.
3. Preparación de arquitectura y entorno.
4. Desarrollo del núcleo administrativo.
5. Desarrollo de inscripciones y pagos.
6. Implementación de recibos e impresión térmica.
7. Implementación de pagos QR.
8. Gestión financiera.
9. Gestión académica.
10. Portal docente.
11. Portal estudiante.
12. Reportes y Dashboard.
13. Pruebas integrales.
14. Implementación y capacitación.
15. Mantenimiento inicial.

9.8 Fase I – Análisis y Requerimientos

Duración estimada: 3 semanas.

Horas aproximadas: 60 horas.

Actividades

- Levantamiento de información.
- Identificación de actores.
- Identificación de módulos.
- Definición de procesos.
- Elaboración del SRS.
- Requerimientos funcionales.
- Requerimientos no funcionales.
- Reglas de negocio.
- Casos de uso.
- Validación de alcance.

Entregables

- Documento SRS.
- Lista de requerimientos.
- Reglas de negocio.
- Casos de uso iniciales.
- Alcance aprobado.

9.9 Fase II – Diseño Técnico

Duración estimada: 3 semanas.

Horas aproximadas: 60 horas.

Actividades

- Definición de arquitectura.
- Diseño de base de datos.
- Modelo entidad-relación.
- Diccionario de datos.
- Diseño de API.
- Diseño de módulos.
- Diseño inicial de interfaces.

Entregables

- Documento de arquitectura.

- Modelo de datos.
- Estructura inicial de API.
- Prototipos de interfaces.
- Estructura del repositorio.

9.10 Fase III – Configuración del Proyecto

Duración estimada: 2 semanas.

Horas aproximadas: 40 horas.

Actividades

Frontend

- Crear proyecto React.
- Configurar TypeScript.
- Configurar rutas.
- Crear estructura de módulos.
- Configurar cliente HTTP.
- Configurar layout base.

Backend

- Crear proyecto Django.
- Configurar Django REST Framework.
- Configurar MariaDB.
- Configurar variables de entorno.
- Crear estructura de aplicaciones.
- Configurar migraciones.
- Crear repositorio Git.

Entregable

Base tecnológica completamente funcional.

9.11 Fase IV – Autenticación, Usuarios, Roles y Permisos

Duración estimada: 3 semanas.

Horas aproximadas: 60 horas.

Funcionalidades

- Login.
- Logout.
- Usuarios.

- Roles.
- Permisos.
- Recuperación de contraseña.
- Protección de rutas.
- Auditoría de acceso.

Entregable

Sistema de autenticación y seguridad funcional.

9.12 Fase V – Gestión de Estudiantes

Duración estimada: 3 semanas.

Horas aproximadas: 60 horas.

Funcionalidades

- Registro de estudiantes.
- Modificación.
- Consulta.
- Búsqueda.
- Tutores.
- Expediente.
- Observaciones.
- Historial.

Entregable

Módulo completo de estudiantes.

9.13 Fase VI – Gestión Académica

Duración estimada: 4 semanas.

Horas aproximadas: 80 horas.

Funcionalidades

- Sucursales.
- Áreas.
- Periodos.
- Cursos.
- Paralelos.
- Materias.
- Aulas.

- Horarios.
- Docentes.
- Asignaciones docentes.
- Cupos.

Entregable

Estructura académica operativa.

9.14 Fase VII – Inscripciones

Duración estimada: 4 semanas.

Horas aproximadas: 80 horas.

Esta será una de las fases de mayor prioridad.

Funcionalidades

- Nueva inscripción.
- Validación de estudiante.
- Selección de área.
- Selección de curso.
- Selección de horario.
- Fecha de inicio.
- Cálculo de fecha final.
- Monto.
- Descuento.
- Tipo de pago.
- Estados.
- Historial.
- Anulación.
- Reportes básicos.

Entregable

Proceso completo de inscripción funcional.

9.15 Fase VIII – Pagos y Recibos

Duración estimada: 4 semanas.

Horas aproximadas: 80 horas.

Funcionalidades

- Pago total.
- Pago parcial.
- Pago en cuotas.
- Plan de pagos.
- Saldos.
- Métodos de pago.
- Recibos.
- Reimpresión.
- Anulación.
- Auditoría.

Entregable

Módulo financiero asociado a estudiantes.

9.16 Fase IX – Impresión Térmica

Duración estimada: 2 semanas.

Horas aproximadas: 40 horas.

Actividades

- Identificar impresora.
- Validar drivers.
- Determinar ancho de papel.
- Diseñar formato.
- Generar comprobante.
- Realizar pruebas físicas.
- Ajustar impresión.

Entregable

Impresión de recibo validada con hardware real.

9.17 Fase X – Pagos mediante QR

Duración estimada: 3 a 4 semanas.

Horas aproximadas: 60 a 80 horas.

La duración dependerá de la documentación y servicios disponibles del proveedor financiero.

Actividades

- Analizar proveedor.
- Obtener credenciales.
- Generar solicitudes.
- Mostrar código QR.
- Consultar estado.
- Recibir confirmación.
- Evitar duplicidades.
- Asociar pago.
- Generar recibo.
- Conciliación.

Entregable

Integración QR operativa.

9.18 Fase XI – Gestión Financiera Institucional

Duración estimada: 4 semanas.

Horas aproximadas: 80 horas.

Funcionalidades

- Ingresos.
- Egresos.
- Categorías.
- Caja.
- Apertura.
- Cierre.
- Arqueo.
- Diferencias.
- Salarios.
- Movimientos financieros.
- Balance.

Entregable

Módulo financiero institucional.

9.19 Fase XII – Portal Docente

Duración estimada: 4 semanas.

Horas aproximadas: 80 horas.

Funcionalidades

- Dashboard docente.
- Cursos.
- Materias.
- Horarios.
- Estudiantes.
- Asistencia.
- Material académico.

Entregable

Primera versión funcional del Portal Docente.

9.20 Fase XIII – Prácticas y Calificaciones

Duración estimada: 4 semanas.

Horas aproximadas: 80 horas.

Funcionalidades

- Crear prácticas.
- Adjuntar archivos.
- Fechas límite.
- Entregas.
- Revisiones.
- Calificaciones.
- Retroalimentación.
- Historial de notas.

Entregable

Gestión de tareas y evaluación académica.

9.21 Fase XIV – Portal del Estudiante

Duración estimada: 3 semanas.

Horas aproximadas: 60 horas.

Funcionalidades

- Dashboard.
- Cursos.
- Horarios.

- Materiales.
- Prácticas.
- Entregas.
- Calificaciones.
- Asistencia.
- Pagos.
- Recibos.

Entregable

Portal del estudiante completamente funcional.

9.22 Fase XV – Reportes y Dashboard

Duración estimada: 3 semanas.

Horas aproximadas: 60 horas.

Funcionalidades

- Reportes de estudiantes.
- Inscripciones.
- Pagos.
- Recibos.
- QR.
- Ingresos.
- Egresos.
- Docentes.
- Asistencia.
- Calificaciones.
- Dashboard ejecutivo.

Entregable

Herramientas de análisis y supervisión.

9.23 Fase XVI – Pruebas Integrales

Duración estimada: 3 semanas.

Horas aproximadas: 60 horas.

Actividades

- Pruebas funcionales.
- Pruebas de permisos.

- Pruebas financieras.
- Pruebas académicas.
- Pruebas de integración.
- Pruebas de impresión.
- Pruebas QR.
- Corrección de errores.

Entregable

Versión candidata para producción.

9.24 Fase XVII – Implementación

Duración estimada: 2 semanas.

Horas aproximadas: 40 horas.

Actividades

- Preparar servidor.
- Configurar Nginx.
- Configurar HTTPS.
- Configurar MariaDB.
- Desplegar React.
- Desplegar Django.
- Migrar estructura.
- Configurar respaldos.
- Crear usuarios iniciales.
- Validar producción.

9.25 Fase XVIII – Capacitación

Duración estimada: 1 semana.

Horas aproximadas: 20 horas.

Usuarios

- Administrativos.
- Caja.
- Dirección.
- Docentes.
- Personal financiero.

Entregables

- Manual básico.
- Capacitación.
- Material de apoyo.

9.26 CRONOGRAMA DETALLADO DE EJECUCIÓN DEL PROYECTO

El cronograma del Sistema Integral de Gestión Académica y Administrativa de Sumo (SIGA – SUMO) ha sido elaborado considerando una jornada laboral de **4 horas diarias, de lunes a viernes**, equivalente a **20 horas semanales**.

La planificación se encuentra estructurada por semanas de trabajo y establece para cada periodo:

- Objetivo de la semana.
- Actividades principales.
- Horas estimadas.
- Entregable esperado.
- Resultado verificable.
- Estado de control.
- Porcentaje referencial de avance.

El objetivo de este cronograma no es únicamente establecer fechas, sino permitir que la dirección del instituto pueda verificar periódicamente el avance real del desarrollo.

9.27 Distribución General del Tiempo

La disponibilidad estimada del proyecto será:

Periodo	Horas
Día laboral	4 h
Semana laboral	20 h
Mes aproximado	80 h
3 meses	240 h
6 meses	480 h
12 meses	960 h

Estas horas incluyen:

- Análisis.
- Diseño.
- Programación.
- Pruebas.
- Corrección de errores.
- Documentación.

- Reuniones de seguimiento.

Por este motivo, las cuatro horas diarias no serán consideradas exclusivamente como tiempo de programación.

9.28 Distribución de Actividades dentro de una Semana

Cada semana dispondrá aproximadamente de 20 horas.

Se propone la siguiente distribución:

Actividad	Horas aproximadas
Planificación y revisión	2 h
Desarrollo / análisis principal	12 h
Pruebas	3 h
Correcciones	1 h
Documentación	1 h
Preparación de avance	1 h
Total	20 h

La distribución podrá modificarse dependiendo de la etapa del proyecto.

Por ejemplo, durante la elaboración del SRS será mayor el tiempo de documentación, mientras que durante la implementación será mayor el tiempo destinado al desarrollo.

9.29 Etapa 1 – Análisis y Definición del Sistema

Semana 1 – Levantamiento inicial de información

Objetivo: Comprender cómo funciona actualmente la institución.

Horas estimadas: 20 horas.

Actividades

- Reunión inicial con dirección.
- Identificación de procesos administrativos.
- Identificación de procesos académicos.
- Análisis del sistema utilizado actualmente.
- Recopilación de información del formulario de inscripción.
- Identificación de tipos de usuarios.
- Identificación de sucursales.
- Identificación de cursos y áreas.
- Identificación del proceso de pagos.
- Identificación del proceso de pago a docentes.
- Identificación de necesidades financieras.

Entregable

Documento inicial de levantamiento de información.

Resultado verificable

Listado de procesos principales de la institución.

Avance esperado

2 % del proyecto general.

Semana 2 – Definición del alcance

Objetivo: Determinar qué funcionalidades serán desarrolladas dentro de SIGA.

Horas: 20 horas.

Actividades

- Identificación de módulos.
- Definición de usuarios.
- Definición de alcance inicial.
- Identificación de funcionalidades prioritarias.
- Clasificación de necesidades administrativas.
- Clasificación de necesidades académicas.
- Identificación de futuras funcionalidades.
- Definición inicial del MVP.

Entregables

- Alcance del proyecto.
- Lista de módulos.
- Definición inicial del MVP.

Resultado verificable

Dirección podrá conocer claramente qué incluirá el sistema.

Avance acumulado

4 %.

Semana 3 – Requerimientos Funcionales

Objetivo: Documentar formalmente las funcionalidades del sistema.

Horas: 20 horas.

Actividades

- Gestión de usuarios.
- Gestión de estudiantes.
- Gestión de docentes.
- Inscripciones.
- Pagos.
- Recibos.
- QR.
- Finanzas.
- Asistencia.
- Materiales.
- Prácticas.
- Calificaciones.
- Reportes.

Entregable

Listado estructurado de requerimientos funcionales.

Resultado verificable

Documento SRS actualizado.

Avance acumulado

6 %.

Semana 4 – Requerimientos No Funcionales y Reglas de Negocio

Objetivo: Determinar condiciones de seguridad, rendimiento y funcionamiento institucional.

Actividades

- Requerimientos de seguridad.
- Rendimiento.
- Compatibilidad.
- Respaldo.
- Auditoría.
- Reglas de inscripción.
- Reglas de pagos.
- Reglas de recibos.
- Reglas académicas.
- Reglas financieras.

Entregable

Capítulos correspondientes del SRS.

Avance acumulado

8 %.

9.30 Etapa 2 – Diseño Técnico

Semana 5 – Arquitectura del Sistema

Objetivo: Definir cómo se construirá técnicamente SIGA.

Actividades

- Arquitectura React.
- Arquitectura Django.
- API REST.
- MariaDB.
- Organización del frontend.
- Organización del backend.
- Seguridad.
- Gestión de archivos.
- Diseño de despliegue.

Entregable

Documento de arquitectura.

Tecnología definida

Frontend: React + TypeScript

Backend: Python + Django REST Framework

Base de datos: MariaDB

Avance acumulado

10 %.

Semana 6 – Modelo Conceptual de Datos

Objetivo: Identificar las entidades principales.

Actividades

Diseño de:

- Usuario.
- Estudiante.
- Tutor.
- Docente.
- Sucursal.
- Área.
- Curso.
- Materia.
- Inscripción.
- Pago.
- Recibo.
- Caja.
- Asistencia.
- Prácticas.
- Calificaciones.

Entregable

Modelo conceptual de datos.

Avance

12 %.

Semana 7 – Modelo Lógico y Relaciones

Actividades

- Claves primarias.
- Claves foráneas.
- Relaciones uno a muchos.
- Relaciones muchos a muchos.
- Restricciones.
- Integridad referencial.
- Estados.

Entregable

Diagrama Entidad–Relación inicial.

Avance

14 %.

Semana 8 – Diseño físico de MariaDB

Actividades

- Tipos de datos.
- Índices.
- Restricciones UNIQUE.
- DECIMAL para montos.
- Estrategias de eliminación.
- Auditoría.
- Diseño de migraciones.

Entregable

Modelo de base de datos listo para implementación.

Avance

16 %.

9.31 Etapa 3 – Preparación del Proyecto

Semana 9 – Configuración del Backend

Actividades

- Instalar Python.
- Crear entorno virtual.
- Instalar Django.
- Instalar Django REST Framework.
- Crear proyecto.
- Configurar MariaDB.
- Configurar variables de entorno.
- Crear aplicaciones Django.
- Crear repositorio Git.

Entregable

Backend ejecutándose correctamente.

Demostración

El servidor Django deberá responder correctamente.

Avance

18 %.

Semana 10 – Configuración del Frontend

Actividades

- Crear proyecto React.
- Configurar TypeScript.
- Configurar rutas.
- Configurar comunicación con API.
- Crear estructura modular.
- Crear layout.
- Crear menú principal.

Entregable

Frontend base funcionando.

Demostración

Aplicación React conectada al backend.

Avance

20 %.

9.32 Etapa 4 – Usuarios y Seguridad

Semana 11 – Autenticación

Desarrollo

- Login.
- Logout.
- Sesiones.
- Protección de rutas.
- Validación de credenciales.

Entregable

Inicio de sesión funcional.

Demostración

Usuario podrá ingresar al sistema.

Avance

22 %.

Semana 12 – Usuarios

Desarrollo

- Registrar usuario.
- Modificar usuario.
- Desactivar usuario.
- Consultar usuarios.
- Recuperar contraseña.

Entregable

Administración básica de usuarios.

Avance

24 %.

Semana 13 – Roles y Permisos

Desarrollo

- Crear roles.
- Asignar roles.
- Crear permisos.
- Validación de permisos.
- Protección de endpoints.

Entregable

Control de acceso basado en roles.

Avance

26 %.

9.33 Etapa 5 – Gestión de Estudiantes

Semana 14 – Registro de Estudiantes

Desarrollo

Formulario con:

- Apellido paterno.
- Apellido materno.
- Nombres.
- CI.

- Correo.
- Teléfono.
- Tutor 1.
- Tutor 2.

Validaciones

- CI duplicado.
- Campos obligatorios.
- Correo.

Entregable

Registro de estudiantes funcional.

Avance

28 %.

Semana 15 – Consulta y Expediente Desarrollo

- Lista de estudiantes.
- Búsqueda.
- Filtros.
- Modificación.
- Estado.
- Expediente.

Entregable

Gestión completa del estudiante.

Avance

30 %.

9.34 Etapa 6 – Estructura Académica

Semana 16 – Sucursales y Áreas Desarrollo

- Registrar sucursales.
- Administrar sucursales.
- Registrar áreas.

- Asociar áreas.

Entregable

Configuración institucional.

Avance

32 %.

Semana 17 – Cursos y Materias**Desarrollo**

- Cursos.
- Materias.
- Relación curso-materia.
- Capacidad.
- Precio referencial.

Entregable

Catálogo académico.

Avance

34 %.

Semana 18 – Paralelos, Aulas y Horarios**Desarrollo**

- Paralelos.
- Aulas.
- Horarios.
- Validación de conflictos.

Entregable

Estructura de horarios.

Avance

36 %.

Semana 19 – Gestión de Docentes

Desarrollo

- Registro de docentes.
- Modificación.
- Consulta.
- Estado.
- Especialidad.

Entregable

Módulo de docentes.

Avance

38 %.

Semana 20 – Asignación Docente

Desarrollo

- Asignar materia.
- Asignar curso.
- Asignar paralelo.
- Horario.
- Carga horaria.

Entregable

Asignación académica funcional.

Avance

40 %.

9.35 Etapa 7 – Inscripciones

Semana 21 – Formulario de Inscripción

Desarrollo

Construcción del formulario principal.

Campos:

- Estudiante.
- Área.

- Curso.
- Horario.
- Sucursal.
- Monto.
- Fecha inicio.
- Fecha fin.
- Modalidad de pago.
- Detalle.

Entregable

Primera versión del formulario.

Avance

42 %.

Semana 22 – Reglas de Inscripción

Desarrollo

- Validar estudiante.
- Validar curso.
- Validar cupos.
- Detectar duplicidad.
- Calcular fecha final.
- Aplicar descuentos.

Entregable

Inscripción validada.

Avance

44 %.

Semana 23 – Historial y Estados

Desarrollo

- Inscripción activa.
- Cancelada.
- Finalizada.
- Historial.
- Cambio de curso.
- Cambio de horario.

Entregable

Gestión completa de inscripción.

Avance

46 %.

9.36 Etapa 8 – Pagos**Semana 24 – Pagos Básicos****Desarrollo**

- Registro de pago.
- Pago completo.
- Métodos de pago.
- Saldo.

Entregable

Primer pago funcional.

Avance

48 %.

Semana 25 – Pagos Parciales**Desarrollo**

- Pago parcial.
- Actualización de saldo.
- Estado de cuenta.
- Historial.

Entregable

Pagos parciales funcionando.

Avance

50 %.

Semana 26 – Plan de Cuotas

Desarrollo

- Crear plan.
- Crear cuotas.
- Vencimientos.
- Estados.

Entregable

Sistema de cuotas.

Avance

52 %.

9.37 Etapa 9 – Recibos

Semana 27 – Generación del Recibo

Desarrollo

- Numeración.
- Información institucional.
- Datos estudiante.
- Concepto.
- Monto.
- Método de pago.

Entregable

Recibo digital.

Avance

54 %.

Semana 28 – Impresión Térmica

Actividades

- Configurar impresora.
- Identificar tamaño.
- Diseñar ticket.
- Pruebas físicas.
- Ajustes.

Entregable

Recibo impreso correctamente.

Avance

56 %.

Semana 29 – Anulaciones

Desarrollo

- Anular recibo.
- Motivo.
- Usuario responsable.
- Reversión de pago.
- Auditoría.

Entregable

Proceso de anulación seguro.

Avance

58 %.

9.38 HITO PRINCIPAL – MVP

Al finalizar aproximadamente la **Semana 29**, SIGA deberá ser capaz de realizar el flujo operativo principal:

```
ESTUDIANTE
  ↓
INSCRIPCIÓN
  ↓
PAGO
  ↓
RECIBO
  ↓
IMPRESIÓN
```

Este será considerado el **primer gran hito operativo del proyecto**.

La institución podrá comenzar a realizar pruebas con procesos reales controlados.

9.39 Etapa 10 – Gestión Financiera

Semana 30 – Ingresos

- Registrar ingresos.
- Categorías.
- Consulta.
- Reportes básicos.

Avance: 60 %.

Semana 31 – Egresos

- Registrar egresos.
- Categorías.
- Justificaciones.
- Auditoría.

Avance: 62 %.

Semana 32 – Caja

- Apertura.
- Sesión.
- Movimiento.
- Cierre.

Avance: 64 %.

Semana 33 – Arqueo

- Total esperado.
- Total declarado.
- Diferencia.
- Justificación.

Avance: 66 %.

Semana 34 – Salarios Docentes

- Tarifa.
- Horas.
- Cálculo.
- Registro.
- Egreso.

Avance: 68 %.

9.40 Etapa 11 – Pagos QR

Semana 35 – Investigación e Integración Inicial

- Seleccionar proveedor.
- Analizar documentación.
- Credenciales.
- Ambiente de pruebas.

Semana 36 – Generación QR

- Solicitud.
- Código QR.
- Identificador.
- Expiración.

Semana 37 – Confirmación

- Confirmación automática/manual.
- Actualización del pago.
- Control de duplicidad.

Semana 38 – Conciliación y Pruebas

- Validar movimientos.
- Pruebas.
- Errores.
- Integración recibos.

Avance esperado

76 %.

El tiempo de esta fase podrá variar dependiendo del proveedor financiero.

9.41 Etapa 12 – Portal Docente

Semana 39

- Login docente.
- Dashboard.
- Cursos.
- Horarios.

Semana 40

- Listas de estudiantes.
- Sesiones de clase.
- Registro de asistencia.

Semana 41

- Consulta de asistencia.
- Justificaciones.
- Reportes.

Semana 42

- Material académico.
- Archivos.
- Publicaciones.

Avance

84 %.

9.42 Etapa 13 – Prácticas y Calificaciones

Semana 43

- Crear prácticas.
- Fechas.
- Archivos.

Semana 44

- Recepción de entregas.
- Entregas tardías.
- Reemplazos.

Semana 45

- Calificación.
- Retroalimentación.

Semana 46

- Evaluaciones.
- Nota final.
- Historial de cambios.

Avance

90 %.

9.43 Etapa 14 – Portal Estudiante

Semana 47

- Dashboard.
- Cursos.
- Horarios.

Semana 48

- Materiales.
- Prácticas.
- Entregas.

Semana 49

- Calificaciones.
- Asistencia.

Semana 50

- Pagos.
- Estado de cuenta.
- Recibos.

Avance

94 %.

9.44 Etapa 15 – Reportes y Dashboard

Semana 51

Reportes administrativos.

Semana 52

Reportes financieros.

Semana 53

Reportes académicos.

Semana 54

Dashboard ejecutivo.

Avance

97 %.

9.45 Etapa 16 – Pruebas Finales e Implementación

Semana 55 – Pruebas Integrales

- Usuarios.
- Inscripciones.
- Pagos.
- Recibos.
- Finanzas.
- Docentes.
- Estudiantes.

Semana 56 – Corrección de Incidencias

- Corrección de errores detectados.
- Ajustes.
- Optimización.

Semana 57 – Producción

- Servidor.
- Nginx.
- HTTPS.
- Base de datos.
- Respaldos.
- Usuarios.

Semana 58 – Capacitación y Cierre Inicial

- Capacitación administrativos.
- Capacitación caja.
- Capacitación docentes.
- Manuales.
- Entrega de primera versión completa.

Avance

100 %.

9.46 Resumen Ejecutivo del Cronograma

Etap	Semanas	Resultado
Análisis	1-4	SRS y alcance
Diseño	5-8	Arquitectura y BD
Configuración	9-10	React + Django
Seguridad	11-13	Login, roles
Estudiantes	14-15	Gestión estudiantes
Académico	16-20	Cursos y docentes
Inscripciones	21-23	Inscripción funcional
Pagos	24-26	Pagos y cuotas
Recibos	27-29	Ticket térmico
Finanzas	30-34	Caja y salarios
QR	35-38	Pago QR
Portal docente	39-42	Gestión docente
Evaluaciones	43-46	Tareas y notas
Portal estudiante	47-50	Portal académico
Reportes	51-54	Reportes y Dashboard
Implementación	55-58	Producción

9.47 Interpretación para Dirección

El cronograma no significa que la institución deberá esperar hasta la semana 58 para comenzar a utilizar el sistema.

La entrega será progresiva.

Primera entrega visible

Semanas **9-13**:

Sistema base, login y usuarios.

Segunda entrega

Semanas **14-20**:

Estudiantes, cursos y docentes.

Primer producto operativo

Semanas **21-29**:

Inscripciones, pagos y recibos.

Este constituye el **MVP institucional**.

Segunda versión operativa

Semanas **30-38**:

Finanzas, caja, salarios y QR.

Tercera versión

Semanas **39-50**:

Docentes y estudiantes.

Versión integral

Semanas **51-58**:

Reportes, pruebas e implementación completa.

9.48 Mecanismo de Control Semanal

Para facilitar el seguimiento por parte de dirección, cada semana se presentará una tabla como la siguiente:

Elemento	Información
Semana	21
Módulo	Inscripciones
Horas planificadas	20 h
Horas trabajadas	20 h
Objetivo	Formulario de inscripción
Avance anterior	40 %
Avance actual	42 %
Estado	En desarrollo
Entregable	Formulario funcional
Bloqueos	Ninguno

9.49 Semáforo de Avance

Se podrá utilizar un sistema sencillo de seguimiento:

VERDE – En tiempo

La actividad está avanzando conforme al cronograma.

AMARILLO – Riesgo

Existe una dificultad que podría generar retraso.

ROJO – Bloqueado

No es posible continuar debido a una dependencia o problema crítico.

AZUL – Finalizado

La actividad fue desarrollada, probada y documentada.

9.50 Control mediante Evidencias

Cada avance deberá disponer, cuando corresponda, de una evidencia verificable.

Ejemplos:

Actividad	Evidencia
Requisitos	Documento SRS
Base de datos	Diagrama ER
Login	Demostración funcional
Estudiantes	Pantalla funcional
API	Endpoint funcionando
Pago	Registro en sistema
Recibo	Ticket impreso
QR	Transacción de prueba
Reporte	PDF/Excel generado
Producción	Sistema publicado

De esta manera, el avance no dependerá únicamente de porcentajes declarados.

9.51 Criterio para Determinar el Porcentaje de Avance

El porcentaje de avance deberá calcularse principalmente según entregables completados. Por ejemplo, un módulo podrá dividirse de la siguiente forma:

Estado	Porcentaje
No iniciado	0 %
Analizado	15 %
Diseñado	30 %
Backend desarrollado	50 %
Frontend desarrollado	70 %
Integrado	80 %
Probado	90 %
Validado	100 %

Esto permitirá medir el avance de manera más objetiva.

9.52 Gestión de Retrasos

Cuando una actividad no pueda completarse dentro de la semana planificada, deberá registrarse:

- Actividad pendiente.
- Porcentaje realizado.
- Motivo del retraso.
- Horas adicionales estimadas.
- Impacto sobre las siguientes tareas.
- Acción correctiva.

No se modificará silenciosamente el cronograma.

Los cambios deberán quedar documentados.

9.53 Dependencias del Instituto

Algunas actividades requerirán participación directa de la institución.

La dirección deberá facilitar cuando corresponda:

- Validación de requisitos.
- Información de cursos.
- Políticas financieras.
- Información salarial.
- Formato de recibos.
- Acceso a impresora térmica.
- Definición del proveedor QR.
- Acceso a información del sistema anterior.
- Usuarios disponibles para pruebas.

Si alguno de estos elementos no estuviera disponible, la tarea podrá marcarse como **bloqueada por dependencia externa** y se continuará temporalmente con otra actividad cuando sea posible.

9.54 Cierre del Cronograma

El cronograma detallado presentado establece una ruta de desarrollo aproximada de 58 semanas considerando una disponibilidad de veinte horas semanales.

La planificación busca equilibrar la complejidad del proyecto con la necesidad de proporcionar resultados visibles a la institución desde etapas tempranas.

El sistema será entregado progresivamente, priorizando el núcleo operativo compuesto por estudiantes, estructura académica, inscripciones, pagos y recibos antes de avanzar hacia funcionalidades académicas y financieras más complejas.

El cronograma deberá considerarse una línea base para el seguimiento del proyecto y podrá ser ajustado cuando exista una modificación aprobada del alcance, una dependencia externa o nueva información que afecte las estimaciones iniciales.

La dirección podrá realizar el seguimiento mediante entregables semanales, horas ejecutadas, evidencias funcionales, porcentaje de avance y estado de cada actividad, permitiendo conocer de manera objetiva la situación del proyecto durante todo su ciclo de desarrollo.

CAPÍTULO 10 PLAN DE PRUEBAS Y ASEGURAMIENTO DE CALIDAD

10.1 Introducción

El presente capítulo establece el plan general de pruebas y aseguramiento de calidad para el Sistema Integral de Gestión Académica y Administrativa de Sumo (SIGA - SUMO).

El objetivo del proceso de pruebas será verificar que las funcionalidades desarrolladas cumplan con los requerimientos funcionales, no funcionales y reglas de negocio definidos en los capítulos anteriores, así como garantizar que el sistema opere de forma segura, consistente y confiable antes de ser utilizado con información real de la institución.

Debido a que SIGA administrará procesos críticos relacionados con estudiantes, inscripciones, pagos, recibos, caja, salarios, asistencia y calificaciones, las pruebas no deberán limitarse únicamente a verificar que las pantallas funcionen correctamente.

También será necesario comprobar:

- Integridad de la información.
- Correcto cálculo de montos.
- Consistencia de saldos.
- Control de permisos.
- Prevención de operaciones duplicadas.
- Trazabilidad.
- Manejo de errores.
- Compatibilidad con dispositivos.
- Correcto funcionamiento de integraciones externas.
- Protección frente a operaciones no autorizadas.
- Recuperación ante fallos.

Las pruebas serán realizadas progresivamente durante todo el desarrollo y no únicamente al finalizar el proyecto.

10.2 Objetivos del Plan de Pruebas

El objetivo principal consiste en establecer una metodología organizada para identificar errores y validar las funcionalidades desarrolladas antes de su liberación a producción.

Los objetivos específicos serán:

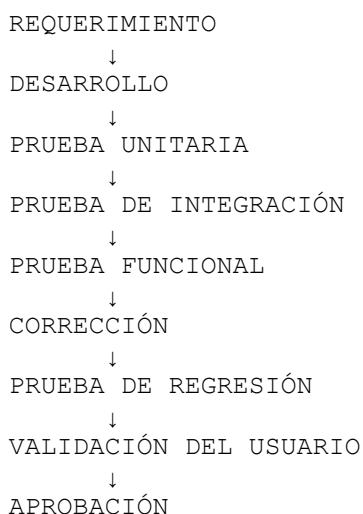
- Verificar el cumplimiento de los requerimientos funcionales.
- Validar las reglas de negocio.
- Verificar requisitos de seguridad.
- Comprobar el funcionamiento de la base de datos.
- Detectar errores antes de producción.
- Validar integraciones.
- Comprobar cálculos financieros.
- Verificar roles y permisos.
- Validar procesos críticos.
- Reducir riesgos operativos.
- Documentar incidencias.
- Garantizar trazabilidad entre requisitos y pruebas.

10.3 Estrategia General de Pruebas

La estrategia de pruebas será incremental.

Cada funcionalidad deberá pasar por diferentes niveles antes de considerarse finalizada.

El flujo propuesto será:



Una funcionalidad no deberá considerarse terminada únicamente porque pueda ejecutarse sin errores visibles.

10.4 Niveles de Prueba

10.4.1 Pruebas Unitarias

Las pruebas unitarias verificarán componentes pequeños e independientes.

En el backend podrán aplicarse principalmente sobre:

- Servicios.
- Validaciones.
- Cálculos.
- Reglas de negocio.
- Modelos.
- Funciones auxiliares.

Ejemplos:

- Calcular fecha final.
- Calcular saldo.
- Validar CI duplicado.
- Calcular nota final.
- Calcular diferencia de caja.
- Determinar si una cuota está vencida.

Una prueba unitaria deberá comprobar un comportamiento específico de manera aislada.

10.4.2 Pruebas de Integración

Las pruebas de integración verificarán que diferentes componentes puedan trabajar conjuntamente.

Ejemplo:

```
Inscripción
↓
Plan de Pago
↓
Pago
↓
Movimiento Financiero
↓
Recibo
```

También se deberán comprobar integraciones como:

- React con Django.
- Django con MariaDB.
- Django con proveedor QR.
- Sistema con impresora térmica.
- Backend con almacenamiento de archivos.

10.4.3 Pruebas Funcionales

Las pruebas funcionales verificarán que el sistema cumpla con las funcionalidades definidas en el SRS.

Cada prueba deberá relacionarse con uno o varios requerimientos funcionales.

Ejemplo:

RF-036 Registrar Estudiante

Prueba:

1. Iniciar sesión como administrativo.
2. Ingresar al módulo estudiantes.
3. Registrar datos válidos.
4. Guardar.
5. Verificar que el estudiante aparezca en el listado.
6. Verificar que el registro exista en base de datos.

10.4.4 Pruebas de Sistema

Estas pruebas evaluarán el comportamiento de SIGA como una solución completa.

Se verificarán flujos de extremo a extremo.

Ejemplo:

```
Registrar estudiante
    ↓
Inscribir
    ↓
Registrar pago
    ↓
Generar recibo
    ↓
Imprimir
    ↓
Consultar reporte financiero
```

10.4.5 Pruebas de Aceptación

Las pruebas de aceptación serán realizadas con participación de usuarios de la institución.

Los usuarios deberán validar que la funcionalidad cumple con el proceso esperado.

Por ejemplo:

- Administrativo valida inscripción.

- Caja valida pagos.
- Dirección valida reportes.
- Docente valida asistencia.
- Estudiante valida prácticas.

Una funcionalidad crítica no deberá considerarse completamente aprobada únicamente mediante pruebas técnicas internas.

10.5 Tipos de Pruebas

10.5.1 Pruebas Positivas

Verificarán que el sistema responda correctamente cuando el usuario proporciona información válida.

Ejemplo:

```
CI no registrado
+
Datos completos
=
Estudiante registrado
```

10.5.2 Pruebas Negativas

Verificarán la respuesta del sistema ante información incorrecta.

Ejemplo:

```
CI duplicado
=
Registro rechazado
```

Otros ejemplos:

- Pago con monto negativo.
- Fecha inválida.
- Curso sin cupos.
- Contraseña incorrecta.
- Archivo no permitido.

10.5.3 Pruebas de Límites

Permitirán comprobar valores extremos.

Ejemplos:

- Pago de 0 Bs.

- Pago igual al saldo.
- Pago mayor al saldo.
- Curso con un cupo disponible.
- Curso lleno.
- Calificación igual a cero.
- Calificación igual al puntaje máximo.
- Fecha final de un mes que inicia el día 31.

10.6 Pruebas del Módulo de Autenticación

Se deberán comprobar como mínimo los siguientes escenarios:

CP-001 – Inicio de sesión correcto

Entrada: Credenciales válidas.

Resultado esperado: Acceso permitido.

CP-002 – Contraseña incorrecta

Resultado esperado: Acceso rechazado.

CP-003 – Usuario inexistente

Resultado esperado: Acceso rechazado sin revelar información sensible.

CP-004 – Usuario desactivado

Resultado esperado: Acceso denegado.

CP-005 – Cierre de sesión

Resultado esperado: La sesión debe quedar invalidada.

CP-006 – Acceso a ruta protegida

Resultado esperado: Usuario no autenticado deberá ser rechazado.

10.7 Pruebas de Roles y Permisos

El sistema deberá comprobar que los usuarios puedan realizar únicamente las operaciones autorizadas.

Ejemplo de matriz:

Acción	Admin	Caja	Docente	Estudiante
Registrar estudiante	✓	X	X	X
Registrar pago	✓	✓	X	X
Anular recibo	✓	Según permiso	X	X
Registrar asistencia	✓	X	✓	X
Consultar calificación propia	✓	X	X	✓

No será suficiente ocultar botones.

El backend deberá impedir la operación cuando el usuario no tenga permiso.

10.8 Pruebas de Estudiantes

Se deberán validar:

- Registro.
- Modificación.
- Consulta.
- Búsqueda.
- Tutores.
- Duplicidad de CI.
- Estados.
- Historial.

CP-020 – Registrar estudiante válido

Resultado esperado: Registro exitoso.

CP-021 – Registrar CI duplicado

Resultado esperado: Operación rechazada.

CP-022 – Registrar sin nombre

Resultado esperado: Validación obligatoria.

CP-023 – Modificar estudiante

Resultado esperado: Datos actualizados y auditoría generada.

10.9 Pruebas de Inscripciones

Este módulo deberá recibir especial atención.

Se deberán probar:

- Curso activo.
- Curso inactivo.
- Cupos disponibles.
- Curso lleno.
- Inscripción duplicada.
- Fecha de inicio.
- Fecha final.
- Descuentos.
- Pago único.
- Pago en cuotas.
- Cancelación.

CP-030 – Inscripción normal

Resultado esperado:

- Inscripción creada.
- Cupo disminuido.
- Obligación financiera creada.

CP-031 – Curso lleno

Resultado esperado:

Inscripción rechazada.

CP-032 – Duplicidad

Resultado esperado:

El sistema deberá detectar la inscripción existente.

10.10 Pruebas de Cálculo de Fechas

Se probarán escenarios especiales.

Ejemplos:

Inicio: 10/01/2026
 Duración: 1 mes
 Esperado: 10/02/2026
 Inicio: 31/01/2026
 Duración: 1 mes
 Esperado: según regla institucional definida

También deberán probarse:

- Febrero.
- Años bisiestos.
- Meses de 30 días.
- Meses de 31 días.

10.11 Pruebas de Pagos

Los pagos constituyen un módulo crítico.

Se deberán validar:

- Pago total.
- Pago parcial.
- Pago en cuotas.
- Métodos de pago.
- Saldos.
- Anulaciones.

CP-040 – Pago completo

Monto total:

Bs 1.000

Pago:

Bs 1.000

Resultado:

Saldo = 0

Estado = PAGADO

CP-041 – Pago parcial

Monto:

Bs 1.000

Pago:

Bs 300

Resultado:

Saldo = Bs 700

CP-042 – Pago superior al saldo

Resultado esperado:

Operación rechazada, salvo política autorizada.

10.12 Pruebas de Transaccionalidad

Se deberá verificar qué ocurre cuando un proceso falla parcialmente.

Ejemplo:

Registrar pago	✓
Actualizar saldo	✓
Crear movimiento	✓
Crear recibo	ERROR

Resultado esperado:

ROLLBACK

La operación completa deberá revertirse cuando corresponda.

No deberá existir:

- Pago sin movimiento.
- Movimiento sin pago.
- Recibo sin pago.

10.13 Pruebas de Recibos

Se deberán validar:

- Generación.
- Numeración.
- Reimpresión.
- Anulación.
- Asociación con pago.

CP-050 – Recibo único

Dos pagos diferentes no deberán recibir el mismo número.

CP-051 – Reimpresión

La reimpresión deberá conservar el número original.

CP-052 – Anulación

El recibo deberá cambiar a ANULADO sin eliminarse.

10.14 Pruebas de Impresora Térmica

Las pruebas deberán realizarse utilizando el hardware real.

Se verificará:

- Conexión.
- Driver.
- Ancho de papel.
- Márgenes.
- Legibilidad.
- Corte de papel, cuando corresponda.
- Logo.
- Caracteres especiales.
- Montos.
- Número de recibo.

Un error físico de impresión no deberá duplicar la operación financiera.

10.15 Pruebas de QR

Las pruebas dependerán del proveedor seleccionado.

Se deberán validar:

- Generación.
- Identificador.
- Monto.
- Expiración.
- Confirmación.
- Rechazo.
- Duplicidad.

CP-060 – QR confirmado

Resultado:

- Transacción confirmada.
- Pago creado.
- Saldo actualizado.
- Ingreso registrado.
- Recibo generado.

CP-061 – QR expirado

Resultado:

No deberá registrarse automáticamente como pago válido.

CP-062 – Confirmación duplicada

Resultado:

El sistema deberá ignorar o rechazar la segunda confirmación.

10.16 Pruebas de Caja

Se probarán:

- Apertura.
- Registro de movimientos.
- Cierre.
- Diferencias.

Ejemplo:

Monto inicial = 100
Ingresos efectivo = 900
Egresos = 100

Esperado en caja = 900

Si el cajero declara:

850

el sistema deberá registrar:

Diferencia = -50

y solicitar justificación.

10.17 Pruebas Financieras

Se deberá comprobar:

- Ingresos.
- Egresos.
- Categorías.
- Balance.

- Salarios.
- Anulaciones.
- Reportes.

Una operación anulada no deberá formar parte de los totales válidos, salvo reportes históricos.

10.18 Pruebas de Docentes

Se deberán validar:

- Registro.
- Asignaciones.
- Horarios.
- Carga horaria.
- Conflictos.

CP-070 – Conflicto de horario

Si un docente ya tiene una clase:

Lunes 08:00 - 10:00

no deberá poder recibir otra asignación:

Lunes 09:00 - 11:00

10.19 Pruebas de Asistencia

Se validarán:

- Sesión de clase.
- Registro masivo.
- Estados.
- Duplicidad.
- Modificación.

No deberá existir más de una asistencia para el mismo estudiante y sesión.

10.20 Pruebas de Materiales

Se deberán comprobar:

- Publicación.
- Archivos.
- Visibilidad.
- Descarga.

- Permisos.

Un estudiante no deberá acceder al material de un curso en el que no está inscrito.

10.21 Pruebas de Prácticas

Se probarán:

- Creación.
- Fechas límite.
- Archivos.
- Entregas.
- Entregas tardías.
- Reemplazos.

Ejemplo:

Fecha límite: 20:00
Entrega: 19:55
Estado: A TIEMPO
Entrega: 20:10
Estado: TARDÍA

cuando se permitan entregas tardías.

10.22 Pruebas de Calificaciones

Se deberán comprobar:

- Nota válida.
- Nota superior al máximo.
- Modificación.
- Publicación.
- Ocultamiento.
- Historial.

Ejemplo:

Puntaje máximo:

20

Nota ingresada:

25

Resultado:

Operación rechazada.

10.23 Pruebas del Portal del Estudiante

Cada estudiante deberá acceder únicamente a:

- Su perfil.
- Sus cursos.
- Sus notas.
- Sus pagos.
- Sus recibos.
- Sus entregas.

Una prueba crítica será intentar acceder manualmente a la información de otro estudiante cambiando identificadores en una URL o solicitud.

Resultado esperado:

Acceso denegado.

10.24 Pruebas de Reportes

Se verificará:

- Filtros.
- Totales.
- Fechas.
- Exportación.
- Permisos.

Los totales mostrados deberán coincidir con las operaciones almacenadas.

10.25 Pruebas de Dashboard

El Dashboard deberá ser contrastado contra reportes equivalentes.

Ejemplo:

Si existen 120 inscripciones activas, el Dashboard no deberá indicar 119 o 121.

Las métricas deberán derivarse de criterios claramente definidos.

10.26 Pruebas de Seguridad

Se deberán realizar pruebas básicas relacionadas con:

- Acceso sin autenticación.
- Escalamiento de privilegios.
- Manipulación de URLs.
- Modificación de identificadores.
- Validación de archivos.
- Inyección.
- XSS.
- CSRF, según arquitectura.
- Sesiones.
- Permisos.

Los detalles técnicos no deberán mostrarse a usuarios finales cuando ocurra un error interno.

10.27 Pruebas de Rendimiento

Se deberán realizar pruebas sobre operaciones frecuentes.

Ejemplos:

- Buscar estudiante.
- Abrir listado.
- Registrar inscripción.
- Consultar pagos.
- Generar reportes.

Se deberá comprobar que el rendimiento continúe siendo aceptable con volúmenes de datos superiores a los iniciales.

10.28 Pruebas de Concurrencia

Las pruebas deberán simular operaciones simultáneas.

Ejemplo:

Dos usuarios intentando obtener el último cupo disponible.

Resultado:

Solo una operación deberá completarse.

Otro escenario:

Dos cajas generando recibos simultáneamente.

Resultado:

No deberán generarse números duplicados.

10.29 Pruebas de Compatibilidad

Se verificará el funcionamiento en:

- Google Chrome.
- Microsoft Edge.
- Mozilla Firefox.

También deberán realizarse pruebas en diferentes resoluciones de pantalla.

10.30 Pruebas de Respaldo

Se deberá verificar que los respaldos:

- Se generen correctamente.
- Contengan los datos necesarios.
- Puedan copiarse.
- Puedan utilizarse durante restauración.

10.31 Pruebas de Restauración

Se deberá realizar periódicamente una restauración en un ambiente controlado.

Procedimiento:

1. Seleccionar respaldo.
2. Restaurar MariaDB.
3. Restaurar archivos.
4. Ejecutar migraciones cuando corresponda.
5. Iniciar aplicación.
6. Consultar información crítica.
7. Verificar integridad.

10.32 Pruebas de Regresión

Cada vez que se modifique una funcionalidad importante deberán repetirse pruebas relacionadas con procesos anteriormente funcionales.

Ejemplo:

Si se modifica el módulo de pagos, deberán revisarse nuevamente:

- Recibos.
- Saldo.

- Caja.
- Reportes.
- QR.
- Movimientos financieros.

Esto evitará que una corrección rompa funcionalidades anteriores.

10.33 Casos de Prueba

Cada caso de prueba podrá documentarse mediante la siguiente estructura:

Campo	Descripción
Código	CP-XXX
Nombre	Nombre de la prueba
Requerimiento	RF relacionado
Precondición	Situación inicial
Datos	Información utilizada
Pasos	Procedimiento
Resultado esperado	Comportamiento correcto
Resultado obtenido	Resultado real
Estado	Aprobado/Rechazado
Evidencia	Captura/log
Observación	Información adicional

10.34 Estados de los Casos de Prueba

Cada prueba podrá tener los estados:

- Pendiente.
- En ejecución.
- Aprobada.
- Fallida.
- Bloqueada.
- Requiere corrección.

10.35 Gestión de Incidencias

Cuando una prueba falle se generará una incidencia.

Cada incidencia deberá registrar:

- Código.
- Módulo.
- Descripción.
- Severidad.
- Prioridad.

- Pasos para reproducir.
- Resultado esperado.
- Resultado obtenido.
- Evidencia.
- Estado.
- Fecha.
- Responsable.

10.36 Clasificación de Errores

Los errores podrán clasificarse por severidad.

Crítica

Impide utilizar una funcionalidad principal o afecta integridad financiera.

Ejemplos:

- Pago duplicado.
- Pérdida de información.
- Acceso financiero no autorizado.

Alta

Afecta una función importante, pero existe una alternativa temporal.

Media

Problema funcional que no impide completamente la operación.

Baja

Problemas visuales o menores.

10.37 Priorización de Correcciones

El orden general será:

1. Errores críticos.
2. Errores de seguridad.
3. Errores financieros.
4. Errores académicos importantes.
5. Errores administrativos.
6. Problemas visuales.

10.38 Entorno de Pruebas

Las pruebas deberán realizarse en un entorno separado del sistema de producción siempre que sea posible.

El entorno deberá disponer de:

- React.
- Django.
- MariaDB.
- Datos ficticios.
- Usuarios de prueba.
- Configuración equivalente a producción.

No deberán utilizarse datos reales innecesariamente durante pruebas técnicas.

10.39 Datos de Prueba

Se deberán crear conjuntos controlados.

Ejemplo:

```
10 estudiantes
5 docentes
3 cursos
3 sucursales
20 inscripciones
30 pagos
```

Posteriormente se aumentará progresivamente el volumen.

10.40 Pruebas con Usuarios Finales

Antes de liberar módulos críticos deberán realizarse sesiones de validación.

Ejemplo:

Administrativo

- Registrar estudiante.
- Inscribir.
- Consultar historial.

Caja

- Registrar pago.
- Imprimir.
- Anular.

Docente

- Registrar asistencia.
- Subir práctica.
- Calificar.

Estudiante

- Descargar material.
- Subir práctica.
- Consultar nota.

10.41 Criterios de Aceptación de un Módulo

Un módulo podrá considerarse aceptado cuando:

- Los requisitos principales estén implementados.
- Los casos críticos estén aprobados.
- No existan errores críticos abiertos.
- Los permisos estén correctamente aplicados.
- La auditoría funcione.
- La documentación correspondiente esté actualizada.
- El usuario responsable haya realizado una validación cuando corresponda.

10.42 Criterios para Paso a Producción

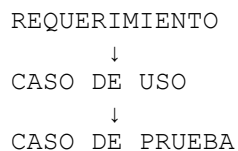
El sistema no deberá pasar a producción hasta comprobar:

- Autenticación.
- Roles.
- Permisos.
- Inscripciones.
- Pagos.
- Recibos.
- Auditoría.
- Respaldos.
- Restauración.
- HTTPS.
- Base de datos.
- Configuración del servidor.

Cuando se trate del MVP, únicamente será obligatorio que los módulos incluidos en dicha versión cumplan los criterios correspondientes.

10.43 Matriz de Trazabilidad de Pruebas

Se establecerá una relación entre:



Ejemplo:

RF	Caso de Uso	Caso de Prueba
RF-036	CU-002	CP-020
RF-116	CU-004	CP-030
RF-156	CU-006	CP-040
RF-175	CU-009	CP-060
RF-217	CU-022	CP-080

Esta matriz facilitará identificar requisitos que todavía no hayan sido probados.

10.44 Automatización de Pruebas

Cuando resulte conveniente se desarrollarán pruebas automatizadas.

Backend:

- Django Test.
- Pytest.

Frontend:

- Herramientas de pruebas compatibles con React.

Las pruebas automatizadas serán especialmente útiles para:

- Cálculos.
- Validaciones.
- Permisos.
- APIs.
- Procesos financieros.
- Reglas críticas.

No todas las pruebas podrán automatizarse.

Por ejemplo:

- Impresión térmica.
- Experiencia del usuario.

- Pruebas físicas.

10.45 Integración con el Ciclo de Desarrollo

Las pruebas formarán parte del trabajo semanal.

Una semana de desarrollo no deberá dejar todas las pruebas para el viernes.

La estrategia será:

```
Desarrollar
  ↓
Probar
  ↓
Corregir
  ↓
Continuar
```

Esto permitirá detectar errores tempranamente.

10.46 Registro de Evidencias

Las pruebas críticas deberán conservar evidencia cuando sea relevante.

Ejemplos:

- Capturas.
- Logs.
- Recibos impresos.
- Reportes.
- Respuestas API.
- Resultados de pruebas automatizadas.

10.47 Responsabilidades

Desarrollador

Será responsable de:

- Diseñar pruebas técnicas.
- Ejecutar pruebas.
- Corregir errores.
- Mantener pruebas automatizadas.
- Documentar resultados.

Usuarios Institucionales

Serán responsables de:

- Validar procesos de negocio.
- Detectar diferencias respecto a la operación real.
- Aprobar funcionalidades cuando corresponda.

Dirección

Será responsable de:

- Validar procesos críticos.
- Aprobar criterios institucionales.
- Priorizar correcciones relevantes.

10.48 Indicadores de Calidad

Se podrán utilizar indicadores como:

Porcentaje de casos aprobados

$\text{Casos aprobados} / \text{Casos ejecutados} \times 100$

Incidencias críticas abiertas

Deberá ser igual a cero antes de producción.

Cobertura de requisitos

$\text{Requisitos probados} / \text{Requisitos implementados} \times 100$

Tasa de corrección

Cantidad de incidencias cerradas durante un periodo.

10.49 Pruebas Prioritarias para el MVP

Antes de liberar el MVP deberán aprobarse obligatoriamente:

1. Inicio de sesión.
2. Roles.
3. Registro de estudiante.
4. Búsqueda.
5. Registro de curso.
6. Inscripción.
7. Fecha final.

8. Pago total.
9. Pago parcial.
10. Pago en cuotas.
11. Saldo.
12. Recibo.
13. Numeración.
14. Impresión térmica.
15. Anulación.
16. Auditoría.
17. Reportes básicos.
18. Respaldo.

10.50 Cierre del Capítulo

El plan de pruebas definido establece una estrategia integral para verificar la calidad del Sistema Integral de Gestión Académica y Administrativa durante todas las etapas de su desarrollo.

La ejecución progresiva de pruebas unitarias, de integración, funcionales, de sistema y de aceptación permitirá identificar errores antes de que afecten a los usuarios finales.

Debido a la naturaleza académica y financiera del sistema, se dará especial prioridad a las pruebas relacionadas con integridad de datos, control de acceso, pagos, recibos, QR, caja, salarios, calificaciones y auditoría.

La aplicación de casos de prueba trazables respecto a los requerimientos permitirá mantener evidencia objetiva del cumplimiento funcional del sistema y facilitará la validación de cada módulo por parte de la institución.

Finalmente, ninguna funcionalidad crítica deberá considerarse terminada hasta haber sido implementada, probada, corregida y validada conforme a los criterios establecidos en este capítulo.

CAPÍTULO 11 IMPLEMENTACIÓN, DESPLIEGUE Y PUESTA EN PRODUCCIÓN

11.1 Introducción

El presente capítulo describe la estrategia propuesta para implementar, desplegar y poner en funcionamiento el Sistema Integral de Gestión Académica y Administrativa de Sumo (SIGA – SUMO) dentro de la infraestructura tecnológica utilizada por la institución.

La puesta en producción constituye una etapa crítica del proyecto, debido a que representa la transición desde un entorno controlado de desarrollo y pruebas hacia un entorno donde el sistema será utilizado con información institucional real.

Durante esta etapa deberán considerarse aspectos relacionados con infraestructura, seguridad, configuración de servidores, migración de base de datos, despliegue del frontend y backend, configuración de dominio y certificados de seguridad, gestión de archivos, respaldos, validación de impresoras térmicas, integración con pagos QR y capacitación de usuarios.

La implementación será realizada de manera progresiva con el objetivo de reducir riesgos operativos y evitar que una falla durante el despliegue afecte los procesos normales de la institución.

La estrategia de implementación deberá garantizar como mínimo:

- Disponibilidad del sistema.
- Integridad de los datos.
- Seguridad de las comunicaciones.
- Correcta configuración de los servicios.
- Respaldo previo de la información.
- Posibilidad de recuperación ante fallos.
- Validación de funcionalidades críticas.
- Capacitación de los usuarios.
- Seguimiento posterior a la puesta en producción.

11.2 Objetivos de la Implementación

El objetivo principal de esta etapa será trasladar de manera segura y controlada SIGA desde los ambientes de desarrollo y pruebas hacia un ambiente de producción preparado para ser utilizado por los usuarios reales de la institución.

Los objetivos específicos serán:

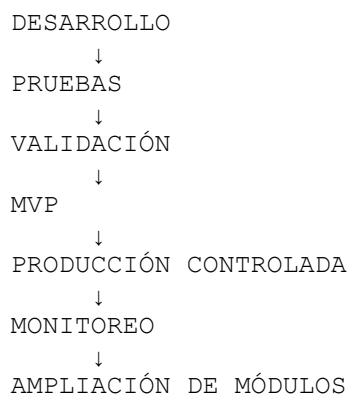
- Preparar la infraestructura de producción.
- Instalar y configurar los componentes tecnológicos necesarios.
- Desplegar el backend desarrollado con Django.
- Desplegar el frontend desarrollado con React.
- Configurar MariaDB.
- Configurar Nginx.
- Implementar HTTPS.
- Configurar variables de entorno.
- Preparar el almacenamiento de archivos.
- Crear mecanismos de respaldo.
- Validar la impresora térmica.
- Configurar las integraciones externas.
- Crear los usuarios institucionales iniciales.
- Migrar información cuando corresponda.
- Ejecutar pruebas de producción.
- Capacitar a usuarios.
- Definir mecanismos de reversión ante fallos.

11.3 Estrategia de Implementación

La implementación se realizará de forma gradual.

No se recomienda habilitar todos los módulos simultáneamente en la primera puesta en producción.

Se utilizará una estrategia de implementación progresiva:



La primera versión productiva deberá priorizar los procesos institucionales esenciales.

Inicialmente podrán habilitarse:

1. Usuarios.
2. Estudiantes.
3. Gestión académica básica.
4. Inscripciones.
5. Pagos.
6. Recibos.
7. Impresión térmica.
8. Reportes básicos.

Posteriormente podrán incorporarse:

- Finanzas.
- QR.
- Portal docente.
- Asistencia.
- Materiales.
- Prácticas.
- Calificaciones.
- Portal estudiante.
- Reportes avanzados.

11.4 Entornos del Sistema

El proyecto deberá manejar, cuando sea técnicamente posible, tres ambientes claramente separados.

11.4.1 Ambiente de Desarrollo

Será utilizado exclusivamente durante la construcción del sistema.

Este ambiente permitirá:

- Escribir código.
- Realizar pruebas locales.
- Crear migraciones.
- Probar funcionalidades nuevas.
- Modificar estructuras.
- Utilizar datos ficticios.

El ambiente de desarrollo podrá ejecutarse en el equipo del desarrollador.

11.4.2 Ambiente de Pruebas

Será utilizado para validar funcionalidades antes de su liberación.

Este entorno deberá aproximarse lo máximo posible al ambiente de producción.

Permitirá comprobar:

- Integraciones.
- Migraciones.
- APIs.
- Roles.
- Permisos.
- Reportes.
- Procesos financieros.
- Impresión.
- Flujos completos.

No deberán utilizarse operaciones financieras reales en este entorno.

11.4.3 Ambiente de Producción

Será el entorno donde se almacenará y procesará la información real.

Este ambiente deberá disponer de:

- Acceso restringido.
- HTTPS.

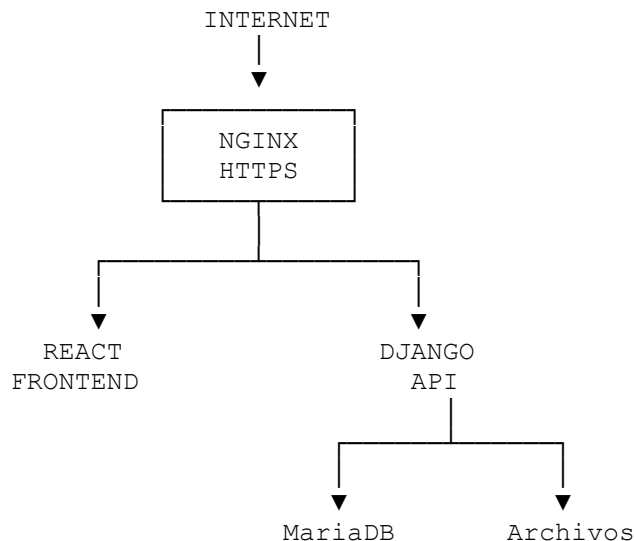
- Respaldos.
- Monitoreo.
- Configuración segura.
- Base de datos productiva.
- Gestión controlada de cambios.

Las modificaciones directas sobre producción deberán minimizarse.

11.5 Infraestructura Inicial

Considerando el volumen inicial del instituto, SIGA podrá comenzar operando en un único servidor virtual correctamente dimensionado.

La arquitectura inicial podrá ser:



Esta arquitectura será suficiente inicialmente y podrá evolucionar posteriormente.

11.6 Características del Servidor

El servidor deberá contar con recursos suficientes para atender las operaciones previstas.

Como referencia inicial se podrá considerar:

- Sistema operativo Linux.
- Procesador de múltiples núcleos.
- Memoria RAM suficiente para aplicación y base de datos.
- Almacenamiento SSD.
- Espacio adicional para archivos.
- Conectividad estable.
- Dirección IP pública o infraestructura equivalente.
- Copias de seguridad externas.

La capacidad definitiva dependerá del proveedor seleccionado y del crecimiento del sistema.

11.7 Sistema Operativo

Se recomienda utilizar una distribución Linux orientada a servidores.

Entre las alternativas podrán considerarse:

- Ubuntu Server.
- Debian.

El sistema operativo deberá mantenerse actualizado con parches de seguridad.

Los servicios innecesarios deberán permanecer deshabilitados.

11.8 Configuración de Usuarios del Servidor

No se recomienda ejecutar la aplicación permanentemente utilizando la cuenta `root`.

Se deberán crear usuarios específicos para:

- Administración.
- Despliegue.
- Aplicación.

Los permisos deberán seguir el principio de mínimo privilegio.

11.9 Preparación de MariaDB

MariaDB será instalado y configurado dentro del ambiente productivo.

Se deberá:

1. Instalar MariaDB.
2. Aplicar configuraciones básicas de seguridad.
3. Crear la base de datos de SIGA.
4. Crear un usuario específico.
5. Asignar únicamente los permisos necesarios.
6. Configurar codificación adecuada.
7. Verificar conexión con Django.
8. Configurar respaldos.

Ejemplo conceptual:

Base de datos:
`sigaproductiva`

Usuario:
siga_app

Las contraseñas no deberán documentarse dentro del SRS ni almacenarse en Git.

11.10 Protección de MariaDB

La base de datos no deberá exponerse directamente a usuarios externos.

El acceso deberá estar limitado al backend y a conexiones administrativas autorizadas.

Se deberán aplicar:

- Firewall.
- Usuarios restringidos.
- Contraseñas robustas.
- Respaldos.
- Monitoreo.
- Actualizaciones.

11.11 Preparación del Backend

El backend desarrollado con Python y Django deberá instalarse utilizando un entorno aislado.

Conceptualmente:

```
Servidor
├── siga/
│   └── backend/
│       └── virtualenv
```

Las dependencias deberán instalarse desde un archivo controlado por versiones.

Ejemplo:

```
requirements.txt
```

o un mecanismo equivalente.

11.12 Variables de Entorno

Las configuraciones sensibles deberán almacenarse mediante variables de entorno.

Ejemplos:

```
SECRET_KEY
DATABASE_NAME
```

```
DATABASE_USER
DATABASE_PASSWORD
DATABASE_HOST
EMAIL_HOST
QR_API_KEY
QR_API_SECRET
ALLOWED_HOSTS
```

Estos valores no deberán estar incluidos directamente en el código fuente.

11.13 Configuración de Django para Producción

La configuración productiva deberá diferenciarse de desarrollo.

Como mínimo deberán revisarse:

- DEBUG desactivado.
- ALLOWED_HOSTS.
- Base de datos productiva.
- HTTPS.
- Cookies seguras.
- Archivos estáticos.
- Archivos multimedia.
- Logging.
- Correo.
- Seguridad.

No deberá desplegarse una configuración de desarrollo en producción.

11.14 Servidor de Aplicación

Django no deberá ejecutarse en producción utilizando directamente el servidor de desarrollo.

Se utilizará un servidor de aplicación compatible, como Gunicorn u otra alternativa adecuada.

La arquitectura será:

```
NGINX
  ↓
GUNICORN
  ↓
DJANGO
```

Nginx recibirá las solicitudes externas y las redirigirá al servidor de aplicación.

11.15 Configuración de Nginx

Nginx será utilizado como proxy inverso y servidor frontal.

Sus principales responsabilidades serán:

- Recibir solicitudes HTTP/HTTPS.
- Redirigir tráfico hacia Django.
- Servir archivos estáticos.
- Servir el frontend React.
- Gestionar certificados TLS.
- Aplicar límites de solicitudes cuando sea necesario.
- Gestionar cabeceras de seguridad.

11.16 Despliegue del Frontend React

El frontend desarrollado con React y TypeScript deberá compilarse antes de ser publicado.

El proceso será conceptualmente:

```
Código React
  ↓
Instalar dependencias
  ↓
Ejecutar build
  ↓
Archivos optimizados
  ↓
Servidor Nginx
```

El navegador de los usuarios descargará la aplicación compilada desde el servidor.

11.17 Comunicación Frontend-Backend

React se comunicará con Django utilizando HTTPS.

Ejemplo:

```
https://siga.instituto.bo/api/
```

La dirección definitiva dependerá del dominio adquirido.

El frontend no deberá conocer credenciales de MariaDB.

Toda información deberá procesarse mediante la API.

11.18 Dominio del Sistema

Se recomienda utilizar un dominio o subdominio institucional.

Ejemplos conceptuales:

`sigainstituto.bo`

o:

`sisteminstituto.bo`

El dominio deberá apuntar al servidor donde se encuentre desplegada la aplicación.

11.19 Certificado HTTPS

El sistema deberá funcionar mediante HTTPS.

Esto permitirá proteger:

- Credenciales.
- Datos personales.
- Información académica.
- Información financiera.
- Tokens o sesiones.

El certificado podrá ser emitido mediante un proveedor reconocido.

Se deberá configurar renovación automática cuando sea posible.

11.20 Configuración del Firewall

El servidor deberá exponer únicamente los servicios necesarios.

Por ejemplo:

- HTTPS.
- SSH restringido.
- HTTP únicamente para redirección a HTTPS, cuando corresponda.

MariaDB no deberá quedar públicamente accesible.

11.21 Gestión de Archivos

El sistema deberá disponer de una ubicación segura para almacenar:

- Materiales.
- Tareas.
- Entregas.
- Fotografías.
- Comprobantes.
- Documentos.

La estructura podrá organizarse conceptualmente por categorías:

```
media/  
├── estudiantes/  
├── docentes/  
├── materiales/  
├── practicas/  
├── entregas/  
└── documentos/
```

Los nombres físicos deberán generarse de manera segura.

11.22 Permisos de Archivos

Los directorios del sistema deberán tener permisos apropiados.

El usuario que ejecuta Django deberá poder escribir únicamente en los directorios necesarios.

No deberán establecerse permisos excesivamente amplios sin justificación.

11.23 Migraciones de Base de Datos

Las migraciones deberán ejecutarse antes de habilitar una nueva versión.

El proceso general será:

```
Código actualizado  
↓  
Respaldar  
↓  
Aplicar migraciones  
↓  
Verificar  
↓  
Reiniciar aplicación
```

Las migraciones deberán probarse previamente en el entorno de pruebas.

11.24 Datos Iniciales

Antes de habilitar el sistema deberán registrarse datos institucionales básicos.

Entre ellos:

- Institución.
- Sucursales.
- Roles.
- Permisos.
- Usuario administrador.

- Métodos de pago.
- Categorías financieras.
- Periodo académico.
- Configuraciones generales.

Estos datos podrán cargarse mediante scripts o procedimientos controlados.

11.25 Creación del Administrador Inicial

Durante la instalación deberá crearse una cuenta administrativa inicial.

Esta cuenta se utilizará para:

- Configuración.
- Creación de usuarios.
- Roles.
- Parámetros.

Posteriormente podrán crearse las cuentas institucionales correspondientes.

11.26 Migración desde el Sistema Anterior

Actualmente parte de la información institucional se encuentra registrada en un sistema previo.

Antes de migrarla deberá realizarse un análisis de:

- Formato.
- Calidad.
- Duplicados.
- Campos disponibles.
- Información faltante.
- Compatibilidad.

La migración no deberá realizarse directamente sin validar previamente los datos.

11.27 Proceso de Migración de Datos

El procedimiento será:

```
EXPORTAR
↓
ANALIZAR
↓
LIMPIAR
↓
TRANSFORMAR
↓
```

IMPORTAR EN PRUEBAS
↓
VALIDAR
↓
IMPORTAR EN PRODUCCIÓN

Se deberá conservar una copia original de la información recibida.

11.28 Datos que Podrán Migrarse

Dependiendo de la disponibilidad podrán migrarse:

- Estudiantes.
- Docentes.
- Cursos.
- Inscripciones.
- Información de contacto.

Los datos financieros históricos únicamente deberán migrarse después de una validación exhaustiva.

11.29 Validación Posterior a la Migración

Se deberá comprobar:

- Cantidad de registros.
- Duplicados.
- Estudiantes.
- Cursos.
- Estados.
- Saldos, cuando corresponda.

La institución deberá participar en la validación antes de considerar aprobada la migración.

11.30 Estrategia de Puesta en Producción

Se recomienda realizar una puesta en marcha progresiva.

Etapas 1

Personal administrativo reducido.

Etapas 2

Caja.

Etapas 3

Todos los administrativos.

Etapas 4

Docentes.

Etapas 5

Estudiantes.

Esto permitirá detectar errores con un número controlado de usuarios.

11.31 Implementación Piloto

Antes de habilitar SIGA simultáneamente en todas las sucursales, podrá seleccionarse una sucursal o grupo reducido para realizar un piloto.

Durante el piloto se evaluará:

- Inscripción.
- Pago.
- Recibo.
- Impresión.
- Rendimiento.
- Facilidad de uso.

Si los resultados son satisfactorios, se ampliará el sistema al resto de las sucursales.

11.32 Periodo de Operación Paralela

Durante una transición inicial podrá considerarse mantener temporalmente:

- SIGA.
- Procedimiento anterior.

El objetivo será comparar resultados y reducir riesgo.

Sin embargo, el periodo paralelo deberá ser limitado para evitar duplicación permanente del trabajo.

11.33 Impresora Térmica en Producción

Antes de habilitar caja se deberán realizar pruebas reales de impresión.

Se validará:

- Modelo.
- Conexión.
- Driver.
- Papel.
- Márgenes.
- Logo.
- Datos.
- Velocidad.
- Corte.

Cada sucursal deberá ser validada si utiliza equipos diferentes.

11.34 Contingencia de Impresión

Si la impresora falla, el pago ya confirmado no deberá registrarse nuevamente.

El usuario deberá poder:

1. Consultar recibo.
2. Resolver problema físico.
3. Reimprimir.

Esta regla evitará duplicidades financieras.

11.35 Configuración QR en Producción

Antes de utilizar pagos QR reales deberán configurarse:

- Credenciales productivas.
- URLs.
- Certificados, cuando corresponda.
- Webhooks o mecanismos de consulta.
- Claves.
- Validación de firmas.
- Tiempo de expiración.

Las credenciales de pruebas deberán mantenerse separadas de las productivas.

11.36 Prueba Financiera Controlada

Antes de liberar pagos QR deberá ejecutarse una operación real de bajo riesgo autorizada por la institución.

Deberá verificarse:

Generar QR
↓
Pagar
↓
Confirmar
↓
Registrar pago
↓
Actualizar saldo
↓
Generar ingreso
↓
Generar recibo

11.37 Respaldo Previo a Producción

Antes de cada actualización importante deberá existir un respaldo válido.

Como mínimo:

- MariaDB.
- Archivos.
- Configuración relevante.

Esto permitirá revertir cambios si ocurre un fallo.

11.38 Plan de Reversión

Toda implementación importante deberá considerar cómo regresar a la versión anterior.

El plan podrá incluir:

1. Detener aplicación.
2. Restaurar código anterior.
3. Restaurar base de datos si fuese necesario.
4. Restaurar configuración.
5. Verificar funcionamiento.
6. Reactivar servicio.

El proceso dependerá de la naturaleza del cambio.

11.39 Monitoreo Posterior al Despliegue

Después de cada despliegue importante se deberá observar:

- Errores.
- Rendimiento.
- Consumo de recursos.
- Accesos.

- Fallos de integración.
- Incidencias reportadas.

Las primeras horas de una nueva versión requerirán especial atención.

11.40 Gestión de Logs

Los logs técnicos podrán contener información sobre:

- Errores de Django.
- Fallos de API.
- Nginx.
- Integraciones QR.
- Procesos de respaldo.

No deberán registrar innecesariamente contraseñas o información sensible.

11.41 Respallos en Producción

Se establecerá una política formal.

Como referencia inicial:

Diario

Base de datos.

Semanal

Base de datos + archivos.

Mensual

Respaldo histórico.

La política definitiva deberá ajustarse al volumen y criticidad de la información.

11.42 Ubicación de Respallos

No se recomienda conservar todas las copias únicamente en el mismo servidor.

Idealmente deberán existir copias en otra ubicación o servicio.

Esto protegerá frente a:

- Daño físico.

- Fallo del servidor.
- Error humano.
- Corrupción.

11.43 Pruebas de Restauración

Deberán realizarse pruebas periódicas de restauración.

Se verificará:

- Base de datos.
- Archivos.
- Usuarios.
- Inscripciones.
- Pagos.
- Recibos.

Un respaldo que no puede restaurarse no deberá considerarse una protección adecuada.

11.44 Capacitación del Personal Administrativo

Antes de utilizar el sistema, el personal administrativo deberá recibir capacitación sobre:

- Login.
- Estudiantes.
- Inscripciones.
- Cursos.
- Búsqueda.
- Pagos.
- Recibos.
- Reportes.

11.45 Capacitación de Caja

Caja deberá recibir capacitación específica sobre:

- Apertura.
- Pagos.
- Pagos parciales.
- Cuotas.
- QR.
- Recibos.
- Reimpresión.
- Anulaciones.
- Cierre.
- Arqueo.

Debido al impacto financiero, esta capacitación será prioritaria.

11.46 Capacitación de Docentes

Los docentes deberán conocer:

- Acceso.
- Horarios.
- Cursos.
- Estudiantes.
- Asistencia.
- Materiales.
- Prácticas.
- Calificaciones.

11.47 Capacitación de Estudiantes

El portal deberá diseñarse para requerir capacitación mínima.

No obstante, se podrá proporcionar:

- Guía rápida.
- Tutorial.
- Preguntas frecuentes.

El estudiante deberá aprender principalmente a:

- Ingresar.
- Consultar curso.
- Descargar material.
- Entregar práctica.
- Consultar nota.
- Consultar pagos.

11.48 Manuales de Usuario

Deberán elaborarse manuales diferenciados por perfil.

Como mínimo:

- Manual Administrativo.
- Manual Caja.
- Manual Docente.
- Manual Estudiante.
- Manual Administrador.

Los manuales deberán incluir capturas cuando el sistema tenga interfaces definitivas.

11.49 Manual Técnico

También deberá elaborarse documentación para mantenimiento.

Deberá incluir:

- Arquitectura.
- Repositorio.
- Instalación.
- Variables de entorno.
- Base de datos.
- Migraciones.
- Despliegue.
- Backups.
- Restauración.
- Integraciones.

11.50 Checklist de Puesta en Producción

Antes de liberar una versión deberán verificarse como mínimo:

- ☐ Servidor configurado.
- ☐ MariaDB operativa.
- ☐ Migraciones ejecutadas.
- ☐ React desplegado.
- ☐ Django desplegado.
- ☐ HTTPS activo.
- ☐ DEBUG desactivado.
- ☐ Usuarios iniciales creados.
- ☐ Roles configurados.
- ☐ Respaldos configurados.
- ☐ Restauración probada.
- ☐ Impresora validada.
- ☐ Reportes principales validados.
- ☐ Pagos probados.
- ☐ Auditoría funcionando.
- ☐ Manuales disponibles.
- ☐ Usuarios capacitados.

11.51 Criterios de Aceptación de Producción

La versión podrá considerarse apta para producción cuando:

- No existan errores críticos abiertos.
- Los procesos principales hayan sido probados.
- Los respaldos funcionen.
- Los permisos estén configurados.
- Los usuarios responsables hayan validado los procesos.
- La infraestructura sea estable.
- Exista un plan de reversión.

11.52 Responsabilidades Durante el Despliegue

Desarrollador

Será responsable de:

- Preparar la aplicación.
- Ejecutar despliegue.
- Configurar servicios.
- Ejecutar migraciones.
- Verificar funcionamiento.
- Resolver incidencias técnicas.

Dirección

Será responsable de:

- Aprobar puesta en producción.
- Coordinar personal.
- Validar políticas.
- Autorizar migración.

Usuarios Clave

Serán responsables de:

- Validar procesos.
- Reportar incidencias.
- Participar en pruebas.

11.53 Mantenimiento Inmediato Posterior

Después de una nueva puesta en producción deberá existir un periodo de seguimiento intensivo.

Se priorizarán:

- Errores críticos.
- Problemas de datos.
- Fallos de impresión.
- Fallos de pago.
- Problemas de permisos.
- Rendimiento.

Las mejoras no urgentes podrán registrarse para siguientes versiones.

11.54 Actualizaciones Futuras

Las nuevas versiones deberán seguir un proceso controlado:

```
Desarrollo
↓
Pruebas
↓
Respaldo
↓
Despliegue
↓
Migración
↓
Validación
↓
Monitoreo
```

No se recomienda modificar directamente el servidor productivo sin control de versiones.

11.55 Versionamiento de la Aplicación

Cada versión importante podrá identificarse mediante numeración.

Ejemplo:

SIGA 1.0.0

Donde conceptualmente:

1 = versión principal
0 = funcionalidad secundaria
0 = corrección

Ejemplo:

1.0.0 MVP inicial
1.1.0 Finanzas
1.2.0 Portal docente

1.2.1 Corrección de asistencia

Esto facilitará conocer qué versión se encuentra instalada.

11.56 Registro de Despliegues

Cada despliegue deberá registrar:

- Versión.
- Fecha.
- Responsable.
- Cambios.
- Migraciones.
- Incidencias.
- Resultado.

Esto permitirá mantener historial de evolución del sistema.

11.57 Indicadores de Puesta en Producción

Se podrán utilizar indicadores como:

- Cantidad de usuarios activos.
- Inscripciones realizadas.
- Pagos registrados.
- Errores críticos.
- Tiempo promedio de respuesta.
- Incidencias reportadas.
- Fallos de impresión.
- Transacciones QR fallidas.

Estos indicadores ayudarán a evaluar la estabilidad.

11.58 Riesgos de Implementación

Entre los principales riesgos se consideran:

- Fallos de conectividad.
- Problemas de servidor.
- Datos incorrectos durante migración.
- Configuración incorrecta.
- Falta de capacitación.
- Fallas de impresora.
- Problemas con proveedor QR.
- Cambios simultáneos no controlados.
- Falta de respaldo.

11.59 Mitigación de Riesgos

Se deberán aplicar acciones como:

- Respaldar antes de cambios.
- Probar migraciones.
- Implementar progresivamente.
- Capacitar usuarios.
- Mantener procedimiento anterior temporalmente cuando sea necesario.
- Mantener logs.
- Probar hardware.
- Mantener documentación.

11.60 Cierre del Capítulo

La implementación y puesta en producción del Sistema Integral de Gestión Académica y Administrativa constituye una etapa crítica que deberá ejecutarse de forma planificada, gradual y controlada.

La estrategia definida permitirá trasladar progresivamente las funcionalidades desarrolladas desde los ambientes de desarrollo y pruebas hacia un entorno productivo seguro, garantizando la integridad de la información y reduciendo los riesgos asociados a la transición tecnológica.

La utilización de Linux, Nginx, Django, React y MariaDB permitirá implementar una infraestructura inicial adecuada para las necesidades actuales, manteniendo la posibilidad de escalar los componentes conforme aumente la cantidad de usuarios y operaciones.

Asimismo, la implementación progresiva mediante un MVP permitirá que la institución obtenga beneficios operativos antes de completar el conjunto total de funcionalidades.

La correcta aplicación de respaldos, migraciones controladas, HTTPS, gestión de configuración, validación de hardware, pruebas de integraciones, capacitación y monitoreo posterior permitirá reducir significativamente los riesgos de interrupciones o pérdida de información.

Finalmente, toda actualización futura deberá seguir procedimientos controlados de desarrollo, pruebas, respaldo, despliegue y validación, garantizando que la evolución de SIGA mantenga los niveles de seguridad, estabilidad y calidad establecidos en el presente documento.

CAPÍTULO 12 MANTENIMIENTO, SOPORTE Y CONTINUIDAD OPERATIVA

12.1 Introducción

El presente capítulo establece los lineamientos generales para el mantenimiento, soporte técnico y continuidad operativa del Sistema Integral de Gestión Académica y Administrativa (SIGA) después de su implementación y puesta en producción.

El desarrollo de un sistema informático no concluye con su publicación. Una vez que SIGA comience a utilizarse con información real de la institución será necesario realizar actividades permanentes orientadas a mantener su estabilidad, corregir errores, actualizar dependencias, proteger la información y adaptar determinadas funcionalidades conforme evolucionen los procesos institucionales.

Debido a que el sistema administrará información académica, administrativa y financiera, las actividades de mantenimiento deberán realizarse mediante procedimientos controlados, evitando modificaciones improvisadas directamente en el entorno productivo.

Asimismo, se deberá establecer una estrategia de continuidad operativa que permita disminuir el impacto de incidentes relacionados con fallos del servidor, pérdida de conectividad, errores de software, problemas de base de datos o indisponibilidad de servicios externos.

El objetivo será garantizar que SIGA pueda continuar prestando sus servicios de manera segura, estable y sostenible durante su vida útil.

12.2 Objetivos del Mantenimiento

El mantenimiento tendrá como objetivo principal conservar el correcto funcionamiento del sistema después de su puesta en producción.

De manera específica se buscará:

- Corregir errores identificados durante la operación.
- Mantener actualizado el software utilizado.
- Mejorar funcionalidades existentes.
- Incorporar nuevas necesidades institucionales de forma controlada.
- Mantener la seguridad de la plataforma.
- Optimizar el rendimiento.
- Verificar periódicamente los respaldos.
- Mantener actualizada la documentación.
- Controlar la evolución de la base de datos.
- Monitorear los servicios externos.
- Reducir los tiempos de indisponibilidad.
- Mantener la trazabilidad de los cambios.

12.3 Tipos de Mantenimiento

El mantenimiento de SIGA se clasificará en diferentes categorías de acuerdo con el objetivo de las modificaciones realizadas.

12.3.1 Mantenimiento Correctivo

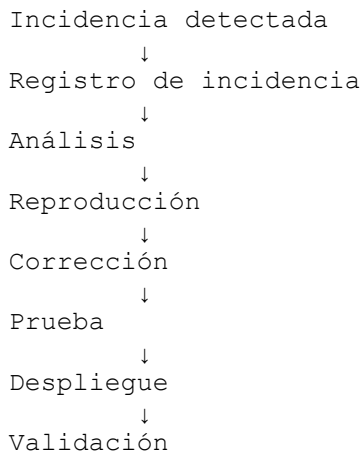
El mantenimiento correctivo estará orientado a solucionar errores o comportamientos incorrectos detectados durante la utilización del sistema.

Ejemplos:

- Error al registrar una inscripción.
- Problema de visualización de un reporte.
- Fallo durante la impresión de recibos.
- Error en algún cálculo.
- Problema de permisos.
- Error al cargar determinado archivo.
- Comportamiento incorrecto de una interfaz.

Las incidencias deberán analizarse antes de aplicar modificaciones en producción.

El procedimiento general será:



12.3.2 Mantenimiento Adaptativo

Este tipo de mantenimiento será necesario cuando cambien elementos externos al sistema.

Ejemplos:

- Nueva versión del sistema operativo.
- Cambios del proveedor de QR.
- Modificaciones en navegadores.
- Cambio de servidor.
- Nueva impresora térmica.
- Cambio de dominio.
- Nuevos servicios institucionales.

El objetivo será mantener SIGA compatible con el entorno tecnológico utilizado por la institución.

12.3.3 Mantenimiento Perfectivo

El mantenimiento perfectivo estará orientado a mejorar funcionalidades existentes sin corregir necesariamente un error.

Ejemplos:

- Mejorar una pantalla.
- Reducir pasos de inscripción.
- Incorporar nuevos filtros.
- Mejorar reportes.
- Optimizar búsquedas.
- Mejorar el Dashboard.
- Agregar nuevas estadísticas.

Estas mejoras deberán priorizarse según su impacto y necesidad institucional.

12.3.4 Mantenimiento Preventivo

El mantenimiento preventivo buscará reducir la posibilidad de fallos futuros.

Entre sus actividades se encuentran:

- Actualización de dependencias.
- Revisión de logs.
- Revisión de almacenamiento.
- Optimización de consultas.
- Verificación de respaldos.
- Revisión de seguridad.
- Eliminación controlada de archivos temporales.
- Pruebas de restauración.
- Revisión de certificados HTTPS.

Este mantenimiento deberá realizarse periódicamente.

12.4 Gestión de Incidencias

Toda incidencia relevante deberá registrarse antes de iniciar su resolución.

Cada incidencia deberá incluir como mínimo:

- Código.
- Fecha.

- Usuario que reporta.
- Módulo afectado.
- Descripción.
- Pasos para reproducir.
- Prioridad.
- Severidad.
- Estado.
- Responsable.
- Solución aplicada.
- Fecha de cierre.

Los estados podrán ser:

- Reportada.
- En análisis.
- En desarrollo.
- En pruebas.
- Resuelta.
- Cerrada.
- Rechazada.

12.5 Clasificación de Incidencias

Crítica

Incendencia que impide utilizar una función esencial o puede comprometer la integridad de la información.

Ejemplos:

- Pagos duplicados.
- Pérdida de información.
- Acceso financiero no autorizado.
- Sistema completamente inaccesible.
- Error grave en recibos.

Alta

Afecta una funcionalidad importante pero no detiene completamente la operación.

Media

Afecta parcialmente una función y existe una alternativa temporal.

Baja

Problemas menores, principalmente visuales o de experiencia del usuario.

12.6 Priorización del Soporte

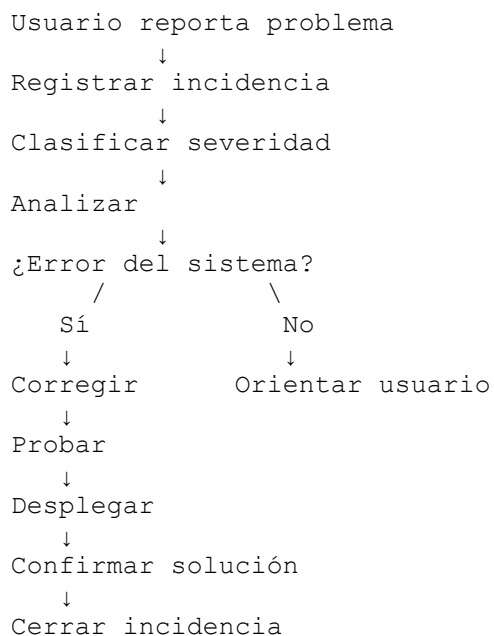
Las incidencias serán atendidas considerando principalmente:

1. Seguridad.
2. Integridad financiera.
3. Disponibilidad.
4. Inscripciones.
5. Pagos y recibos.
6. Información académica.
7. Reportes.
8. Problemas visuales.

La prioridad podrá modificarse cuando dirección determine que una incidencia tiene un impacto institucional mayor.

12.7 Flujo de Atención de Soporte

El proceso general de soporte será:



12.8 Canales de Soporte

Durante la operación podrán establecerse diferentes medios institucionales para reportar incidencias.

Por ejemplo:

- Sistema interno de tickets.
- Correo institucional.
- Formulario.
- Canal autorizado de comunicación.

Se recomienda evitar que las incidencias importantes dependan exclusivamente de conversaciones informales, debido a que dificultan su seguimiento.

12.9 Registro de Cambios

Toda actualización liberada deberá contar con un registro de cambios.

Ejemplo:

Versión: SIGA 1.2.1
Fecha: 15/03/2027

Cambios:

- Corrección de impresión de recibo.
- Optimización de búsqueda de estudiantes.
- Corrección de cálculo de saldo.

El registro permitirá identificar qué funcionalidades fueron modificadas en cada versión.

12.10 Versionamiento

SIGA utilizará una estrategia de versionamiento.

Ejemplo:

1.0.0

Conceptualmente:

- Primer número: cambio principal.
- Segundo número: nuevas funcionalidades compatibles.
- Tercer número: correcciones.

Ejemplos:

1.0.0 Primera versión del MVP
1.1.0 Gestión financiera
1.2.0 Portal docente
1.2.1 Corrección de asistencia
2.0.0 Cambio estructural importante

12.11 Actualización de Dependencias

Las tecnologías utilizadas evolucionarán durante la vida del sistema.

Por ello será necesario revisar periódicamente:

- Python.
- Django.
- Django REST Framework.
- React.
- TypeScript.
- MariaDB.
- Nginx.
- Librerías complementarias.

Las actualizaciones importantes deberán probarse previamente antes de instalarse en producción.

12.12 Gestión de Vulnerabilidades

Las dependencias utilizadas deberán revisarse periódicamente para identificar vulnerabilidades conocidas.

Cuando exista una vulnerabilidad relevante se deberá:

1. Analizar impacto.
2. Identificar versión corregida.
3. Actualizar en pruebas.
4. Ejecutar pruebas de regresión.
5. Desplegar actualización.
6. Verificar funcionamiento.

12.13 Mantenimiento de la Base de Datos

MariaDB deberá recibir mantenimiento periódico.

Las actividades podrán incluir:

- Revisión de espacio.
- Revisión de índices.
- Optimización de consultas.
- Verificación de integridad.
- Monitoreo de rendimiento.
- Respaldos.
- Pruebas de restauración.

No se deberán ejecutar operaciones destructivas sin respaldo.

12.14 Crecimiento de la Información

El volumen de información aumentará progresivamente.

Especialmente:

- Auditoría.
- Pagos.
- Recibos.
- Asistencia.
- Entregas.
- Archivos.
- Logs.
- Notificaciones.

Se deberá monitorear el crecimiento para evitar que afecte el rendimiento o el almacenamiento disponible.

12.15 Gestión de Archivos

Los archivos académicos podrán representar una parte importante del almacenamiento.

Se deberá monitorear:

- Espacio ocupado.
- Archivos duplicados.
- Tamaño máximo.
- Extensiones.
- Integridad.
- Respallos.

Podrá implementarse posteriormente almacenamiento externo si el volumen crece considerablemente.

12.16 Política de Respallos

Los respaldos deberán mantenerse durante toda la operación.

Como estrategia inicial se podrá considerar:

Diario

Respaldo de MariaDB.

Semanal

Respaldo completo de base de datos y archivos.

Mensual

Copia histórica.

La política deberá adaptarse conforme aumente la información.

12.17 Regla 3-2-1 como Referencia

Cuando la infraestructura lo permita, podrá adoptarse como referencia una estrategia de respaldo que mantenga múltiples copias de la información, utilizando diferentes ubicaciones o medios.

El objetivo será evitar que un fallo en el servidor principal implique también la pérdida de los respaldos.

12.18 Pruebas de Recuperación

Se deberán ejecutar pruebas periódicas de restauración.

Una prueba podrá incluir:

1. Crear ambiente temporal.
2. Restaurar MariaDB.
3. Restaurar archivos.
4. Iniciar backend.
5. Consultar estudiantes.
6. Consultar pagos.
7. Consultar recibos.
8. Verificar integridad.

Los resultados deberán documentarse.

12.19 Continuidad Operativa

La continuidad operativa tiene como objetivo mantener los procesos esenciales de la institución incluso cuando ocurra un incidente tecnológico.

Se consideran procesos especialmente críticos:

- Inscripciones.
- Pagos.
- Emisión de recibos.

- Consulta de estudiantes.
- Control financiero.

La institución deberá disponer de procedimientos alternativos para situaciones en las que temporalmente SIGA no pueda utilizarse.

12.20 Escenario de Caída de Internet

Si una sucursal pierde temporalmente la conexión:

- No deberán realizarse operaciones duplicadas en diferentes registros sin control.
- Se podrá utilizar un procedimiento manual temporal autorizado.
- Una vez recuperada la conexión, la información deberá registrarse siguiendo un proceso de regularización.

El procedimiento definitivo deberá ser definido por la institución.

12.21 Escenario de Fallo del Servidor

Ante una indisponibilidad del servidor deberán realizarse acciones como:

1. Identificar causa.
2. Revisar infraestructura.
3. Verificar base de datos.
4. Restaurar servicio.
5. Restaurar respaldo si fuera necesario.
6. Validar operaciones críticas.

12.22 Escenario de Corrupción de Datos

En caso de detectarse corrupción:

1. Suspender operaciones críticas.
2. Determinar alcance.
3. Proteger evidencia.
4. Seleccionar respaldo válido.
5. Restaurar en ambiente controlado.
6. Verificar información.
7. Rehabilitar producción.

12.23 Escenario de Fallo del Proveedor QR

Cuando el proveedor QR se encuentre indisponible, SIGA deberá permitir utilizar otros métodos de pago habilitados.

La indisponibilidad del proveedor QR no deberá impedir el registro de pagos en efectivo cuando la institución lo permita.

12.24 Escenario de Fallo de Impresora

Cuando falle la impresora:

- El pago confirmado deberá conservarse.
- El recibo deberá permanecer registrado.
- No deberá registrarse nuevamente la operación.
- El comprobante podrá reimprimirse una vez solucionado el problema.

12.25 Objetivos de Recuperación

En futuras etapas podrán definirse formalmente métricas como:

RTO – Recovery Time Objective

Tiempo máximo objetivo para restablecer un servicio.

RPO – Recovery Point Objective

Cantidad máxima tolerable de información que podría perderse medida en tiempo.

Estas métricas deberán definirse conjuntamente con la institución según el nivel de criticidad y los costos de infraestructura disponibles.

12.26 Monitoreo

Durante la operación se deberán supervisar:

- Estado del servidor.
- Uso de CPU.
- Uso de RAM.
- Espacio en disco.
- Estado de MariaDB.
- Errores de aplicación.
- Certificado HTTPS.
- Respallos.
- Servicios externos.

Inicialmente el monitoreo podrá ser básico y ampliarse conforme crezca la plataforma.

12.27 Logs

Los registros técnicos deberán conservar información suficiente para investigar fallos.

Entre ellos:

- Logs de Django.
- Logs de Nginx.
- Logs del sistema operativo.
- Errores QR.
- Tareas automáticas.
- Respaldos.

Los logs no deberán registrar innecesariamente contraseñas, tokens o datos sensibles completos.

12.28 Gestión de Capacidad

Se deberá monitorear periódicamente si la infraestructura continúa siendo suficiente.

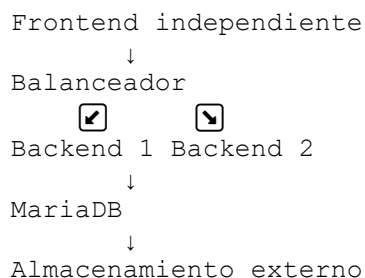
Indicadores:

- Número de usuarios.
- Conexiones simultáneas.
- Tamaño de base de datos.
- Espacio de archivos.
- Tiempo de respuesta.
- Uso de memoria.
- Uso de CPU.

Cuando se detecte crecimiento significativo podrá aumentarse la capacidad del servidor o separar componentes.

12.29 Escalamiento de Infraestructura

Si fuera necesario, la arquitectura podrá evolucionar hacia:



Esta complejidad únicamente deberá incorporarse cuando exista una necesidad real.

12.30 Mantenimiento de Documentación

La documentación deberá actualizarse cada vez que existan cambios importantes.

Esto incluye:

- Arquitectura.
- Base de datos.
- API.
- Manual técnico.
- Manuales de usuario.
- Requerimientos.
- Reglas de negocio.
- Cronograma cuando corresponda.

La documentación desactualizada no deberá utilizarse como única referencia del sistema.

12.31 Capacitación Continua

Cuando se incorporen nuevas funcionalidades importantes se deberá comunicar y, cuando corresponda, capacitar a los usuarios afectados.

Por ejemplo:

- Nuevo proceso de caja.
- Nuevo módulo QR.
- Nuevo sistema de calificaciones.
- Nuevos reportes.

12.32 Solicitudes de Mejora

Las nuevas solicitudes deberán registrarse y analizarse antes de ser incorporadas.

Cada solicitud podrá indicar:

- Descripción.
- Solicitante.
- Beneficio.
- Prioridad.
- Impacto.
- Horas estimadas.
- Dependencias.

Posteriormente podrá formar parte de una nueva versión.

12.33 Roadmap de Evolución

Una posible evolución será:

SIGA 1.0
MVP administrativo

SIGA 1.1

Finanzas

SIGA 1.2

Portal docente

SIGA 1.3

Portal estudiante

SIGA 1.4

QR y automatizaciones

SIGA 2.0

Nuevas funcionalidades estratégicas

La secuencia definitiva dependerá de las prioridades institucionales.

12.34 Propiedad y Custodia de la Información

La información operativa generada por el sistema deberá considerarse información institucional.

La institución deberá definir responsables de:

- Accesos.
- Respaldos.
- Solicitudes de información.
- Conservación.
- Eliminación cuando corresponda.

El desarrollador deberá contar únicamente con los accesos necesarios para ejecutar sus responsabilidades técnicas.

12.35 Gestión de Accesos Técnicos

Los accesos administrativos al servidor, base de datos y servicios externos deberán controlarse.

Cuando una persona deje de requerir acceso, sus credenciales deberán revocarse.

Se deberán evitar cuentas técnicas compartidas cuando exista una alternativa razonable.

12.36 Revisión Periódica de Usuarios

La institución deberá revisar periódicamente las cuentas registradas para identificar:

- Personal retirado.
- Docentes inactivos.

- Estudiantes finalizados.
- Cuentas sin uso.
- Roles incorrectos.

Las cuentas innecesarias deberán desactivarse sin eliminar su historial.

12.37 Continuidad ante Ausencia del Desarrollador

La documentación técnica deberá permitir que otro profesional pueda comprender y mantener el sistema en caso de ser necesario.

Para ello deberán mantenerse disponibles:

- Repositorio.
- Documentación.
- Procedimientos de instalación.
- Esquema de base de datos.
- Variables requeridas, sin divulgar secretos en documentación pública.
- Procedimientos de respaldo.
- Procedimiento de despliegue.

Este punto contribuirá a reducir la dependencia de una sola persona.

12.38 Indicadores de Mantenimiento

Se podrán utilizar indicadores como:

- Número de incidencias mensuales.
- Incidencias críticas.
- Tiempo medio de resolución.
- Disponibilidad.
- Restauraciones exitosas.
- Crecimiento del almacenamiento.
- Errores por módulo.
- Cantidad de versiones publicadas.

12.39 Revisión Periódica del Sistema

Se recomienda realizar revisiones periódicas para determinar:

- Estado técnico.
- Seguridad.
- Rendimiento.
- Necesidades de usuarios.
- Dependencias desactualizadas.
- Crecimiento de datos.

- Necesidades de infraestructura.

Los resultados podrán incorporarse al backlog de mantenimiento.

12.40 Cierre del Capítulo

El mantenimiento y la continuidad operativa constituyen componentes fundamentales para garantizar la sostenibilidad del Sistema Integral de Gestión Académica y Administrativa después de su implementación.

La aplicación de procesos de mantenimiento correctivo, adaptativo, perfectivo y preventivo permitirá mantener la estabilidad y seguridad del sistema durante su evolución.

Asimismo, la gestión organizada de incidencias, actualizaciones, respaldos, restauraciones, documentación y monitoreo contribuirá a disminuir los riesgos asociados con fallos tecnológicos o pérdida de información.

La estrategia de continuidad operativa permitirá establecer procedimientos para responder ante situaciones como indisponibilidad del servidor, fallos de conectividad, problemas de impresión o interrupciones en servicios externos.

Finalmente, la documentación técnica, el control de versiones y la planificación de mejoras deberán mantenerse durante toda la vida útil del sistema, permitiendo que SIGA evolucione de manera ordenada conforme cambien las necesidades de la institución.

CAPÍTULO 13 CONCLUSIONES Y RECOMENDACIONES

13.1 Introducción

El presente capítulo expone las conclusiones obtenidas durante el análisis, planificación y diseño del Sistema Integral de Gestión Académica y Administrativa (SIGA), así como las principales recomendaciones que deberán considerarse durante las etapas posteriores de implementación, pruebas, despliegue y mantenimiento.

La elaboración del presente documento permitió transformar las necesidades inicialmente identificadas en una propuesta estructurada de software, estableciendo requerimientos, reglas de negocio, arquitectura, modelo de datos, casos de uso, planificación, mecanismos de prueba y procedimientos de implementación.

13.2 Conclusiones Generales

El análisis realizado permite concluir que la institución requiere una plataforma integral que permita centralizar procesos que actualmente se encuentran distribuidos entre procedimientos manuales y herramientas con capacidades limitadas.

SIGA se plantea como una solución orientada a integrar los procesos académicos, administrativos y financieros dentro de una única plataforma.

La implementación permitirá mejorar principalmente:

- Registro de estudiantes.
- Inscripciones.
- Control de pagos.
- Emisión de recibos.
- Administración de cursos.
- Gestión docente.
- Seguimiento académico.
- Control financiero.
- Reportes.
- Disponibilidad de información.

13.3 Centralización de Información

Una de las principales ventajas del sistema será centralizar la información generada en las diferentes sucursales.

Esta centralización permitirá reducir registros duplicados, facilitar consultas y generar reportes consolidados.

La dirección podrá disponer de información de las diferentes sedes mediante una misma plataforma, reduciendo la necesidad de recopilar información manualmente.

13.4 Mejora del Proceso de Inscripción

La inscripción constituye uno de los procesos prioritarios identificados.

SIGA permitirá integrar en un mismo flujo:

```
Estudiante
  ↓
Curso
  ↓
Horario
  ↓
Inscripción
  ↓
Pago
  ↓
Recibo
```

Esto permitirá reducir pasos manuales y mantener un historial completo de cada estudiante.

13.5 Mejora del Control Financiero

La incorporación de pagos, cuotas, recibos, ingresos, egresos, caja y salarios permitirá mejorar el seguimiento financiero de la institución.

Las operaciones podrán mantener trazabilidad, evitando que anulaciones o correcciones impliquen eliminar información histórica.

Los reportes proporcionarán mayor visibilidad sobre:

- Ingresos.
- Egresos.
- Deudas.
- Caja.
- Sucursales.
- Pagos docentes.

13.6 Beneficios Académicos

La plataforma permitirá incorporar procesos académicos que actualmente no se encuentran sistematizados completamente.

Entre ellos:

- Asistencia.
- Materiales.
- Prácticas.
- Entregas.
- Calificaciones.

Los docentes podrán disponer de herramientas para administrar sus cursos y los estudiantes podrán consultar información desde su propio portal.

13.7 Importancia de la Arquitectura Seleccionada

La selección de React, TypeScript, Python, Django REST Framework y MariaDB proporciona una arquitectura adecuada para desarrollar el sistema de forma modular.

La separación entre frontend y backend permitirá mantener una API independiente que podrá ser reutilizada en futuras aplicaciones.

Asimismo, Django proporcionará herramientas relacionadas con seguridad, autenticación, ORM, migraciones y administración que permitirán acelerar el desarrollo.

13.8 Importancia del Modelo de Datos

El diseño de la base de datos representa uno de los componentes centrales del proyecto.

La correcta relación entre estudiantes, inscripciones, pagos, recibos, cursos y demás entidades permitirá garantizar consistencia y trazabilidad.

Las decisiones relacionadas con pagos, anulaciones y movimientos financieros deberán implementarse cuidadosamente debido al impacto que pueden tener sobre la información institucional.

13.9 Importancia de la Seguridad

Debido al tipo de información administrada, la seguridad deberá ser considerada durante todo el desarrollo.

No deberá limitarse únicamente al inicio de sesión.

Será necesario aplicar mecanismos relacionados con:

- Autenticación.
- Roles.
- Permisos.
- Validación.
- HTTPS.
- Auditoría.
- Respaldos.
- Protección de archivos.
- Control de accesos.

13.10 Importancia de la Auditoría

Las operaciones críticas deberán ser trazables.

Especialmente:

- Pagos.
- Anulaciones.
- Recibos.
- Cambios financieros.
- Calificaciones.
- Usuarios.
- Permisos.

La auditoría permitirá disponer de evidencia sobre quién realizó cada operación y cuándo ocurrió.

13.11 Importancia del Desarrollo Incremental

Debido al tamaño del sistema, no se recomienda intentar implementar todos los módulos simultáneamente.

La estrategia propuesta consiste en comenzar con un MVP compuesto principalmente por:

- Usuarios.
- Estudiantes.
- Estructura académica básica.
- Inscripciones.
- Pagos.
- Recibos.
- Impresión térmica.

Posteriormente podrán incorporarse progresivamente las demás funcionalidades.

13.12 Conclusión sobre el Cronograma

La planificación desarrollada considera una jornada de trabajo parcial y un desarrollo individual.

Por ello, el sistema completo requiere una planificación de largo plazo.

Sin embargo, la división por versiones permitirá disponer de funcionalidades utilizables antes de completar la totalidad del alcance.

El cronograma deberá actualizarse conforme exista información real sobre la velocidad de desarrollo y las dependencias externas.

13.13 Conclusión sobre Pagos QR

La integración QR constituye una dependencia externa importante.

Su implementación definitiva dependerá de:

- Proveedor.
- API disponible.
- Credenciales.
- Documentación.
- Mecanismos de confirmación.
- Condiciones comerciales.

Por esta razón, el diseño deberá mantener esta integración desacoplada del módulo financiero principal.

13.14 Conclusión sobre Impresión Térmica

La impresión térmica deberá validarse utilizando el hardware real.

El diseño funcional podrá desarrollarse anticipadamente, pero la aceptación definitiva requerirá comprobar:

- Modelo.
- Driver.
- Conexión.
- Papel.
- Formato.
- Compatibilidad.

13.15 Recomendación – Validar Reglas Pendientes

Antes de implementar los procesos financieros críticos deberá validarse con la dirección:

- Política de descuentos.
- Número de cuotas.
- Vencimientos.
- Devoluciones.
- Anulación de pagos.
- Anulación de recibos.
- Cambios de curso.
- Excedentes.
- Salarios.
- Cierres de caja.

Estas reglas no deberán quedar implícitas.

13.16 Recomendación – Priorizar el MVP

Se recomienda mantener el MVP como primera prioridad.

El objetivo inicial deberá ser conseguir un flujo completamente funcional:

```
Registrar estudiante
    ↓
Inscribir
    ↓
Registrar pago
    ↓
Generar recibo
    ↓
Imprimir comprobante
```

Una vez estable, se podrá ampliar hacia otros módulos.

13.17 Recomendación – Validaciones Frecuentes

Se recomienda mantener reuniones periódicas de validación con los responsables institucionales.

No deberá esperarse hasta terminar un módulo grande para mostrarlo.

Las demostraciones tempranas permitirán detectar diferencias antes de invertir más tiempo.

13.18 Recomendación – Controlar Cambios de Alcance

Toda nueva funcionalidad deberá ser analizada antes de incorporarse.

La solicitud deberá evaluar:

- Necesidad.
- Prioridad.
- Impacto técnico.
- Tiempo.
- Dependencias.
- Módulos afectados.

Esto permitirá mantener control sobre el cronograma.

13.19 Recomendación – No Medir Avance Solo por Código

El avance deberá considerar:

- Análisis.
- Diseño.
- Backend.
- Frontend.
- Integración.
- Pruebas.
- Validación.
- Documentación.

Un módulo no deberá considerarse terminado únicamente porque exista una pantalla funcional.

13.20 Recomendación – Mantener Git desde el Inicio

Todo el código deberá mantenerse bajo control de versiones desde las primeras etapas.

Cada funcionalidad deberá generar commits identificables.

Esto permitirá:

- Recuperar cambios.
- Analizar errores.
- Mantener historial.
- Facilitar despliegues.

13.21 Recomendación – Mantener Ambientes Separados

Se recomienda diferenciar:

- Desarrollo.
- Pruebas.
- Producción.

Los cambios importantes deberán probarse antes de afectar información real.

13.22 Recomendación – Proteger Información Financiera

Las operaciones financieras deberán implementarse utilizando:

- Transacciones.
- Validaciones.
- Auditoría.
- Permisos específicos.
- Eliminación lógica.
- Reversión controlada.

No deberán modificarse pagos confirmados directamente sin un procedimiento definido.

13.23 Recomendación – Probar Respaldos

No será suficiente automatizar copias.

Periódicamente se deberá comprobar que los datos pueden restaurarse correctamente.

13.24 Recomendación – Mantener Documentación Técnica

La documentación deberá actualizarse conforme evolucione SIGA.

Especialmente:

- Arquitectura.
- Modelo de datos.
- API.
- Instalación.
- Despliegue.
- Respaldos.

- Reglas de negocio.

Esto reducirá la dependencia de conocimiento no documentado.

13.25 Recomendación – Capacitar a Usuarios

La introducción de una nueva herramienta modifica procesos de trabajo.

Por ello se recomienda capacitar principalmente a:

- Administración.
- Caja.
- Dirección.
- Docentes.

La capacitación deberá realizarse antes de habilitar cada módulo.

13.26 Recomendación – Implementación Piloto

Se recomienda utilizar inicialmente el sistema con un grupo controlado antes de habilitarlo para toda la institución.

El piloto permitirá detectar:

- Dificultades de uso.
- Problemas de impresión.
- Reglas no contempladas.
- Errores de flujo.
- Problemas de rendimiento.

13.27 Recomendación – Mantener Auditoría desde la Primera Versión

La auditoría no deberá dejarse para una etapa final.

Desde el MVP deberán registrarse al menos las operaciones críticas.

Especialmente:

- Inscripciones.
- Pagos.
- Recibos.
- Anulaciones.
- Usuarios.

13.28 Recomendación – Evitar Complejidad Prematura

Aunque la arquitectura está preparada para crecer, no se recomienda implementar infraestructura compleja antes de necesitarla.

Inicialmente un único servidor correctamente configurado puede ser suficiente.

El escalamiento deberá realizarse cuando exista evidencia de crecimiento real.

13.29 Recomendación – Evaluar Métricas después del MVP

Una vez utilizado SIGA durante un periodo real se recomienda analizar indicadores como:

- Número de usuarios.
- Cantidad de operaciones.
- Tiempo de respuesta.
- Incidencias.
- Cantidad de pagos.
- Volumen de archivos.
- Uso del servidor.

Esto permitirá ajustar las estimaciones técnicas futuras.

13.30 Recomendación – Revisar el SRS Periódicamente

El presente documento deberá considerarse una línea base y no un documento inmutable.

Cuando existan modificaciones aprobadas deberá actualizarse:

- Versión.
- Fecha.
- Sección modificada.
- Responsable.
- Motivo.

Esto permitirá mantener correspondencia entre documentación y sistema real.

13.31 Resultado Esperado del Proyecto

Una vez completado, SIGA deberá permitir que la institución disponga de una plataforma centralizada capaz de administrar los principales procesos académicos, administrativos y financieros.

El resultado esperado será una reducción de procesos manuales y una mejora en:

- Organización.
- Control.
- Disponibilidad de información.

- Seguimiento académico.
- Control financiero.
- Trazabilidad.
- Generación de reportes.
- Atención a estudiantes.

13.32 Visión de Evolución

El sistema deberá poder evolucionar posteriormente hacia funcionalidades adicionales.

Entre ellas podrán considerarse:

- Aplicación móvil.
- Comunicación automatizada.
- WhatsApp.
- Facturación electrónica.
- Nuevos métodos de pago.
- Analítica avanzada.
- Business Intelligence.
- Integraciones externas.

Estas ampliaciones deberán evaluarse como nuevos proyectos o versiones.

13.33 Conclusión Final

La propuesta del Sistema Integral de Gestión Académica y Administrativa establece una solución integral para centralizar y automatizar los procesos institucionales identificados durante el análisis.

La documentación desarrollada proporciona una base para orientar el proyecto desde la definición de requerimientos hasta su implementación y mantenimiento, incluyendo aspectos funcionales, técnicos, financieros, académicos y operativos.

El éxito del proyecto dependerá no solamente de la implementación tecnológica, sino también de la validación continua con los responsables institucionales, la definición clara de reglas de negocio, la participación de los usuarios, el control de cambios y el desarrollo incremental.

La priorización de un Producto Mínimo Viable permitirá obtener resultados utilizables en etapas tempranas, reduciendo riesgos y permitiendo que el sistema evolucione a partir de experiencia real.

Finalmente, la aplicación de buenas prácticas relacionadas con arquitectura, seguridad, trazabilidad, pruebas, respaldos, documentación y mantenimiento permitirá que SIGA se convierta en una plataforma sostenible y capaz de acompañar el crecimiento futuro de la institución.

