

Especificación de Requisitos de Software

Proyecto: Sistema de Gestión de Estudiantes Universitarios

Revisión: 0.1

Fecha: 14 de octubre de 2025

Autor: Mayra Moyano

FICHA DEL DOCUMENTO

Fecha	Revisión	Autor	Verificado dep. calidad.
14/10/2025	0.1	Mayra Moyano	Pendiente

CONTENIDO

1. INTRODUCCIÓN

- 1.1 Propósito
- 1.2 Alcance
- 1.3 Personal involucrado
- 1.4 Definiciones, acrónimos y abreviaturas
- 1.5 Referencias
- 1.6 Resumen

2. DESCRIPCIÓN GENERAL

- 2.1 Perspectiva del producto
- 2.2 Funcionalidad del producto
- 2.3 Características de los usuarios
- 2.4 Restricciones
- 2.5 Suposiciones y dependencias
- 2.6 Evolución previsible del sistema

3. REQUISITOS ESPECÍFICOS

- 3.1 Requisitos comunes de los interfaces
 - 3.1.1 Interfaces de usuario
 - 3.1.2 Interfaces de hardware
 - 3.1.3 Interfaces de software

- 3.1.4 Interfaces de comunicación
- 3.2 Requisitos funcionales
 - 3.2.1 Gestión de Estudiantes
 - 3.2.2 Gestión de Carreras
 - 3.2.3 Gestión de Materias
 - 3.2.4 Gestión de Inscripciones
 - 3.2.5 Gestión de Calificaciones
 - 3.2.6 Gestión de Condiciones Académicas
- 3.3 Requisitos no funcionales
 - 3.3.1 Requisitos de rendimiento
 - 3.3.2 Seguridad
 - 3.3.3 Fiabilidad
 - 3.3.4 Disponibilidad
 - 3.3.5 Mantenibilidad
 - 3.3.6 Portabilidad
- 3.4 Otros requisitos

4. APÉNDICES

1. INTRODUCCIÓN

1.1 Propósito

El propósito de este documento es definir los requisitos funcionales y no funcionales del **Sistema de Gestión de Estudiantes Universitarios**, una aplicación web diseñada para centralizar y automatizar la administración de datos académicos y personales de los estudiantes de la institución.

Este documento está dirigido a:

- El equipo de desarrollo.
- Los analistas de sistemas.
- Los responsables de pruebas y calidad.
- Los stakeholders y usuarios finales (administradores, docentes, personal administrativo).

1.2 Alcance

El sistema cubrirá las siguientes áreas clave:

- Registro y mantenimiento de información personal y académica de los estudiantes.
- Gestión de carreras y materias ofrecidas por la universidad.

- Control de inscripciones y matrículas.
- Registro de calificaciones y condiciones académicas de los estudiantes.
- Generación de reportes básicos.

1.3 Personal involucrado

Nombre	Rol	Categoría profesional	Responsabilidades	Información de contacto	Aprobación
Mayra Moyano	Analista / Desarrollador	Ingeniería de Software	Diseño, desarrollo y documentación del sistema.	mayrayazmi.nmoyano@gmail.com	<input type="checkbox"/>

1.4 Definiciones, acrónimos y abreviaturas

- **SRS:** Especificación de Requisitos de Software.
- **DER:** Diagrama Entidad-Relación.
- **PK:** Primary Key (Clave Primaria).
- **FK:** Foreign Key (Clave Foránea).
- **UI:** Interfaz de Usuario.
- **API:** Application Programming Interface.
- **CRUD:** Create, Read, Update, Delete (Crear, Leer, Actualizar, Eliminar).

1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
IEEE Std 830-1998	IEEE Recommended Practice for Software Requirements Specifications	https://standards.ieee.org/standard/830-1998.html	1998	IEEE
Plantilla IEEE 830	Plantilla de documento SRS	plantilla_formato_ieee830.docx	2025	Coloriuris

1.6 Resumen

Este documento se organiza en tres secciones principales: Introducción, Descripción General y Requisitos Específicos. La sección de Requisitos Específicos detalla todas las funcionalidades requeridas, organizadas por módulos, junto con sus requisitos no funcionales. El apéndice contiene el modelo de datos (DER) que soporta la implementación.

2. DESCRIPCIÓN GENERAL

2.1 Perspectiva del producto

El sistema es un producto independiente, aunque está diseñado para ser escalable y potencialmente integrable con otros sistemas de la universidad (como SIRH o sistemas de biblioteca). Su núcleo es una base de datos relacional que almacena toda la información académica y personal de los estudiantes.

2.2 Funcionalidad del producto

El sistema permitirá a los usuarios autorizados:

- Registrar, consultar, actualizar y eliminar datos de estudiantes.
- Gestionar las carreras y materias disponibles.
- Registrar inscripciones de estudiantes a carreras y materias.
- Asignar y registrar calificaciones finales.
- Definir y asignar condiciones académicas a los estudiantes por materia.
- Generar reportes básicos sobre estado de estudiantes, inscripciones y calificaciones.

2.3 Características de los usuarios

Tipo de usuario	Formación	Habilidades	Actividades
Administrador	Nivel universitario	Manejo básico de computadoras y sistemas web	Gestionar usuarios, configurar catálogos (países, géneros, condiciones), generar reportes.
Docente	Nivel universitario	Manejo básico de computadoras y sistemas web	Consultar estudiantes, asignar calificaciones, ver

Tipo de usuario	Formación	Habilidades	Actividades
			condiciones académicas.
Personal Administrativo	Secundaria completa	Manejo básico de computadoras y sistemas web	Registrar estudiantes, gestionar inscripciones, actualizar datos personales.

2.4 Restricciones

- El sistema debe desarrollarse utilizando tecnologías web modernas (Node.js, React, PostgreSQL).
- Debe ser compatible con los navegadores Chrome, Firefox y Edge.
- La base de datos debe ser PostgreSQL.
- El diseño de la interfaz debe seguir un patrón de diseño consistente (usando Tailwind CSS o similar).
- No se permitirá el uso de lenguajes de programación obsoletos o sin soporte.

2.5 Suposiciones y dependencias

- Se asume que el entorno de producción contará con acceso a Internet y un servidor con capacidad suficiente para ejecutar Node.js y PostgreSQL.
- Se asume que el personal administrativo recibirá capacitación básica para el uso del sistema.
- El sistema depende de la existencia de un servicio de autenticación y autorización (JWT).
- Se asume que los datos de países, localidades, áreas telefónicas y géneros estarán previamente cargados en la base de datos.

2.6 Evolución previsible del sistema

En futuras versiones, se planea:

- Dashboard con estadísticas avanzadas y gráficos interactivos.
 - Soporte para múltiples idiomas.
-

3. REQUISITOS ESPECÍFICOS

3.1 Requisitos comunes de los interfaces

3.1.1 Interfaces de usuario

- Todas las pantallas deben tener un diseño limpio, intuitivo y accesible.
- El sistema debe ser responsive y funcionar correctamente en dispositivos móviles, tablets y desktops.
- Las acciones críticas (eliminar, guardar cambios) deben tener confirmaciones visuales.
- Los formularios deben validar los campos antes de enviarlos al servidor.
- Los mensajes de error deben ser claros y descriptivos.

3.1.2 Interfaces de hardware

- El sistema no requiere hardware especializado. Puede ejecutarse en cualquier dispositivo con navegador web moderno.
- El servidor de backend debe contar con al menos 2 GB de RAM y 1 CPU.

3.1.3 Interfaces de software

- El frontend interactúa con el backend mediante una API RESTful.
- El backend utilizará Prisma como ORM para interactuar con PostgreSQL.
- El sistema no tendrá dependencias con software propietario.

3.1.4 Interfaces de comunicación

- El sistema utilizará el protocolo HTTP/HTTPS para todas las comunicaciones.
 - Las solicitudes desde el frontend y backend serán en formato JSON.
 - El sistema no requiere comunicación con otros sistemas externos en esta versión.
-

3.2 Requisitos funcionales

3.2.1 Gestión de Estudiantes

RF 3.2.1.1: Registro de Estudiante

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir registrar un nuevo estudiante ingresando su nombre, apellido, DNI, email, teléfono, domicilio, fecha de nacimiento, género, país,

localidad, área telefónica, cohorte, secundario, CUIL, y campos booleanos para examen mayores 25, solicitud de beca, trabajador y persona a cargo. El estado inicial del estudiante será "Activo".

RF 3.2.1.2: Consulta de Estudiante

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir buscar y visualizar los datos de un estudiante por DNI, nombre, apellido o ID. Debe mostrar toda la información registrada, incluyendo su historial de inscripciones y calificaciones.

RF 3.2.1.3: Actualización de Estudiante

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Media
- **Descripción:** El sistema debe permitir modificar cualquier dato del estudiante, excepto su ID. Debe registrar la fecha de última modificación.

RF 3.2.1.4: Eliminación de Estudiante

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Baja
- **Descripción:** El sistema debe permitir marcar a un estudiante como "Inactivo" en lugar de eliminarlo físicamente. Debe registrar la causa de inactividad.

3.2.2 Gestión de Carreras

RF 3.2.2.1: Registro de Carrera

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir registrar una nueva carrera con su nombre, duración (en años) y título otorgado.

RF 3.2.2.2: Consulta de Carrera

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir listar todas las carreras y ver los detalles de una carrera específica.

RF 3.2.2.3: Actualización de Carrera

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Media
- **Descripción:** El sistema debe permitir modificar el nombre, duración o título de una carrera.

3.2.3 Gestión de Materias

RF 3.2.3.1: Registro de Materia

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir registrar una nueva materia con su nombre y asociarla a una carrera específica.

RF 3.2.3.2: Consulta de Materia

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir listar todas las materias y ver los detalles de una materia específica, incluyendo la carrera a la que pertenece.

RF 3.2.3.3: Actualización de Materia

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Media
- **Descripción:** El sistema debe permitir modificar el nombre de una materia o cambiar la carrera a la que pertenece.

3.2.4 Gestión de Inscripciones

RF 3.2.4.1: Inscripción a Carrera

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir inscribir a un estudiante en una carrera. Debe registrar la fecha de inscripción y el estado de la inscripción (Activo/Inactivo).

RF 3.2.4.2: Consulta de Inscripciones

- **Tipo:** Requisito

- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir ver todas las inscripciones de un estudiante, incluyendo la carrera, la fecha y el estado.

RF 3.2.4.3: Cancelación de Inscripción

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Media
- **Descripción:** El sistema debe permitir cancelar una inscripción, marcándola como "Inactiva" y registrando la causa de inactividad.

3.2.5 Gestión de Calificaciones

RF 3.2.5.1: Registro de Calificación

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir registrar la nota final de un estudiante en una materia específica. La nota debe ser un número decimal.

RF 3.2.5.2: Consulta de Calificaciones

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir ver todas las calificaciones de un estudiante, mostrando la materia, la nota y la carrera asociada.

RF 3.2.5.3: Actualización de Calificación

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Media
- **Descripción:** El sistema debe permitir modificar la nota final de un estudiante en una materia.

3.2.6 Gestión de Condiciones Académicas

RF 3.2.6.1: Definición de Condiciones

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Media

- **Descripción:** El sistema debe permitir definir las condiciones académicas posibles (ej. "Regular", "Libre", "Aprobado", "Desaprobado") con un nombre descriptivo.

RF 3.2.6.2: Asignación de Condición

- **Tipo:** Requisito
- **Fuente:** Stakeholder
- **Prioridad:** Alta
- **Descripción:** El sistema debe permitir asignar una condición académica a un estudiante para una materia específica.

RF 3.2.6.3: Consulta de Condiciones

- **Tipo:** Requisito
 - **Fuente:** Stakeholder
 - **Prioridad:** Alta
 - **Descripción:** El sistema debe permitir ver las condiciones académicas de un estudiante por materia.
-

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

- El sistema debe responder a una consulta de búsqueda de estudiantes en menos de 2 segundos con una carga de hasta 1000 usuarios concurrentes.
- El sistema debe soportar al menos 100 transacciones de inserción/actualización por minuto.

3.3.2 Seguridad

- El sistema debe autenticar a todos los usuarios mediante un sistema de login con contraseña cifrada (bcrypt).
- El sistema debe autorizar el acceso a funcionalidades según el rol del usuario (Administrador, Docente, Personal).
- Todos los datos sensibles (DNI, email, teléfono) deben estar cifrados en la base de datos.
- El sistema debe registrar en logs todas las operaciones de creación, actualización y eliminación de datos.

3.3.3 Fiabilidad

- El sistema debe tener una disponibilidad del 99.9% durante el horario laboral (8:00 a 22:00).
- En caso de fallo del servidor, el sistema debe recuperar su estado en menos de 15 minutos.

3.3.4 Disponibilidad

- El sistema debe estar disponible 24/7, excepto durante mantenimientos programados (que no excederán 2 horas por semana).

3.3.5 Mantenibilidad

- El código debe estar bien documentado con comentarios y seguir convenciones de nombramiento claras.
- El sistema debe estar modularizado para facilitar la actualización de componentes individuales.
- Debe haber un plan de respaldo diario de la base de datos.

3.3.6 Portabilidad

- El sistema debe ser capaz de ejecutarse en cualquier sistema operativo que soporte Node.js y PostgreSQL (Windows, Linux, macOS).
 - El código debe ser independiente del sistema operativo y del servidor web.
-

3.4 Otros requisitos

- **Requisitos Legales:** El sistema debe cumplir con la normativa de protección de datos personales vigente en el país de operación.
 - **Requisitos Culturales:** El sistema debe usar un lenguaje respetuoso y neutral, evitando términos que puedan ser considerados ofensivos o discriminatorios.
-

4. APÉNDICES

Apéndice A: Modelo de Datos (Diagrama Entidad-Relación)

El modelo de datos se compone de las siguientes entidades y relaciones:

