Especificación de Requisitos del Software (SRS)

**Fecha:** 27/08/2025

**Autor:**

**Danluis Enrique Romani Mora**

**Eduardo Chaparro Huaman**

**Fabrizcio Galileo Argandoña Montalvo**

**Versión:** 1.2

# 1. Introducción

## 1.1 Propósito

En este documento se define cuales son los requisitos para el desarrollo de nuestra aplicación web que está diseñada para la Identificación de Inteligencias múltiples en los estudiantes y el apoyo pedagógico a los docentes, así mismo se busca que exista una integración entre la aplicación y los usuarios que la utilicen.

## 1.2 Alcance

La aplicación estará disponible para diferentes navegadores, maximizando el número de dispositivos que puedan ejecutar esta aplicación, por otro lado, esta llegara a identificar los tipos de inteligencias que tienen los estudiantes.

## 1.3 Definiciones, Acrónimos y Abreviaturas

* **UI:** Interfaz de Usuario
* **API:** Interfaz de Programación de Aplicaciones
* **DBMS:** Sistema de Gestor de Bases de Datos
* **IA:** Inteligencia Artificial.

## 1.4 Referencias

* IEEE 830-1998 Standard for Software Requirements Specifications

## 1.5 Descripción General

El objetivo de este proyecto se basa en diseñar, desarrollar e implementar un software basado en la teoría de las inteligencias múltiples, con el cual se pueda personalizar las experiencias educativas, optimizar la toma de decisiones pedagógicas y fomentar una educación más adaptativa y equitativa con el apoyo de inteligencia artificial. Con esta los usuarios podrán dar evaluaciones así como monitorear el rendimiento académico en tiempo real.

# 2. Descripción General del Producto

## 2.1 Perspectiva del Producto

La aplicación web, basada en la arquitectura de tres capas utilizando tecnologías como Spring Boot para las interfaces y java en el desarrollo de la lógica interna de la aplicación, para la base de datos se utilizará MySQL debido a la versatilidad y ligereza de esta base de datos.

## 2.2 Funcionalidades Principales

* Ingreso al software mediante un inicio de sesión .
* Identificación del tipo de inteligencia predominante del alumno por medio de test
* Integración de un chatbot con inteligencia artificial.
* Visualización de datos del alumno en un dashboard.
* Listado de alumnos y acciones CRUD
* Recomendaciones pedagógicas automáticas.
* Panel de Docente Interactivo

## 2.3 Características de los Usuarios

* **Usuarios principales:** Personas que desean mejorar su organización personal.
* **Beneficios esperados:** Mayor productividad, mejor seguimiento de tareas y recordatorios automáticos.

## 2.4 Restricciones

* Compatible con navegadores modernos (Chrome, Firefox, Edge).
* La aplicación será de uso individual en la primera versión.

## 2.5 Suposiciones y Dependencias

* Se asumirá que los usuarios tienen acceso a internet.
* La aplicación dependerá de servicios en la nube para almacenamiento y autenticación.

# 3. Requisitos Específicos

## 3.1 Requisitos Funcionales

1. Ingreso al software mediante un inicio de sesión .
   1. RF1.1: El sistema debe permitir el inicio de sesión para el personal docente.
   2. RF1.2: El sistema debe permitir el inicio de sesión para los alumnos.
   3. RF1.3 El sistema debe permitir el inicio de sesión para un usuario con rol de administrador.
   4. RF1.4 El sistema debe permitir un cierre de sesión seguro.
2. Identificación del tipo de inteligencia predominante del alumno por medio de test:
   1. RF2.1: El sistema debe mostrar un test basado en la teoría de inteligencias múltiples de Howard Gardner.
   2. RF2.2: El sistema debe permitir que el alumno responda todas las preguntas del test.
   3. RF2.3: El sistema debe calcular automáticamente los puntajes de cada tipo de inteligencia.
   4. RF2.4: El sistema debe identificar y mostrar cuál es la inteligencia predominante del alumno.
   5. RF2.5: El sistema debe almacenar los resultados en la base de datos asociados al alumno.
3. Integración de un chatbot con inteligencia artificial.
   1. RF3.1:El sistema debe permitir que alumnos y docentes envíen consultas al chatbot a través de una interfaz web .
   2. RF3.2: El sistema debe enviar las consultas del usuario al servicio de IA (API DeepSeek) y mostrar las respuestas en tiempo real.
   3. RF3.3:El sistema debe permitir que el chatbot acceda a información relevante del perfil del usuario (rol, resultados de test, historial académico) para personalizar las respuestas.
   4. RF3.4: El sistema debe manejar escenarios de error (falla en API, tiempo de espera superado, conexión interrumpida) mostrando mensajes claros al usuario.
   5. RF3.5: El sistema debe almacenar las consultas realizadas por los docentes al chatbot y las respuestas entregadas, con fines de auditoría.
4. Visualización de datos del alumno en un dashboard.
   1. RF4.1:El sistema debe mostrar un dashboard para cada alumno con sus resultados de inteligencias múltiples, historial de pruebas y progreso académico.
   2. RF4.2: El dashboard debe incluir gráficos dinámicos e indicadores visuales
5. Listado de alumnos y acciones CRUD
   1. RF5.1: El sistema debe permitir al docente y al administrador visualizar un listado completo de alumnos registrados.
   2. RF5.2: El sistema debe permitir realizar operaciones CRUD sobre los alumnos (crear, leer, actualizar, eliminar).
   3. RF5.3: El sistema debe permitir buscar alumnos por nombre o código.
   4. RF5.4: Las operaciones de eliminación deben solicitar confirmación para evitar errores.
6. Recomendaciones pedagógicas automáticas.
   1. RF6.1:El sistema debe generar recomendaciones pedagógicas automáticas basadas en los resultados de las pruebas de inteligencias múltiples de cada alumno.
   2. RF6.2: El sistema debe almacenar las recomendaciones generadas para futuras consultas.

## 3.2 Requisitos No Funcionales

1. RNF1: La aplicación debe responder en menos de 10 segundos en operaciones normales.
2. RNF2: Se deberá seguir un diseño responsivo para compatibilidad con móviles.
3. RNF 3: El test no tardará en cargar los contenidos tanto para el alumno como para el docente.

## 3.3 Requisitos de Interfaz de Usuario

* La interfaz debe ser intuitiva y minimalista.
* Debe permitir la personalización(oe que es esta mamada) del esquema de colores.
* Debe

## 3.4 Requisitos de Hardware y Software

* Servidor con soporte Java JDK > 17
* MySql.
* Angular.
* Compatible con navegadores con Chromium (Chrome, Edge, Brave, Opera…).

# 4. Riesgos y Limitaciones

## 4.1 Riesgos

* Posible sobrecarga del servidor si la cantidad de usuarios crece rápidamente.
* Dependencia de servicios de terceros para autenticación y almacenamiento.
* Cambio de politicas de las APIs en uso

## 4.2 Limitaciones

* No incluirá funcionalidades de colaboración en tiempo real en la primera versión.
* No habrá integración con asistentes virtuales en esta etapa.

# 5. Alcance del Proyecto

## 5.1 Lo que incluirá

* Gestor de tareas con categorías y recordatorios.
* Aplicación web con interfaz responsiva.

## 5.2 Lo que NO incluirá (por ahora)

* Aplicaciones nativas para iOS y Android.
* Sincronización con otros servicios de productividad.

# Referencias

Define qué incluirá y qué quedará fuera en la primera versión.

# Control de Cambios

| **Nro.** | **Fecha** | **Autor(es)** | **Descripción** |
| --- | --- | --- | --- |
|  | 27/08/2025 | Danluis Enrique Romani Mora  Eduardo Chaparro Huaman  Fabrizcio Galileo Argandoña Montalvo | Se creó el documento y se sentaron las bases del proyecto que se va a desarrollar. |
|  | 31/08/2025 | Eduardo Chaparro Huaman | Se agregaron Requerimientos no funcionales respecto a la experiencia del usuario. |
| 3. | 05/09/2025 | Danluis Enrique Romani Mora  Eduardo Chaparro Huaman  Fabrizcio Galileo Argandoña Montalvo | Se corrigieron los Requerimientos funcionales para estar acorde a las funcionalidades. |
|  | 06/09/2025 | Danluis Enrique Romani Mora  Eduardo Chaparro Huaman  Fabrizcio Galileo Argandoña Montalvo | Se agregaron Requisitos funcionales y se revisaron los requisitos anteriores |