

# Especificación de Requisitos de Software

**Proyecto:** Sistema de Gestión de Aprendizaje (LMS)

**Versión:** 1.0

**Fecha:** Julio 2025

# 1. Introducción

## 1.1 Propósito

El propósito de este documento es definir los requisitos del sistema para el desarrollo de una **plataforma web educativa** estilo aula virtual, basada en los contenidos adquiridos durante la carrera de Desarrollo de Software. Esta plataforma estará dirigida principalmente a estudiantes de nivel inicial y medio que buscan reforzar o aprender desde cero los contenidos de materias clave como Fundamentos de Programación, Base de Datos, Arquitectura de Software, entre otras.

El documento será utilizado por la creadora del sitio (Mayra Moyano), colaboradores educativos y desarrolladores que se integren en futuras versiones para asegurar que el producto final cumpla con los objetivos de aprendizaje, funcionalidad y calidad esperados.

## 1.2 Alcance

El sistema será una plataforma web alojada públicamente, accesible desde cualquier navegador. Permitirá a los usuarios:

- Navegar por materias organizadas por año o tema.
- Acceder a contenido gratuito (videos, apuntes, ejercicios).
- Realizar pruebas y actividades interactivas.
- Registrarse y acceder a contenido exclusivo mediante suscripción o pago único.
- Descargar recursos como PDFs, plantillas, códigos fuente y ejercicios.

En futuras versiones se prevé la incorporación de:

- Seguimiento de progreso del usuario.
- Foros o comunidades de aprendizaje.
- Certificados de finalización.

## 1.3 Personal involucrado

Nombre	Rol	Responsabilidad	Contacto
Mayra Moyano	Desarrolladora / Creadora de contenido	Diseño, implementación y curaduría de contenidos	Mayra Moyano

## 1.4 Definiciones, acrónimos y abreviaturas

Término	Definición
HTML	Lenguaje de Marcado de Hipertexto

CSS	Hojas de Estilo en Cascada
LMS	Learning Management System (Sistema de Gestión de Aprendizaje)
MVP	Producto Mínimo Viable

## 1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
IEEE Std 830-1998	Recomendación para especificación de requisitos de software	<a href="#">w plantill...</a>	1998	IEEE
Apuntes carrera Desarrollo de Software	Recopilación personal	Archivos propios	2023-20 25	Mayra Moyano

## 1.6 Resumen

Este documento presenta una descripción detallada de la plataforma de aula virtual en desarrollo. En él se define el propósito, el alcance, las funciones previstas, las restricciones técnicas, así como los requisitos funcionales y no funcionales. Está basado en el estándar IEEE 830 y servirá como guía tanto para la planificación del sistema como para su futura evolución.

# 2. Descripción General

## 2.1 Perspectiva del producto

La plataforma de aula virtual es un producto independiente, desarrollado desde cero como iniciativa personal. Está destinada a ofrecer una experiencia de aprendizaje estructurada, con contenidos organizados por materias y niveles.

No forma parte de un sistema mayor, aunque a futuro podría integrarse con herramientas externas como sistemas de pago, foros comunitarios o plataformas de certificación. El sistema será accesible a través de navegadores web y estará optimizado para dispositivos móviles y de escritorio.

Se prevé usar tecnologías web como HTML, CSS, JavaScript para el frontend, y Django para el backend.

## 2.2 Funcionalidad del producto

Las principales funcionalidades previstas del sistema incluyen:

- Mostrar un **catálogo de cursos** organizados por año y temática.
- Permitir a los usuarios **visualizar contenidos gratuitos** (videos, textos, ejercicios, imágenes, enlaces).
- Ofrecer **descargas** de recursos como PDFs, plantillas o guías prácticas.
- Implementar un sistema de **registro e inicio de sesión**, opcional para acceder a contenido exclusivo.
- Incorporar una sección de **cursos pagos o membresías** para monetizar la plataforma.
- Incluir un **buscador interno** para facilitar el acceso a contenidos específicos.
- En versiones futuras: seguimiento de progreso, foros, comentarios, y sistema de insignias o certificados.

## 2.3 Características de los usuarios

Los usuarios previstos del sistema incluyen:

Tipo de usuario	Formación	Habilidades	Actividades esperadas
Estudiantes principiantes	Nivel secundario completo o en curso	Uso básico de computadoras e internet	Acceder a materiales, ver videos, hacer ejercicios
Estudiantes universitarios	Carreras afines (sistemas, programación, software)	Intermedios o avanzados en programación	Usar la plataforma como refuerzo académico
Autodidactas	Formación diversa	Interés en aprender por su cuenta	Seguir cursos, realizar prácticas, descargar guías

## 2.4 Restricciones

El desarrollo del sistema deberá considerar las siguientes limitaciones:

- El diseño debe ser **responsive** (adaptado a móviles, tablets y computadoras).
- El contenido debe ser **ligero y accesible**, sin necesidad de hardware especializado.
- El sistema deberá cumplir con **estándares básicos de accesibilidad web (WCAG nivel A/AA)**.
- Las tecnologías utilizadas deben ser **open source o gratuitas** en la etapa inicial.
- El lenguaje de desarrollo deberá facilitar la **escalabilidad** (ej. Python con Django, WordPress con plugins LMS).
- La carga del sitio deberá optimizarse para conexiones de internet **de media o baja velocidad**, comunes en zonas rurales o con acceso limitado.

## 2.5 Suposiciones y dependencias

El desarrollo y funcionamiento del sistema se basan en las siguientes suposiciones y dependencias:

- Se supone que los usuarios tendrán acceso a una conexión a internet y a un dispositivo con navegador web actualizado.
- Se espera que el contenido se desarrolle e incremente de forma progresiva, por lo que inicialmente el sistema contará con una cantidad limitada de cursos.
- La plataforma dependerá de servicios de terceros como:
  - Hosting web (ej. GitHub Pages, Netlify, o proveedor de hosting para WordPress).
  - Pasarelas de pago (Cafecito, PayPal, Stripe).
  - Reproductores de video (YouTube o Vimeo embebido).
- Se asume que la creadora del proyecto será la única responsable del contenido inicial, por lo tanto el sistema deberá ser fácil de mantener y expandir por una sola persona.
- Se depende del uso de herramientas gratuitas para diseño, edición de video, y gestión de contenido (ej. Canva, OBS, Visual Studio Code).

## 2.6 Evolución previsible del sistema

El sistema está concebido para evolucionar de forma modular, escalando progresivamente en funcionalidad, contenido y usuarios. Se espera incorporar en futuras versiones:

- Un sistema de progreso del estudiante, con indicadores visuales e historial de avance.
- Certificados de finalización automáticos al completar cursos o unidades.
- Foros de discusión o chats para generar comunidad y soporte entre estudiantes.
- Gamificación: insignias, logros y puntos por completar actividades.
- Un sistema más robusto de gestión de cursos y contenidos, con área de administración para agregar o modificar cursos sin modificar el código directamente.
- Opción de multi-idioma para llegar a usuarios fuera del ámbito hispanohablante.
- Colaboraciones con otros docentes o especialistas para ampliar el contenido.

## 3. Requisitos Específicos

### 3.1 Requisitos comunes de interfaces

#### 3.1.1 Interfaces de usuario

- El sistema contará con una **interfaz amigable, moderna y accesible**, orientada a estudiantes principiantes e intermedios.
- Deberá tener un **menú de navegación claro** con secciones como: Inicio, Cursos, Blog, Recursos, Sobre mí, Contacto.
- Se deberá mantener coherencia en los estilos visuales (colores, tipografía, iconografía).
- La página principal incluirá un resumen de cursos destacados, enlaces rápidos y testimonios.

Cada curso tendrá su propia página con:

- Descripción del curso.
- Contenido dividido por unidades.
- Enlaces a recursos descargables.
- Videos integrados desde YouTube o servidores propios.

### 3.1.2 Interfaces de hardware

No se requieren interfaces de hardware específicas.

El sistema debe funcionar correctamente en navegadores modernos de computadoras, tablets y smartphones.

### 3.1.3 Interfaces de software

- Uso de frameworks frontend como React o backend como Django (opcional).
- Integración con servicios externos: YouTube para videos embebidos. PayPal, Cafecito o Stripe para pagos.

### 3.1.4 Interfaces de comunicación

- Se integrarán formularios de contacto mediante correo electrónico o API de servicios como Formspree o EmailUS.
- Para versiones avanzadas: posibilidad de chat embebido con Tawk.to o bots educativos.
- Los protocolos utilizados serán HTTP/HTTPS.

## 3.2 Requisitos funcionales

#### RF 1 – Visualización de cursos

- Tipo: Requisito
  - Fuente: Usuario final
  - Prioridad: Alta
- Descripción:** El sistema debe permitir a cualquier usuario acceder a un listado de cursos disponibles, categorizados por año o temática.

#### RF 2 – Acceso a contenido

- Tipo: Requisito
  - Fuente: Usuario final
  - Prioridad: Alta
- Descripción:** El sistema debe permitir a los usuarios visualizar contenido multimedia (texto, videos, imágenes, PDF).

#### RF 3 – Registro y login

- Tipo: Requisito
  - Fuente: Plataforma
  - Prioridad: Media
- Descripción:** El sistema debe ofrecer una opción de registro para que los usuarios puedan crear una cuenta y guardar su progreso.

#### RF 4 – Descarga de recursos

- Tipo: Requisito
  - Fuente: Usuario final
  - Prioridad: Media
- Descripción:** Los usuarios registrados podrán descargar recursos adicionales asociados a cada curso (plantillas, apuntes, código fuente).

## RF 5 – Donaciones

- Tipo: Requisito
- Fuente: Plataforma
- Prioridad: Alta

**Descripción:** El sistema deberá permitir el pago por determinados cursos mediante plataformas como Cafecito o PayPal, y desbloquear el contenido premium.

## 3.3 Requisitos no funcionales

### 3.3.1 Rendimiento

- El sistema debe permitir al menos 50 usuarios simultáneos sin pérdida de rendimiento en la primera versión.
- El tiempo de carga de las páginas debe ser menor a 3 segundos en condiciones estándar.

### 3.3.2 Seguridad

- Uso de conexión HTTPS obligatoria.
- Protección básica contra inyecciones de código, formulario de contacto con validaciones, y reCAPTCHA si aplica.

### 3.3.3 Fiabilidad

- El sistema deberá tener una disponibilidad de al menos 95% mensual, y deberá ser capaz de recuperarse de caídas con una copia de seguridad mensual.

### 3.3.4 Disponibilidad

- La plataforma debe estar disponible 24/7, salvo períodos programados de mantenimiento.

### 3.3.5 Mantenibilidad

- El código debe estar modularizado, documentado y versionado con Git para facilitar mantenimiento.
- La administradora (vos) debe poder subir nuevos cursos o editar textos desde una interfaz visual o CMS.

### 3.3.6 Portabilidad

- El sistema deberá ser accesible desde cualquier navegador moderno (Chrome, Firefox, Edge, Safari).
- En caso de desarrollarse desde cero, deberá poder desplegarse fácilmente en plataformas como Netlify, Vercel o Heroku.

## 3.4 Otros requisitos

- El diseño debe promover un ambiente educativo accesible, con colores legibles, buena jerarquía visual y navegación sencilla.
- El sistema deberá estar preparado para incluir versiones en inglés y español en un futuro.
- El sitio cumplirá con la Ley de Protección de Datos Personales en cuanto al manejo de información de los usuarios (nombre, email).

## 4. Apéndices

En esta sección se incluye información complementaria que puede ser útil para comprender o implementar correctamente los requisitos, pero que no forma parte del cuerpo principal del documento.

Término	Definición
Aula virtual	Plataforma web que permite acceder a contenidos educativos en línea
MVP (Producto Mínimo Viable)	Versión funcional básica del producto con las características esenciales para su funcionamiento inicial
Responsivo	Diseño web adaptado a diferentes tamaños de pantalla (PC, móvil, tablet)
LMS (Learning Management System)	Sistema de gestión de aprendizaje

### Apéndice B – Herramientas utilizadas

Herramienta	Propósito
Canva	Diseño de portadas, imágenes y recursos visuales
OBS Studio	Grabación de clases en video
GitHub Pages / Netlify	Publicación del sitio web
Visual Studio Code	Edición de código fuente
WordPress + TutorLMS	Alternativa para construir el aula sin programar desde cero
Google Forms o EmailUS	Formularios de contacto y retroalimentación

### Apéndice C – Enlaces de referencia



Enlace	Descripción
<a href="https://freecodecamp.org">freecodecamp.org</a>	Plataforma gratuita de aprendizaje de programación
<a href="https://undraw.co">undraw.co</a>	Ilustraciones gratuitas para uso web
<a href="https://storyset.com">storyset.com</a>	Recursos gráficos educativos
<a href="https://www.ieee.org/standards/publications/830-1998">IEEE 830-1998</a>	Estándar para especificación de requisitos de software

## Apéndice D – Plan tentativo de desarrollo

Fase	Actividades clave	Duración estimada
Fase 1 – Preparación	Recolección de contenidos, bocetos de diseño, organización de materias	1 semana
Fase 2 – Desarrollo MVP	Maquetado, estructura básica, carga de 2 o 3 cursos iniciales	2 semanas
Fase 3 – Pruebas	Revisión, feedback de usuarios beta, correcciones	1 semana
Fase 4 – Lanzamiento	Publicación, promoción en redes, seguimiento	1 semana
Fase 5 – Evolución	Incorporación de nuevos cursos, mejoras, contenido premium	Continua