

EXPEUEW7

# Especificación de requisitos de software

---

*Proyecto: [EduTechAPP]*

Revisión: [01.00]

[24-04-2025]

## Contenido

Ficha del documento	4
1. Introducción	5
1.1. Propósito	5
1.2. Ámbito del Sistema	5
1.3. Definiciones, Acrónimos y Abreviaturas	5
1.4. Referencias	5
1.5. Visión General del Documento	5
2. Descripción General	6
2.1. Perspectiva del Producto	6
2.2. Funciones del Producto	6
2.3. Características de los Usuarios	6
2.4. Restricciones	6
2.5. Suposiciones y Dependencias	7
2.6. Requisitos Futuros	7
3. Requisitos Específicos	8
3.1. Requisitos comunes de los interfaces	9
3.1.1 Interfaces de usuario	9
3.1.2 Interfaces de hardware	9
3.1.3 Interfaces de software	9
3.1.4 Interfaces de comunicación	9
3.2. Requisitos funcionales	9
3.3. Requisitos no funcionales	10
3.3.1 Requisitos de rendimiento	10
3.3.2 Seguridad	10
3.3.3 Fiabilidad	10
3.3.4 Disponibilidad	11
3.3.5 Mantenibilidad	11

3.3.6	Portabilidad	11
3.4	Otros Requisitos	11

## Ficha del documento

Fecha	Revisión	Autor	Modificación

Documento validado por las partes en fecha:

Por el cliente

Por la empresa suministradora

[Firma]

[Firma]

\_\_\_\_\_  
Sr./Sra.

\_\_\_\_\_  
Sr./Sra.

## 1. Introducción

En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software (ERS). Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

### 1.1. Propósito

Este documento especifica los requisitos de software para la transformación digital de EduTech Innovators SPA, centrada en la migración de su sistema monolítico a una arquitectura de microservicios con base de datos MySQL. Su objetivo es mejorar el rendimiento, la disponibilidad y la escalabilidad del sistema.

### 1.2. Ámbito del Sistema

El sistema EduTechAPP proporcionará una plataforma educativa en línea con funcionalidades para la gestión de cursos, usuarios, instructores y evaluaciones. Además, incluirá herramientas de soporte técnico, control de contenido y monitoreo del rendimiento. No se contemplan funcionalidades de creación de contenido educativo ni procesamiento de pagos externos.

### 1.3. Definiciones, Acrónimos y Abreviaturas

ERS: Especificación de Requisitos de Software

API: Interfaz de Programación de Aplicaciones

UX: Experiencia del Usuario

### 1.4. Referencias

Especificación de Requisitos según estándar de IEEE 830.

Caso EduTech Innovators SPA

### 1.5. Visión General del Documento

Este documento detalla el contexto del proyecto, los requisitos funcionales y no funcionales, así como las interfaces y restricciones que definirán el desarrollo de la nueva plataforma.

## 2. Descripción General

### 2.1. Perspectiva del Producto

El sistema basado en microservicios permitirá una mayor escalabilidad y disponibilidad. Se integrará con herramientas externas para análisis de datos, notificaciones al momento, autenticación segura y monitoreo en tiempo real.

### 2.2. Funciones del Producto

- Gestión de usuarios y permisos
- Administración de cursos y contenido educativo
- Creación y evaluación de pruebas
- Soporte a estudiantes mediante foros y chat
- Monitoreo del rendimiento del sistema

### 2.3. Características de los Usuarios

El sistema será utilizado por administradores, gerentes de cursos, instructores, personal de soporte técnico y estudiantes. Se diseñará con una interfaz intuitiva adaptada a cada perfil.

### 2.4. Restricciones

El desarrollo del sistema estará sujeto a las siguientes limitaciones:

**Políticas de la empresa:** Cumplimiento con normativas internas de EduTech Innovators SPA.

**Limitaciones del hardware:** Dependencia de servidores en la nube para garantizar la escalabilidad.

**Interfaces con otras aplicaciones:** Integración con sistemas de autenticación y comunicación mediante APIs.

**Operaciones paralelas:** Soporte para múltiples solicitudes concurrentes sin afectar el rendimiento.

**Funciones de auditoría:** Registro de actividades para garantizar la trazabilidad de los datos.

**Funciones de control:** Implementación de roles y permisos de acceso basados en usuarios.

**Lenguaje(s) de programación:** Uso de tecnologías compatibles con arquitecturas de microservicios y SQL para la base de datos.

**Protocolos de comunicación:** Implementación de RESTful APIs y WebSockets para comunicación en tiempo real.

**Requisitos de habilidad:** Personal técnico capacitado en desarrollo de software basado en microservicios.

**Criticalidad de la aplicación:** Alta disponibilidad y mínima tolerancia a fallos para garantizar la continuidad del servicio.

**Consideraciones acerca de la seguridad:** Protección de datos mediante encriptación y autenticación multifactor.

## 2.5. Suposiciones y Dependencias

Integración con plataformas de gestión educativa

Uso de APIs externas para validación de usuarios

## 2.6. Requisitos Futuros

Esta subsección esbozará futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.

### 3. Requisitos Específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

- El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.
- Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.
- Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.
- Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:
  - **Corrección:** La ERS es correcta si y sólo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
  - **No ambiguos:** Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.
  - **Completo:** Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
  - **Consistentes:** Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.
  - **Clasificados:** Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
  - **Verificables:** La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable.



Expresiones como a veces, bien, adecuado, etc. Introducen ambigüedad en los requisitos. Requisitos como "en caso de accidente la nube tóxica no se extenderá más allá de 25Km" no es verificable por el alto costo que conlleva.

- **Modificables:** La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos facilitan enormemente esta tarea.
- **Trazables:** La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica que componentes del sistema son los que realizan el requisito R.

## Por Avanzar

---

## Listo

### 3.1 Requisitos comunes de los interfaces

Las interfaces de usuario deben contar con un diseño intuitivo y adaptable, garantizando una experiencia fluida y accesible para todos los usuarios. Esto implica la organización clara de los elementos visuales, una navegación sencilla y la implementación de principios de usabilidad que faciliten la interacción.

#### 3.1.1 Interfaces de usuario

Interfaces de usuario con diseño intuitivo y adaptado para una experiencia fluida y accesible para todos los usuarios, esto implica un diseño orientando a los patrones propios de los colores institucionales con una navegación sencilla, elementos visuales claros y la implementación de usabilidad que faciliten la interacción del usuario,

### 3.1.2 Interfaces de hardware

- Soporte para despliegue en servidores en la nube
- Algunas pueden integrar cifrado o autenticaciones para la protección de los datos.

### 3.1.3 Interfaces de software

Productos de software externos integrables:

- **Sistema de Autenticación** (Auth0 o Firebase Authentication)

Gestionar la autenticación de usuarios y permisos de acceso.

API REST con soporte para OAuth 2.0 y JWT.

- **Plataforma de Mensajería** (Twilio, SendGrid)

-Propósito del interfaz: Envío de notificaciones por correo electrónico y SMS a los usuarios.

-Definición del interfaz: API REST con soporte para múltiples formatos (JSON, XML)..

- **Sistema de Monitoreo y Alertas** (Datadog, Prometheus)

-Propósito del interfaz: Seguimiento del rendimiento del sistema y detección de fallos.

-Definición del interfaz: Integración mediante agentes y consultas API para visualización en dashboards.

### 3.1.4 Interfaces de comunicación

- El sistema deberá contar con interfaces de comunicación que permitan la integración con herramientas de mensajería, notificaciones y soporte en tiempo real.
- Se deberá garantizar la comunicación fluida entre los distintos módulos del sistema y con los usuarios finales a través de API REST

## 3.2 Requisitos funcionales

son los requisitos mínimos para que un sistema pueda cumplir su propósito.

### 3.2.1 Gestión de Usuarios (Administrador del Sistema)

- Crear, actualizar, desactivar y eliminar cuentas de usuarios.
- Asignar o modificar permisos de las cuentas

### 3.2.2 Monitorización y Mantenimiento del Sistema

- Ver el estado del sistema y recibir alertas sobre fallos.
- Realizar copias de seguridad y restaurar datos.

### 3.3.3 Gestión de Cursos:

- **Gerentes de Cursos I**

- Crear, actualizar y eliminar cursos.
- Gestionar instructores y evaluar el contenido antes de su publicación.
- Crear, actualizar y corregir evaluaciones y pruebas.
- Monitorear el progreso de los estudiantes.

- **logística académica II**

- Gestión del presupuesto de los cursos.
- Manejan problemas logísticos rápidamente como algún cambio de hora a última hora o fallos.
- Asegura que los horarios de los cursos no se superpongan, y la disponibilidad de los materiales necesarios.

### 3.3.4 Interacción con Estudiantes

- Responder preguntas a través de foros o chats.

### 3.3.5 Gestión de Incidencias (Logística de Soporte)

- Recibir, asignar y resolver incidentes técnicos.
- Mejorar recursos tecnológicos.

### 3.3.6 Acciones de clientes vía web

- Registrarse.
- Explorar y buscar recursos.
- Inscripción en cursos y consulta de progreso.
- Gestión del perfil y detalles de pago.
- Solicitar soporte mediante formulario o chat.
- Dejar reseñas y calificaciones.
- Aplicar cupones y descuentos en inscripciones.

### 3.3 Requisitos no funcionales

las características mínimas de calidad del sistema, como rendimiento, seguridad y usabilidad.

#### 3.3.1 Requisitos de rendimiento

- El sistema debe de soportar un mínimo de mil usuarios conectados a la vez, sin que afecte el rendimiento.
- Se debe garantizar que el servicio esté disponible el mayor tiempo posible, generando más satisfacción al cliente.
- El sistema debe estar diseñado para escalar cuando sea necesario adicionar más recursos cuando la demanda lo requiera.

#### 3.3.2 Seguridad

- Se debe implementar un sistema de autenticación seguro para validar el acceso hacia el usuario.
- Se debe generar un informe detallado de las actividades realizadas dentro del sistema para un monitoreo.
- La información almacenada debe estar protegida con un sistema de cifrado para accesos no autorizados.

#### 3.3.3 Fiabilidad

- El sistema debe seguir funcionando incluso si un componente deja de funcionar, asegurando que los usuarios no se vean afectados.
- La información debe ser respaldada de manera automática, asegurando la disponibilidad en caso de incidentes.

#### 3.3.4 Disponibilidad

- En caso de una falla, el sistema debe ser capaz de volver a funcionar rápidamente usando copias de seguridad o recursos adicionales.
- Se debe implementar una herramienta que detecte los posibles fallos antes de que afecte a los usuarios.
- La plataforma debe estar disponible el mayor tiempo posible, evitando innecesarias interrupciones.

#### 3.3.5 Mantenibilidad

- Las actualizaciones deben de seguir un proceso organizado para asegurar que se pueda rastrear todos los cambios realizados.
- Debe de existir un plan para hacer un respaldo de los datos y copias de seguridad en caso de fallos, para proteger la información del sistema.
- El sistema debe ser fácil de mantener y actualizar, permitiendo que los cambios se realicen de manera rápida y sin afectar el rendimiento.

- La información obsoleta debe ser depurada periódicamente para evitar la acumulación de datos innecesarios.

### **3.3.6 Portabilidad**

- La plataforma debe poder ejecutarse en diferentes sistemas operativos sin requerir modificaciones.
- Se debe de ocupar lenguajes de programación con una compatibilidad amplia para implementar en diferentes entornos.
- Agregar soluciones que faciliten la implementación en distintas estructuras, permitiendo un despliegue eficiente.

## **3.4 Otros Requisitos**

Cumplimiento con normativas educativas y de protección de datos

Accesibilidad para usuarios con discapacidad