

ESPECIFICACIÓN DE REQUERIMIENTOS

Sistema de Gestión Documental Universitaria TESCHI

CARÁTULA

INSTITUCIÓN: Tecnológico de Estudios Superiores de Chimalhuacán (TESCHI)
PROYECTO: Sistema de Gestión Documental Universitaria
MATERIA: Ingeniería de Software
DOCENTE: Modesto Castro Yolanda
SEMESTRE: 7ISC23

INTEGRANTES DEL EQUIPO:

1. **Gálvez Romero Irvin Osvaldo** - Administrador de Base de Datos / Desarrollador Full-Stack
2. **Cruz Contreras Ángel Valentín** - Desarrollador Frontend / UI/UX
3. **Sánchez Vargas Kevin Antonio** - Analista de Sistemas / Desarrollador Backend
4. **Juárez Vargas Alberto** - Líder de Proyecto / DevOps

FECHA: [Fecha actual]

VERSIÓN: 1.0

1. INTRODUCCIÓN

1.1 Propósito del Documento

Este documento especifica los requerimientos funcionales y no funcionales del Sistema de Gestión Documental Universitaria TESCHI, diseñado para facilitar la administración de documentos académicos, gestión de estudiantes y control de procesos administrativos de la institución.

1.2 Alcance del Proyecto

El sistema permitirá a los administradores gestionar estudiantes, revisar documentos académicos y mantener un control centralizado de la información. Los estudiantes podrán subir documentos requeridos y consultar su estado. Los profesores tendrán acceso a la información de sus estudiantes y podrán revisar documentos.

1.3 Definiciones y Acrónimos

- **TESCHI:** Tecnológico de Estudios Superiores de Chimalhuacán
- **SGD:** Sistema de Gestión Documental
- **API:** Application Programming Interface
- **UI/UX:** User Interface / User Experience
- **CRUD:** Create, Read, Update, Delete
- **JWT:** JSON Web Token
- **REST:** Representational State Transfer

1.4 Referencias

- Estándares de desarrollo web W3C
 - Guías de accesibilidad WCAG 2.1
 - Estándares de seguridad OWASP
 - Metodología ágil de desarrollo
-

2. DESCRIPCIÓN GENERAL

2.1 Perspectiva del Producto

El Sistema de Gestión Documental Universitaria TESCHI es una aplicación web que se integra con los procesos administrativos existentes de la institución, proporcionando una plataforma centralizada para la gestión de documentos académicos y administración de estudiantes.

2.2 Funcionalidades del Producto

- Gestión de usuarios (estudiantes, profesores, administradores)
- Subida y gestión de documentos académicos
- Sistema de autenticación y autorización
- Dashboard administrativo con estadísticas
- Sistema de notificaciones por correo electrónico
- Gestión de carreras académicas
- Reportes y consultas

2.3 Clases de Usuarios y Características

- **Administradores:** Acceso completo al sistema, gestión de usuarios y documentos
- **Profesores:** Acceso a información de estudiantes y revisión de documentos
- **Estudiantes:** Subida de documentos y consulta de estado

2.4 Restricciones Generales

- Debe funcionar en navegadores modernos (Chrome, Firefox, Safari, Edge)
 - Debe ser responsive para dispositivos móviles
 - Debe cumplir con estándares de accesibilidad
 - Debe ser compatible con archivos PDF, DOC, DOCX, JPG, PNG
-

3. REQUERIMIENTOS FUNCIONALES

3.1 Gestión de Usuarios

RF-001: Registro de Usuarios

Descripción: El sistema debe permitir el registro de nuevos usuarios con diferentes roles.

Prioridad: Alta

Complejidad: Media

Criterios de Aceptación:

- El sistema debe validar que el email sea único
- Debe requerir campos obligatorios: nombre, apellidos, email, teléfono
- Debe generar credenciales automáticamente para estudiantes y profesores
- Debe enviar credenciales por correo electrónico
- Debe asignar roles según el tipo de usuario

Entrada:

- Datos personales del usuario
- Tipo de usuario (estudiante, profesor, administrador)
- Información académica (matrícula, carrera)

Salida:

- Usuario registrado en el sistema
- Credenciales enviadas por correo
- Confirmación de registro

RF-002: Autenticación de Usuarios

Descripción: El sistema debe autenticar usuarios mediante email y contraseña.

Prioridad: Alta

Complejidad: Media

Criterios de Aceptación:

- Debe validar credenciales de acceso
- Debe generar token de sesión JWT
- Debe redirigir según el rol del usuario
- Debe requerir cambio de contraseña en primer login
- Debe implementar bloqueo por intentos fallidos

Entrada:

- Email y contraseña del usuario

Salida:

- Token de autenticación
- Redirección al dashboard correspondiente

RF-003: Gestión de Perfiles

Descripción: Los usuarios deben poder actualizar su información personal.

Prioridad: Media

Complejidad: Baja

Criterios de Aceptación:

- Debe permitir actualizar datos personales
- Debe validar cambios antes de guardar
- Debe mantener historial de cambios
- Debe notificar cambios importantes

3.2 Gestión de Estudiantes

RF-004: Administración de Estudiantes

Descripción: Los administradores deben poder gestionar la información de estudiantes.

Prioridad: Alta

Complejidad: Alta

Criterios de Aceptación:

- Debe permitir crear, leer, actualizar y eliminar estudiantes
- Debe incluir búsqueda y filtrado de estudiantes
- Debe mostrar información completa del estudiante
- Debe permitir dar de baja (soft delete) y eliminar permanentemente (hard delete)
- Debe generar matrículas automáticamente

Entrada:

- Datos del estudiante
- Criterios de búsqueda
- Acciones de gestión

Salida:

- Lista de estudiantes
- Información detallada
- Confirmaciones de acciones

RF-005: Asignación de Carreras

Descripción: Los estudiantes deben ser asignados a carreras académicas.

Prioridad: Alta

Complejidad: Media

Criterios de Aceptación:

- Debe permitir seleccionar carrera al registrar estudiante
- Debe validar que la carrera esté activa
- Debe permitir cambio de carrera
- Debe mantener historial de cambios

3.3 Gestión de Documentos

RF-006: Subida de Documentos

Descripción: Los estudiantes deben poder subir documentos académicos.

Prioridad: Alta

Complejidad: Media

Criterios de Aceptación:

- Debe permitir subir archivos PDF, DOC, DOCX, JPG, PNG
- Debe validar tamaño máximo de 10MB
- Debe categorizar documentos por tipo
- Debe generar nombre único para archivos
- Debe almacenar metadatos del documento

Entrada:

- Archivo del documento
- Tipo de documento
- Descripción opcional

Salida:

- Documento almacenado
- Confirmación de subida
- Notificación a administrador

RF-007: Revisión de Documentos

Descripción: Los administradores deben poder revisar y aprobar/rechazar documentos.

Prioridad: Alta

Complejidad: Media

Criterios de Aceptación:

- Debe mostrar lista de documentos pendientes
- Debe permitir previsualizar documentos
- Debe permitir descargar documentos
- Debe permitir aprobar o rechazar con comentarios
- Debe notificar resultado al estudiante

Entrada:

- Lista de documentos
- Acción de revisión
- Comentarios opcionales

Salida:

- Estado actualizado del documento
- Notificación al estudiante
- Historial de revisiones

RF-008: Consulta de Documentos

Descripción: Los usuarios deben poder consultar el estado de sus documentos.

Prioridad: Media

Complejidad: Baja

Criterios de Aceptación:

- Debe mostrar estado de cada documento
- Debe permitir filtrar por estado
- Debe mostrar fecha de subida y revisión
- Debe permitir descargar documentos aprobados

3.4 Dashboard y Reportes

RF-009: Dashboard Administrativo

Descripción: Los administradores deben tener acceso a un dashboard con estadísticas.

Prioridad: Alta

Complejidad: Media

Criterios de Aceptación:

- Debe mostrar estadísticas de estudiantes
- Debe mostrar estadísticas de documentos
- Debe incluir gráficos y métricas
- Debe actualizarse en tiempo real
- Debe permitir exportar reportes

RF-010: Dashboard de Estudiante

Descripción: Los estudiantes deben tener acceso a su dashboard personal.

Prioridad: Media

Complejidad: Baja

Criterios de Aceptación:

- Debe mostrar información personal
- Debe mostrar estado de documentos
- Debe incluir notificaciones
- Debe permitir acceso rápido a funciones

3.5 Sistema de Notificaciones

RF-011: Notificaciones por Correo

Descripción: El sistema debe enviar notificaciones por correo electrónico.

Prioridad: Alta

Complejidad: Media

Criterios de Aceptación:

- Debe enviar credenciales de acceso
 - Debe notificar cambios de estado de documentos
 - Debe enviar recordatorios
 - Debe usar plantillas profesionales
 - Debe manejar errores de envío
-

4. REQUERIMIENTOS NO FUNCIONALES

4.1 Rendimiento

RNF-001: Tiempo de Respuesta

Descripción: El sistema debe responder a las peticiones en tiempo adecuado.

Especificación:

- Páginas web: < 3 segundos
- API endpoints: < 2 segundos
- Consultas de base de datos: < 1 segundo
- Carga de archivos: < 10 segundos

RNF-002: Capacidad

Descripción: El sistema debe soportar la carga esperada de usuarios.

Especificación:

- Usuarios concurrentes: 100
- Documentos por usuario: 50
- Tamaño total de archivos: 1TB
- Transacciones por minuto: 1000

4.2 Seguridad

RNF-003: Autenticación y Autorización

Descripción: El sistema debe implementar medidas de seguridad robustas.

Especificación:

- Autenticación mediante JWT
- Contraseñas encriptadas con bcrypt
- Sesiones con expiración automática
- Bloqueo por intentos fallidos
- Validación de entrada en todos los campos

RNF-004: Protección de Datos

Descripción: Los datos deben estar protegidos contra accesos no autorizados.

Especificación:

- Conexiones HTTPS obligatorias
- Validación de archivos subidos
- Sanitización de datos de entrada
- Logs de auditoría
- Backup regular de datos

4.3 Usabilidad

RNF-005: Interfaz de Usuario

Descripción: La interfaz debe ser intuitiva y fácil de usar.

Especificación:

- Diseño responsive para móviles
- Navegación intuitiva
- Mensajes de error claros
- Ayuda contextual
- Accesibilidad WCAG 2.1 AA

RNF-006: Compatibilidad

Descripción: El sistema debe funcionar en diferentes navegadores y dispositivos.

Especificación:

- Chrome 90+
- Firefox 88+
- Safari 14+
- Edge 90+
- Dispositivos móviles iOS y Android

4.4 Confiabilidad

RNF-007: Disponibilidad

Descripción: El sistema debe estar disponible la mayor parte del tiempo.

Especificación:

- Uptime: 99.5%
- Tiempo de recuperación: < 4 horas
- Backup automático diario
- Monitoreo 24/7

RNF-008: Recuperación

Descripción: El sistema debe poder recuperarse de fallos.

Especificación:

- Backup de base de datos cada 24 horas
- Restauración completa en < 2 horas
- Logs detallados para debugging
- Procedimientos de recuperación documentados

4.5 Mantenibilidad

RNF-009: Código Limpio

Descripción: El código debe ser mantenible y documentado.

Especificación:

- Código documentado al 80%
- Pruebas unitarias al 70%
- Estándares de codificación consistentes
- Documentación técnica actualizada

RNF-010: Escalabilidad

Descripción: El sistema debe poder crecer con las necesidades.

Especificación:

- Arquitectura modular
- Base de datos optimizada
- Caché para consultas frecuentes
- Posibilidad de escalado horizontal

5. RESTRICCIONES DEL SISTEMA

5.1 Restricciones de Hardware

- Servidor con mínimo 4GB RAM
- Almacenamiento mínimo 100GB
- Procesador de 2 núcleos mínimo
- Conexión a internet estable

5.2 Restricciones de Software

- Node.js 16+
- Base de datos SQLite/PostgreSQL
- Navegador web moderno
- Sistema operativo Windows/Linux/macOS

5.3 Restricciones de Interfaz

- Debe funcionar sin JavaScript deshabilitado
- Debe ser compatible con lectores de pantalla
- Debe soportar zoom hasta 200%
- Debe funcionar con teclado únicamente

5.4 Restricciones de Seguridad

- No almacenar contraseñas en texto plano
 - Validar todos los archivos subidos
 - Implementar rate limiting
 - Logs de auditoría obligatorios
-

6. REQUERIMIENTOS DE INTERFAZ

6.1 Interfaz de Usuario

- Diseño moderno y profesional
- Colores institucionales de TESCHI
- Iconografía consistente
- Tipografía legible
- Layout responsive

6.2 Interfaz de Hardware

- Compatible con mouse y teclado
- Soporte para pantallas táctiles
- Funcionamiento en tablets
- Optimizado para móviles

6.3 Interfaz de Software

- API REST estándar
- Formato JSON para intercambio de datos
- Autenticación mediante headers HTTP
- Manejo de errores consistente

6.4 Interfaz de Comunicaciones

- Protocolo HTTPS
 - Codificación UTF-8
 - Formato de correo HTML
 - Notificaciones push opcionales
-

7. REQUERIMIENTOS DE DATOS

7.1 Estructura de Base de Datos

- Normalización hasta 3NF
- Índices para consultas frecuentes
- Claves foráneas para integridad
- Triggers para auditoría

7.2 Almacenamiento de Archivos

- Sistema de archivos organizado por usuario
- Nombres únicos para evitar conflictos
- Metadatos almacenados en base de datos
- Compresión de archivos grandes

7.3 Backup y Recuperación

- Backup diario automático
- Retención de 30 días
- Pruebas de recuperación mensuales
- Almacenamiento en ubicación segura

8. REQUERIMIENTOS DE IMPLEMENTACIÓN

8.1 Tecnologías Requeridas

- Frontend: React 18+ con TypeScript
- Backend: Node.js con Express
- Base de datos: SQLite/PostgreSQL
- ORM: Prisma
- Autenticación: JWT
- Estilos: Material-UI

8.2 Estándares de Desarrollo

- Código en TypeScript
- ESLint para linting
- Prettier para formateo
- Git para control de versiones
- Testing con Jest

8.3 Despliegue

- Servidor Linux/Windows
- Nginx como proxy reverso
- PM2 para gestión de procesos
- SSL/TLS obligatorio
- Dominio personalizado

9. CRITERIOS DE ACEPTACIÓN

9.1 Funcionalidad

- Todos los requerimientos funcionales implementados
- Casos de uso completados al 100%
- Integración entre módulos funcional
- Pruebas de usuario exitosas

9.2 Calidad

- Código sin bugs críticos
- Rendimiento dentro de especificaciones
- Seguridad validada
- Documentación completa

9.3 Entrega

- Sistema desplegado en producción
- Usuarios capacitados
- Documentación entregada
- Soporte inicial proporcionado

Documento elaborado por: Equipo de Desarrollo TESCHI

Fecha de creación: [Fecha actual]

Última actualización: [Fecha actual]

Próxima revisión: [Fecha + 1 mes]