|  |
| --- |
| DUOC UC - Escuela de informatica y telecomunicaciones |
| Especificación de Requisitos de Software |
| *Proyecto:* |
|  |
| **Revisión*: [1.0]*** |
| **[Seleccionar fecha]** |

|  |
| --- |
| Especificación de Requisitos según estándar de IEEE 830. |

Contenido

[Ficha del documento 3](#_Toc503382603)

[1. Introducción 4](#_Toc503382604)

[1.1. Propósito 4](#_Toc503382605)

[1.2. Ámbito del Sistema 4](#_Toc503382606)

[1.3. Definiciones, Acrónimos y Abreviaturas 4](#_Toc503382607)

[1.4. Referencias 4](#_Toc503382608)

[1.5. Visión General del Documento 4](#_Toc503382609)

[2. Descripción General 5](#_Toc503382610)

[2.1. Perspectiva del Producto 5](#_Toc503382611)

[2.2. Funciones del Producto 5](#_Toc503382612)

[2.3. Características de los Usuarios 5](#_Toc503382613)

[2.4. Restricciones 5](#_Toc503382614)

[2.5. Suposiciones y Dependencias 6](#_Toc503382615)

[2.6. Requisitos Futuros 6](#_Toc503382616)

[3. Requisitos Específicos 7](#_Toc503382617)

[3.1 Requisitos comunes de las interfaces 8](#_Toc503382618)

[3.1.1 Interfaces de usuario 8](#_Toc503382619)

[3.1.2 Interfaces de hardware 8](#_Toc503382620)

[3.1.3 Interfaces de software 8](#_Toc503382621)

[3.1.4 Interfaces de comunicación 8](#_Toc503382622)

[3.2 Requisitos funcionales 9](#_Toc503382623)

[3.3 Requisitos no funcionales 9](#_Toc503382624)

[3.3.1 Requisitos de rendimiento 9](#_Toc503382625)

[3.3.2 Seguridad 10](#_Toc503382626)

[3.3.3 Fiabilidad 10](#_Toc503382627)

[3.3.4 Disponibilidad 10](#_Toc503382628)

[3.3.5 Mantenibilidad 10](#_Toc503382629)

[3.3.6 Portabilidad 10](#_Toc503382630)

[3.4 Otros Requisitos 10](#_Toc503382631)

# Ficha del documento

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| *07/01/25* | *1* | *Antonio Sepúlveda* |  |
|  |  |  |  |

Documento validado por las partes en fecha:

|  |  |  |
| --- | --- | --- |
| Por el cliente |  | Por la empresa suministradora |
| [Firma] |  | [Firma] |
| Sr./Sra. |  | Sr./Sra. |

# 1. Introducción

## 1.1. Propósito

El propósito de este documento es especificar los requisitos funcionales y no funcionales necesarios para el diseño, desarrollo, implementación y mantenimiento del **Sistema de Ventas y Facturación Online para la Librería "El Lector"**. Este documento servirá como referencia principal para desarrolladores, testers, gerentes de proyecto, analistas y demás partes interesadas involucradas en el proyecto, asegurando que todos los elementos del sistema cumplan con las expectativas del cliente y las necesidades del negocio.

# 1. Introducción

## 1.1. Propósito

El propósito de este documento es definir los requisitos funcionales y no funcionales para el desarrollo de un Sistema de Control de Ventas y Facturación. Está dirigido a desarrolladores, testers, gerentes de proyecto y partes interesadas que participen en el diseño e implementación del sistema.

## 1.3. Definiciones, Acrónimos y Abreviaturas

* **ERS:** Especificación de Requisitos del Software.
* **Factura Electrónica:** Documento fiscal generado y timbrado digitalmente.

## 1.4. Referencias

Requisitos del documento fuente: "Requerimientos.docx"

### **1.5. Visión General del Documento**

Este documento describe los objetivos generales del sistema, incluyendo la implementación de un sistema de ventas y facturación online para la librería. Contiene los requisitos funcionales y no funcionales necesarios, además de una descripción del contexto operativo, estableciendo los fundamentos para el desarrollo, validación e implementación del sistema.

# 2. Descripción General

## 2.1. Perspectiva del Producto

El sistema operará como una solución independiente y estará diseñado para ser accesible a través de navegadores web. Incluirá integraciones con sistemas de facturación locales y funcionalidades para administración de inventario.

## 2.2. Funciones del Producto

Las principales funciones del sistema incluyen:

* Gestión de productos: registro, modificación y eliminacion.
* Generación de facturas electrónicas.
* Consultas y reportes financieros e inventarios.

## 2.3. Características de los Usuarios

El sistema será utilizado por:

* **Administrador:** Nivel medio en uso de PC y Excel.
* **Cajero:** Nivel usuario en PC.

## 2.4. Restricciones

* Lenguaje de desarrollo: Node, JavaScript, Tailwind.
* Disponibilidad en navegadores estándar (Chrome, Firefox, etc).
* Generación de reportes en formato PDF y Excel.

• Funciones de control.

• Lenguaje(s) de programación.

• Protocolos de comunicación.

• Requisitos de habilidad.

• Criticidad de la aplicación.

• Consideraciones acerca de la seguridad.

## 2.5. Suposiciones y Dependencias

* El sistema operará en servidores con conexión estable a internet.
* Las regulaciones fiscales locales no sufrirán cambios abruptos.

## 2.6. Requisitos Futuros

* Incorporación de inteligencia de negocios para análisis predictivo.
* Integraciones con APIS más enfocadas al objetivo del proyecto y su eficiencia.

# 3. Requisitos Específicos

Esta sección describe los requisitos del sistema en detalle, asegurando que sean claros, verificables y adaptados a las necesidades de los usuarios y del negocio.

## 3.1 Requisitos comunes de las interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

### 3.1.1 Interfaces de usuario

* El sistema incluirá una interfaz gráfica web optimizada para facilitar la navegación, accesible a través de navegadores modernos como Google Chrome y Mozilla Firefox.
* **Diseño responsivo**: El diseño será adaptable a distintos dispositivos, incluyendo computadoras, tablets y smartphones.
* **Normativas de accesibilidad**: Cumplirá con las pautas WCAG 2.1 para asegurar accesibilidad a usuarios con discapacidad visual.
  + **Contraste adecuado** entre texto y fondo.
  + Compatibilidad con **lectores de pantalla** y navegación mediante teclado.

##### **Componentes principales de la interfaz:**

* **Menú superior fijo**:
  + Acceso rápido a las funciones principales como catálogo, pedidos, facturación, y administración de inventario.
  + Botón para iniciar sesión o acceder a la cuenta del usuario.
* **Área central dinámica**:
  + Muestra formularios, reportes, información del catálogo y detalles de los libros.
  + Utiliza pestañas y secciones para organizar la información.
* **Notificaciones en pantalla**:
  + Mensajes claros para confirmar acciones (compra exitosa, pedido registrado).
  + Alertas de errores o validaciones (datos faltantes, fallos de autenticación).
* **Catálogo interactivo de libros**:
  + Permite la visualización en lista o cuadrícula, mostrando portada, título, precio y disponibilidad de los libros.
  + Filtros y barra de búsqueda para personalizar los resultados.
* **Carrito de compras**:
  + Resumen de los libros seleccionados con cantidades ajustables.
  + Precio total actualizado en tiempo real.
* **Botones y formularios**:
  + Diseño claro con texto descriptivo (e.g., "Agregar al carrito").
  + Validaciones visuales en campos obligatorios.

### 3.1.2 Interfaces de hardware

* **Base de datos MySQL**: Todas las operaciones de consultas, actualizaciones, y eliminación de datos estarán optimizadas para cumplir con los tiempos de respuesta esperados.
* **Facturación electrónica**: Integración con un servicio externo autorizado para la emisión de facturas. La API debe ser compatible con las regulaciones legales locales (e.g., SII en Chile).
* **Manejo de imágenes**: Acceso directo al almacenamiento de imágenes del servidor (carpeta uploads) vinculado a cada libro.

### 3.1.3 Interfaces de software

* Soporte para servidores web estándar que cumplan con los requisitos de:
  + Procesadores x86 o ARM de al menos 2 GHz.
  + 8 GB de memoria RAM mínima para entornos productivos.
  + Almacenamiento SSD para rapidez en acceso a la base de datos y carga de imágenes.

### 3.1.4 Interfaces de comunicación

* El sistema asegurará una comunicación segura y eficiente entre el cliente y el servidor mediante los siguientes estándares y tecnologías:
* **Protocolo HTTPS**:
  + Toda la comunicación entre el cliente (navegador del usuario) y el servidor estará cifrada usando HTTPS.
  + Implementación de certificados SSL/TLS para proteger datos sensibles como credenciales de usuario, información de pago y detalles de pedidos.
* **WebSockets**:
  + Utilización de WebSockets para garantizar sincronización en tiempo real entre usuarios en funcionalidades críticas:
    - **Gestión de inventarios**: Actualización automática de stock cuando se registre una venta.
    - **Notificaciones en vivo**: Alertas para administradores sobre libros agotados o pedidos recibidos.
* **RESTful API**:
  + Arquitectura REST para todas las comunicaciones principales del sistema, asegurando interoperabilidad y escalabilidad.
  + Los endpoints estarán protegidos mediante autenticación por tokens JWT.
* **Integración externa**:
  + **Facturación electrónica**: Comunicación con sistemas de facturación autorizados del SII (Servicio de Impuestos Internos) en Chile mediante sus APIs correspondientes.
  + **Envíos de correo electrónico**: Uso de servicios SMTP como parte de las funcionalidades de notificación a clientes y recuperación de contraseñas.

## 3.2 Requisitos funcionales

Definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.

En ellas se incluye:

* Comprobación de validez de las entradas
* Secuencia exacta de operaciones
* Respuesta a situaciones anormales (desbordamientos, comunicaciones, recuperación de errores)
* Parámetros
* Generación de salidas
* Relaciones entre entradas y salidas (secuencias de entradas y salidas, fórmulas para la conversión de información)
* Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)

Los requisitos funcionales pueden ser divididos en sub-secciones.

#### **3.2.1 Requisito Funcional 1: Gestión de Productos**

**Descripción:**

* El sistema permitirá a los administradores registrar, consultar, actualizar y eliminar productos.
* Se gestionará el inventario actualizando las cantidades en tiempo real tras cada venta.

**Actores:** Administrador de Inventario.

**Comportamiento:**

* Validación de datos requeridos: título, categoría, precio y stock.
* Control de alerta para inventarios bajos.

#### **3.2.2 Requisito Funcional 2: Gestión de Ventas**

**Descripción:**

* Se implementará una funcionalidad para seleccionar productos, calcular el precio final y realizar pedidos.
* Generará automáticamente facturas electrónicas para cada transacción.

**Actores:** Vendedores, Clientes.

**Comportamiento:**

* Registro de productos seleccionados para el pedido.
* Validación de métodos de pago y finalización exitosa de la venta.

#### **3.2.3 Requisito Funcional 3: Gestión de Clientes**

**Descripción:**

* Permitirá el registro y consulta de datos básicos de los clientes, como nombre, dirección, teléfono, y correo electrónico.

**Actores:** Administradores, Clientes.

**Comportamiento:**

* Asociación de ventas a perfiles de clientes.
* Consulta de historiales de compras.

#### **3.2.4 Requisito Funcional 4: Reportes de Ventas**

**Descripción:**

* El sistema generará reportes detallados diarios, semanales y mensuales sobre las ventas.

**Actores:** Gerente Financiero, Administrador.

**Comportamiento:**

* Cálculo dinámico de totales, impuestos aplicados y utilidades generadas.

## 3.3 Requisitos no funcionales

#### **3.3.1 Requisitos de rendimiento**

1. **Capacidad de usuarios simultáneos:**
   1. El sistema deberá soportar hasta 100 usuarios conectados simultáneamente en horas punta sin experimentar degradación significativa en el rendimiento.
2. **Velocidad de transacción:**
   1. El 95% de las transacciones deben completarse en menos de 2 segundos bajo condiciones normales de carga.
   2. Operaciones de búsqueda y consulta en el catálogo deberán ejecutarse en menos de 1 segundo en el 90% de los casos.
3. **Procesamiento de pedidos:**
   1. El sistema deberá ser capaz de procesar al menos 50 pedidos por minuto de manera concurrente sin interrupciones.
4. **Carga de inventario:**
   1. La actualización de inventarios tras una venta debe reflejarse en el sistema en un plazo máximo de 500 milisegundos.
5. **Tolerancia a picos de carga:**
   1. El sistema debe poder manejar incrementos súbitos de tráfico, como campañas promocionales, sin interrupciones, permitiendo hasta un 50% más de usuarios simultáneos en esos periodos.
6. **Transferencia de datos:**
   1. Toda la comunicación entre el servidor y el cliente deberá ser eficiente, con un tiempo máximo de 1.5 segundos para cargar detalles de productos o pedidos de tamaño estándar.

### 3.3.2 Seguridad

* **Cifrado de Datos:**
* Toda la comunicación entre cliente y servidor debe estar protegida mediante el protocolo HTTPS, empleando certificados SSL/TLS actualizados.
* Los datos críticos, como contraseñas, deberán almacenarse utilizando un algoritmo de hash seguro (como bcrypt).
* **Control de Acceso:**
* Solo usuarios autenticados podrán realizar operaciones sensibles, como la gestión de inventarios, emisión de facturas y administración del catálogo de libros.
* El sistema debe soportar roles (e.g., cliente, administrador) y permisos específicos asignados a cada funcionalidad.
* **Autenticación y Autorización:**
* El acceso al sistema se realizará mediante credenciales seguras (usuario/contraseña), protegidas contra ataques de fuerza bruta mediante límites de intentos y bloqueo temporal de cuentas.
* Las API sensibles requerirán tokens de autenticación JWT (JSON Web Tokens) para su acceso.
* **Registros y Auditorías:**
* Se deberá registrar cada operación relevante en logs de actividad, incluyendo:
  + Fecha y hora de la acción.
  + Usuario responsable.
  + Operación realizada.
  + Resultado de la operación.
* Los logs deberán almacenarse de manera segura y ser auditables.
* **Protección contra Amenazas Comunes:**
* Implementar medidas de protección contra ataques de inyección SQL, Cross-Site Scripting (XSS), y Cross-Site Request Forgery (CSRF).
* Validar y sanitizar todas las entradas de usuario para prevenir la ejecución de código malicioso.
* **Respaldo de Información:**
* Generar copias de seguridad automáticas de los datos críticos (inventario, pedidos, facturación, clientes) diariamente, con retención durante un periodo mínimo de 30 días.
* Implementar comprobaciones periódicas de integridad en la base de datos para detectar inconsistencias o manipulaciones indebidas.
* **Notificaciones y Alertas:**
* Configurar el sistema para enviar alertas automáticas a los administradores en caso de actividades inusuales, como múltiples intentos fallidos de inicio de sesión o cambios inesperados en el inventario

### 3.3.3 Fiabilidad

* **Tolerancia a Fallos:**
* El sistema debe ser capaz de manejar errores inesperados y realizar una recuperación controlada sin comprometer datos críticos.
* En caso de falla de conexión con la base de datos, se deberá informar al usuario y registrar la incidencia en el sistema de logs.
* **Errores Permisibles:**
* El tiempo promedio entre fallos críticos (Mean Time Between Failures - MTBF) deberá ser mayor a **6 meses**.
* El tiempo de recuperación promedio (Mean Time To Repair - MTTR) debe ser inferior a **4 horas** en caso de fallo crítico.
* **Pruebas de Resiliencia:**
* Las funcionalidades del sistema se validarán regularmente mediante pruebas de estrés y simulaciones de escenarios de fallo (por ejemplo, interrupciones de red, pérdida de datos temporales).

### 3.3.4 Disponibilidad

* **Requisito de Disponibilidad:**
* El sistema deberá estar disponible el **99.5% del tiempo** mensual, con un máximo de 3.6 horas de inactividad permitidas por mes.
* **Planificación de Mantenimiento:**
* Las interrupciones programadas para mantenimiento no superarán las **4 horas consecutivas** y se comunicarán a los usuarios con al menos **48 horas de antelación**.
* **Monitoreo en Tiempo Real:**
* Implementar un sistema de monitoreo que detecte interrupciones y notifique automáticamente a los administradores mediante alertas por correo o sistemas de notificación.
* **Gestión de Copias de Respaldo:**
* El sistema deberá restaurarse a partir de un respaldo en un tiempo no mayor a **30 minutos** tras un fallo que implique pérdida de disponibilidad.

1. **Tipo de Mantenimiento:**
   1. **Mantenimiento Correctivo:** Resolución de errores detectados en tiempo de ejecución o después del despliegue.
   2. **Mantenimiento Evolutivo:** Incorporación de nuevas funcionalidades según los cambios en los requisitos del negocio o nuevas normativas.
   3. **Mantenimiento Preventivo:** Actualización de las dependencias y herramientas utilizadas para evitar futuros problemas de compatibilidad.
2. **Responsables de Mantenimiento:**
   1. **Desarrolladores:** Responsables de corregir errores en el código fuente, implementar nuevas funcionalidades y realizar pruebas después de los cambios.
   2. **Administradores de Sistema:** Encargados de gestionar el entorno de producción, realizar actualizaciones programadas y supervisar los logs de actividad.
3. **Frecuencia del Mantenimiento:**
   1. **Correctivo:** Realizado inmediatamente después de detectar un incidente crítico o prioritario.
   2. **Preventivo:** Programado trimestralmente para actualización de librerías, entornos y copias de seguridad.
   3. **Estadísticas y Reportes:** Generación de reportes mensuales de acceso, ventas y comportamiento del sistema.

### 3.3.6 Portabilidad

* **Atributos del Software para la Portabilidad:**
* Uso de tecnologías basadas en **JavaScript (Next.js en frontend)** y **Node.js** para el backend, lo que permite ejecutar el sistema en múltiples plataformas.
* El sistema será compatible con navegadores modernos como **Google Chrome, Mozilla Firefox** y cualquier navegador basado en WebKit.
* Dependencia reducida del entorno específico del servidor, utilizando contenedores de **Docker** para facilitar la portabilidad.
* **Dependencias del Servidor:**
* **Porcentaje dependiente:** Aproximadamente el **85% del sistema** depende de la infraestructura del servidor (backend, base de datos y API).
* El frontend es accesible como un cliente independiente que se comunica con el backend a través de la red (REST/HTTPS).
* **Adaptabilidad a Nuevas Plataformas:**
* El uso de un sistema operativo universal como **Linux** para el servidor.
* Disponibilidad de contenedores configurados para despliegue rápido en plataformas de alojamiento populares como **AWS, Azure** o servidores locales.
* **Lenguaje y Herramientas de Desarrollo:**
* El sistema está desarrollado en **JavaScript y TypeScript**, lenguajes ampliamente soportados y con gran portabilidad.
* Uso de **Prisma** para la base de datos, lo que permite adaptar el sistema a múltiples motores de bases de datos sin cambios significativos.

## 3.4 Otros Requisitos

- Cumplir con las normativas legales.

- Integración de plataformas de terceros.