

---

## **Especificación de requisitos de software**

**Proyecto: Desarrollo de una aplicación web progresiva para la administración, venta de productos e integración de un modelo predictivo de demanda para la Asociación**

**KALLARI**

Revisión 0.2



Octubre 2025

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
04/10/2025	Rev. 1.0	Scarlet Cayapa	
20/10/2025	Rev. 2.0	Scarlet Cayapa	

Documento validado por las partes en fecha: 04/10/2025

Por el cliente	Por la empresa suministradora
Fdo. D./ Dña [Nombre]	Fdo. Srta. Scarlet Cayapa



## Contenido

<b>FICHA DEL DOCUMENTO</b>	<b>2</b>
<b>CONTENIDO</b>	<b>3</b>
<b>1 INTRODUCCIÓN</b>	<b>5</b>
1.1 Propósito	5
1.2 Alcance	5
1.3 Personal involucrado	6
1.4 Definiciones, acrónimos y abreviaturas	6
1.5 Referencias	6
1.6 Resumen	7
<b>2 DESCRIPCIÓN GENERAL</b>	<b>7</b>
2.1 Perspectiva del producto	7
2.2 Funcionalidad del producto	7
2.3 Características de los usuarios	8
2.4 Restricciones	8
2.5 Suposiciones y dependencias	8
2.6 Evolución previsible del sistema	9
<b>3 REQUISITOS ESPECÍFICOS</b>	<b>9</b>
<b>3.1 Requisitos comunes de los interfaces</b>	<b>9</b>
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	10
<b>3.2 Requisitos funcionales</b>	<b>10</b>
3.2.1 Requisito funcional 1	12
3.2.2 Requisito funcional 2	12
3.2.3 Requisito funcional 3	13
3.2.4 Requisito funcional n	13
<b>3.3 Requisitos no funcionales</b>	<b>14</b>
3.3.1 Requisitos de rendimiento	14
3.3.2 Seguridad	14



**Especificación de requisitos de software**

---

3.3.3	Fiabilidad	14
3.3.4	Disponibilidad	15
3.3.5	Mantenibilidad	15
3.3.6	Portabilidad	15
<b>3.4</b>	<b>Otros requisitos</b>	<b>15</b>
<b>4</b>	<b>APÉNDICES</b>	<b>15</b>



# 1 Introducción

La presente Especificación de Requisitos de Software (SRS) define de forma clara, completa y estructurada los requisitos funcionales y no funcionales para el desarrollo de una Aplicación Web Progresiva (PWA) destinada a la Asociación KALLARI. Este documento se ajusta al estándar IEEE Std 830-1998 y sirve como acuerdo técnico formal entre el equipo de desarrollo y los stakeholders del proyecto.

El sistema propuesto denominado Sistema Integral de Gestión KALLARI (SIG-KALLARI) integra módulos de administración de inventario, gestión de ventas, registro de clientes y la integración de un modelo predictivo de demanda, todo ello dentro de una arquitectura web moderna basada en React (frontend), Express.js (backend) y PostgreSQL (base de datos). La solución está diseñada específicamente para operar en entornos con conectividad intermitente, aprovechando las capacidades offline de las PWAs mediante Service Workers y almacenamiento local persistente.

Esta especificación establece los cimientos para el diseño, implementación, pruebas y validación del software, garantizando que el producto final responda eficazmente a las necesidades operativas de la asociación, mejore la trazabilidad de sus transacciones y reduzca la dependencia de registros manuales o herramientas desconectadas.

El documento se organiza en secciones que abarcan desde una descripción general del sistema hasta los requisitos específicos, incluyendo interfaces, funcionalidades clave, atributos de calidad (rendimiento, seguridad, fiabilidad, etc.) y consideraciones técnicas y operativas relevantes, proporcionando así una guía integral para todas las fases del ciclo de vida del desarrollo del software.

## 1.1 Propósito

Este documento de Especificación de Requisitos de Software (SRS) define de forma clara, completa y estructurada los requisitos que debe cumplir la Aplicación Web Progresiva (PWA) destinada a la Asociación KALLARI. El sistema tiene como objetivo centralizar la administración de inventario, la gestión de ventas y la predicción de demanda mediante un modelo predictivo integrado.

Este documento sirve como contrato técnico entre el equipo de desarrollo y los stakeholders, y guía las fases de diseño, implementación, pruebas y validación del software. Está dirigido a:

- Equipo de desarrollo: para implementar la solución con React (frontend), Express (backend) y PostgreSQL (base de datos).
- Equipo de pruebas: para verificar el cumplimiento funcional y no funcional.
- Stakeholders y usuarios finales: para validar que la solución resuelve sus necesidades operativas.
- Gestores de proyecto: para planificar entregas iterativas.
- Personal de calidad: para asegurar el cumplimiento de estándares técnicos y de usabilidad.

## 1.2 Alcance

**Nombre del sistema:** "Sistema Integral de Gestión KALLARI" (SIG-KALLARI)

**Tipo de producto:** Aplicación Web Progresiva (PWA), accesible desde navegadores modernos en dispositivos móviles y de escritorio.

Módulos principales:



- Gestión de Inventario
- Gestión de ventas y clientes
- Integración de modelo predictivo de demanda
- Sincronización offline/online con base de datos central

Características clave:

- Funcionalidad offline mediante Service Workers y almacenamiento local (IndexedDB)
- Sincronización automática al recuperar conectividad
- Interfaz responsiva y optimizada para dispositivos móviles
- Arquitectura cliente-servidor con API RESTful

### 1.3 Personal involucrado

Nombre	Scarlet Cayapa
Rol	Full Stack Developer
Categoría profesional	Desarrollador Junior
Responsabilidades	Análisis de requisitos, diseño arquitectónico, desarrollo frontend/backend, pruebas, documentación
Información de contacto	scarletcayapa@gmail.com
Aprobación	Validación técnica

### 1.4 Definiciones, acrónimos y abreviaturas

Término / Acrónimo	Definición
PWA	Aplicación Web Progresiva – aplicación web que ofrece funcionalidad similar a una app nativa, incluyendo offline y notificaciones push.
SIG-KALLARI	Sistema Integral de Gestión KALLARI – nombre del sistema a desarrollar.
MVP	Producto Mínimo Viable – versión inicial con funcionalidades esenciales.
CSV	Comma-Separated Values – formato para intercambio de datos tabulares.
REST	Representational State Transfer – estilo arquitectónico para APIs web
IndexedDB	Base de datos NoSQL en el navegador para almacenamiento local en PWAs
Service Worker	Script en segundo plano que permite funcionalidad offline en PWAs

### 1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
IEEE-830-1998	IEEE Recommended Practice for Software Requirements Specifications	<a href="https://standards.ieee.org/ieee/830/1222/">https://standards.ieee.org/ieee/830/1222/</a>	1998	IEEE Standards Association



MDN-PWA	Documentación de Aplicaciones Web Progresivas	<a href="https://developer.mozilla.org/es/docs/Web/Progressive_web_apps">https://developer.mozilla.org/es/docs/Web/Progressive_web_apps</a>	2025	Mozilla Developer Network
React Docs	Documentación oficial de React	<a href="https://react.dev/">https://react.dev/</a>	2025	Meta
Express Docs	Documentación oficial de Express.js	<a href="https://expressjs.com">https://expressjs.com</a>	2025	OpenJS Foundation
PostgreSQL Docs	Documentación oficial de PostgreSQL	<a href="https://www.postgresql.org/docs">https://www.postgresql.org/docs</a>	2025	PostgreSQL Global Dev Group

## 1.6 Resumen

Este documento se estructura en tres secciones principales:

- Sección 2: Descripción general del sistema, contexto operativo, usuarios y restricciones.
- Sección 3: Requisitos específicos, tanto funcionales como no funcionales, con énfasis en la arquitectura PWA y el stack tecnológico (React, Express, PostgreSQL).
- Sección 4: Apéndices con formatos, diagramas y especificaciones complementarias.

# 2 Descripción general

## 2.1 Perspectiva del producto

SIG-KALLARI es una PWA autocontenida, diseñada para operar en entornos con conectividad limitada o intermitente, común en las zonas rurales donde opera la Asociación KALLARI. La aplicación se compone de un frontend en React, un backend en Express, y una base de datos relacional en PostgreSQL. Utiliza Service Workers e IndexedDB para permitir operación offline y sincronización diferida.

## 2.2 Funcionalidad del producto

Módulo de Gestión de Inventario:

- Registro de productos (código, nombre, categoría, precio, stock)
- Control de entradas/salidas con actualización automática de stock
- Alertas de stock mínimo
- Búsqueda por nombre o código

Módulo de Ventas y Clientes:

- Registro y gestión de clientes (nombre, identificación, contacto)
- Creación de órdenes de venta con cálculo automático de impuestos (IVA 15%)
- Historial de ventas con filtros por fecha y cliente
- Generación, firma electrónica y autorización de facturas electrónicas mediante integración con los Web Services del SRI, incluyendo la generación del comprobante en PDF.

Módulo Predictivo de Demanda:



- Análisis de productos más vendidos (unidades y valor)
- Tendencias mensuales de ventas
- Recomendaciones de reposición basadas en rotación histórica
- Alertas de baja rotación

Funcionalidad Offline:

- Operación completa sin conexión (ventas, registro de productos, etc.)
- Almacenamiento local persistente
- Sincronización automática de datos (ventas, clientes, inventario) al recuperar la conectividad, y opción de sincronización manual forzada por el usuario.

## 2.3 Características de los usuarios

Tipo de usuario	Contador
Formación	Tercer nivel
Habilidades	Ofimática, manejo básico de sistemas
Actividades	Revisa y exporta reportes de ventas e inventario, supervisa los ingresos registrados y utiliza los datos generados para la contabilidad externa.

Tipo de usuario	Vendedor
Formación	Tercer nivel
Habilidades	Uso básico de smartphones y apps
Actividades	Registra ventas, consulta inventario, atiende clientes

## 2.4 Restricciones

Técnicas:

- Frontend: React 18+ con soporte PWA (manifest, service workers)
- Backend: Express.js sobre Node.js 18+
- Base de datos: PostgreSQL 12+
- Almacenamiento offline: IndexedDB + librerías como Dexie.js
- Compatible con navegadores modernos (Chrome, Edge, Firefox, Safari)

Operativas:

- No requiere instalación; accesible vía URL
- Funciona en dispositivos móviles con Android 8+ o iOS 14+
- Requiere conexión inicial para registro y descarga de assets

De desarrollo:

- Plazo MVP: 6 meses
- Equipo: desarrollador full-stack único
- Presupuesto limitado a herramientas open-source

## 2.5 Suposiciones y dependencias

Suposiciones:

- Los usuarios tienen acceso a smartphones con navegadores actualizados
- Existe conectividad intermitente (no nula) para sincronización periódica
- El cliente (Asociación KALLARI) debe proveer un certificado digital válido (P12 o token) para la firma electrónica de comprobantes, conforme a los requisitos del SRI." "El sistema depende de la disponibilidad de los Web Services del SRI para la autorización de facturas.





Dependencias:

- Servidor backend accesible vía HTTPS
- Base de datos PostgreSQL en la nube o local con respaldo
- Librerías de análisis predictivo (e.g., TensorFlow.js o modelos preentrenados en backend)

Factores de riesgo:

- Cambios en políticas de navegadores que afecten Service Workers
- Pérdida de datos si no se sincroniza antes de limpiar caché del navegador

## 2.6 Evolución previsible del sistema

Fase 2:

- Integración con pasarelas de pago (Stripe, transferencias)
- Notificaciones push para alertas de stock o ventas
- Panel de administración web para gestión avanzada

Fase 3:

- Migración a arquitectura microservicios
- Dashboard ejecutivo con Power BI o Metabase

## 3 Requisitos específicos

### 3.1 Requisitos comunes de los interfaces

#### 3.1.1 Interfaces de usuario

- Diseño responsive (móvil primero)
- Soporte offline total en módulos de inventario y ventas
- Validación en tiempo real en formularios
- Indicador visual de estado de conexión (online/offline/sincronizando)
- Menú de navegación por módulos: Inventario, Ventas, Predicción, Reportes.

#### 3.1.2 Interfaces de hardware

- Compatible con pantallas táctiles
- Soporte opcional para lectores de códigos de barras vía cámara (API de escaneo)
- Impresión de comprobantes vía Bluetooth o red (dependiente del dispositivo).

#### 3.1.3 Interfaces de software

Software externo	Propósito del interfaz	Detalles del interfaz
PostgreSQL	Base de datos central	Conexión vía Pool de Express
IndexedDB	Almacenamiento local offline	Usado por frontend para datos pendientes
API REST (Express)	Comunicación cliente-servidor	Endpoints para productos, ventas, clientes, sincronización



TensorFlow.js (opcional)	Modelo predictivo en frontend	O bien modelo en backend con API de predicción
-----------------------------	-------------------------------------	--

### 3.1.4 Interfaces de comunicación

- HTTPS obligatorio para todas las llamadas al backend
- Formato JSON para intercambio de datos
- Mecanismo de cola para operaciones offline (sync queue).

## 3.2 Requisitos funcionales

### 3.2.1 Requisito funcional 1

Número de requisito	RF-01		
Nombre de requisito	Registro de usuarios		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Reunión con líder del proyecto		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

El sistema debe permitir al administrador registrar un usuario nuevo y asignarle un rol específico, además enviarle las credenciales por correo electrónico al usuario registrado. El registro debe validarse para evitar duplicados y guardar la fecha de creación.

### 3.2.2 Requisito funcional 2

Número de requisito	RF-02		
Nombre de requisito	Iniciar sesión		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Reunión con usuarios finales		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

El sistema debe proveer una interfaz para que los usuarios registrados se autenticuen usando su usuario (correo) y contraseña. El sistema debe validar estas credenciales contra la base de datos. Si las credenciales son correctas, el sistema debe autorizar el acceso y redirigir al usuario a la interfaz principal.

### 3.2.3 Requisito funcional 3

Número de requisito	RF-03		
Nombre de requisito	Bloqueo del sistema		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Reunión con usuarios finales		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

El sistema debe bloquear a un usuario luego de fallar los 3 intentos para iniciar sesión.



### 3.2.4 Requisito funcional 4

Número de requisito	RF-04
Nombre de requisito	Recuperar contraseña
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con usuarios finales
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir a los usuarios recuperar la contraseña mediante un código OTP enviado al correo.

### 3.2.1 Requisito funcional 5

Número de requisito	RF-05
Nombre de requisito	Editar usuario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con líder del proyecto
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir al usuario editar su información registrada.

### 3.2.1 Requisito funcional 6

Número de requisito	RF-06
Nombre de requisito	Eliminar cuenta
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con líder del proyecto
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir a un usuario (Vendedor, Contador) eliminar su propia cuenta. El sistema debe permitir a un Administrador eliminar la cuenta de cualquier usuario.

### 3.2.1 Requisito funcional 7

Número de requisito	RF-07
Nombre de requisito	Crear categoría
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con usuarios finales
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir crear una o varias categorías para los productos.

### 3.2.1 Requisito funcional 8

Número de requisito	RF-08
Nombre de requisito	Editar categoría
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con usuarios finales
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir editar una o varias categorías para los productos.



### 3.2.1 Requisito funcional 9

Número de requisito	RF-09
Nombre de requisito	Eliminar categoría
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con usuarios finales
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir eliminar una o varias categorías para los productos.

### 3.2.2 Requisito funcional 10

Número de requisito	RF-10
Nombre de requisito	Registro de productos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con usuarios finales
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir al usuario registrar nuevos productos con los siguientes campos obligatorios: código único (auto-generado o manual), nombre, precio unitario, stock inicial y categoría. El registro debe validarse para evitar duplicados y guardar la fecha de creación.

### 3.2.1 Requisito funcional 11

Número de requisito	RF-11
Nombre de requisito	Editar productos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con usuarios finales
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir editar los siguientes campos de los productos: nombre, precio unitario, stock inicial y categoría. El registro debe validarse para evitar duplicados y guardar la fecha de creación.

### 3.2.1 Requisito funcional 12

Número de requisito	RF-12
Nombre de requisito	Eliminar productos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con usuarios finales
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir eliminar uno o varios productos.

### 3.2.2 Requisito funcional 13

Número de requisito	RF-13
Nombre de requisito	Control de movimientos de inventario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación inicial del proyecto
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe registrar automáticamente las salidas (ventas) de productos, actualizando el stock disponible en tiempo real. Cada movimiento debe asociarse



a una factura o documento de compra, y se debe mantener un historial de movimientos por producto.

### 3.2.3 Requisito funcional 14

Número de requisito	RF-14
Nombre de requisito	Alerta de stock mínimo
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesidad operativa de gestión de stock
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe mostrar una alerta visual (icono o color destacado) en la interfaz cuando el stock de un producto cae por debajo del umbral definido por el usuario. Además, debe generar un reporte diario de productos en estado “bajo stock”.

### 3.2.4 Requisito funcional 15

Número de requisito	RF-15
Nombre de requisito	Creación de facturas electrónicas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Normativa SRI / Reunión con contadores
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe generar facturas electrónicas conforme al formato XML exigido por el SRI, incluyendo clave de acceso, secuencial, datos del emisor y receptor, ítems con precios e IVA, y firma digital. Debe enviar el comprobante al SRI vía web service y almacenar la respuesta de autorización.

### 3.2.5 Requisito funcional 16

Número de requisito	RF-16
Nombre de requisito	Gestión de clientes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesidad de ventas y facturación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir registrar y buscar clientes por nombre, RUC o cédula. Cada cliente debe tener campos opcionales como dirección, teléfono y correo. La búsqueda debe ser instantánea y filtrar resultados mientras se escribe.

### 3.2.6 Requisito funcional 17

Número de requisito	RF-17
Nombre de requisito	Cálculo automático de impuestos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Normativa tributaria (IVA 15%)
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Al agregar productos a una factura, el sistema debe calcular automáticamente el subtotal, el IVA (15% sobre productos gravados) y el total final. Debe permitir excluir productos de impuestos si así lo indica su categoría.

### 3.2.7 Requisito funcional 18

Número de requisito	RF-18
---------------------	-------



### Especificación de requisitos de software

Nombre de requisito	Análisis de productos más vendidos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesidad de toma de decisiones
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe generar un reporte mensual que muestre los 10 productos con mayor volumen de ventas en unidades y valor monetario. Este reporte debe ser exportable en PDF o CSV.

### 3.2.8 Requisito funcional 19

Número de requisito	RF-19
Nombre de requisito	Recomendaciones de reposición
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Optimización de inventario
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

Basado en el histórico de ventas y rotación de productos, el sistema debe sugerir cantidades mínimas de reposición para productos con alta demanda. Las recomendaciones deben considerar el período de entrega estimado.

### 3.2.9 Requisito funcional 20

Número de requisito	RF-20
Nombre de requisito	Sincronización con la Base de datos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesidad de respaldo en la nube
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe gestionar una cola de sincronización. Sincronizar automáticamente los datos locales (creados offline) con la base de datos central en cuanto detecte una conexión a internet estable. Adicionalmente, el sistema debe ofrecer un botón o función visible para que el usuario pueda activar manualmente la sincronización en cualquier momento. El sistema debe gestionar reintentos automáticos en caso de fallo.

## 3.3 Requisitos no funcionales

### 3.3.1 Requisitos de rendimiento

- Tiempo de carga inicial < 3s en 3G
- Operaciones offline instantáneas (< 300ms)
- Sincronización de 100 registros en < 5s en conexión 2G.

### 3.3.2 Seguridad

- Autenticación JWT con refresh tokens
- HTTPS obligatorio (TLS 1.3)
- Sanitización de entradas para prevenir XSS e inyecciones SQL.

### 3.3.3 Fiabilidad

- Mecanismo de reintento y cola persistente para operaciones offline
- Tasa de pérdida de datos < 0.1% tras fallos de sincronización.



### 3.3.4 Disponibilidad

- Funcionalidad offline 100% operativa
- Backend con uptime  $\geq 99\%$ .

### 3.3.5 Mantenibilidad

- Código modular en React y Express
- Documentación de API con OpenAPI/Swagger
- Scripts de migración de base de datos versionados.

### 3.3.6 Portabilidad

- Compatible con Chrome, Firefox, Safari, Edge
- Instalable como app en Android/iOS
- Código 100% web estándar (sin dependencias nativas).

## 3.4 Otros requisitos

### 3.4.1 Requisitos legales

- El sistema debe cumplir con la Normativa del SRI para Facturación Electrónica (Resolución 024-2023 y actualizaciones).
- Los comprobantes autorizados deben almacenarse en formato XML y PDF durante 5 años, conforme al Art. 97 del Código Orgánico Tributario.

### 3.4.2 Requisitos culturales

- La interfaz debe estar completamente en español (Ecuador).
- Fechas en formato dd/mm/aaaa, moneda en dólares estadounidenses (USD).

## 4 Apéndices

**Apéndice A:** Diagrama de arquitectura (frontend React PWA + backend Express + PostgreSQL)

**Apéndice B:** Especificación de API REST (endpoints, métodos, ejemplos de request/response)

**Apéndice C:** Esquema de base de datos relacional (tablas: productos, clientes, ventas, movimientos, usuarios)

**Apéndice D:** Estrategia de sincronización offline/online (algoritmo de reconciliación)