

Informe de Requisitos de Software (SRS) para la Aplicación de Automatización de Entregas RPP

Autor: Manus AI

Fecha: 29 de agosto de 2025

1. Introducción

Este Informe de Requisitos de Software (SRS) detalla las funcionalidades y restricciones de la aplicación de automatización de entregas de Raciones para Preparar (RPP) en La Guajira, Colombia. El objetivo principal de esta aplicación es optimizar y digitalizar el proceso manual de gestión de entregas y generación de informes que actualmente realizan los trabajadores sociales, mejorando la eficiencia, la precisión y la transparencia en la distribución de la asistencia alimentaria. Este documento servirá como una guía fundamental para el equipo de desarrollo, asegurando que el producto final cumpla con las expectativas de los usuarios y las necesidades del negocio.

2. Propósito del Documento

El propósito de este SRS es:

- Definir de manera clara y concisa los requisitos funcionales y no funcionales de la aplicación.
- Servir como un acuerdo entre los stakeholders (usuarios, equipo de desarrollo, gerencia) sobre el alcance y las características del sistema.
- Proporcionar una base para el diseño, desarrollo, pruebas y mantenimiento del software.
- Facilitar la comunicación y la comprensión mutua entre todas las partes involucradas en el proyecto.

3. Alcance del Producto

La aplicación de automatización de entregas RPP abarcará dos componentes principales: una aplicación móvil para trabajadores sociales y una plataforma web para coordinadores de centros zonales y supervisores regionales. El sistema permitirá la gestión integral del

ciclo de vida de las entregas de RPP, desde el registro de beneficiarios y la captura de evidencias en campo, hasta la generación automatizada de informes y el análisis de datos.

4. Usuarios y Características del Sistema

El sistema está diseñado para ser utilizado por diferentes perfiles de usuario, cada uno con roles y permisos específicos:

4.1. Trabajador Social (Usuario Móvil)

- **Perfil:** Personal de campo encargado de la entrega directa de RPP a los beneficiarios en zonas rurales.
- **Necesidades Clave:** Facilidad de uso, operación offline, captura rápida de datos y evidencias.
- **Funcionalidades Principales:**
 - **Gestión de Beneficiarios:**
 - Visualizar lista de beneficiarios asignados a su centro/ruta.
 - Buscar beneficiarios por nombre o número de identificación.
 - Acceder al perfil de cada beneficiario (nombre, identificación, grupo poblacional, historial de entregas).
 - **Registro de Entregas:**
 - Seleccionar beneficiario para registrar una entrega.
 - Confirmar la ración correspondiente al grupo poblacional del beneficiario.
 - Registrar la fecha y hora de la entrega.
 - Capturar evidencia fotográfica de la entrega (beneficiario con el paquete).
 - Asociar automáticamente la foto al registro de entrega.
 - Registrar la firma o huella del responsable de la recepción (opcional).
 - Guardar el registro de entrega localmente en el dispositivo.
 - **Sincronización Offline:**
 - Realizar todas las operaciones de registro sin conexión a internet.
 - Sincronizar automáticamente los datos y fotos pendientes con el servidor central cuando se detecte conexión.
 - Visualizar el estado de sincronización de los registros.
 - **Validación de Datos:**

- Validación en tiempo real de la entrada de datos (ej. formato de identificación, cantidades de ración).
- Advertencias para datos fuera de rango o inconsistencias.

4.2. Coordinador de Centro Zonal (Usuario Web)

- **Perfil:** Personal administrativo encargado de supervisar las operaciones en un centro zonal específico.
- **Necesidades Clave:** Supervisión de actividades, generación de informes detallados, análisis a nivel de centro.
- **Funcionalidades Principales:**
 - **Gestión de Usuarios:**
 - Administrar cuentas de trabajadores sociales asignados a su centro.
 - Asignar beneficiarios y rutas a trabajadores sociales.
 - **Visualización de Datos:**
 - Acceder a datos consolidados de entregas por centro, trabajador social y período.
 - Visualizar evidencias fotográficas de las entregas.
 - **Generación de Informes:**
 - Generar automáticamente el "Formato de Evidencias RPP" para un período y centro específico.
 - Generar reportes de "Encuentro Comunitario" (si aplica).
 - Exportar informes en formatos PDF o Word.
 - **Análisis Básico:**
 - Visualizar métricas básicas de entregas (número de entregas, beneficiarios atendidos).

4.3. Supervisor Regional (Usuario Web)

- **Perfil:** Personal de alto nivel encargado de la supervisión y toma de decisiones estratégicas a nivel regional.
- **Necesidades Clave:** Visión macro, análisis de tendencias, KPIs, informes ejecutivos.
- **Funcionalidades Principales:**
 - **Visión General:**
 - Acceder a paneles de control (dashboards) con KPIs agregados a nivel regional.

- Visualizar el progreso general de los programas de RPP en su región.
- **Análisis Avanzado:**
 - Herramientas de análisis de datos para identificar tendencias, patrones y áreas de mejora.
 - Generar informes ejecutivos personalizados.
- **Gestión de Usuarios (Regional):**
 - Administrar cuentas de coordinadores de centros zonales.
 - Asignar centros zonales a coordinadores.

5. Requisitos Funcionales

Los requisitos funcionales describen las funciones que el sistema debe realizar. Se agrupan por módulos principales:

5.1. Módulo de Gestión de Usuarios y Roles

- **RF1.1:** El sistema debe permitir la creación, edición y eliminación de cuentas de usuario (Trabajador Social, Coordinador, Supervisor).
- **RF1.2:** El sistema debe asignar roles y permisos específicos a cada tipo de usuario.
- **RF1.3:** El sistema debe permitir la asignación de centros/rutas/beneficiarios a los trabajadores sociales por parte de los coordinadores.
- **RF1.4:** El sistema debe permitir la asignación de centros zonales a los coordinadores por parte de los supervisores.
- **RF1.5:** El sistema debe implementar un sistema de autenticación seguro (login/logout).

5.2. Módulo de Gestión de Beneficiarios

- **RF2.1:** El sistema debe permitir la importación masiva de datos de beneficiarios desde archivos estructurados (ej. Excel).
- **RF2.2:** El sistema debe permitir la visualización y búsqueda de beneficiarios por nombre, identificación y grupo poblacional.
- **RF2.3:** El sistema debe mostrar el historial de entregas de RPP para cada beneficiario.
- **RF2.4:** El sistema debe manejar casos de números de identificación "SD" (Sin Dato) mediante la asignación de un identificador único temporal.

5.3. Módulo de Registro de Entregas (Aplicación Móvil)

- **RF3.1:** El sistema debe permitir al trabajador social seleccionar un beneficiario de su lista asignada.
- **RF3.2:** El sistema debe pre-cargar la información de la ración correspondiente al grupo poblacional del beneficiario.
- **RF3.3:** El sistema debe permitir la captura de la fecha y hora de la entrega.
- **RF3.4:** El sistema debe permitir la captura de una o varias fotografías como evidencia de la entrega.
- **RF3.5:** El sistema debe asociar automáticamente las fotografías al registro de entrega y al beneficiario.
- **RF3.6:** El sistema debe permitir el registro de la firma o huella del responsable de la recepción.
- **RF3.7:** El sistema debe guardar los registros de entrega localmente en el dispositivo móvil.
- **RF3.8:** El sistema debe validar la entrada de datos en tiempo real y alertar sobre inconsistencias.

5.4. Módulo de Sincronización Offline

- **RF4.1:** El sistema debe permitir la operación completa de registro de entregas sin conexión a internet.
- **RF4.2:** El sistema debe almacenar de forma segura los datos y evidencias fotográficas localmente en el dispositivo.
- **RF4.3:** El sistema debe sincronizar automáticamente los datos pendientes con el servidor central cuando se detecte una conexión a internet estable.
- **RF4.4:** El sistema debe resolver conflictos de datos durante la sincronización (ej. "last write wins" o reglas de resolución predefinidas).
- **RF4.5:** El sistema debe mostrar el estado de sincronización de los datos al usuario.

5.5. Módulo de Generación de Informes

- **RF5.1:** El sistema debe generar automáticamente el "Formato de Evidencias RPP" en formato PDF o Word, incluyendo datos de beneficiarios, entregas y evidencias fotográficas.
- **RF5.2:** El sistema debe generar reportes de "Encuentro Comunitario" (si aplica) en formato PDF o Word.
- **RF5.3:** El sistema debe permitir la selección de filtros (centro, período, trabajador social) para la generación de informes.

- **RF5.4:** El sistema debe permitir la exportación de datos en formatos comunes (CSV, Excel) para análisis externos.

5.6. Módulo de Análisis y Dashboards

- **RF6.1:** El sistema debe proporcionar dashboards interactivos con KPIs relevantes para coordinadores y supervisores.
- **RF6.2:** El sistema debe permitir la visualización de métricas agregadas por centro, región y período.
- **RF6.3:** El sistema debe ofrecer herramientas básicas de análisis de datos para identificar tendencias y patrones.

6. Requisitos No Funcionales

Los requisitos no funcionales describen las cualidades del sistema y las restricciones bajo las cuales debe operar.

6.1. Rendimiento

- **RNF6.1.1:** La aplicación móvil debe tener un tiempo de respuesta inferior a 2 segundos para la mayoría de las operaciones offline.
- **RNF6.1.2:** La sincronización de datos debe completarse en un tiempo razonable (ej. menos de 5 minutos para 100 registros y 50 fotos) una vez que se establece la conexión.
- **RNF6.1.3:** La generación de informes complejos debe completarse en menos de 30 segundos.
- **RNF6.1.4:** El sistema debe soportar al menos 1000 usuarios concurrentes en la plataforma web sin degradación significativa del rendimiento.

6.2. Escalabilidad

- **RNF6.2.1:** El sistema debe ser capaz de escalar para soportar un crecimiento futuro en el número de beneficiarios (millones), trabajadores sociales (miles) y volumen de datos (terabytes de fotos).
- **RNF6.2.2:** La arquitectura debe permitir la adición de nuevas funcionalidades y módulos sin requerir una re-arquitectura mayor.

6.3. Seguridad

- **RNF6.3.1:** El sistema debe implementar autenticación y autorización robustas para todos los usuarios.

- **RNF6.3.2:** Todos los datos sensibles (información de beneficiarios, fotos) deben ser encriptados en tránsito y en reposo.
- **RNF6.3.3:** El sistema debe cumplir con la Ley 1581 de 2012 de protección de datos personales en Colombia.
- **RNF6.3.4:** Se deben realizar auditorías de seguridad periódicas para identificar y mitigar vulnerabilidades.
- **RNF6.3.5:** El sistema debe implementar mecanismos de prevención de acceso no autorizado y manipulación de datos.

6.4. Usabilidad

- **RNF6.4.1:** La interfaz de usuario de la aplicación móvil debe ser intuitiva y fácil de usar para trabajadores sociales con alfabetización digital limitada.
- **RNF6.4.2:** El flujo de trabajo de registro de entregas debe ser lineal y requerir un mínimo de pasos.
- **RNF6.4.3:** La aplicación debe utilizar iconos y elementos visuales claros para facilitar la navegación.
- **RNF6.4.4:** Se debe proporcionar documentación clara y concisa, así como material de capacitación.

6.5. Fiabilidad

- **RNF6.5.1:** El sistema debe tener una disponibilidad del 99.9% para la plataforma web y una alta tolerancia a fallos para la aplicación móvil (operación offline).
- **RNF6.5.2:** El sistema debe implementar mecanismos de respaldo y recuperación de datos para prevenir la pérdida de información.
- **RNF6.5.3:** La sincronización de datos debe garantizar la integridad y consistencia de la información.

6.6. Mantenibilidad

- **RNF6.6.1:** El código fuente debe ser modular, bien documentado y fácil de entender.
- **RNF6.6.2:** El sistema debe ser fácil de mantener y actualizar, permitiendo la implementación de nuevas funcionalidades y corrección de errores de manera eficiente.

6.7. Compatibilidad

- **RNF6.7.1:** La aplicación móvil debe ser compatible con las últimas versiones de Android e iOS.

- **RNF6.7.2:** La plataforma web debe ser compatible con los navegadores web modernos (Chrome, Firefox, Edge, Safari).

7. Requisitos de Interfaz

7.1. Interfaz de Usuario (UI)

- **Aplicación Móvil:** Interfaz limpia, minimalista, con botones grandes y claros, y un flujo de navegación intuitivo. Uso de elementos visuales para indicar el estado de la sincronización y la validación de datos.
- **Plataforma Web:** Diseño responsive, paneles de control interactivos, tablas de datos filtrables y exportables, y formularios claros para la gestión de usuarios e informes.

7.2. Interfaces de Comunicación

- **API RESTful:** Para la comunicación entre la aplicación móvil/web y el backend. Debe ser segura y bien documentada.
- **Integración con Servicios de Almacenamiento en la Nube:** Para el almacenamiento de fotos (Amazon S3, Google Cloud Storage, Firebase Storage).
- **Possible Integración con Sistemas del ICBF:** A través de APIs o intercambio de archivos, si es requerido en el futuro.

8. Requisitos de Datos

8.1. Entidades de Datos Clave

- **Usuarios:** ID, nombre, rol, credenciales de acceso, centro/región asignada.
- **Centros Operativos:** ID, nombre, ubicación (regional, centro zonal, municipio), modalidad, nombre del servicio, nombre EAS, nombre UDS, dirección, código de punto de entrega.
- **Beneficiarios:** ID, centro operativo ID (FK), nombres y apellidos, número de identificación, fecha de ingreso, grupo poblacional.
- **Entregas:** ID, beneficiario ID (FK), fecha de entrega, mes de entrega, tipo de ración, firma/huella responsable, nombre responsable, identificación responsable.
- **Productos Entregados:** ID, entrega ID (FK), producto, cantidad entregada, lote.
- **Evidencias Fotográficas:** ID, entrega ID (FK), ruta de archivo (URL), fecha de captura, estado de sincronización.

8.2. Volúmenes de Datos

- **Beneficiarios:** Miles a millones de registros.
- **Entregas:** Cientos de miles a millones de registros anuales.
- **Fotos:** Terabytes de almacenamiento a lo largo del tiempo.

9. Requisitos de Rendimiento y Escalabilidad (Detallado)

Para asegurar que el software sea super rápido y de alto rendimiento, se implementarán las siguientes estrategias:

- **Optimización de Base de Datos:** Uso de índices, optimización de consultas, particionamiento de tablas si es necesario.
- **Caching:** Implementación de mecanismos de caché en el backend y en la aplicación móvil para reducir la latencia y la carga de la base de datos.
- **Procesamiento Asíncrono:** Uso de colas de mensajes y procesamiento asíncrono para tareas que no requieren respuesta inmediata (ej. subida de fotos, generación de informes complejos).
- **Balanceo de Carga:** Distribución del tráfico entre múltiples servidores para manejar un alto volumen de solicitudes.
- **Microservicios:** Arquitectura basada en microservicios para permitir el escalado independiente de componentes y una mayor resiliencia.
- **Redes de Entrega de Contenido (CDN):** Para la distribución eficiente de las evidencias fotográficas y otros activos estáticos.

10. Tecnologías Sugeridas

Basado en los requisitos y las tendencias del mercado, se sugieren las siguientes tecnologías:

- **Frontend Móvil:** React Native (con Expo para desarrollo rápido) o Flutter. Ambas ofrecen una excelente experiencia de desarrollo multiplataforma y buen soporte para funcionalidades nativas como la cámara y el almacenamiento offline.
 - **Sincronización Offline:** WatermelonDB (para React Native) o Moor (para Flutter) para la gestión de bases de datos locales y sincronización robusta.
- **Backend:**
 - **Framework:** Node.js con Express.js o Python con FastAPI (para APIs de alto rendimiento) o Django (para desarrollo más rápido con ORM).

- **Base de Datos:** PostgreSQL (relacional, robusta, escalable, buen soporte para JSONB para flexibilidad).
- **ORM/ODM:** SQLAlchemy (para Python) o Prisma (para Node.js) para la interacción con la base de datos.
- **Almacenamiento de Archivos:** Amazon S3, Google Cloud Storage o Firebase Storage para el almacenamiento escalable y seguro de las evidencias fotográficas.
- **Procesamiento Asíncrono/Colas:** Redis (para caching y colas de tareas) o RabbitMQ.
- **Generación de Informes:** Librerías de generación de PDF/Word en el backend (ej. `python-docx`, `ReportLab` para Python, o soluciones basadas en HTML a PDF como Puppeteer con Node.js).
- **Despliegue:** Contenedores (Docker) y orquestación (Kubernetes) para un despliegue escalable y gestionable. Plataformas de nube como AWS, Google Cloud Platform o Azure.

11. Consideraciones de Diseño y Arquitectura (Visión General)

La arquitectura del sistema será una arquitectura de microservicios, lo que permitirá la independencia de los componentes, facilitando el desarrollo, despliegue y escalado. Se seguirá un enfoque API-first, donde todas las interacciones entre el frontend y el backend se realizarán a través de APIs RESTful bien definidas. La seguridad será un pilar fundamental, con autenticación basada en tokens (JWT) y encriptación de extremo a extremo. El diseño offline-first de la aplicación móvil será clave, con una base de datos local que se sincronizará de forma inteligente con la base de datos central. Se implementarán mecanismos de monitoreo y logging para asegurar la observabilidad del sistema y facilitar la detección y resolución de problemas.

12. Plan de Pruebas (Visión General)

Se implementará un plan de pruebas integral que incluirá:

- **Pruebas Unitarias:** Para verificar la funcionalidad de componentes individuales.
- **Pruebas de Integración:** Para asegurar que los diferentes módulos y servicios interactúen correctamente.
- **Pruebas Funcionales:** Para validar que el sistema cumple con los requisitos funcionales definidos.

- **Pruebas de Rendimiento y Carga:** Para evaluar la capacidad del sistema bajo diferentes volúmenes de usuarios y datos.
- **Pruebas de Seguridad:** Para identificar vulnerabilidades y asegurar la protección de datos.
- **Pruebas de Usabilidad:** Con usuarios reales (trabajadores sociales) para asegurar una experiencia de usuario óptima.
- **Pruebas Offline/Sincronización:** Para validar la robustez de la funcionalidad offline y la fiabilidad de la sincronización de datos.

13. Requisitos de Mantenimiento y Soporte

- **RM13.1:** El sistema debe contar con un equipo de soporte técnico disponible para resolver incidencias y consultas de los usuarios.
- **RM13.2:** Se deben establecer procedimientos claros para la gestión de errores y la resolución de problemas.
- **RM13.3:** Se deben planificar actualizaciones periódicas del software para introducir nuevas funcionalidades, corregir errores y mejorar el rendimiento.

14. Glosario

- **API:** Interfaz de Programación de Aplicaciones.
- **CDN:** Red de Entrega de Contenido.
- **ICBF:** Instituto Colombiano de Bienestar Familiar.
- **KPI:** Indicador Clave de Rendimiento.
- **ORM:** Mapeo Objeto-Relacional.
- **RPP:** Ración Para Preparar.
- **SRS:** Informe de Requisitos de Software.
- **UI:** Interfaz de Usuario.
- **UX:** Experiencia de Usuario.

15. Referencias

- Estudio de Mercado para la Aplicación de Automatización de Entregas RPP en La Guajira (documento previo).
- Documentación de React Native, Flutter, Node.js, FastAPI, Django, PostgreSQL, WatermelonDB, Moor, Amazon S3, Google Cloud Storage, Firebase Storage, etc.

Este SRS es un documento vivo y puede ser actualizado a medida que el proyecto avance y se obtenga nueva información.