



**Gobierno Bolivariano**  
de Venezuela

Ministerio del Poder Popular  
para la Educación Universitaria

Universidad Nacional Experimental  
para las Telecomunicaciones e Informática (UNETI)



**REPÚBLICA BOLIVARIANA DE VENEZUELA**

**UNIVERSIDAD NACIONAL EXPERIMENTAL**

**PARA LAS TELECOMUNICACIONES E INFORMÁTICA**

**VICERRECTORADO ACADÉMICO**

**PROGRAMA NACIONAL DE FORMACIÓN EN INGENIERÍA EN INFORMÁTICA**

# **Documentación del proyecto**

**Winder Rivas**  
C.I: 22.029.261

**Caracas, diciembre del 2025**



# Contenido

1. Gestión del Proyecto .....	3
1.1 Plan de Desarrollo de Software .....	3
1.2 Planes de Iteraciones.....	4
1.3 Lista de Riesgos y estrategia de mitigación.....	5
1.4 Glosario del Proyecto.....	6
2. Requisitos .....	7
2.1 Documento de Especificaciones del Sistema (DES).....	7
2.2 Lista de Requisitos Priorizados del Usuario (Product Backlog) .....	9
2.3 Lista de Tareas por Iteración o Sprint (Sprint Backlog).....	10
2.4 Documento de Gestión de Requisitos por Iteración .....	11
2.5 Especificación de Requisitos de Software (ERS) / Modelo de Casos de Uso.....	13
Requisitos Funcionales.....	16
Requisitos No Funcionales.....	18
3. Análisis y Diseño.....	19
3.1 Vista de Casos de Uso a Implementar.....	19
3.2 Descripción Detallada de Arquitectura .....	19
3.3 Modelo de Análisis .....	22
3.4 Modelo de Diseño.....	23
3.5 Modelo de Datos / Diccionario de Datos .....	25
4. Implementación .....	27
4.1 Documento de Arquitectura del Sistema .....	27
4.2 Manual de Operaciones.....	30
4.3 Código Fuente y Versiones Ejecutables .....	32
5 pruebas.....	34
5.1 Plan de Pruebas a Aplicar por Iteración o Sprint .....	34
5.2 Casos de Pruebas a Aplicar por Iteración o Sprint.....	35
5.3 Informes de Pruebas Realizadas por Iteración o Sprint.....	37
5.4 Pruebas Unitarias.....	39
6. Capacitación.....	40
6.1 Plan de Capacitación y Contenido Programático.....	40
6.2 Manual de Instalación .....	42
6.3 Manual de Usuarios .....	44



## 1. Gestión del Proyecto

### 1.1 Plan de Desarrollo de Software

El Objetivo general del proyecto se centra en diseñar e implementar una base de datos centralizada que consolide la información operativa de Movilnet C.A., junto con un sistema automatizado de generación de reportes en tiempo real, mejorando la eficiencia operativa y la toma de decisiones.

El proyecto seguirá una metodología estructurada en 6 fases principales, con entregas parciales cada 2-4 semanas. Se prioriza la integración temprana de componentes críticos para validación continua con el usuario final.

#### Fases Principales:

Fase	Nombre	Duración	Entregables Clave
Fase 1	Análisis y Diseño	2 semanas	Esquema BD, Requerimientos, Prototipo UI
Fase 2	Desarrollo de Base de Datos	3 semanas	BD Google Sheets, Scripts de migración, Validaciones
Fase 3	Desarrollo de Interfaz Web	4 semanas	App HTML/JS, Integración con Google Apps Script, Formularios funcionales
Fase 4	Módulos de Reportes	3 semanas	Reportes automatizados, Dashboards básico
Fase 5	Seguridad y Despliegue	2 semanas	Roles de acceso, Red interna, Capacitación inicial
Fase 6	Mantenimiento Continuo	Indefinido	Monitoreo, Optimización, Mejoras incrementales

#### Tecnologías Empleadas:

BackEnd: Google Apps Script

FrontEnd: HTML5, CSS3, JavaScript (ES6+)

Base de Datos: Google Sheet (transición)

Herramientas: Google Workspace, Visual Studio Code



## 1.2 Planes de Iteraciones

En el siguiente cuadro se representa las iteraciones por las que pasara el proyecto junto a sus actividades, entregables y el tiempo en el que se estima que serán entregadas/evaluadas según los planificado con la institución:

Iteración	Objetivo	Actividades	Entregables	Tiempo Promedio
1	Validar estructura de datos y diseño visual.	Análisis de formularios OSAC y Canal Virtual Diseño de esquema relacional preliminar Creación de prototipo HTML/CSS con navegación básica Definición de mapeo de campos entre UI y BD	Documento de especificaciones técnicas Prototipo estático en HTML Mapeo inicial de datos (MASTER_FIELD_MAPPING)	Semanas 1-2
2	Sistema básico de registro operativo.	Desarrollo de funciones guardarRegistroOSAC() y guardarRegistroCanal() Implementación de validaciones frontend y backend Conexión con Google Sheets como almacenamiento temporal Autocompletado de datos con buscarLinea()	Formularios completamente funcionales Sistema de guardado en Google Sheets Validaciones de campos obligatorios y formatos	Semanas 3-5
3	Mejora de rendimiento y despliegue controlado	Optimización de consultas con índices Implementación de caché para datos estáticos Configuración de seguridad por roles Capacitación a usuarios pilotos	Sistema optimizado para 100+ usuarios concurrentes Documentación de usuario final Plan de rollback en caso de fallos	Semanas 9-10
4	Ampliar funcionalidades y preparar para escalamiento	Implementación de módulo de consultas avanzadas Desarrollo de dashboard ejecutivo Integración con sistemas existentes (si aplica) Pruebas de carga y estrés	Dashboard de métricas en tiempo real Sistema preparado para escalamiento nacional Documentación técnica completa	Semanas 11-14



### 1.3 Lista de Riesgos y estrategia de mitigación

Riesgo	Impacto	Probabilidad	Estrategia de Mitigación
Inconsistencias en datos históricos	Medio	Alta	<ul style="list-style-type: none"> <li>Validación en 3 fases (migración, limpieza, verificación)</li> <li>Mantener backup original durante 3 meses</li> </ul>
Límites de Google Apps Script	Medio	Media	<ul style="list-style-type: none"> <li>Optimizar llamadas por lotes</li> <li>Implementar tiempo de espera exponencial</li> <li>Plan B: migrar a Node.js si es necesario</li> </ul>
Problemas de conectividad BD	Crítico	Baja	<ul style="list-style-type: none"> <li>Conexiones con reintentos automáticos</li> <li>Modo offline temporal en Google Sheets</li> <li>Monitoreo 24/7 de conectividad</li> </ul>
Resistencia al cambio	Menor	Media	<ul style="list-style-type: none"> <li>Comunicación temprana y frecuente</li> <li>Involucramiento de usuarios clave en diseño</li> </ul>
Falta de capacitación	Medio	Alta	<ul style="list-style-type: none"> <li>Plan de capacitación por roles</li> <li>Videos tutoriales y guías paso a paso</li> <li>Acompañamiento a la gestión en el primer mes</li> </ul>
Accesos no autorizados	Crítico	Baja	<ul style="list-style-type: none"> <li>Autenticación de dos factores</li> <li>Acceso solo por red corporativa</li> <li>Auditorías semanales de logs</li> </ul>
Pérdida de datos	Crítico	Baja	<ul style="list-style-type: none"> <li>Backup diario/semanal automático</li> <li>Réplica en tiempo real</li> <li>Recuperación desastrosa probada mensualmente</li> </ul>
Sobrecarga en horario pico	Medio	Media	<ul style="list-style-type: none"> <li>Escalado automático de recursos</li> <li>Colas de procesamiento asíncrono</li> <li>Monitoreo de rendimiento en tiempo real</li> </ul>



## 1.4 Glosario del Proyecto

Términos del código	
Término	Significado
Google Apps Script	Plataforma de desarrollo en la nube de Google para automatizar procesos y crear aplicaciones web
MASTER_FIELD_MAPPING	Objeto JavaScript que define la relación entre campos del formulario y columnas en la base de datos
Datos Operativos	Nombre de la hoja principal en Google Sheets que almacena todos los registros del sistema
doGet()	Función especial en Apps Script que maneja solicitudes HTTP GET y sirve la interfaz web
KPI	Key Performance Indicator (Indicador Clave de Desempeño)
BD	Base de Datos

Términos de Formulario:	
Término	Significado
Campos Obligatorios	Datos que deben completarse para que el registro sea válido (indicados con *)
Autocompletado	Función que busca y completa datos basados en la línea telefónica ingresada
Validación en Tiempo Real	Verificación inmediata de formatos y consistencia durante el llenado
Toggle Buttons	Botones de selección exclusiva para campos como Género o Atención Preferencial
Registro Completo	Caso que no requiere escalado y se considera finalizado
Registro Incompleto	Caso que requiere seguimiento o escalado a segundo nivel
Consulta en Tiempo Real	Búsqueda inmediata en base de datos con resultados instantáneos
Reporte Automatizado	Documento generado automáticamente con datos actualizados del sistema
Acceso por Red Corporativa	Restricción que limita el uso del sistema solo desde la red interna de Movilnet
Roles y Permisos	Niveles de acceso diferenciados según función del usuario
Datos Sensibles	Información personal protegida (cédula, teléfono, email)
Sesión Activa	Período durante el cual un usuario puede interactuar con el sistema sin Reautenticarse



Términos de Dominio (Movilnet):	
Término	Significado
OSAC	Oficina de Servicio y Atención al Cliente - Canal de atención presencial
Canal Virtual	Canales digitales de atención (chat web, redes sociales, email)
Línea a Gestionar	Identificador completo de línea telefónica (prefijo + 7 dígitos)
Gestor/Ejecutivo	Empleado de Movilnet que atiende al cliente y registra la gestión
Escalado	Proceso de derivar un caso a un nivel superior de soporte (SI/NO)
Prefijo	Código numérico inicial que identifica región o tipo de servicio (ej.: 416)

## 2. Requisitos

### 2.1 Documento de Especificaciones del Sistema (DES)

#### Propósito del Sistema

El Sistema de Gestión Movilnet es una plataforma web que centraliza el registro de atenciones al cliente a través de los canales OSAC (presencial) y Canal Virtual (digital), automatiza la generación de reportes y consolida datos operativos para análisis y toma de decisiones.

#### Alcance del Sistema

El sistema incluye:

- Registro de atenciones en tiempo real
- Consulta histórica de gestiones
- Generación automatizada de reportes
- Dashboard de indicadores clave (KPIs)
- Gestión de usuarios y permisos
- Integración con Google Sheets (transicional)
- Base de datos PostgreSQL (definitiva)



El sistema excluye:

- Sistema de facturación
- Gestión de inventario
- Comunicación con sistemas externos de terceros
- Aplicación móvil nativa

### **Arquitectura General**

- Frontend: Aplicación web responsiva (HTML, CSS, JavaScript)
- Backend: Google Apps Script (fase inicial), PostgreSQL (fase final)
- Almacenamiento: Google Sheets (transitorio), PostgreSQL (producción)
- Seguridad: Autenticación corporativa, acceso por red interna

### **Restricciones Técnicas**

- Debe funcionar en navegadores modernos (Chrome 80+, Firefox 75+)
- Compatible con dispositivos móviles y tabletas
- Tiempo de respuesta < 3 segundos para operaciones críticas
- Disponibilidad del 99% en horario laboral





## 2.2 Lista de Requisitos Priorizados del Usuario (Product Backlog)

Épica	ID	Requisito	Prioridad	Sprint Objetivo
<b>Épica 1: Gestión de Registros de Atención</b>	R01	Como ejecutivo, necesito registrar atenciones OSAC para documentar la gestión	Alta	Sprint 1-2
	R02	Como ejecutivo, necesito registrar atenciones por Canal Virtual para gestiones digitales	Alta	Sprint 1-2
	R03	Como ejecutivo, quiero autocompletar datos al ingresar una línea para reducir errores	Alta	Sprint 2
	R04	Como ejecutivo, necesito validaciones en tiempo real para evitar registros incorrectos	Media	Sprint 1
	R05	Como ejecutivo, quiero ver un resumen del registro antes de guardar	Baja	Sprint 3
<b>Épica 2: Consultas y Búsquedas</b>	R06	Como supervisor, necesito buscar registros por línea telefónica	Alta	Sprint 3
	R07	Como supervisor, quiero filtrar registros por fecha, ejecutivo o tipo	Alta	Sprint 3
	R08	Como gerente, necesito ver el histórico completo de un cliente	Media	Sprint 4
	R09	Como ejecutivo, quiero consultar mis registros del día	Baja	Sprint 4
<b>Épica 3: Reportes Automatizados</b>	R10	Como gerente, necesito reporte diario de atenciones completadas	Alta	Sprint 4
	R11	Como supervisor, quiero reporte de casos escalados pendientes	Alta	Sprint 4
	R12	Como analista, necesito reporte mensual por tipo de gestión	Media	Sprint 4
	R13	Como director, quiero dashboard con KPIs en tiempo real	Media	Sprint 4
<b>Épica 4: Gestión de Usuarios y Seguridad</b>	R14	Como administrador, necesito gestionar usuarios y permisos	Media	Sprint 3
	R15	Como sistema, debo restringir acceso por red corporativa	Alta	Sprint 2
	R16	Como usuario, quiero cerrar sesión de manera segura	Baja	Sprint 1
<b>Épica 5: Integración y Datos</b>	R18	Como sistema, necesito backup automático diario	Media	Sprint 3
	R19	Como administrador, quiero exportar datos a Excel	Baja	Sprint 3



## 2.3 Lista de Tareas por Iteración o Sprint (Sprint Backlog)

Fase	Objetivo	Tareas	Responsable	Estimación	Estado
<b>Sprint 1: Registro Básico y Estructura (2 semanas)</b>	<b>Formulario OSAC funcional con guardado básico</b>	Diseñar interfaz formulario OSAC	Frontend	16h	Realizado
		Implementar navegación entre secciones	Frontend	8h	Realizado
		Crear función doGet() para servir HTML	Backend	4h	Realizado
		Configurar hoja Google Sheets "Datos Operativos"	Backend	4h	Realizado
		Implementar función guardarRegistroOSAC()	Backend	12h	Realizado
		Validaciones básicas de campos obligatorios	Frontend	8h	Realizado
		Estilos responsivos para móviles	Frontend	12h	Realizado
<b>Sprint 2: Formulario Canal Virtual y Validaciones (2 semanas)</b>	<b>Segundo formulario funcional con validaciones avanzadas</b>	Diseñar formulario Canal Virtual	Frontend	12h	Realizado
		Adaptar función guardarRegistroCanal()	Backend	8h	Realizado
		Implementar autocompletado con buscarLinea()	Backend	16h	Realizado
		Validación de número telefónico (7 dígitos)	Frontend	6h	Realizado
		Lógica de escalado automático (SI/NO)	Frontend	8h	Realizado
		Selectores dinámicos (motivo → detalle)	Frontend	10h	Realizado
		Restringir acceso por red corporativa	Backend	6h	Por Aprobar
<b>Sprint 3: Consultas y Búsquedas (3 semanas)</b>	<b>Módulo de consultas operativo</b>	Diseñar interfaz de consultas	Frontend	16h	Pendiente
		Implementar búsqueda por línea	Backend	12h	Pendiente
		Crear filtros por fecha y ejecutivo	Frontend	14h	Pendiente
		Optimizar rendimiento de consultas	Backend	10h	Pendiente



<b>Sprint 4: Reportes Básicos (3 semanas)</b>	<b>Primeros reportes automatizados</b>	Diseñar reporte diario de atenciones	Frontend	16h	Pendiente
		Implementar reporte de casos escalados	Backend	14h	Pendiente
		Diseñar dashboard básico	Frontend	20h	Pendiente
		Configurar generación automática (cron)	Backend	8h	Pendiente
		Pruebas de generación de reportes	QA	10h	Pendiente

## 2.4 Documento de Gestión de Requisitos por Iteración

### Sprint 1: Gestión de Requisitos

Requisitos Implementados:

- R01: Registro OSAC funcional
- R04: Validaciones en tiempo real
- R16: Cierre de sesión seguro

Cambios y Ajustes:

- Original: Campos de ubicación complejos
- Ajustado: Simplificado a estado/ciudad con carga dinámica
- Justificación: Reducir tiempo de registro manteniendo datos esenciales

Lecciones Aprendidas:

- Google Apps Script tiene límites de ejecución (6 min)
- Importancia de validación inmediata en frontend
- Necesidad de feedback visual durante guardado



## Sprint 2: Gestión de Requisitos

### Requisitos Implementados:

- R02: Registro Canal Virtual
- R03: Autocompletado parcial
- R15: Restricción de red (parcial)

### Cambios y Ajustes:

- Original: Lógica de escalado manual
- Ajustado: Automático basado en motivo de contacto
- Justificación: Reducir errores y estandarizar proceso

### Riesgos Identificados:

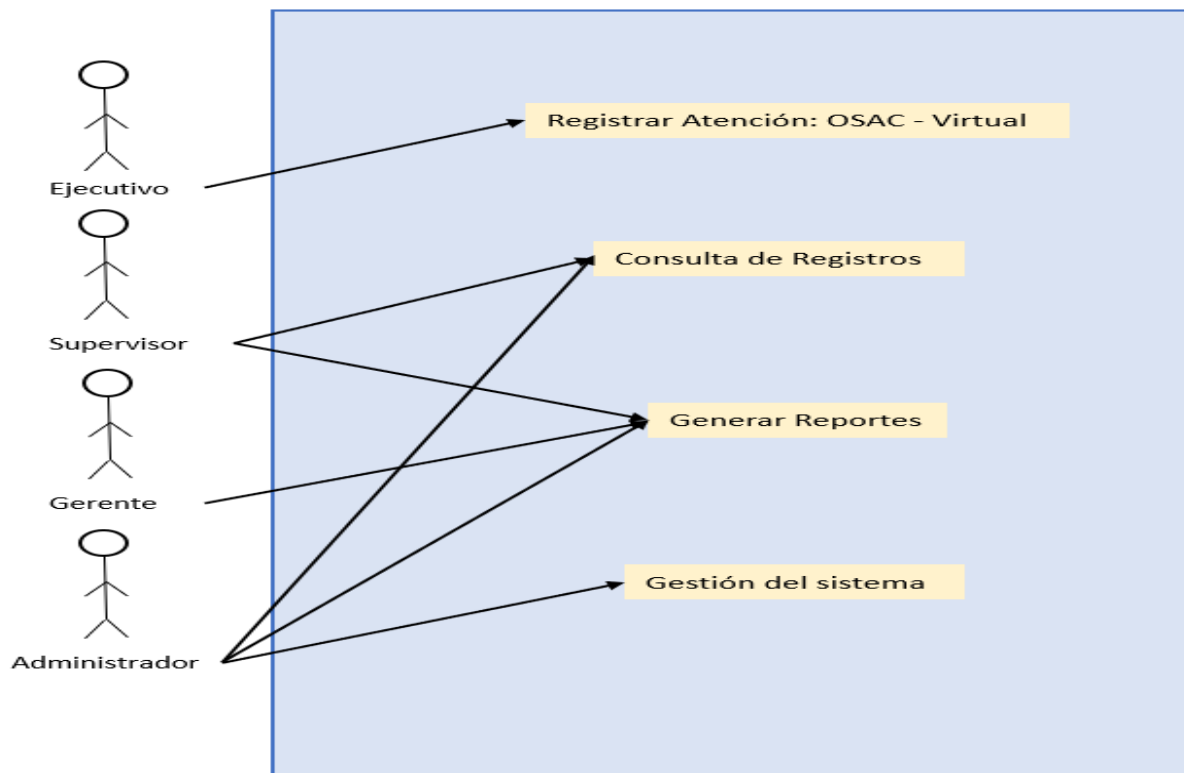
- Complejidad en carga de datos maestros (estados, ciudades)
- Dependencia de conexión a internet para autocompletado
- Variabilidad en formatos de datos históricos

## 2.5 Especificación de Requisitos de Software (ERS) / Modelo de Casos de Uso

### Actores del Sistema

Actor	Descripción	Responsabilidades
<b>Ejecutivo de Atención</b>	Personal que registra las atenciones al cliente	Registrar, consultar propias gestiones
<b>Supervisor de Área</b>	Persona que supervisa equipo de ejecutivos	Consultar registros de su equipo, generar reportes básicos
<b>Gerente de Operaciones</b>	Responsable de resultados operativos	Acceso a todos los datos, reportes gerenciales, dashboards
<b>Administrador del Sistema</b>	Personal técnico que gestiona el sistema	Mantenimiento, gestión de usuarios, backups
<b>Cliente (Indirecto)</b>	Usuario final que recibe la atención	No interactúa directamente, pero es sujeto de los registros

### Diagrama de Casos de Uso Resumido



### Especificación Detallada de Casos de Uso



## CU-001: Registrar Atención OSAC

1. Actor Primario: Ejecutivo de Atención
2. Precondiciones: Usuario autenticado, sección "Registro de atención" activa

### Flujo Principal:

3. Sistema presenta formulario OSAC con campos organizados por secciones
4. Ejecutivo selecciona oficina de atención de lista desplegable
5. Sistema carga ejecutivos disponibles para esa oficina
6. Ejecutivo ingresa datos de línea (prefijo + número)
7. Sistema autocompleta datos del cliente si la línea existe
8. Ejecutivo selecciona motivo y detalle de contacto
9. Sistema activa/desactiva campo de escalado según motivo
10. Ejecutivo ingresa observaciones y hace clic en "Guardar"
11. Sistema valida datos, guarda en BD y confirma éxito

Postcondiciones: Registro guardado en BD, número de caso generado

## CU-002: Consultar Registros Históricos

Actor Primario: Supervisor

Precondiciones: Usuario autenticado, permisos de consulta

### Flujo Principal:

1. Supervisor navega a sección "Consultas"
2. Sistema presenta formulario de búsqueda con filtros



3. Supervisor ingresa criterios (línea, fecha, ejecutivo)
4. Sistema valida criterios y ejecuta búsqueda
5. Sistema muestra resultados en tabla paginada
6. Supervisor puede ver detalles de cada registro
7. Supervisor puede exportar resultados a CSV

#### Criterios de Aceptación:

- Búsqueda por línea retorna resultados en < 2 segundos
- Filtros combinados funcionan correctamente
- Paginación maneja hasta 10,000 registros
- Exportación mantiene formato y codificación correcta

#### CU-003: Generar Reporte Diario

Actor Primario: Gerente

Precondiciones: Usuario autenticado, permisos de reportes

#### Flujo Principal:

1. Gerente navega a sección "Reportes"
2. Sistema muestra opciones de reportes disponibles
3. Gerente selecciona "Reporte Diario"
4. Sistema solicita fecha (por defecto día actual)
5. Gerente confirma parámetros
6. Sistema genera reporte con: Total de atenciones del día Desglose por tipo (OSAC/Canal Virtual), casos completados vs incompletos, top 5 motivos de contacto



7. Sistema presenta reporte en pantalla y opción de descarga PDF

Criterios de Aceptación:

- Generación completa en < 30 segundos
- Datos consistentes con registros en BD
- Descarga del formato en .PDF
- Incluye metadatos (fecha generación, usuario)

Requisitos Funcionales

ID	Requisito	Descripción
RF-01	Registro de Atención OSAC	Sistema debe permitir registrar atenciones presenciales con todos los campos obligatorios y opcionales
RF-02	Registro de Atención Canal Virtual	Sistema debe permitir registrar atenciones digitales con campos específicos para canal virtual
RF-03	Autocompletado de Datos	Al ingresar una línea telefónica, sistema debe buscar y autocompletar datos del cliente si existen registros previos
RF-04	Validación en Tiempo Real	Sistema debe validar formatos y reglas de negocio durante el llenado del formulario
RF-05	Selectores Dinámicos	Los campos desplegables deben cargar opciones basadas en selecciones previas (ej: estado → ciudad → municipio)
RF-06	Gestión de Escalado	Sistema debe determinar automáticamente si un caso requiere escalado basado en el motivo de contacto
RF-07	Búsqueda por Línea	Sistema debe permitir buscar registros históricos por número de línea telefónica completa
RF-08	Filtros Avanzados	Sistema debe permitir filtrar registros por múltiples criterios (fecha, ejecutivo, tipo, resultado)
RF-09	Paginación de Resultados	Al consultar muchos registros, sistema debe mostrar resultados paginados con navegación
RF-10	Exportación de Consultas	Sistema debe permitir exportar resultados de búsqueda a formatos CSV o Excel





<b>RF-11</b>	Reporte Diario de Atenciones	Sistema debe generar reporte automático con estadísticas diarias de atenciones
<b>RF-12</b>	Reporte de Casos Escalados	Sistema debe listar casos pendientes de resolución que requieren escalado
<b>RF-13</b>	Reporte Mensual por Tipo	Sistema debe generar consolidado mensual desglosado por tipo de gestión y motivo
<b>RF-14</b>	Dashboard de KPIs	Sistema debe mostrar dashboard con indicadores clave en tiempo real
<b>RF-15</b>	Gestión de Usuarios	Administrador debe poder crear, editar y desactivar usuarios del sistema
<b>RF-16</b>	Asignación de Permisos	Sistema debe permitir asignar diferentes niveles de acceso por rol de usuario
<b>RF-17</b>	Cierre de Sesión Seguro	Sistema debe permitir cerrar sesión y limpiar datos de sesión localmente
<b>RF-18</b>	Resumen de Registro	Sistema debe mostrar resumen del registro antes de confirmar el guardado
<b>RF-19</b>	Consulta de Registros Propios	Ejecutivo debe poder consultar sus propios registros del día
<b>RF-20</b>	Histórico Completo por Cliente	Sistema debe mostrar todas las gestiones históricas de un cliente específico
<b>RF-21</b>	Backup Automático	Sistema debe realizar backup automático diario de la base de datos
<b>RF-22</b>	Restricción por Red	Sistema solo debe ser accesible desde la red corporativa interna
<b>RF-23</b>	Notificaciones de Error	Sistema debe mostrar mensajes de error claros y sugerencias de solución
<b>RF-24</b>	Auditoría de Acciones	Sistema debe registrar quién, cuándo y qué acción realizó en el sistema

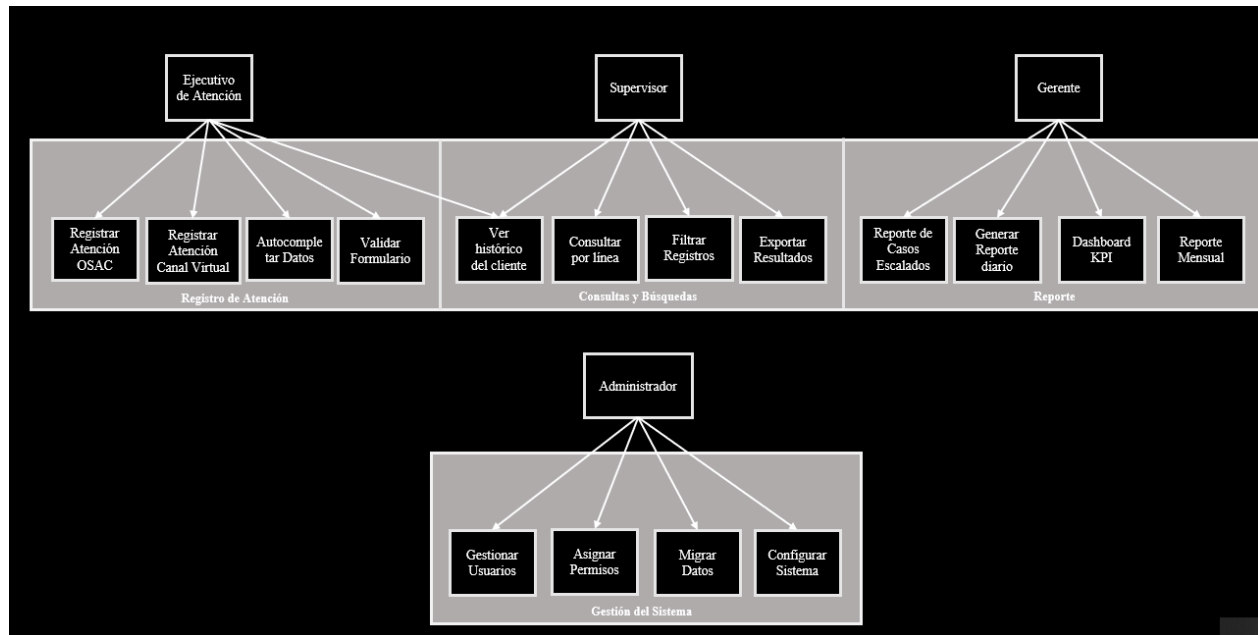


## Requisitos No Funcionales

ID	Requisito	Descripción
RNF-01	Tiempo de Respuesta Guardado	Sistema debe guardar registros en tiempo aceptable
RNF-02	Tiempo de Búsqueda	Sistema debe responder búsquedas rápidamente
RNF-03	Tiempo de Carga Inicial	La aplicación debe cargar completamente en tiempo razonable
RNF-04	Disponibilidad Horario Laboral	Sistema debe estar disponible durante horas de trabajo
RNF-05	Tolerancia a Fallos	Sistema debe recuperarse de fallos menores sin pérdida de datos
RNF-06	Autenticación	Solo usuarios autorizados pueden acceder al sistema
RNF-07	Confidencialidad	Datos sensibles deben estar protegidos
RNF-08	Control de Acceso	Usuarios solo ven/accionan según sus permisos
RNF-09	Interfaz Intuitiva	La interfaz debe ser clara y fácil de entender
RNF-10	Accesibilidad	Sistema debe ser usable por personas con discapacidades leves
RNF-11	Navegadores Soportados	Sistema debe funcionar en navegadores corporativos
RNF-12	Dispositivos Soportados	Sistema debe ser responsivo para diferentes tamaños de pantalla
RNF-13	Usuarios Concurrentes	Sistema debe soportar múltiples usuarios simultáneos
RNF-14	Volumen de Datos	Sistema debe manejar crecimiento de datos
RNF-15	Integridad de Datos	Datos deben mantenerse consistentes y completos
RNF-16	Tiempo Medio Entre Fallos	Sistema debe ser estable y confiable
RNF-17	Documentación	El sistema debe estar documentado para mantenimiento
RNF-18	Modularidad	El código debe ser modular para facilitar cambios
RNF-19	Dependencias de Plataforma	Sistema debe minimizar dependencias de plataforma específica
RNF-20	Uso de Recursos	Sistema debe usar recursos de manera eficiente
RNF-21	Ancho de Banda	Sistema debe minimizar transferencia de datos

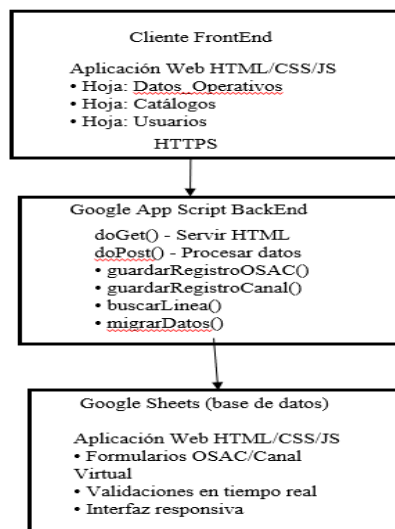
### 3. Análisis y Diseño

#### 3.1 Vista de Casos de Uso a Implementar



#### 3.2 Descripción Detallada de Arquitectura

##### Arquitectura General del Sistema





## Arquitectura de Componentes

### Frontend (HTML/JS/CSS):

- Componente UI: Gestión de formularios y navegación
- Componente Validación: Reglas de negocio en cliente
- Componente Comunicación: AJAX con Google Apps Script
- Componente Renderizado: Manejo de DOM dinámico

### Backend (Google Apps Script):

- Componente Routing: Manejo de endpoints (doGet, doPost)
- Componente Negocio: Lógica de guardado y búsqueda
- Componente Integración: Conexión con Google Sheets
- Componente Seguridad: Validación de acceso y datos

### Capa de Datos:

- Google Sheets: Almacenamiento transitorio y catálogos
- Componente Migración: Transferencia incremental de datos



## Flujo de Datos

1. Cliente carga página → doGet() sirve Index.html
2. Usuario llena formulario → Validaciones en JavaScript
3. Submit formulario → AJAX a función Apps Script
4. Apps Script procesa → Valida y transforma datos
5. Guarda en Google Sheets → (o en Respaldo)
6. Responde al cliente → Confirmación o error
7. Cliente muestra feedback → Éxito o mensaje de error

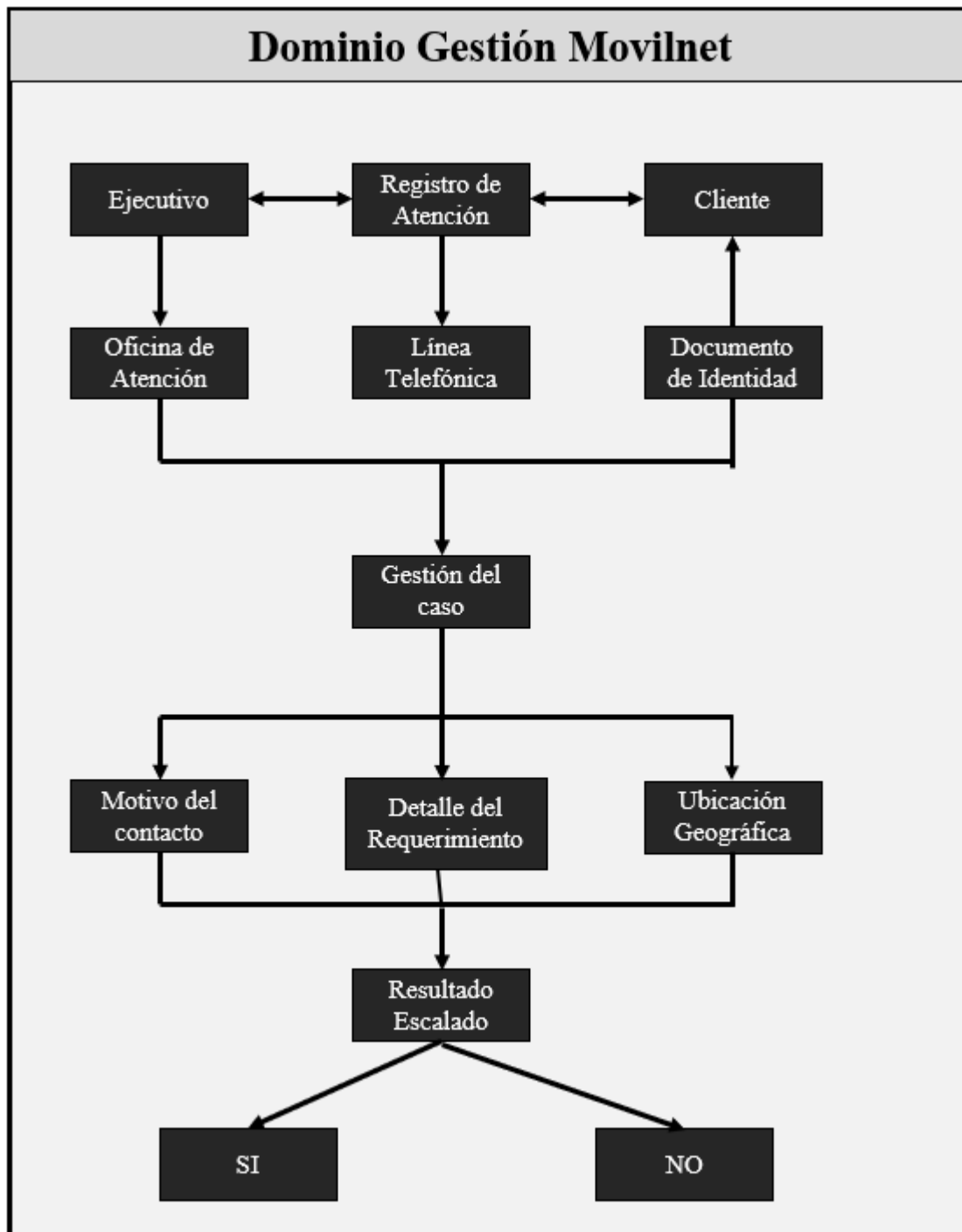
## Patrones de Diseño Aplicados

MVC (Modelo-Vista-Controlador):

- Modelo: Estructura de datos en Google Sheets
- Vista: HTML/CSS con templates dinámicos
- Controlador: Google Apps Script con funciones de negocio
- Singleton: Para conexión a base de datos
- Factory: Para creación de diferentes tipos de registros
- Observer: Para actualizaciones en tiempo real en dashboard
- Strategy: Para diferentes algoritmos de validación

### 3.3 Modelo de Análisis

#### Modelo de Dominio





## Reglas de Negocio

- RN-01: Un registro debe tener mínimo: canal, ejecutivo, fecha, línea, motivo y detalle
- RN-02: El campo "Escalado" solo es editable si motivo  $\in$  {Requerimiento, Falla Operativa, Reclamo}
- RN-03: Si escalado = "NO", entonces resultado = "COMPLETO", caso contrario "INCOMPLETO"
- RN-04: La línea a gestionar se forma concatenando prefijo2 + número (7 dígitos)
- RN-05: Solo usuarios autenticados desde red corporativa pueden acceder
- RN-06: Los catálogos (estados, ciudades, etc.) se cargan dinámicamente según selecciones anteriores

## 3.4 Modelo de Diseño

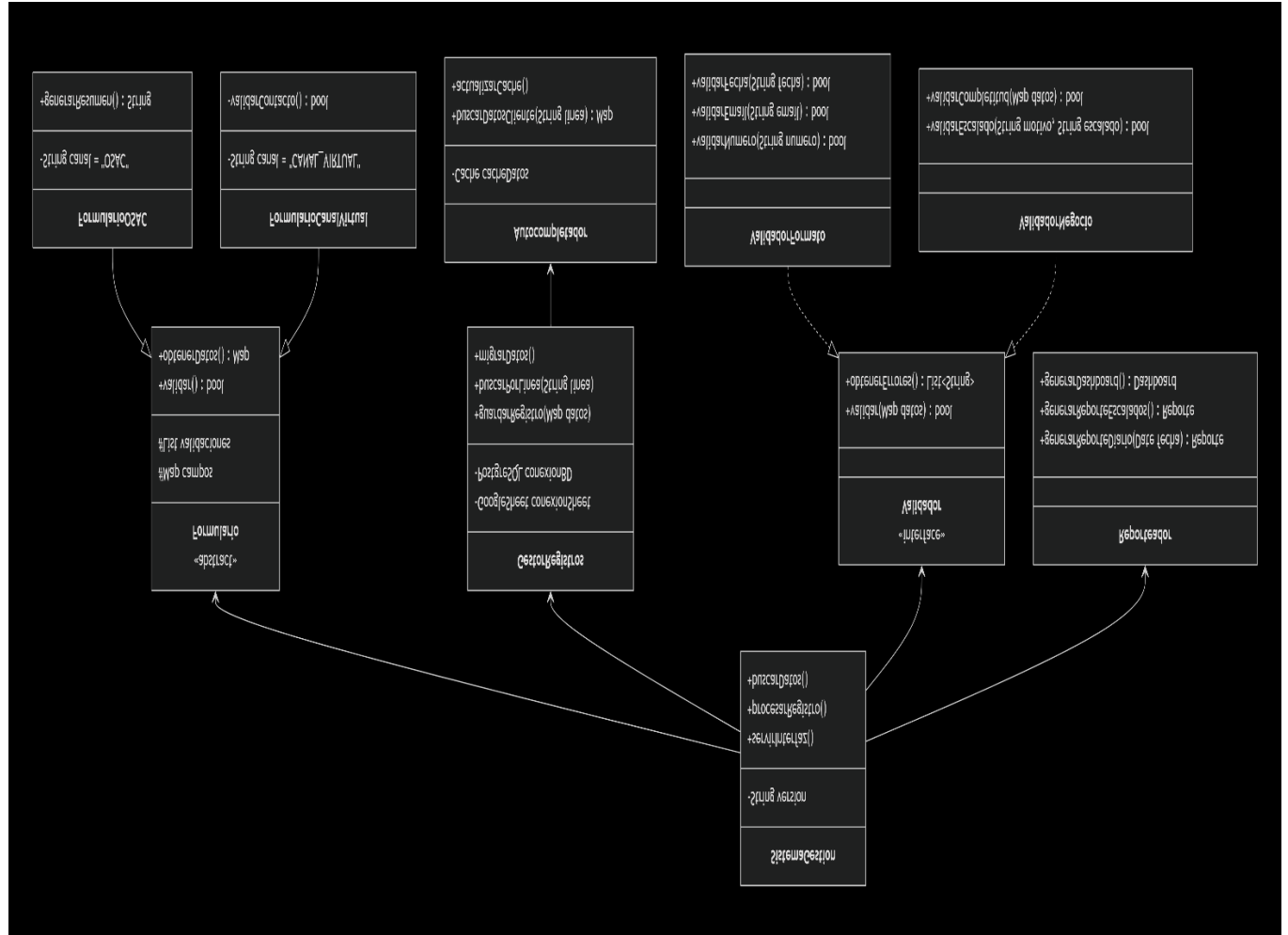
### Diseño de Interfaz de Usuario

Inicio → [Registro → (OSAC | Canal Virtual)] | [Consultas] | [Reportes] | [Configuración]

#### Componentes UI Principales:

- Header: Logo Movilnet + Usuario + Cerrar Sesión
- MainMenu: Navegación principal con estado activo
- FormContainer: Contenedor de formularios con pestañas OSAC/Canal
- FormSection: Secciones temáticas (Canal, Línea, Cliente, Caso, Ubicación)
- DynamicSelect: Selectores con carga dinámica y dependencias
- ToggleGroup: Grupos de botones para opciones exclusivas (SI/NO)
- ValidationFeedback: Mensajes de error/éxitos contextuales

## Diagrama de Clases (Nivel de Diseño)

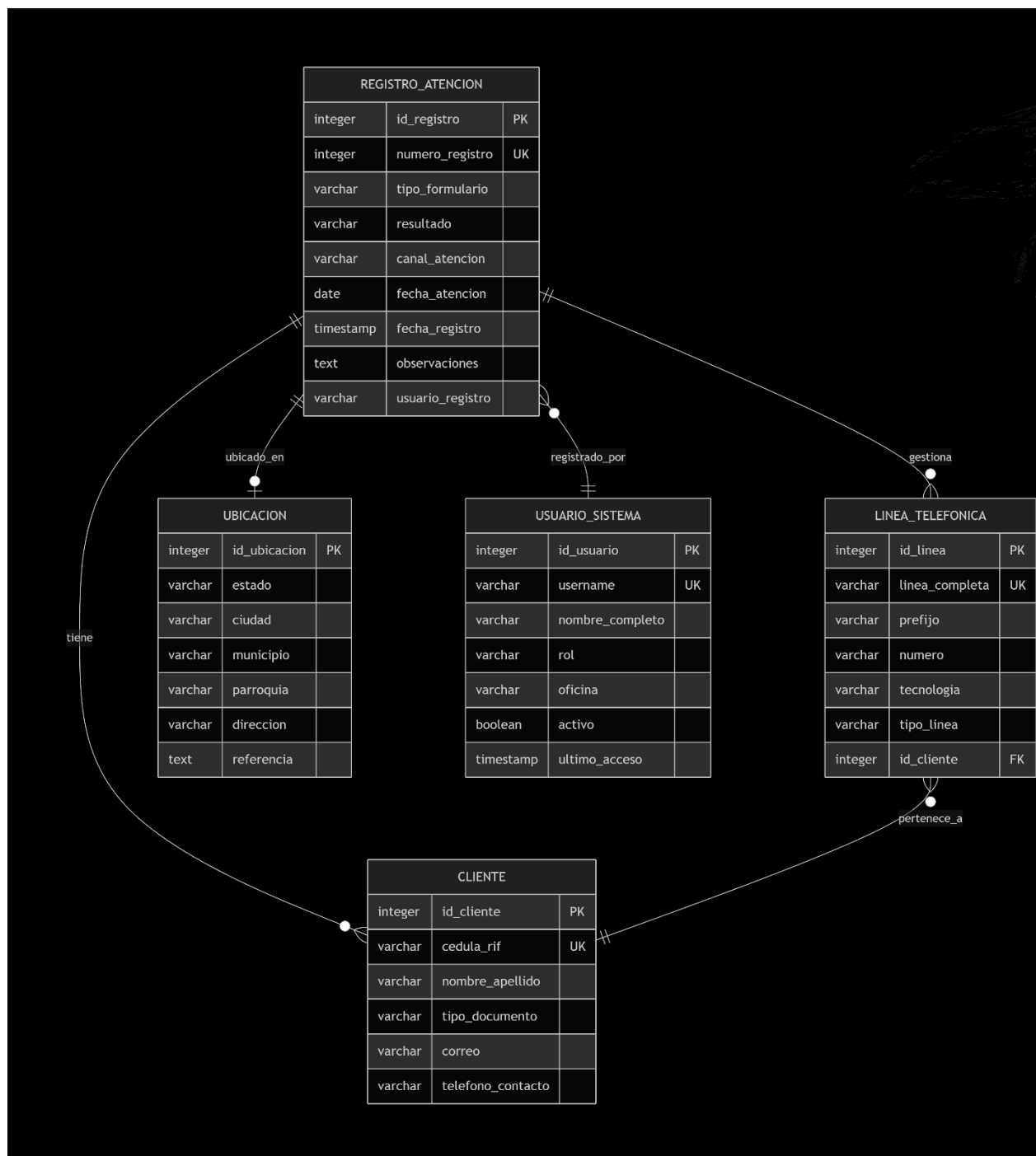






### 3.5 Modelo de Datos / Diccionario de Datos

#### Diagrama Entidad-Relación (Nivel Lógico)





## Diccionario de Datos

Campo	Tipo	Descripción
id_registro	SERIAL	Identificador único autoincremental
numero_registro	INTEGER	Número consecutivo de registro
tipo_formulario	VARCHAR	Tipo de formulario (OSAC/CANAL_VIRTUAL)
resultado	VARCHAR	Resultado de la gestión
canal_atencion	VARCHAR	Canal de atención
fecha_atencion	DATE	Fecha de la atención
fecha_registro	TIMESTAMP	Fecha-hora del registro
prefijo	VARCHAR	Prefijo telefónico
contacto	VARCHAR	Número de contacto
cola_atencion	VARCHAR	Cola u oficina de atención
gestor	VARCHAR	Ejecutivo que atendió
numero_contrato	VARCHAR	Número de contrato
prefijo2	VARCHAR	Segundo prefijo
numero	VARCHAR	Número telefónico (7 dígitos)
linea_gestionar	VARCHAR	Línea completa
tecnologia	VARCHAR	Tecnología de la línea
tipo_linea	VARCHAR	Tipo de línea
tipo_cliente	VARCHAR	Tipo de cliente
segmento	VARCHAR	Segmento del cliente
nombre_apellido	VARCHAR	Nombre completo
tipo_documento	VARCHAR	Tipo de documento
cedula_rif	VARCHAR	Número de identificación
numero_sim	VARCHAR	Número de SIM
correo	VARCHAR	Correo electrónico
telefono_contacto	VARCHAR	Teléfono de contacto
motivo_contacto	VARCHAR	Motivo del contacto
detalle_requerimiento	VARCHAR	Detalle del requerimiento
escalado	VARCHAR	¿Requiere escalado?
estatus	VARCHAR	Estatus del caso
estado	VARCHAR	Estado
ciudad	VARCHAR	Ciudad
municipio	VARCHAR	Municipio
parroquia	VARCHAR	Parroquia
direccion	VARCHAR	Dirección
referencia	TEXT	Punto de referencia
genero	VARCHAR	Género
atencion_preferencial	VARCHAR	¿Atención preferencial?



<b>discapacidad</b>	VARCHAR	¿Posee discapacidad?
<b>adulto_mayor</b>	VARCHAR	¿Adulto mayor?
<b>observaciones</b>	TEXT	Observaciones adicionales
<b>usuario_registro</b>	VARCHAR	Usuario que registró

## 4. Implementación

### 4.1 Documento de Arquitectura del Sistema

El sistema de Gestión Movilnet se implementa bajo un modelo de arquitectura Client-Server (o de Tres Capas). La implementación se aprovecha completamente del entorno de Google Workspace, utilizando Google Apps Script como la capa de lógica de negocio y Google Sheets como la capa de persistencia de datos.

#### Modelo de Implementación

El modelo se divide en tres capas principales:

- Capa de Presentación (Frontend):
- Responsable de la Interfaz de Usuario (UI) y la interacción con el usuario.
- Desarrollada con HTML5, CSS3, y JavaScript (ES6+).

El archivo principal es Index.html que contiene los formularios de registro OSAC y Canal Virtual.

Utiliza funciones de JavaScript para validaciones en tiempo real , autocompletado de datos , y manejo de la navegación.

#### Capa de Lógica de Negocio (Backend/Middleware):

- Implementada mediante Google Apps Script.



- Actúa como un controlador que maneja las solicitudes del cliente y las interacciones con la base de datos.
- Contiene funciones clave como doGet() para servir la UI , y guardarRegistroOSAC()/guardarRegistroCanal() para el procesamiento de datos.
- Contiene la lógica de mapeo de campos a columnas de la base de datos a través del objeto MASTER\_FIELD\_MAPPING.

#### **Capa de Datos (Persistencia):**

- Actualmente utiliza Google Sheets como la base de datos transaccional (con planes de posible migración).
- La hoja principal de almacenamiento se identifica como "Datos Operativos".
- La interacción se realiza exclusivamente a través de las funciones de Apps Script, asegurando la integridad de los datos según el mapeo definido en MASTER\_FIELD\_MAPPING.

#### **Modelo de Despliegue**

El sistema se despliega íntegramente en la infraestructura de Google, accesible a través de la red corporativa. El acceso se realiza mediante el navegador web, ya que Google Apps Script sirve el código HTML/JS/CSS.



## Componentes del Sistema

Componente	Descripción	Tecnologías/Archivos	Función
<b>UI Forms</b>	Interfaz para los registros OSAC y Canal Virtual.	Index.html (Formularios)	Captura de datos del usuario.
<b>Client Script</b>	Lógica de validación de formularios y manejo de eventos.	Index.html (<script> block)	Validaciones en tiempo real y comunicación con el Backend.
<b>Apps Script Controller</b>	Punto de entrada del Backend.	Code.gs (doGet ())	Sirve la interfaz web y gestiona la comunicación.
<b>Apps Script Data Functions</b>	Lógica para guardar, buscar y autocompletar registros.	Code.gs (guardarRegistroOSAC (), etc.)	Procesamiento y persistencia de datos.
<b>Data Mappers</b>	Objeto de configuración que mapea los campos de la UI a las columnas de la BD.	Code.gs (MASTER_FIELD_MAPPING)	Garantiza la consistencia en el almacenamiento.
<b>Data Storage</b>	Base de datos principal.	<b>Google Sheet</b> (Hoja "Datos Operativos")	Almacenamiento centralizado de registros.



## 4.2 Manual de Operaciones

### Configuración y Despliegue Inicial

Publicación de la Aplicación: El Apps Script Controller (Code.gs) debe publicarse como una aplicación web con acceso restringido a la red corporativa o usuarios autorizados de Google Workspace. La función doGet() es la que sirve la UI.

ID de la Hoja de Datos: El valor de la variable a\_sheetId en Code.gs debe coincidir con el ID de la hoja de Google Sheets designada como base de datos

### Tareas de Mantenimiento y Monitoreo

Tarea	Frecuencia	Responsable	Procedimiento
<b>Copia de Seguridad (Backup)</b>	Diario/Semanal automático	Administrador de Sistema	Realizar una copia de la hoja de Google Sheets. Se mantiene un <i>backup original</i> durante 1 mes para la mitigación de inconsistencias.
<b>Monitoreo de Conectividad</b>	24/7 (según mitigación)	Operaciones/TI	Verificar la disponibilidad del servicio de Google Apps Script y la conexión con Google Sheets. Se debe asegurar que las conexiones críticas utilicen reintentos automáticos.
<b>Monitoreo de Rendimiento</b>	En tiempo real	Operaciones/TI	Observar la latencia de guardado de registros (RNF-01) y la velocidad de las búsquedas (RNF-02). Si se detecta sobrecarga en horas pico, evaluar el escalado o la implementación de colas de procesamiento asíncrono.



Tarea	Frecuencia	Responsable	Procedimiento
<b>Auditoría de Acceso</b>	Semanal	Seguridad	Revisar los logs de acceso al sistema y a la hoja de Google Sheets para detectar accesos no autorizados.
<b>Actualización de Datos Estáticos</b>	Bajo demanda	Administrador de Aplicación	Actualizar las listas de colas, gestores, motivos, y datos de ubicación (estados, ciudades, etc.) definidas en los scripts, recargando el caché si aplica.

### Resolución de Problemas (Troubleshooting)

Problema: Fallos por límites de Google Apps Script.

Solución: Optimizar las llamadas por lotes a la API de Sheets. Si el problema persiste, se tiene un Plan B de migrar el Backend a Node.js.

Problema: Pérdida de datos.

Solución: Utilizar el proceso de Recuperación Desastrosa (probado mensualmente) a partir del backup más reciente.



## 4.3 Código Fuente y Versiones Ejecutables

### Código Fuente

- El código fuente se encuentra dividido en dos componentes principales: la lógica de servidor (Backend) y la interfaz de cliente (Frontend), adjuntados en los documentos originales.
- Backend/Apps Script (Archivo Code.gs - contenido en Code.docx):
- Contiene la lógica de negocio, manipulación de datos y la configuración del mapeo.
- Funciones clave: doGet(), guardarRegistroOSAC(), guardarRegistroCanal(),  
MASTER\_FIELD\_MAPPING.

### Frontend/Web UI (Archivo Index.html - contenido en Index.docx):

- Contiene la estructura HTML, el CSS de estilos (definido en el bloque <style>) y el JavaScript de cliente.
- Lógica importante en el cliente: configurarEventListeners(), validarFormulario(),  
generarLineaGestionar(), y la data estática (prefijosData, tiposClienteData).
- Versiones Ejecutables (Incrementos y Versión Final)

El proyecto se desarrolló siguiendo una metodología que involucra entregas incrementales (Sprints).





## Incrementos (Iteraciones)

Iteración	Entregable Clave	Lógica Implementada
<b>Iteración 1</b> <b>(Semanas</b> <b>1-2)</b>	<b>Prototipo estático y</b> <b>Formulario OSAC</b> <b>funcional</b>	Diseño de la UI, <code>doGet()</code> para servir HTML, configuración de Google Sheets, implementación de <code>guardarRegistroOSAC()</code> con validaciones básicas.
<b>Iteración 2</b> <b>(Semanas</b> <b>3-5)</b>	<b>Segundo Formulario</b> <b>(Canal Virtual)</b> <b>funcional y Sistema de</b> <b>guardado en Google</b> <b>Sheets</b>	Implementación de <code>guardarRegistroCanal()</code> , lógica de <b>autocompletado</b> (función <code>buscarYAutocompletar</code> implícita en <code>generarLineaGestionar</code> ), validaciones avanzadas <sup>4646464646464646</sup> .
<b>Iteración 3</b> <b>(Semanas</b> <b>9-10)</b>	<b>Sistema optimizado y</b> <b>Documentación de</b> <b>usuario final</b>	Optimización de consultas, implementación de seguridad por roles, caché para datos estáticos (mencionado en mitigación de riesgos).
<b>Iteración 4</b> <b>(Semanas</b> <b>11-14)</b>	<b>Dashboard de métricas</b> <b>en tiempo real y</b> <b>Documentación técnica</b> <b>completa</b>	Implementación de módulos de consultas avanzadas y desarrollo del Dashboard ejecutivo.



## 5 pruebas

### 5.1 Plan de Pruebas a Aplicar por Iteración o Sprint

El plan de pruebas se enfoca en verificar la funcionalidad y la integridad de los datos del sistema, siguiendo el enfoque incremental de la metodología ágil.

Fase	Objetivo Principal	Tipo de Pruebas	Alcance	Criterios de Éxito
<b>Iteración 1: Formulario OSAC Básico</b>	Validar la captura de datos y el guardado en la base de datos (Google Sheet) para el formulario principal.	<b>Pruebas Unitarias</b> (Functions), <b>Pruebas de Integración</b> (Frontend-Backend), <b>Pruebas Funcionales</b> (Caja Negra).	Módulo de Registro OSAC, función <code>guardarRegistroOSAC()</code> , mapeo de datos.	100% de los casos de prueba funcionales exitosos. Los datos se almacenan correctamente en la hoja.
<b>Iteración 2: Formulario Canal Virtual y Autocompletado</b>	Validar el nuevo módulo de registro, las nuevas validaciones y la funcionalidad de autocompletado de la línea a gestionar.	<b>Pruebas de Regresión</b> (Módulo OSAC), <b>Pruebas de Integración</b> (Nuevo Módulo), <b>Pruebas Funcionales</b> .	Módulo de Registro Canal Virtual, función <code>guardarRegistroCanal()</code> , función <code>generarLineaGestionar()/buscarYAutocompletar</code> .	100% de los casos de prueba exitosos. El registro OSAC sigue funcionando sin errores. El autocompletado es preciso.
<b>Iteración 3: Optimización y Seguridad</b>	Verificar la robustez, el rendimiento y la aplicación de políticas de seguridad.	<b>Pruebas de Rendimiento</b> (Carga), <b>Pruebas de Seguridad</b> (Acceso), <b>Pruebas de Regresión</b> .	Optimización de consultas, gestión de roles/permisos, manejo de caché.	El tiempo de respuesta del servidor (latencia de guardado) cumple con el RNF-01 (< 3 segundos). Los usuarios no autorizados no pueden acceder.



## 5.2 Casos de Pruebas a Aplicar por Iteración o Sprint

### Iteración 1: Casos de Prueba Funcionales (Registro OSAC)

ID Caso	Componente	Descripción de la Prueba	Datos de Prueba	Resultado Esperado
T-OSAC-001	UI/Guardado	Registro exitoso con todos los campos obligatorios.	Campos obligatorios llenos, escalado = <b>NO</b> .	El registro se guarda en la Hoja "Datos Operativos". Columna Resultado = <b>COMPLETO</b> .
T-OSAC-002	Validación	Intentar guardar con un campo obligatorio vacío (Ej: Cédula/RIF).	Campo <code>cedula_rif</code> vacío.	El sistema muestra un mensaje de error en la UI. No se realiza el envío al Backend.
T-OSAC-003	Mapeo de Datos	Verificar el mapeo correcto de un campo crítico (Ej: Cola de Atención).	<code>cola_atencion</code> = 'OSAC Central'.	El valor 'OSAC Central' se registra correctamente en la columna 'Cola De Atención' de la Hoja.
T-OSAC-004	Lógica Negocio	Registro que requiere escalado.	Campos llenos, escalado = <b>SÍ</b> .	El registro se guarda. Columna Resultado = <b>INCOMPLETO</b> .



## Iteración 2: Casos de Prueba Funcionales (Canal Virtual y Autocompletado)

ID Caso	Componente	Descripción de la Prueba	Datos de Prueba	Resultado Esperado
T-CV-001	UI/Guardado	Registro exitoso en el Formulario Canal Virtual.	Llenar campos obligatorios del formulario Canal Virtual.	El registro se guarda en la Hoja con <i>Tipo</i> = <b>Canal Virtual</b> .
T-CV-002	Regresión OSAC	Verificar que el Formulario OSAC sigue funcionando.	Realizar T-OSAC-001 de nuevo.	El resultado debe ser <b>éxito</b> , como en la Iteración 1.
T-CV-003	Autocompletado	Generación correcta de la Línea a Gestionar.	<code>prefijo2 = '416', numero = '1234567'.</code>	El campo <code>linea_gestionar</code> se actualiza automáticamente a <code>'4161234567'</code> .
T-CV-004	Lógica Autocompl.	Autocompletado con número incompleto.	<code>prefijo2 = '416', numero = '123456' (6 dígitos).</code>	El campo <code>linea_gestionar</code> debe quedar <b>vacío</b> o no actualizarse.



### 5.3 Informes de Pruebas Realizadas por Iteración o Sprint

El informe de pruebas documenta la ejecución de los casos definidos, el estado final y los defectos encontrados.

#### Informe de Pruebas - Iteración 1 (Formulario OSAC Básico)

ID Caso	Descripción	Resultado (Éxito/Fallo)	Observaciones	Estado (Corregido/Abierto)
T-OSAC-001	Registro exitoso.	Éxito	Guardado correcto en la fila 2 de la BD.	Corregido
T-OSAC-002	Campo obligatorio vacío (Cédula).	Fallo (Defecto 001)	La validación de <i>Cédula/RIF</i> permitía el envío si el campo no tenía el atributo <i>required</i> en el HTML.	Corregido (Se añadió el atributo <i>required</i> y la validación en JS).
T-OSAC-003	Mapeo de Datos.	Éxito	El campo se mapeó correctamente según <i>MASTER_FIELD_MAPPING</i> .	Corregido
T-OSAC-004	Registro con escalado.	Éxito	El campo <i>Resultado</i> se almacenó como <b>INCOMPLETO</b> .	Corregido



## Informe de Pruebas - Iteración 2 (Canal Virtual y Autocompletado)

ID Caso	Descripción	Resultado (Éxito/Fallo)	Observaciones	Estado (Corregido/Abierto)
T-CV-001	Registro Canal Virtual.	Éxito	Guardado correcto en la Hoja.	Corregido
T-CV-002	Regresión OSAC.	Éxito	La corrección del Defecto 001 en Iteración 1 no afectó la funcionalidad OSAC.	Corregido
T-CV-003	Autocompleta do Línea.	Éxito	La concatenación de prefijo2 y número fue correcta.	Corregido
T-CV-004	Línea incompleta.	Fallo (Defecto 002)	La función <code>generarLineaGestionar</code> no limpiaba el campo si la longitud del número era incorrecta.	Corregido (Se añadió la condición <code>else { lineaGestionar Input.value = ''; } al código JavaScript</code> ).



## 5.4 Pruebas Unitarias

Las pruebas unitarias se enfocan en verificar las funciones del Google Apps Script (Code.gs) de forma aislada para asegurar que la lógica de negocio funcione correctamente antes de la integración con el Frontend.

ID Prueba	Función a Probar	Descripción de la Prueba	Input de Prueba	Resultado Esperado
U-GS-001	MASTER_FIELD_MAPPING	Verificar si existe el mapeo para un campo esencial.	Clave: 'cedula_rif'.	Valor: 'N° De Cédula O Rif'.
U-GS-002	guardarRegistroOSAC()	Verificar la lógica de <b>Resultado</b> (COMPLETO/INCOMPLETO).	formData con escalado = 'NO'.	La columna <b>Resultado</b> debe ser 'COMPLETO' en el array de la nueva fila (newRow).
U-GS-003	guardarRegistroOSAC()	Verificar el valor por defecto de la columna <b>Tipo</b> .	formData de tipo 'osac'.	La columna <b>Tipo</b> debe ser 'OSAC'.
U-GS-004	getDataSheet()	Verificar que la función obtiene la hoja correcta.	ID de la hoja: a_sheetId.	Retorna el objeto SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Datos Operativos").
U-GS-005	obtenerEncabezados()	Verificar la lista completa y el orden de los encabezados.	Hoja "Datos Operativos".	Retorna un array con el orden exacto de las columnas de la BD.



## 6. Capacitación

La capacitación es fundamental para asegurar la correcta adopción del Sistema de Gestión Movilnet por parte de los operadores (Ejecutivos de Atención OSAC y de Canal Virtual) y del personal técnico que realizará el soporte y mantenimiento.

### 6.1 Plan de Capacitación y Contenido Programático

El plan de capacitación está diseñado con un enfoque modular, adaptado a los dos perfiles principales:

Usuarios Finales (Ejecutivos, Supervisores) y Administradores del Sistema (TI).

#### Plan de Capacitación General

Sesión	Audiencia Objetivo	Duración Estimada	Objetivo de Aprendizaje
<b>Sesión 1: Inducción al Sistema</b>	Usuarios Finales y Administradores	1.5 horas	Comprender la arquitectura, la importancia de la calidad del dato y la navegación básica.
<b>Sesión 2: Operación Diaria</b>	Usuarios Finales (Agentes)	2.5 horas	Manejar correctamente los formularios, aplicar validaciones y registrar transacciones de forma eficiente.
<b>Sesión 3: Despliegue y Soporte</b>	Administradores del Sistema (TI)	3.0 horas	Conocer el ciclo de vida del código (Apps Script), el modelo de datos (Google Sheets) y los procedimientos de mantenimiento y <i>troubleshooting</i> .





## Contenido Programático de las Sesiones

Módulo	Tema	Contenido Específico	Audiencia
<b>Módulo 1: Introducción y Fundamentos</b>	1.1 Visión General del Proyecto	Objetivos del sistema (DES), beneficios y requisitos no funcionales (RNF-01, RNF-02).	Todos
	1.2 Arquitectura del Sistema	Explicación de la arquitectura de 3 capas (Frontend, Apps Script, Google Sheets).	Todos
	1.3 Calidad del Dato	Relevancia de los datos a capturar (según MASTER_FIELD_MAPPING) y las consecuencias de datos erróneos.	Todos
<b>Módulo 2: Operación del Agente</b>	2.1 Acceso y Navegación	Acceso a la URL de la Web App, interfaz principal y cambio entre formularios (OSAC/Canal Virtual).	Agentes
	2.2 Registro OSAC	Llenado del formulario principal, significado de cada campo obligatorio, uso correcto de las listas desplegables.	Agentes
	2.3 Registro Canal Virtual	Campos específicos del Canal Virtual. Función de <b>Autocompletado</b> (generarLineaGestionar) y su uso.	Agentes
	2.4 Validaciones y Errores	Respuesta del sistema ante datos faltantes (T-OSAC-002), interpretación de mensajes de error de Apps Script.	Agentes
	2.5 Flujo de Escalado	Diferencia entre registro <b>COMPLETO</b> y	Agentes



Módulo	Tema	Contenido Específico	Audiencia
		<b>INCOMPLETO</b> (según el campo escalado).	
<b>Módulo 3: Gestión Técnica</b>	3.1 Entorno de Google Apps Script	Acceso al editor de <i>scripts</i> , estructura de los archivos ( <code>Code.gs</code> , <code>Index.html</code> ), y cómo realizar un cambio menor.	Administradores
	3.2 Despliegue y Versiones	Procedimiento de publicación como Web App (ver 6.2), manejo de versiones y rollback.	Administradores
	3.3 Mantenimiento de la Base de Datos	Estructura de la Hoja de Google Sheets ("Datos Operativos"), orden de las columnas (según <code>MASTER_FIELD_MAPPING</code> ) y la importancia de no modificar los encabezados.	Administradores
	3.4 Procedimientos de Soporte	Monitoreo de logs de ejecución y gestión de errores comunes (ej. "Service invoked too many times").	Administradores

## 6.2 Manual de Instalación

Este manual está dirigido al personal de TI responsable del despliegue y mantenimiento del ambiente del Sistema de Gestión Movilnet.

### 1. Requisitos Previos

Cuenta de Google Workspace con permisos para crear y publicar aplicaciones en Google Apps Script.

Acceso al ID de la Hoja de Google Sheets centralizada (1tT-

ximGZnuNIItSpszy3Y6kmNRF1E5tXEJDFmJWNqLo8).



## 2. Preparación de la Hoja de Cálculo

Verifique que la hoja de Google Sheets denominada "Datos Operativos" exista y contenga los encabezados de columna exactamente en el orden definido por el sistema.

Asegúrese de que el ID de esta hoja coincida con el valor de la variable `a_sheetId` en el archivo `Code.gs`.

## 3. Despliegue del Código

Abra el editor de Google Apps Script (mediante la Hoja de Cálculo o directamente desde `script.google.com`).

Cree un nuevo proyecto de script.

Capa de Lógica (`Code.gs`): Copie y pegue el contenido completo del documento `Code.docx` en el archivo `Code.gs` del proyecto de Apps Script, reemplazando cualquier contenido existente.

Capa de Presentación (`Index.html`): Cree un nuevo archivo HTML en el proyecto de Apps Script con el nombre `Index.html`. Copie y pegue el contenido completo del documento `Index.docx` en este nuevo archivo.

Verificación de ID: Revise la primera línea de `Code.gs` y confirme que const `a_sheetId` contenga el ID correcto de la Hoja de Datos.

## 4. Publicación como Aplicación Web

En el editor de Apps Script, haga clic en Desplegar > Nueva implementación.

Seleccione el Tipo de implementación como Aplicación web.

Configure los siguientes parámetros:



Ejecutar como: Usuario que accede a la aplicación (es importante para que se ejecute con los permisos del agente).

Quién tiene acceso: Solo yo (para pruebas iniciales) o Cualquier usuario de Movilnet C.A. (para despliegue final).

Haga clic en Desplegar.

Se generará una URL de la aplicación web. Esta es la URL que se proporcionará a los agentes.

Opcional (Despliegue de Producción): Actualice la implementación existente seleccionando el número de versión anterior en lugar de crear una nueva, para mantener la misma URL.

## **6.3 Manual de Usuarios**

Este manual está dirigido a los Ejecutivos de Atención (OSAC y Canal Virtual) para guiarles en el uso diario del sistema.

### **1. Acceso al Sistema**

Abra su navegador web y acceda a la URL de la Aplicación Web proporcionada por el equipo de TI.

La pantalla principal mostrará la interfaz del Sistema de Gestión Movilnet.

### **2. Navegación entre Módulos**

El sistema presenta pestañas o botones que permiten alternar entre el Formulario OSAC (para atenciones presenciales/directas) y el Formulario Canal Virtual (para atenciones remotas).

Seleccione el formulario que corresponda al tipo de atención que está realizando.



### 3. Registro de Atención (Procedimiento General)

El objetivo es llenar todos los campos obligatorios para asegurar la integridad de los datos.

Datos del Gestor: Ingrese su Cola de Atención y Gestor (su nombre).

Datos de la Línea (Módulo Canal Virtual):

Ingrese el Prefijo2 (ej. 416, 426).

Ingrese el Número (7 dígitos).

Función de Autocompletado: Al completar el número de 7 dígitos, el campo Línea A Gestionar se completará automáticamente. Si falta un dígito, el campo se vaciará como señal de validación.

Datos del Cliente:

Seleccione el Tipo de Documento y el N° De Cédula O Rif.

Llene los datos demográficos (Nombre Y Apellido, Teléfono De Contacto, Correo Electrónico).

Datos de la Gestión:

Seleccione el Motivo del Contacto (lista predefinida).

Escriba el Detalle del Requerimiento de forma concisa.

Indique si el caso requiere Escalado (SÍ o NO). Esto definirá el Resultado como INCOMPLETO o COMPLETO.



#### 4. Finalizar y Guardar el Registro

Una vez llenados todos los campos obligatorios, haga clic en el botón Guardar Registro.

Mensajes de Confirmación:

Éxito: Si el registro se guarda correctamente, el sistema mostrará un mensaje de éxito y los campos se limpiarán.

Error: Si falta un campo obligatorio o hay un error de conexión, se mostrará un mensaje de error que indica qué campos deben corregirse.

#### 5. Consultas

Los registros guardados se almacenan en la Hoja de Google Sheets "Datos Operativos".

Para verificar un registro, solicite el acceso de lectura a dicha hoja. El sistema registra cada atención con un número de fila (columna N°).