



**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**

# **Documento de Diseño**

**Winder Rivas**  
C.I: 22.029.261

**Caracas, agosto del 2025**



## Introducción

La aplicación web propuesta tiene como propósito centralizar la gestión de datos operativos de Movilnet C.A., eliminando la dispersión de registros en múltiples sistemas y formatos. Su objetivo es consolidar la información de clientes, empleados y reportes de gestión en una base de datos unificada, accesible en tiempo real, que permita generar informes automatizados, mejorar la atención al cliente y optimizar la toma de decisiones estratégicas.

Esta solución responde a necesidades críticas identificadas en el diagnóstico: duplicación de datos, retrasos en la atención, falta de trazabilidad y ausencia de una infraestructura tecnológica integrada.



# Arquitectura Aplicación

La aplicación se construirá inicialmente sobre el ecosistema de Google Workspace, aprovechando Google Apps Script como motor backend y Google Sheets como base de datos operativa. Esta arquitectura permite una rápida implementación, bajo costo de infraestructura, y una integración nativa con formularios, correos y otros servicios de Google.

## Componentes principales

Componente	Descripción
<b>Frontend (Vista)</b>	Interfaz web desarrollada en HTML/CSS/JavaScript, embebida en Web Apps de Apps Script o desplegada externamente.
<b>Backend (Controlador)</b>	Scripts en Google Apps Script que gestionan la lógica de negocio, validación y conexión con la hoja de cálculo.
<b>Base de datos (Modelo)</b>	Google Sheets estructurado como base de datos relacional simulada, con hojas separadas por entidad.

## Estructura de la base de datos en Google Sheets

Campo	Descripción
Nº	Número de registro único.
Resultado	Resultado de la gestión (Completo, Incompleto)
PREFIJO / Nº CONTACTO (Solo WhatsApp)	Datos de entrada del cliente
Canal / Cola / Ejecutivo de Atención	Información del canal, cola y agente que atendió.
Fecha de Atención	Fecha en que se realizó la atención.
Número de Contrato / Prefijo2 / Número	Datos de la operación que realiza el cliente
Línea a Gestionar / Tecno. / Tipo de Línea	Información técnica de la línea gestionada.
Tipo de Cliente / Segmento	Clasificación del cliente.
Nombre / Cédula / Género / tipo de documento	Datos personales del titular o usuario.



¿Requiere Atención Preferencial? / ¿Posee discapacidad? / ¿Adulto Mayor?	Indicadores de atención especial.
Motivo / Detalle del requerimiento	Razón del contacto y descripción del caso.
Número de SIM / Correo / Teléfono	Datos de opcionales y de contacto
Observaciones / Escalado / Estatus	Comentarios adicionales y estado del caso.
Estado / Ciudad / Municipio / Parroquia	Ubicación geográfica del cliente.
Dirección / Punto de referencia	Detalles de ubicación para reclamos o fallas operativas.

## Lógica de negocio

- Validación dinámica de campos obligatorios y formatos (Apps Script).
- Filtrado automático por estado, canal, ejecutivo, etc.
- Generación de reportes por fecha, segmento, tipo de cliente.
- Control de acceso por rol desde hoja de usuarios.
- Envío de correos automáticos para casos escalados o pendientes.

## Justificación

Esta arquitectura permite:

- Centralizar la información operativa en una única hoja estructurada.
- Automatizar tareas repetitivas mediante Apps Script.
- Facilitar la trazabilidad de casos y atención al cliente.
- Preparar el terreno para una migración futura a SQL, donde cada grupo de campos podrá convertirse en una tabla relacionada (Ej. Clientes, Atención, Ubicación, Requerimientos).

La estructura actual está diseñada para ser fácilmente transformable en un modelo relacional, manteniendo la integridad de los datos y la lógica de negocio modular.



# Diseño de la Interfaz de Usuario

Principios de diseño de la aplicación en cuanto a la interfaz de usuario (UI):

- **Intuitiva:** navegación clara, sin curvas de aprendizaje.
- **Coherente:** estilos visuales homogéneos en toda la aplicación.
- **Accesible:** compatible con lectores de pantalla, contraste adecuado, y navegación por teclado.
- **Estéticamente agradable:** uso de la paleta corporativa de Movilnet (Cereza, blanco, gris), tipografía legible (Arial Narrow, 12), y disposición limpia.

## Justificación basada en usuarios

Los usuarios principales son agentes de atención, analistas de gestión y supervisores. La interfaz está pensada para:

- Minimizar errores en la carga de datos gracias a los campos listados, con formato y obligatorios.
- Agilizar la búsqueda y filtrado de registros.
- Facilitar la visualización rápida del estado de atención.
- Permitir la edición y seguimiento (aunque lo permita la aplicación no está pensada para realizar seguimiento) de casos desde cualquier dispositivo.

## Estructura de navegación

Sección principal	Función
Inicio	Panel de resumen con KPIs
Registro de atención	Formulario dinámico con validación en tiempo real.
Consultas	Filtros avanzados por estado, ejecutivo, fecha, tipo de cliente.
Reportes	Visualización de estadísticas por segmento, canal, y tipo de requerimiento.
Usuarios	Gestión de roles, permisos y credenciales.



## Disposición de elementos

- **Menú principal fijo:** accesible desde cualquier vista, con íconos y etiquetas.
- **Panel principal:** Botón de Inicio, formularios, tablas con paginación.
- **Centro:** resumen de los últimos casos registrados

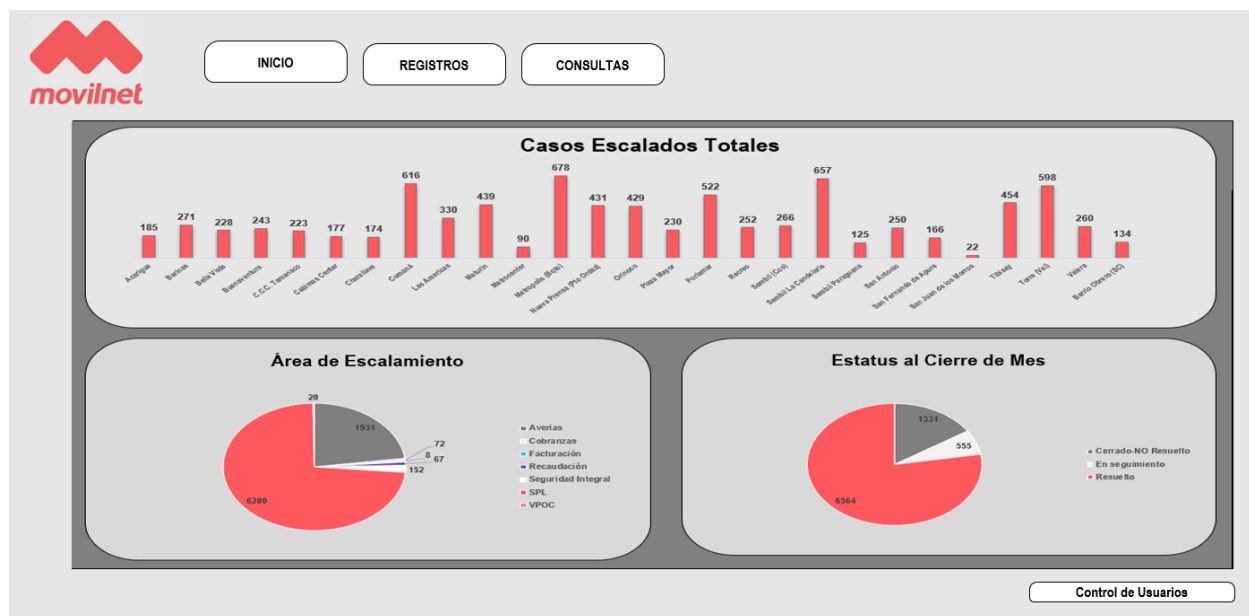
## Estilo visual

- **Paleta de colores:**
  - Blanco (#ffffff): botones primarios, alertas.
  - Gris claro (#F2F2F2): fondo de formularios.
  - Blanco: Gris Claro.
  - Negro/gris oscuro: texto principal.
  - Gráficos Cerezo/Gris/Blanco
- **Tipografía:** Arial Narrow, 12–18px, con jerarquía clara en títulos y subtítulos.
- **Iconografía:** Stock image (de referencia y genéricas).

## Placeholder

The placeholder shows a web application interface. At the top left is the Movilnet logo. To its right are three navigation buttons: INICIO, REGISTROS, and CONSULTAS. Below these is a table with 14 columns. The table headers are: N°, Resultado, Prefijo, N° Entrada, Canal de Atención, Cola de Atención, Ejecutivo de Atención, Género, ¿Requiere Atención Preferencial?, ¿Posee discapacidad?, ¿Adulto Mayor?, Motivo del contacto, Detalle del requerimiento, and Teléfono de Contacto. Below the table is a large empty rectangular area. At the bottom right of the interface is a button labeled 'Control de Usuarios'.

N°	Resultado	Prefijo	N° Entrada	Canal de Atención	Cola de Atención	Ejecutivo de Atención	Género	¿Requiere Atención Preferencial?	¿Posee discapacidad?	¿Adulto Mayor?	Motivo del contacto	Detalle del requerimiento	Teléfono de Contacto
----	-----------	---------	------------	-------------------	------------------	-----------------------	--------	----------------------------------	----------------------	----------------	---------------------	---------------------------	----------------------



## Accesibilidad

- Navegación por teclado (tabindex definido).
- Etiquetas ARIA para lectores de pantalla.
- Contraste mínimo AA garantizado.
- Mensajes de error descriptivos y visibles.

## Diseño de la Base de Datos

La aplicación utilizará una hoja de cálculo estructurada como tabla única en Google Sheets, simulando una base de datos relacional. Cada fila representa un caso de atención, y cada columna corresponde a un atributo del registro. Aunque no se implementan claves foráneas de forma nativa, se aplicarán validaciones y segmentaciones mediante Apps Script para mantener la integridad referencial.



## Estructura de la tabla principal

Campo	Tipo de dato	Descripción
Nº	<b>Número entero</b>	Identificador único del registro.
Resultado	<b>Texto</b>	Estado final del caso (Ej. Resuelto, Pendiente).
PREFIJO / Nº CONTACTO	<b>Número entero</b>	Número de contacto del cliente.
Canal de Atención / Cola / Ejecutivo	<b>Lista</b>	Información del canal, cola y agente que atendió.
Fecha de Atención	<b>Fecha</b>	Fecha en que se realizó la atención.
Número de Contrato / Prefijo2 / Número	<b>Número entero</b>	Datos contractuales del cliente.
Línea a Gestionar / Tecno. / Tipo de Línea	<b>Lista</b>	Información técnica de la línea gestionada.
Tipo de Cliente / Segmento	<b>Texto</b>	Clasificación del cliente.
Nombre y Apellido / Cédula / Género / Nac.	<b>Texto / Selección</b>	Datos personales del titular o usuario.
¿Requiere Atención Preferencial? / ¿Posee discapacidad? / ¿Adulto Mayor?	<b>Selección</b>	Indicadores de atención especial.
Motivo del contacto / Detalle	<b>Lista</b>	Razón del contacto y descripción del requerimiento.
Número de SIM / Correo / Teléfono	<b>Texto</b>	Datos de contacto y SIM.
Observaciones / Escalado / Estatus	<b>Texto</b>	Comentarios adicionales y estado del caso.
Estado / Ciudad / Municipio / Parroquia	<b>Lista</b>	Ubicación geográfica del cliente.
Dirección / Punto de referencia	<b>Texto largo</b>	Detalles de ubicación para reclamos o fallas operativas.





## Normalización y eficiencia

Aunque la estructura inicial es una tabla plana, se aplicarán principios de normalización lógica:

- **1FN:** Todos los campos contienen valores atómicos.
- **2FN:** Se evitarán dependencias parciales mediante segmentación lógica en Apps Script.
- **3FN:** Se eliminarán dependencias transitivas en futuras migraciones a SQL.

Para mejorar la eficiencia:

- Se usarán **validaciones de entrada** en formularios para evitar errores.
- Se aplicarán **filtros automáticos** por canal, ejecutivo, estado y fecha.
- Se implementarán **índices simulados** mediante Apps Script para búsquedas rápidas.

## Justificación del diseño de seguridad

Este modelo responde directamente a los requisitos del diagnóstico:

- Permite registrar todos los atributos relevantes de cada caso en una sola vista.
- Facilita la trazabilidad de atención por ejecutivo, canal y ubicación.
- Soporta segmentación por tipo de cliente, atención preferencial y estado del caso.
- Es fácilmente exportable y transformable en un modelo relacional SQL.

## Diseño de la Seguridad

La seguridad de la aplicación se abordará de forma integral, considerando tanto el acceso de usuarios como la protección de los datos sensibles gestionados en la hoja de cálculo. Dado que la aplicación se ejecutará sobre Google Apps Script, se aprovecharán las capacidades nativas de Google Workspace para autenticación, además de implementar controles personalizados para autorización, trazabilidad y prevención de ataques comunes.



## Autenticación de usuarios

Mecanismo	Descripción
<b>Autenticación OAuth 2.0 (Google)</b>	Los supervisores accederán mediante su cuenta corporativa de Google, validada por el sistema.
<b>Control de sesión</b>	Se verificará la identidad del usuario en cada carga de página mediante <code>Session.getActiveUser()</code> .
<b>Registro de actividad</b>	Se almacenará el correo del usuario y la hora de acceso en una hoja de auditoría.

## Autorización y control de acceso

Rol de usuario	Permisos principales
<b>Administrador</b>	Acceso total: configuración, usuarios, reportes, edición de registros.
<b>Supervisores</b>	Acceso a reportes, filtros avanzados, y edición de registros.

La lógica de autorización se implementará en Apps Script, validando el rol antes de ejecutar cualquier acción sensible. Los formularios y botones se mostrarán u ocultarán dinámicamente según el rol.

## Protección de datos sensibles

- **Cifrado en tránsito:** Toda la comunicación se realiza sobre HTTPS, garantizado por el entorno de Google Apps Script.
- **Cifrado en reposo:** Aunque Google Sheets no permite cifrado nativo por celda, los datos sensibles como contraseñas se almacenarán como hash (bcrypt) y no serán visibles en la interfaz.
- **Separación de datos:** Los datos de autenticación y roles estarán en una hoja separada, protegida contra edición directa.



## Prevención contra ataques comunes

Amenaza	Mecanismo de mitigación
<b>Acceso no autorizado</b>	Validación de usuario activo y rol antes de ejecutar cualquier acción.

Además, se implementará un sistema de alertas para detectar accesos fuera del horario habitual o desde ubicaciones no autorizadas.

## Justificación técnica

La seguridad propuesta responde directamente a los riesgos identificados en el diagnóstico:

- Protección de datos personales y contractuales del cliente.
- Control de acceso por rol para evitar manipulación indebida.
- Prevención de errores humanos mediante validaciones dinámicas.
- Trazabilidad completa de acciones para auditoría interna.

La arquitectura sobre Google Apps Script permite implementar estas medidas sin necesidad de servidores externos, aprovechando la robustez de Google Workspace y complementándola con lógica personalizada.

## Diseño de la Escalabilidad

La escalabilidad de la aplicación se ha diseñado para garantizar un rendimiento óptimo tanto en su fase inicial como en su evolución futura. Aunque la solución parte de una arquitectura ligera sobre Google Apps Script y Google Sheets, se han previsto mecanismos para soportar un crecimiento sostenido en volumen de datos, cantidad de usuarios y complejidad funcional.



## Escalabilidad en el entorno actual

Estrategia	Descripción
<b>Segmentación de datos</b>	División de registros por hojas mensuales o por canal de atención para evitar sobrecarga.
<b>Paginación y carga diferida</b>	Visualización de registros en bloques, evitando renderizar grandes volúmenes simultáneamente.
<b>Indexación lógica</b>	Uso de Apps Script para simular índices en campos clave (fecha, ejecutivo, estado).

Estas técnicas permiten manejar cientos de registros diarios sin afectar la experiencia del usuario ni comprometer la integridad de los datos.

## Escalabilidad futura

Se planea una migración de infraestructura a prevención en un futuro cuando el volumen de datos o la cantidad de usuarios supere los límites operativos de Google Sheets, se migrará a una arquitectura más robusta como lo sería una infraestructura SQL y un servidor para el Backend:

Componente futuro	Ventaja principal
<b>Base de datos relacional (PostgreSQL/MySQL)</b>	Manejo eficiente de millones de registros, relaciones complejas y consultas optimizadas.
<b>Backend en Node.js o Python</b>	Mayor control sobre concurrencia, seguridad y lógica de negocio avanzada.
<b>API RESTful</b>	Comunicación estructurada entre frontend y backend, permitiendo escalabilidad horizontal.
<b>Balanceadores de carga</b>	Distribución automática de peticiones entre múltiples servidores.
<b>Contenedores (Docker)</b>	Despliegue ágil y replicable en entornos de producción.



## Justificación técnica

La escalabilidad propuesta responde directamente a los desafíos identificados en el diagnóstico:

- **Crecimiento de usuarios:** la aplicación será utilizada por múltiples oficinas, ejecutivos y analistas en simultáneo.
- **Aumento de registros:** se espera una acumulación progresiva de casos, requerimientos y reportes.
- **Evolución funcional:** se prevé la incorporación de nuevos módulos (permisos, incidencias, trazabilidad avanzada).

La arquitectura modular y el uso de Apps Script como capa intermedia permiten una transición progresiva sin reescribir la lógica de negocio, manteniendo la continuidad operativa.