TIC

Reto
Hackathon
Ficha Técnica

Operador: ANDES

28 y 29 de octubre de 2025
Cúcuta, Norte de Santander

TALENTO TECH ORIENTE

TIC

Reto

¿Cómo diseñar un prototipo funcional, sencillo y escalable que permita a la administración del Conjunto Residencial Balcones de Salesia organizar, consolidar y analizar la información contable–administrativa (cartera, ingresos/egresos, conciliaciones, censo de propietarios, residentes y vehículos, y reportes generales) reduciendo tiempos, errores y duplicidades derivados del manejo manual actual?

Este reto ha sido planteado por Condominio Balcones de Salesia – Propiedad Horizontal .

Descripción del problema:

Descripción de la entidad

El Condominio balcones de Salesia administra 13 apartamentos y requiere optimizar sus procesos administrativos y contables. Hoy la información se gestiona de forma manual y fragmentada, lo que provoca demoras, baja trazabilidad y dificultad para generar reportes confiables para copropietarios y el Consejo. Se busca un MVP digital acorde al nivel de los campistas que siente bases para automatizar tareas críticas y mejorar la toma de decisiones.

Contextualización del problema para el Reto Tecnológico

- Procesos afectados: registro de pagos y gastos, conciliaciones bancarias, seguimiento de cartera morosa, elaboración de informes mensuales, actualización de censos(residentes, propietarios y vehículos), comunicaciones y trazabilidad documental.
- Dificultades actuales: duplicidad de datos, errores de digitación, archivos dispersos, demoras en cierres, poca visibilidad del flujo de caja y del estado de la cartera, escasa estandarización de procedimientos.
- Meta del reto: contar con una solución simple de bajo umbral de entrada (hoja de cálculo + interfaz web ligera o app de escritorio/web sencilla) que centralice datos, automatice tareas repetitivas básicas y genere reportes claros en tiempo real o bajo demanda.

Descripción del reto

Desarrollar un MVP (prototipo) que, con herramientas de nivel básico/intermedio, permita a la administración.

1. Registrar y centralizar información de unidades (aptos), copropietarios, cuotas ordinarias/extraordinarias, pagos, gastos, proveedores, censos, documentación actualizada y presupuesto anual.
2. Calcular automáticamente saldos por apartamento (corriente, vencido, intereses si aplica) y alertar sobre morosidad.
3. Generar reportes básicos: estado de cartera (por antigüedad), flujo de caja simple, egresos por categoría, conciliación básica (cargue manual/CSV).
4. Visualizar tableros simples (gráficas y tablas) para la toma de decisiones.
5. Incorporar nociones mínimas de ciberseguridad: credenciales por rol, bitácora de cambios y respaldo/exportación.

Objetivo del reto:

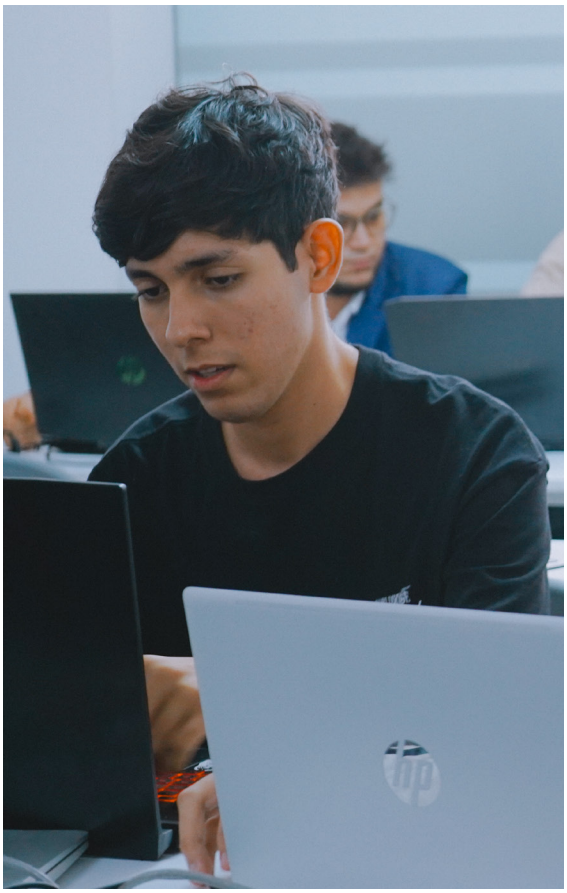
Entregar un prototipo funcional y documentado que:

- Unifique la información clave (cartera, ingresos/egresos, contactos, censos, presupuesto).
- Estandarice el registro y la consulta de datos.
- Automatice cálculos y genere reportes comprensibles.
- Aplique buenas prácticas básicas de seguridad y protección de datos acorde al nivel del equipo.
- Sea escalable para futuras mejoras (p. ej., integración de pasarelas de pago, mensajería, IA para predicciones).

Alcance (MVP para hackathon)

- **Base de datos simple** (Google Sheets o SQLite) con tablas: Apartamentos, Copropietarios, Pagos, Gastos, Proveedores, Configuración de cuotas.
- **Interfaz mínima para:**
  - Crear/editar registros.
  - Cargar pagos/gastos (manual o CSV).
  - Ver dashboard con cartera por antigüedad, pagos del mes, egresos por rubro.
- **Reportes exportables (PDF/CSV) de**: Cartera, Flujo simple, Egresos por categoría, Resumen mensual.
- **Seguridad básica:** usuario/contraseña, roles (Administrador/Consulta), bitácora de cambios, respaldo (exportación periódica).
- **Datos de prueba precargados** (13 aptos, 3–6 meses de movimientos simulados).

Nota: No se exige infraestructura compleja. Se valora simplicidad, claridad y que el prototipo quede utilizable.



Requerimientos funcionales:

1. **Gestión de cartera:** cálculo automático de saldos y morosidad; filtro por apartamento, torre, rango de fechas.
2. **Ingreso de pagos y gastos:** formulario guiado, validaciones básicas y adjuntos opcionales (comprobantes).
3. **Reportes y tableros:** cartera por antigüedad (30/60/90+), flujo de caja simple, egresos por rubro, top morosos.
4. **Conciliación básica:** cargue de extracto (CSV) y marcado manual de correspondencias.
5. **Usuarios y roles:** Admin (CRUD y configuración) y Consulta (solo lectura/reportes).
6. **Exportación y respaldo:** descarga de CSV/PDF y copia de seguridad programable (por ejemplo, duplicado de Sheet).
7. **Comunicaciones (opcional):** generación de aviso/estado de cuenta en PDF por apartamento.

Requerimientos NO funcionales:

- **Usabilidad:** interfaz intuitiva, formularios claros, mensajes de validación.
- **Escalabilidad:** diseño de datos que permita crecer (más aptos, rubros, periodos).
- **Rendimiento:** respuesta fluida con volumen moderado (mas o menos 5 mil registros).
- **Seguridad de datos:** control de acceso por rol, gestión de contraseñas, respaldo/exportación.
- **Documentación:** guía rápida de uso + README técnico de despliegue.

Atributos de la solución:

- **Innovación útil:** enfoque simple pero efectivo para digitalizar la operación.
- **Flexibilidad:** parametrización de cuotas, intereses (si aplica), rubros y periodos.
- **Transparencia:** visualizaciones claras y trazabilidad (bitácora).
- **Escalabilidad:** posibilidad de integrar futuras funciones (pagos, mensajería, IA).
- **Seguridad básica:** Debe ofrecer visualización en tiempo real de las reservas, mesas disponibles y ocupación, permitiendo una gestión eficiente y planificación precisa.
- **Seguridad:** buenas prácticas iniciales (roles, copias, logs).

Criterios de evaluación:

- **Innovación y valor práctico:** creatividad con impacto real en la gestión.
- **Viabilidad técnica:** factible con herramientas y conocimientos básico/intermedios.
- **Impacto en el problema:** mejora tangible en tiempos, errores y reportes.
- **Usabilidad (15%):** facilidad de uso para personal administrativo.
- **Escalabilidad:** bases para crecer sin rediseños profundos. Los criterios están alineados con formatos tipo “ficha del reto” para hackathons académicos y comunitarios.

Recursos disponibles para los equipo:

- **Bases de datos:** Acceso a un conjunto de datos predefinidos o simplificados que contienen información que permitirá a los campistas trabajar en un entorno simulado para realizar pruebas y desarrollar soluciones. En caso de no contar con esta base de datos pueden crear campos hipotéticos de acuerdo al requerimiento del reto.

- **Herramientas de análisis de datos:** Acceso a programas o plataformas de software que permiten a los campistas realizar análisis básicos, como Microsoft Excel, Google Sheets, o herramientas de visualización de datos simples.

- **Soporte de mentores:** Disponibilidad de expertos que pueden responder preguntas y guiar a los campistas en el uso de herramientas de programación y análisis de datos.

- Sesiones de orientación con directores académicos para conocer más a fondo las dificultades y expectativas en la asignación de recursos y cursos.

Restricciones o consideraciones especiales:

- Los campistas tienen un **nivel básico** en inteligencia artificial, análisis de datos y programación, por lo que las soluciones deben enfocarse en un prototipo simple, que pueda ser escalable en el futuro.

- Permitir al equipo operativo visualizar rápidamente la disponibilidad de mesas, la asignación y el cliente de manera eficiente, adaptándose a los horarios extendidos y bloques específicos de eventos, manteniendo la flexibilidad para futuras expansiones en la cantidad de sedes y recursos disponibles.

- El tiempo para desarrollar el prototipo está limitado a los días del hackathon, por lo que se espera una **funcionalidad mínima viable.**

Posibles herramientas y técnicas (nivel básico):

Análisis de datos: uso de herramientas simples para organizar la información de los perfiles de los clientes.

Inteligencia artificial: implementación de algoritmos de búsqueda básica para sugerir perfiles basados en las necesidades de los usuarios externos.

Programación: desarrollo de una interfaz básica (frontend) para la visualización de los perfiles, con un backend que gestione las bases de datos, utilizando lenguajes como python.

Prototipo funcional (MVP)

- Puede ser navegable (web/app/archivo ejecutable) o un video de demostración (1–2 min) que evidencie los flujos clave: registro de pagos/gastos, cálculo de cartera, reportes básicos y respaldo/exportación.
- Debe incluir datos de prueba (no reales).

Manual de uso / PDF demostrativo

Documento breve (3–6 páginas) con:

- Descripción del problema y objetivo del MVP.
- Guía de uso paso a paso con capturas de pantalla.
- Alcance del prototipo (qué sí/qué no hace).
- Requisitos para ejecutar (si aplica).
- Próximos pasos sugeridos para evolución post-hackathon.

Nota: *El repositorio o código fuente es opcional, no es obligatorio para la evaluación del hackathon en nivel básico/intermedio.*

Ideas de extensión (opcional)

- **Notificaciones** (correo/WhatsApp) con estado de cuenta.
- **Clasificador simple** (IA) para priorizar cobro según historial (reglas o árbol básico).
- **Predicción básica** de flujo de caja con medias móviles.
- **Carga masiva** desde plantillas CSV.
- **Módulo PQRS** simplificado con estados.

TIC

¡Gracias por participar en el Hackathon Talento Tech Oriente – Cúcuta!

www.talentotechoriente.com