



# PRD - E26124

Propiedad	Valor
Desarrolladores	Giovanni Jose Leon Posada Ismael David Trocha Sarmiento Juan Camilo Vergara Marin
Fecha de Entrega	TBD (Sujeto a iteraciones de Beta)
Tipo de Aplicación	Mobile App (iOS/Android) + Backend API

## Contexto

### El Problema

En Cartagena, los micro-negocios y trabajadores independientes carecen de visibilidad geográfica y canales de comunicación en tiempo real. Muchos emprendedores gestionan más de un negocio a la vez y necesitan una herramienta centralizada para administrarlos de forma profesional e independiente.

### La Oportunidad

Transformar el descubrimiento de servicios locales mediante una interfaz basada en mapas con un componente social. La plataforma permitirá una arquitectura **Multi-Entidad**, donde un usuario puede ser dueño de múltiples proyectos independientes (cada uno con su propia ubicación única, catálogo, estados y suscripción).

### Métricas de Éxito (KPIs)

- Registrar y verificar 100 perfiles en 60 días.
- **Tasa de Multi-Tenancy:** Lograr que el 15% de los usuarios gestionen más de un emprendimiento.

- **Engagement Social:** Promedio de 3 vistas de "Estados" por usuario activo al día.
- **Conversión de Chat:** Lograr que el 20% de los chats iniciados terminen en una calificación de servicio finalizado.

## Alcance

### Incluye

- **Arquitectura Multi-Entidad:** Un usuario gestiona N emprendimientos de forma aislada.
- **Mapa Interactivo:** Vista principal con un pin por cada negocio/independiente registrado.
- **Sistema de Estados (Stories):** Publicaciones de 24h con funcionalidad de "Boost" (promoción pagada).
- **Chat en Tiempo Real:** Mensajería interna para consultas y cierre de ventas.
- **Blog de Perfil:** Noticias de solo lectura con reacciones (emojis).
- **Selector de Negocios:** Panel para que el dueño alterne entre sus diferentes emprendimientos.
- **Sistema de Feedback Verificado:** Reseñas habilitadas tras confirmar la finalización del servicio.
- **Pagos Locales:** Integración con pasarela colombiana (ePayco/Wompi) para suscripciones y cobros únicos.
- **Sistema de Verificación de Ubicación:** Métodos diferenciados para asegurar la veracidad de los pines en el mapa.

### No Incluye

- **Gestión Multi-Sede:** Un negocio solo tiene una ubicación física en V1.
- **Seguimiento en tiempo real:** Ubicación estática del negocio o profesional.
- **Consolidación de facturas:** Cada proyecto se paga por separado.
- **Pagos Internacionales:** Optimizado solo para tarjetas y métodos colombianos.

# Requerimientos Funcionales

Área	Historia de Usuario	Prioridad
Usuario	Como usuario, quiero crear múltiples perfiles de negocio independientes desde una sola cuenta.	Critica
Usuario	Como usuario, quiero ver un mapa con los servicios cercanos a mi ubicación para solucionar una necesidad.	Critica
Usuario	Como usuario, quiero enviar mensajes en tiempo real al emprendedor para resolver dudas.	Critica
Emprendedor	Como emprendedor, quiero subir pruebas de mi ubicación (fotos/documentos) para obtener el sello de "Verificado".	Alta
Emprendedor	Como emprendedor, quiero marcar un servicio como "finalizado" para que el cliente pueda calificarme.	Alta
Admin	Como administrador, quiero revisar las pruebas de ubicación enviadas para aprobar o rechazar la aparición del pin en el mapa.	Critica
Usuario	Como usuario, quiero reportar un pin si descubro que el negocio no existe en esa ubicación.	Media

# Especificaciones Técnicas

- **Frontend:** Expo (React Native) + `expo-maps`.
- **Backend:** Golang con Gin.
- **Comunicación:** WebSockets (Golang) para el chat.
- **Base de Datos:** PostgreSQL + PostGIS (Relación 1:N entre Usuario y Entidad).
- **Almacenamiento:** AWS S3 para fotos de catálogo, estados, blogs y documentos de verificación.
- **Cache:** Redis para sesiones de chat y persistencia temporal de estados.
- **Pagos:** **ePayco** o **Wompi** (API para cobros recurrentes vía Tokenización/Suscripciones y pagos únicos para Boosts).

## Cronograma

### Fase 1: Core y Multi-Tenancy (4 semanas)

- Diseño de base de datos multi-entidad.
- API base en Golang y servidor de WebSockets.
- Integración de pasarela de pagos colombiana (ePayco/Wompi) con soporte para PSE.

### Fase 2: Mapa y Social (6 semanas)

- Mapa con pines dinámicos y perfiles.
- Sistema de Chat e interfaz de Estados/Blog.
- Implementación del módulo de verificación de perfiles.

### Fase 3: Lanzamiento (3 semanas)

- Ajustes de UX basados en feedback.
- Configuración de producción en AWS.
- Publicación en Stores.

## Ideas

## Estrategia de Verificación de Ubicación

Para evitar pines falsos, se implementarán las siguientes reglas de validación:

- **Para Negocios Físicos (Entidades):**
  1. **Prueba Visual:** Foto de la fachada del local con el nombre visible.
  2. **GPS Check-in:** El administrador debe realizar el registro del pin estando físicamente en el local (validación por coordenadas del dispositivo).
  3. **Documental:** Carga de recibo de servicios públicos o RUT.
- **Para Perfiles Independientes:**
  1. **Identidad:** Foto de documento de identidad + Selfie (tipo KYC básico).
  2. **Zona de Cobertura:** En lugar de un punto exacto, se valida que vivan en la ciudad mediante un recibo de servicios, pero en el mapa se muestra un radio de acción.
- **Sello de Confianza:** Los perfiles que completen la validación tendrán un pin con un check dorado, dándoles prioridad en el buscador.

## Diferenciación Visual en Mapa

- **Iconografía:** Pines personalizados por categoría.
- **Estados Activos:** Borde de color en el pin si hay una historia de 24h activa.

## Roadmap Futuro (V2 & V3)

- **Gestión Multi-Sede (V2):** Múltiples ubicaciones por marca.
- **Gestión de Equipos (V2):** Colaboradores para chats.
- **Pagos de Servicios (V3):** Transacciones completas in-app.

## Preguntas Abiertas

- **Límite de Negocios:** Un usuario podrá crear una cantidad ilimitada de negocios bajo su cuenta.
- **Moderación:** ¿Cuánto tiempo máximo daremos al equipo de Admin para aprobar una nueva ubicación? (Establecido: 48 horas inicialmente).

## Riesgos y Mitigación

Riesgo (El problema)	Impacto	Plan de Mitigación (La solución)
Latencia en el Chat	Medio	Implementar Redis Pub/Sub para escalar los mensajes en el backend de Go.
Mezcla de datos	Alto	Middleware estricto en el backend para validar <code>user_id</code> vs <code>entity_id</code> .
Pines Fraudulentos	Alto	Implementar sistema de reportes por parte de la comunidad y verificación por GPS obligatoria al crear el pin.