

INGEMA CONSULTORES S.A.C.

MANUAL TECNICO INTERNO

# INGE+ iOS

Actualizacion para nuevo servidor, Web central y sincronizacion Web / Android / iOS

Empresa	INGEMA CONSULTORES S.A.C.
Proyecto	InGe+ Web + Mobile
Documento	INGE+ iOS
Dirigido a	Desarrollador iOS / Responsable Capacitor, Codemagic, IPA / IA de apoyo tecnico
Fecha	07 / 07 / 2026
Revision	Rev. 01 - Actualizacion alineada al avance Web/Supabase
Codigo	MAN-PRG-INGE-IOS-SYNC-02

Uso interno. Manual preparado para el desarrollador y para la IA que lo asista en su chat tecnico. **No contiene claves reales.**

## Como debe usar este manual el desarrollador y su IA

Este documento no es solo una guía humana. También debe cargarse en el chat de la IA que apoye al desarrollador mobile. La IA debe usarlo como contexto principal antes de proponer código, migraciones, pantallas o cambios de arquitectura.

El punto más importante es que Android, iOS y Web no pueden avanzar como tres aplicaciones separadas. InGe+ debe comportarse como un solo ecosistema: mismo Supabase, mismas tablas, mismos estados, mismas rutas de Storage, misma lógica de usuarios y mismo criterio offline.

### Prompt recomendado para pegar en el chat IA del desarrollador

Actúa como asistente técnico del desarrollo iOS de InGe+.

Antes de escribir código, lee este manual completo y respeta estas reglas:

1. La fuente de verdad es Supabase/Postgres usado por InGe+ Web.
2. No inventes tablas, campos, rutas, estados ni permisos.
3. No uses `service_role`, `secret keys` ni claves administrativas en cliente móvil.
4. La app móvil debe soportar modo offline tipo Dropbox: selección local, cola de sincronización, estados y conflictos.
5. Todo cambio debe ser compatible con Web, Android e iOS.
6. Si falta el esquema actual de Supabase o el archivo de tipos, solicita esos datos antes de programar.
7. Trabaja por hitos pequeños y entrega checklist QA por cada cambio.

## Resumen ejecutivo actualizado desde el desarrollo Web

- El proyecto InGe+ ya no debe pensarse como Desktop + Mobile. La dirección actual es Web + Mobile.
- La Web es la plataforma central siempre actualizada: login, perfiles, proyectos, detalle de proyecto, calicatas por proyecto, ficha individual editable y documentos por proyecto.
- Android e iOS son clientes de campo: captura, consulta, fotos, GPS, offline y sincronización.
- El software de PC antiguo queda como referencia funcional/visual, no como fuente de verdad de datos.
- La base central actual de trabajo es Supabase: Auth, Postgres, Storage, RLS y ambientes DEV/STAGING/PROD.
- Las fichas técnicas como calicatas deben tratarse como objetos internos de InGe+, parecidos conceptualmente a archivos editables: tienen metadata, contenido, versión, estado, historial, permisos y posible exportación a Excel/PDF.

## Estado actual de InGe+ Web que debe respetar Mobile

Bloque Web	Estado actual	Impacto para Mobile
<b>Auth + profiles</b>	Login con Supabase Auth y validación de <code>profiles.status = ACTIVO</code> .	Mobile debe repetir el mismo flujo: no basta con login de Auth.
<b>Projects</b>	Listado, estadísticas, creación desde Web, detalle por id y breadcrumb.	Mobile debe listar/abrir los mismos <code>project.id</code> y no crear IDs locales finales distintos.
<b>Project detail</b>	Ruta <code>/projects/[id]</code> con módulos Calicatas y Documentos.	Mobile debe navegar por proyecto y asociar todo a <code>project_id</code> .
<b>Calicatas</b>	Tabla <code>public.calicatas</code> creada; lista por	Mobile debe crear/editar calicatas contra

	proyecto; ficha individual; edicion basica.	la misma tabla y conservar modo offline.
<b>Documents</b>	Ruta por proyecto ya existe como contenedor inicial; repositorio real queda para siguiente fase.	Mobile debe preparar document_nodes/Storage, pero no inventar estructura distinta.
<b>Breadcrumb</b>	Referencia visual de ubicacion: Proyectos /Codigo + nombre / Seccion.	Mobile debe usar etiquetas equivalentes para que el usuario sepa en que proyecto/modulo esta.

## Esquema central vigente y esquema planificado

Para evitar divergencias, el equipo mobile debe separar dos niveles: tablas ya activas en el avance Web y tablas planificadas para siguientes hitos. Lo activo se puede consumir ahora; lo planificado debe prepararse sin romper la compatibilidad.

### Tablas activas del hito actual

Tabla	Campos base esperados	Regla operativa
<b>profiles</b>	id, email, full_name, role, status, position, phone, avatar_url, created_at, updated_at	Validar status = ACTIVO antes de permitir acceso.
<b>projects</b>	id, code, name, client, location, description, status, created_by, created_at, updated_at	Todo modulo tecnico debe colgar de project_id.
<b>calicatas</b>	id, project_id, code, title, location, progresiva, easting, northing, altitude_m, depth_m, groundwater_depth_m, description, observations, status, created_by, created_at, updated_at	Es la primera ficha tecnica conectada. La estructura final puede ampliarse despues.

### Tablas planificadas que Mobile debe considerar

Tabla/Grupo	Uso previsto	Estado
<b>document_nodes</b>	Arbol documental tipo Dropbox/Drive: carpetas, archivos, fichas/modulos, rutas y metadata.	Siguiente fase.
<b>document_versions</b>	Versiones de documentos y archivos exportados.	Siguiente fase.
<b>file_sync_states</b>	Estado offline por usuario, dispositivo, proyecto, carpeta o archivo.	Clave para modo offline.
<b>calicata_cortes</b>	Estratos/cortes/capas de cada calicata.	Fase de ficha tecnica completa.
<b>calicata_photos</b>	Fotos clasificadas, GPS, sello, ruta Storage y metadata.	Fase fotos/Storage.
<b>exports</b>	Excel/PDF generados desde Web o	Fase exportacion.

	backend.	
activity_logs	Auditoria de crear, editar, eliminar, exportar y sincronizar.	Obligatorio antes de produccion.
sync_queue	Registro central/opcional de operaciones pendientes o eventos de sync.	Complementa la cola local mobile.

## Roles, permisos y estados compartidos

Rol	Uso mobile
GERENCIA	Puede revisar, aprobar y consultar informacion amplia.
ADMINISTRADOR_GENERAL	Gestiona configuracion, usuarios, proyectos y control interno.
COORDINADOR_PROYECTO	Controla proyecto, revisa fichas y coordina campo.
ADMINISTRACION	Documentos administrativos, cuentas, reportes y valorizaciones.
LOGISTICA	Recursos, proveedores, inventario y evidencia logistica.
ESPECIALISTA_TECNICO	Revisa y edita fichas tecnicas.
CAMPO	Captura fichas, fotos, GPS y datos en campo.
ASISTENTE	Carga registros, apoya fichas y documentos.
LECTOR	Solo consulta.
Estado de ficha/documento	Significado
BORRADOR	Editable, no enviado a revision.
EN_REVISION	Pendiente de revision tecnica o administrativa.
OBSERVADO	Tiene correcciones solicitadas.
REVISADO	Verificado por responsable tecnico/coordinacion.
APROBADO	Validado por responsable autorizado.
EXPORTADO	Tiene Excel/PDF generado y registrado.
ARCHIVADO	Cerrado historicamente, editable solo con trazabilidad.

## Modo offline tipo Dropbox: criterio actualizado

La duda del equipo mobile debe resolverse con esta regla: InGe+ Mobile no debe descargar toda la empresa ni depender solo de cache. Debe permitir seleccionar proyectos, carpetas, documentos o fichas como disponibles sin conexion, igual que Dropbox permite marcar contenido para uso offline.

La diferencia tecnica es que en InGe+ hay datos estructurados, no solo archivos. Por eso se necesita base local, almacenamiento local de archivos y una cola de sincronizacion.

Concepto	Comportamiento esperado
Cache temporal	Puede existir para acelerar pantallas, pero no garantiza trabajo offline.
Disponible offline	El usuario lo selecciona. Se guarda durablemente metadata, ficha y archivos necesarios.
Siempre sincronizado	La app mantiene ese proyecto/carpeta actualizado automaticamente cuando haya red.
Local only	Dato creado en campo que todavia no subio.
Pending upload	Cambio local pendiente de subir.
Pending download	Cambio remoto pendiente de bajar.
Synced	Servidor y local coinciden.
Conflict	Cambio local y remoto simultaneo; no sobrescribir.
Error	Fallo registrado con reintento y mensaje tecnico.

### Flujo obligatorio offline

1. Usuario inicia sesion online al menos una vez y se valida en profiles.
2. La app descarga perfil, proyectos asignados y estructura minima.
3. Usuario marca proyecto, carpeta, documento o modulo como disponible offline.
4. La app descarga metadata y archivos seleccionados, no todo el servidor.
5. En campo, toda creacion o edicion se guarda primero localmente.
6. Cada cambio genera un item en sync queue local.
7. Al recuperar internet, se suben primero datos estructurados y luego archivos/fotos.
8. Despues de subir, se confirma updated\_at/version y se marca synced.
9. Si hay cambio remoto incompatible, se marca conflict y se evita pisar informacion.

### Modelo local minimo

Entidad local	Contenido minimo
LocalProfile	id, email, full_name, role, status, cachedAt.
LocalProject	id, code, name, client, location, status, syncMode, lastSyncedAt.
LocalCalicata	id, project_id, code, payload campos, status, localVersion, remoteUpdatedAt, syncStatus.
LocalDocumentNode	id, project_id, parent_id, name, type, storagePath, localPath, syncStatus.
LocalFile/Photo	id, entityType, entityId, localPath, storagePath, gpsJson, mimeType, syncStatus.
SyncQueueItem	id, operation, entityType, entityId, payloadJson, retryCount,

	lastError, createdAt.
<b>LocalConflict</b>	entityType, entityId, localPayload, remotePayload, detectedAt, resolutionStatus.

## Algoritmo recomendado de sincronizacion

Orden	Operacion	Regla
1	Validar sesion	Si no hay access token valido, refrescar o pedir login.
2	Subir cambios locales de datos	INSERT/UPDATE de projects/calicitatas/document_nodes segun corresponda.
3	Subir archivos/fotos	Storage despues de tener id remoto confirmado.
4	Actualizar metadata de archivos	Guardar storagePath, version, checksum si se implementa.
5	Descargar cambios remotos	Traer registros modificados desde lastSyncedAt.
6	Detectar conflictos	Comparar updated_at/version/localDirty. Nunca pisar silenciosamente.
7	Actualizar estados locales	synced, conflict o error con lastError.

### Regla de conflicto

Si la ficha local fue editada offline y la misma ficha tambien fue modificada en Web/iOS/Android antes de sincronizar, la app no debe escoger automaticamente una version. Debe marcar conflicto y conservar ambas cargas de datos hasta que se resuelva desde Web o desde una pantalla de resolucion controlada.

## Storage y rutas compartidas

Uso	Ruta recomendada
<b>Documentos de proyecto</b>	project-files/orgs/{org_id}/projects/{project_id}/documents/...
<b>Fotos de calicata</b>	project-files/orgs/{org_id}/projects/{project_id}/calicatas/{calicata_id}/photos/...
<b>Exportaciones</b>	exports/orgs/{org_id}/projects/{project_id}/...
<b>Avatares</b>	avatars/users/{user_id}/avatar.png
<b>Plantillas</b>	templates/...

- La tabla debe guardar metadata; Storage guarda el binario.
- No usar ruta como unica fuente de verdad.

- No subir archivos sin project\_id o entity\_id asociado.
- No guardar archivos pesados como base64 dentro de Postgres.

## Indicaciones específicas para iOS

Si el proyecto iOS es...	Recomendacion
Capacitor/WebView	Mantener Capacitor + Supabase JS, variables por ambiente, SQLite local mediante plugin, Filesystem para fotos/documentos, Network plugin y Codemagic.
Swift nativo	Xcode, SwiftUI, URLSession/Supabase client, Core Data/SQLite-GRDB, Keychain, CoreLocation, PhotosUI/AVFoundation.
Hibrido temporal	No duplicar logica de servidor. Centralizar config y usar las mismas tablas y estados que Web/Android.

## Estructura recomendada iOS/Capacitor

```

ingepius-ios/
  www/
  config/server-config.js
  lib/supabaseClient.js
  data/localDb.js
  data/syncQueue.js
  features/projects/
  features/calicatas/
  features/documents/
ios/App/
capacitor.config.json
codemagic.yaml

```

## Variables Codemagic/ambiente

Variable	Uso
APP_ENV	DEV, STAGING o PROD.
SUPABASE_URL	URL publica del proyecto Supabase del ambiente.
SUPABASE_ANON_KEY o SUPABASE_PUBLISHABLE_KEY	Clave publica para cliente. No service_role.
WEB_BASE_URL	URL de la Web si se abren deep links o enlaces internos.
BUNDLE_ID	Identificador iOS aprobado.

## Hitos iOS recomendados

Hito	Objetivo	Cierre
I1	Config DEV/STAGING/PROD y eliminar servidor antiguo.	IPA muestra ambiente correcto.
I2	Auth + profiles.	Usuario ACTIVO entra; no ACTIVO queda bloqueado.
I3	Projects + detalle.	Mismos proyectos que Web y Android.
I4	Calicatas por proyecto.	Lista, abre, crea y edita ficha basica.
I5	Offline local.	Base local y cola de sync sobreviven cierre de app.
I6	Fotos/GPS/Storage.	Permisos iOS y Storage funcionando con metadata.

## Advertencias iOS sobre offline

- iOS limita procesos en segundo plano. La sincronizacion debe ejecutarse al abrir la app, al volver a primer plano y cuando Network detecte conectividad.
- No depender de que iOS mantenga subidas largas en background si el usuario cierra la app.
- Los tokens deben guardarse en Keychain si la implementacion es nativa; en Capacitor, usar almacenamiento seguro disponible o estrategia validada.
- Las fotos/documentos offline deben guardarse en Filesystem/app storage, no solo en memoria o cache de WebView.

## Ejemplo JS conceptual para cliente Supabase

```
const supabase = createClient(
  window.INGEPLUS_CONFIG.supabaseUrl,
  window.INGEPLUS_CONFIG.supabaseAnonKey,
  { auth: { persistSession: true, autoRefreshToken: true } }
);

const { data: projects } = await supabase
  .from('projects')
  .select('id, code, name, client, location, status, created_at, updated_at')
  .order('created_at', { ascending: false });
```

## Seguridad obligatoria

Prohibido	Motivo	Alternativa correcta
service_role o secret key en mobile	Compromete toda la base si se extrae la APK/IPA.	Usar publishable/anon key + token de usuario + RLS.

<b>Desactivar RLS para que funcione rapido</b>	Expone informacion interna.	Crear policias por authenticated, role y project_members.
<b>Crear tablas distintas para mobile</b>	Rompe sincronizacion Web/Android/iOS.	Usar el esquema central o pedir migracion formal.
<b>Probar contra PROD sin aprobacion</b>	Riesgo sobre datos reales.	Usar DEV y luego STAGING.
<b>Sobrescribir conflictos automaticamente</b>	Puede perder datos de campo.	Marcar conflict y resolver con trazabilidad.

## QA cruzado Web + Android + iOS

Prueba	Accion	Resultado esperado
<b>Web -&gt; Mobile</b>	Crear proyecto desde Web.	iOS lo ve al sincronizar.
<b>Mobile -&gt; Web</b>	Crear calicata desde iOS, primero online y luego offline.	Web la muestra dentro del proyecto correcto.
<b>Offline real</b>	Activar modo avion, editar ficha y tomar foto.	No se pierde informacion y queda pending_upload.
<b>Reconexion</b>	Desactivar modo avion.	Datos suben, archivos suben y estado queda synced.
<b>Conflicto</b>	Editar la misma ficha en Web y mobile antes de sincronizar.	Se marca conflict; no se pisa una version.
<b>RLS</b>	Usuario no ACTIVO o sin permiso intenta entrar.	No accede a modulos internos.
<b>Entorno</b>	Confirmar DEV/STAGING/PROD visible en pantalla debug interna.	No hay confusion de servidor.

## Entregables obligatorios del desarrollador

- Rama y commits separados por hito: config, auth, projects, calicatas, offline, storage, QA.
- Lista de archivos modificados y descripcion breve de cada cambio.
- Evidencia de que ya no quedan URLs o endpoints del servidor antiguo.
- Capturas o video corto de login, perfil, proyectos, calicatas y cierre de sesion.
- Prueba de modo offline real con internet cortado y posterior sincronizacion.
- Reporte de errores pendientes clasificados: schema, RLS, Storage, permisos, UI, sync o build.
- APK/IPA de QA segun plataforma, compilado contra DEV y no contra PROD.

### Criterio de cierre Rev. 01

El hito iOS se considera cerrado solo cuando el cliente mobile se conecta al mismo servidor que la Web, valida profiles, lista projects, abre calicatas por project\_id, puede trabajar un registro en modo offline local y luego sincroniza sin duplicar ni perder datos.

## Checklist rapido para iniciar el siguiente chat de iOS

- Cargar este PDF en el chat del desarrollador iOS.
- Adjuntar tambien el esquema actual de Supabase o `src/types/database.ts` actualizado.
- Pedir a la IA que confirme si la app sigue en Capacitor o si se migrara a Swift nativo.
- Configurar Codemagic con variables de entorno; no pegar claves reales en repositorio.
- Primer entregable: IPA DEV conectado a Supabase, Auth + profiles + projects, sin servidor antiguo.