

Análisis Profundo y Plan de Reconstrucción (SGMI -> Nuevo Proyecto)

1. Análisis del Proyecto Actual (SGMI)

El proyecto actual es una aplicación monolítica que gestiona miembros institucionales, importa datos desde CSV/Google Sheets y genera carnets digitales (PDF con domPDF/wkhtmltopdf) con funcionalidades de login mediante código QR.

Problemas Arquitectónicos y "Vibe Coding" Detectados:

- **Base de Datos Monolítica:** Todo está centralizado en la tabla `users` (más de 20 campos de negocio como `cedula`, `numero_carnet`, `profesion`, `estado_carnet`, mezclados con credenciales de acceso). Esto viola el Principio de Responsabilidad Única.
- **Acoplamiento de Lógica:** Controladores masivos manejando lógica de subida de archivos, parseo de CSV o diagnósticos (`MemberCsvImportController`, `MemberCardController`).
- **Autenticación Hacky:** El login por QR (`MemberQrLoginController`) usa sesiones temporales forzadas y tokens en la tabla `users`, una práctica riesgosa.
- **Frontend Acoplado:** Uso de Livewire (Volt/Flux) mezclado con vistas Blade regulares (`resources/js/app.js`, `routes/web.php`).

2. Propuesta de Nueva Arquitectura Profesional

El nuevo proyecto eliminará la deuda técnica adoptando el stack solicitado y basándose 100% en la **documentación oficial de Laravel 12**:

- **Backend / Core API:** Laravel 12 (Modo API o Inertia).
- **Frontend:** React Starter Kit (Laravel Breeze/Inertia React).
- **Autenticación:** WorkOS (B2B SSO, Passwordless, o gestión de identidades), delegando el manejo de sesiones inseguras.
- **Caché / Colas / Sesiones:** Valkey (High-performance KV store en reemplazo de Redis).
- **Gestión de Contenido:** Strapi CMS para contenido no transaccional (noticias, plantillas de carnets, emails).

Estructura de Datos Mejorada (Data Modeling)

Se debe normalizar la base de datos para separar "Identidad" de "Negocio":

1. `users` (Tabla reducida administrada en sincronización con WorkOS): `id`, `workos_id`, `email`, `name`, `role`.
2. `member_profiles` (Nueva Entidad de Negocio): `id`, `user_id`, `cedula`, `tipo_documento`, `genero`, `fecha_nacimiento`, `telefono`.
3. `member_professional_data`: `member_profile_id`, `profesion`, `empresa`, `cargo`.
4. `digital_cards` (Carnets): `id`, `member_profile_id`, `numero_carnet`, `fecha_emision`, `fecha_vencimiento`, `estado_carnet`, `qr_token`, `verification_hash`.

3. Plan de Acción por Fases

Fase 1: Inicialización y Foundation (Infraestructura)

- Instalar Laravel 12 puro (`laravel new`) y React Starter Kit (Breeze/Inertia).
- Configurar Valkey como driver principal de caché, sesiones y colas (según docs oficiales).

- Configurar entorno para integrar Strapi (API Keys, Webhooks).

Fase 2: Integración de Identidad (WorkOS)

- Configurar autenticación mediante Socialite extendido o SDK de WorkOS.
- Modelar tabla `users` para que funcione como espejo de WorkOS Auth.
- Implementar Middleware y Policies (Authorization) basados en roles nativos o sincronizados desde WorkOS.

Fase 3: Modelado de Identidad y Base de Datos (Negocio)

- Crear migraciones, factorías y modelos estrictos con Tipado Fuerte de PHP 8.3/8.4.
- Implementar patrón Repository/Service para abstraer la lógica de negocio.
- Crear las tablas `member_profiles` y `digital_cards`.

Fase 4: Refactorización de Automatizaciones (Scripts de Importación)

- Trasladar `GoogleSheetsSyncService` a un Job asíncrono gestionado a través de Valkey (Colas).
- Implementar el procesamiento en lotes (Batches) nativo de Laravel para la importación de CSV.

Fase 5: Desarrollo Frontend (React)

- Creación de componentes UI modulares en React, consumiendo la API de Laravel o mediante Inertia.
- Interfaces protegidas y generación moderna de QR en el cliente (evitando cargas de servidor).

4. Master Prompt para la Inicialización del Nuevo Proyecto

Copia y pega el siguiente prompt cuando tengas el nuevo proyecto en blanco (Laravel 12 + React) en el IDE del AI:

Rol: Actúa como un Lead Full Stack Architect experto en Laravel 12, React, Valkey, WorkOS y Strapi CMS. Tu objetivo es construir un sistema de Gestión de Miembros institucional desde cero, escribiendo un código inmaculado, usando tipado estricto (PHP 8.3+) y patrones de diseño empresariales que eliminen cualquier rastro de "vibe coding".

Reglas Críticas de Desarrollo:

1. **Documentación Oficial:** DEBES basar absolutamente toda la implementación backend consultando previamente el directorio local `docs/docs-12.x`. No asumas funciones deprecadas de Laravel 10/11.
2. **Arquitectura MVC / Servicios:** Ningún controlador debe exceder las 50 líneas. Extrae la lógica compleja a Clases de Servicio (`app/Services`) y el procesamiento masivo a Jobs (`app/Jobs`) que correrán sobre **Valkey**.
3. **Base de Datos Normalizada:** NO agregues atributos de perfiles a la tabla `users`. Crea un modelo relacional estricto: `User` (Identity via WorkOS) -> `MemberProfile` (Demographics) -> `DigitalCard` (Credentials & QR).
4. **Autenticación:** Implementa el scaffolding para usar WorkOS. Ignora los métodos tradicionales basados solo en base de datos.
5. **Frontend:** Usa únicamente React (Inertia o SPA puro). Prohibido el uso de Livewire/Volt/Blade para lógica visual. TailwindCSS v4 será la base del diseño.

Fase 1 - Ejecución Inmediata:

- Lee el contexto de `docs/docs-12.x/cache.md` y `docs/docs-12.x/queues.md` para configurar correctamente los servicios de **Valkey**.

- *Genera los Modelos, Migraciones (con Foreign Keys estrictas) y Factories para: `User` , `MemberProfile` , `DigitalCard` y `ImportLog` .*
 - *Prepara un `GoogleSheetsImportJob` en forma de Batch processing usando la API nativa de colas de Laravel 12.*
 - *Al terminar, preséntame el reporte de lo construido para proceder a la escritura de la lógica y endpoints React.*
-

Verification Plan

Manual Verification

- Solicitar al usuario revisión de este documento (`implementation_plan.md`) para confirmar que el diseño de bases de datos, stack sugerido (WorkOS, Valkey, Strapi) y el Master Prompt satisfacen su visión técnica del nuevo proyecto.
- Ningún código del proyecto original `sgmi` ha sido ni será modificado, tal como fue solicitado.