

## # 📄 Documento Final: Sistema SaaS de Béisbol (DRS)

### ## 1. Introducción

Este documento es la hoja de ruta definitiva para el desarrollo del ecosistema \*\*BB-SaaS\*\*. Define las responsabilidades de cada actor y la estructura técnica para evitar ambigüedades en la implementación.

### ## 2. Objetivos del Sistema

- \* Operar bajo un modelo \*\*Multi-tenant\*\* donde la Liga es la raíz.
- \* Descentralizar la carga de datos hacia los equipos.
- \* Proveer un portal público de alto impacto visual (estilo MLB) para consulta de estadísticas.

### ## 3. Stakeholders y Roles

1. \*\*Administrador SaaS:\*\* Dueño de la plataforma.
2. \*\*Dueño de Liga (Tenant):\*\* Cliente principal, crea equipos y gestiona suscripciones.
3. \*\*Dueño de Equipo:\*\* Usuario administrativo que gestiona su personal (Secretaría/Coaches).
4. \*\*Secretaría:\*\* Encargada del registro de \*\*Rosters (Jugadores)\*\* y documentos.
5. \*\*Coach:\*\* Encargado del \*\*Registro de Estadísticas\*\* y asistencia en campo.

### ## 4. Requerimientos Funcionales: El Flujo de Trabajo

- \* \*\*La Liga\*\* registra un equipo y el sistema envía credenciales al \*\*Dueño de Equipo\*\*.
- \* \*\*El Dueño de Equipo\*\* crea accesos para su \*\*Secretaría\*\* y sus \*\*Coaches\*\*.
- \* \*\*La Secretaría\*\* da de alta a los jugadores.
- \* \*\*El Coach\*\* registra los eventos de cada partido.

### ## 5. Requerimientos No Funcionales

- \* \*\*Trazabilidad:\*\* Cada dato guardado registra quién lo creó (Secretaría o Coach).
- \* \*\*Aislamiento:\*\* El personal de un equipo no puede ver la administración interna de otro.

### ## 6. Arquitectura del Sistema

Basada en Laravel y FilamentPHP. La base de datos utiliza tablas pivot para manejar la \*\*Jerarquía Flexible\*\* (un jugador en múltiples categorías).

### ## 7. Entidades y Jerarquía Operativa

La estructura sigue este orden: \*\*Liga > Equipo > Personal Operativo > Sub-equipo/Categoría > Jugador.\*\*

### ## 8. Gestión de Usuarios y Notificaciones

El sistema automatiza el envío de credenciales por correo electrónico en cada nivel de la cadena, asegurando que cada nuevo usuario reciba su enlace de acceso al panel correspondiente.

## ## 9. Registro de Estadísticas

El Coach es el responsable de la carga de acciones (Hits, Carreras, Errores). El sistema consolida estos datos bajo el contexto: `Jugador + Sub-equipo + Categoría + Temporada`.

## ## 10. Modelo de Suscripciones (SaaS)

La Liga es el pagador. El sistema maneja tres estados:

- \* \*\*Activa:\*\* Todo funciona.
- \* \*\*En Gracia:\*\* Acceso con advertencias de pago.
- \* \*\*Suspendida:\*\* Bloqueo total para la Liga y todos sus Equipos dependientes.

## ## 11. Reglas de Visibilidad

- \* \*\*Público:\*\* Rankings, perfiles y estadísticas globales (con Ads).
- \* \*\*Privado:\*\* Gestión técnica y reportes detallados del equipo (sin Ads).

## ## 12. Modelo de Datos (Esquema)

Relación N:M entre jugadores, categorías y sub-equipos para permitir que un mismo atleta juegue en diferentes niveles según las reglas de la liga.

## ## 13. Reglas de Oro

1. \*\*La Liga es el Tenant.\*\*
2. El Dueño de Equipo gestiona personas; el Staff gestiona datos.
3. Toda suspensión es reversible y \*\*no destruye datos\*\*.
4. Si una funcionalidad no está en este documento, no se desarrolla.

## ## 14. Roadmap de Desarrollo

1. \*\*Fundaciones:\*\* Multi-tenancy y Roles.
2. \*\*Suscripciones:\*\* Pasarela de pagos para Ligas.
3. \*\*Gestión de Equipos:\*\* Invitaciones y delegación a Secretaría/Coaches.
4. \*\*Rosters:\*\* Carga de jugadores.
5. \*\*Estadísticas:\*\* Motor de registro de jugadas.
6. \*\*Portal Público:\*\* SEO y Visualización MLB.
7. \*\*Monetización:\*\* Integración de Ads.
8. \*\*Analítica:\*\* Dashboards de rendimiento avanzado.

---