



Documento de Requerimientos del Sistema

BB-SaaS – Plataforma SaaS de Gestión y Analítica de Béisbol

1. Introducción

Este documento define de forma **definitiva y extensible** los requerimientos funcionales, no funcionales y conceptuales del sistema **BB-SaaS**, una plataforma SaaS orientada a la gestión operativa, estadística y analítica de ligas de béisbol.

El objetivo principal es construir un producto **escalable, auditabile y comercialmente sostenible**, evitando ambigüedades técnicas y reduciendo deuda futura.

2. Objetivos del Sistema

- Operar bajo un modelo **SaaS Multi-tenant**, donde **la Liga es el Tenant raíz**.
 - Delegar la carga operativa hacia equipos y staff.
 - Proveer un **portal público de alto impacto visual** (estilo MLB).
 - Permitir estadísticas históricas, comparables y recalcables.
 - Facilitar monetización progresiva por planes y módulos.
-

3. Stakeholders y Roles

3.1 Roles Globales

1. Administrador SaaS

Dueño de la plataforma. Control total del sistema, planes y monitoreo.

2. Dueño de Liga (Tenant)

Cliente principal. Administra equipos, temporadas, competencias y suscripción.

3.2 Roles Operativos

1. Dueño de Equipo

Administrador interno del equipo. Gestiona staff y participa en competencias.

2. Secretaría

Responsable del registro administrativo:

3. Jugadores

4. Documentación

5. Inscripciones por temporada/categoría

6. Coach

Responsable **exclusivo** del registro de estadísticas de juego.

4. Principios de Diseño (Reglas de Oro)

1. **La Liga es el Tenant.**
 2. El Dueño de Equipo gestiona personas; el Staff gestiona datos.
 3. Las estadísticas **no se editan**, se corrigen mediante eventos compensatorios.
 4. Toda suspensión es reversible y **no destruye datos**.
 5. Si una funcionalidad no está en este documento, no se desarrolla.
-

5. Modelo Operativo y Flujo de Trabajo

1. La **Liga** se registra y activa su suscripción.
 2. La Liga crea **Temporadas**.
 3. Dentro de una temporada se crean **Competencias / Torneos**.
 4. Las competencias definen **Categorías**.
 5. La Liga registra o invita **Equipos** a competir.
 6. El sistema envía credenciales al **Dueño de Equipo**.
 7. El Dueño de Equipo crea accesos para Secretaría y Coaches.
 8. La Secretaría registra jugadores y los inscribe por categoría y temporada.
 9. El Coach registra eventos de juego durante los partidos.
-

6. Arquitectura Conceptual

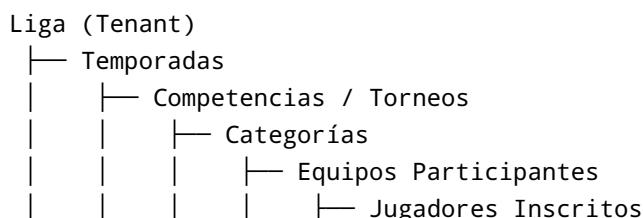
6.1 Separación de Estructuras

Estructura Organizativa * Liga * Equipo * Usuarios / Roles

Estructura Deportiva * Temporadas * Competencias * Categorías * Partidos

Esta separación garantiza flexibilidad, reutilización y escalabilidad.

7. Jerarquía del Sistema



Un jugador puede:

- * Estar en múltiples categorías
- * Participar en distintas temporadas
- * Cambiar de equipo entre temporadas

8. Gestión de Usuarios y Permisos

- Los usuarios **no están atados permanentemente a una Liga**.
- Las relaciones usuario-liga-equipo son **N:M**.
- Los permisos son **contextuales**, definidos por:
 - Liga
 - Temporada
 - Competencia
 - Equipo

Ejemplos:

- * Un Coach solo puede registrar estadísticas en su categoría asignada.
- * La Secretaría no puede modificar estadísticas.

9. Modelo de Estadísticas (Event-Driven)

9.1 Principio Fundamental

Las estadísticas **no se almacenan como totales**, se **derivan de eventos**.

9.2 Modelo de Eventos

```
Partido
  └── Innings
      └── Plays (Eventos)
          ├── tipo (hit, strike, run, error, etc.)
          ├── jugador
          ├── pitcher (opcional)
          └── metadata (JSON)
```

9.3 Cálculo de Estadísticas

- Las estadísticas se calculan dinámicamente.
 - Se cachean para rendimiento.
 - Pueden recalcularse ante cambios o correcciones.
-

10. Reglas de Trazabilidad y Auditoría

- Todo registro guarda:
- Usuario creador
- Rol
- Contexto (liga, temporada, equipo)
- Correcciones generan nuevos eventos, nunca sobrescritura.

11. Modelo de Suscripciones SaaS

La Liga es el único pagador.

Estados de Suscripción

- **Activa:** Acceso completo según plan.
- **En Gracia:** Acceso limitado con advertencias.
- **Suspendida:** Bloqueo total del tenant.

Límites por Plan (ejemplos)

- Número de equipos
- Temporadas activas
- Acceso a estadísticas avanzadas
- Exportación de datos
- Acceso a API

12. Reglas de Visibilidad

Público (con Ads)

- Rankings
- Estadísticas agregadas
- Perfil básico del jugador

Privado (sin Ads)

- Gestión operativa
- Reportes detallados
- Analítica avanzada
- Historial sensible

13. Modelo de Datos (Resumen)

- Relaciones N:M:
- Jugadores ↔ Categorías
- Jugadores ↔ Equipos
- Usuarios ↔ Ligas
- Uso de tablas pivot con contexto temporal.
- Metadata flexible en eventos deportivos.

14. Gestión de Juegos y Programación de Calendarios

14.1 Principio Rector

El sistema sigue el principio:

El sistema propone. La Liga decide.

La programación de juegos es **automática por defecto**, con capacidad de **ajuste manual controlado**, garantizando flexibilidad sin perder integridad de datos.

14.2 Entidad Schedule / Fixture

Cada competencia gestiona uno o más calendarios (**Schedule**), responsables de la generación y organización de los partidos.

Atributos conceptuales:

- Tipo: **automatic** | **manual** | **mixed**
- Estado: **draft** | **published** | **locked**
- Reglas de generación (**ruleset** en formato JSON)
- Fecha de generación

Un calendario en estado **draft** no genera partidos oficiales.

14.3 Modalidades de Programación

a) Automática (Default)

Utilizada para fases regulares y ligas estándar.

La Liga define: * Equipos participantes * Categoría * Fechas disponibles * Días permitidos * Sedes o campos

El sistema genera: * Cruces equilibrados * Asignación de fechas * Balance local / visitante

Antes de publicarse, el calendario puede ser **previsualizado y ajustado manualmente**.

b) Manual

Utilizada para: * Playoffs * Torneos cortos * Invitaciones especiales

Permite: * Crear partidos individuales * Reprogramar fechas * Suspender o cancelar juegos

Todas las acciones manuales pasan por validaciones de pertenencia y contexto competitivo.

c) Mixta

La competencia puede combinar calendarios:

- Fase regular: automática
- Eliminatorias: manual o semi-automática

Todos los calendarios conviven bajo la misma competencia y temporada.

14.4 Estados del Partido

Cada partido mantiene un ciclo de vida controlado:

- scheduled
- confirmed
- in_progress
- suspended
- finished
- voided

Reglas: * Solo partidos `in_progress` aceptan eventos estadísticos * Partidos `voided` no computan estadísticas * Partidos `suspended` conservan los eventos registrados

14.5 Reprogramaciones y Auditoría

Las reprogramaciones no modifican el historial original.

Cada cambio registra: * Fecha anterior * Nueva fecha * Motivo * Usuario responsable

Esto garantiza trazabilidad completa y consistencia histórica.

14.6 Automatización Avanzada (Diseño Futuro)

El sistema queda preparado para: * Generación automática de playoffs * Brackets dinámicos * Semillas basadas en rankings * Rondas de descanso (bye)

Estas funcionalidades se apoyan en estadísticas derivadas y rankings confiables.

15. Roadmap de Desarrollo

1. Fundaciones: Multi-tenancy y RBAC
2. Temporadas, Competencias y Categorías
3. Gestión de Equipos y Staff
4. Registro de Jugadores (Rosters)
5. Motor de Eventos y Partidos
6. Cálculo y Cache de Estadísticas

7. Portal Público (SEO + Visual)
 8. Suscripciones y Monetización
 9. Analítica Avanzada y Dashboards
-

15. Cierre

Este documento define el **contrato funcional y técnico** del sistema BB-SaaS.
Cualquier desarrollo fuera de este alcance deberá ser evaluado como fase posterior o módulo premium.

BB-SaaS se concibe como un producto **profesional, auditabile y escalable**, preparado para ligas amateur hoy y organizaciones profesionales mañana.