

Documento: Qué implica un sistema para administrar hoteles (PMS + E-commerce)

1) Resumen ejecutivo

Un PMS (Property Management System) centraliza operaciones del hotel: reservas, front desk, disponibilidad, facturación, housekeeping, y funciones avanzadas como CRM y channel management. Es un mercado con demanda real para hoteles pequeños/medianos que buscan alternativas más económicas y personalizables frente a productos grandes.

[AltexSoft+1](#)

2) Objetivo del sistema

Crear una plataforma modular que permita:

- Administrar 1 o varios establecimientos (hoteles, hostels, apart-hotels).
 - Vender directamente al huésped (sitio público / e-commerce / motor de reservas).
 - Gestionar consumos internos (F&B, room service, spa) y facturarlos al check-out.
 - Escalar hacia SaaS multi-tenant y conectar con OTAs (Booking, Airbnb, Expedia) mediante un Channel Manager.
-

3) Módulos funcionales (detalle y features clave)

A. Core (esencial)

1. Gestión de establecimientos y habitaciones

- Entidades genéricas: Establecimiento (hotel), Ubicaciones, Plantillas de habitación, Inventario de unidades (habitaciones físicas), atributos (cama, vista, capacidad). (Recomendado pensar en **Establishment** para genérico).

2. Reservas & Motor de venta (Booking Engine)

- Calendario, reglas de tarifas, overbooking protection, políticas de cancelación, reservas grupales. Integración con pasarelas de pago. [Hotel Tech Report](#)

3. Front-desk / Check-in / Check-out

- Asignación de habitaciones, folios, pagos parciales, estancias prolongadas.

4. Facturación y Pagos

- Registro de folios por reserva, impuestos locales, integración con pasarelas (MercadoPago / Stripe / Adyen). Cumplir PCI para tarjetas. [PCI Security Standards Council](#)
-

B. Operaciones internas

5. Housekeeping (Limpieza)

- Estado de habitación, asignación de tareas, mobile dispatch para personal, actualización en tiempo real. Muy importante para evitar errores operativos. [RoomRaccoon+1](#)

6. Mantenimiento

- Tickets, prioridad, historial de reparaciones.

7. Staff & RRHH (roles, turnos)

- Permisos, acceso por rol (recepción, limpieza, cocina, admin).
-

C. Consumos / Servicios (crucial para rentabilidad)

8. Servicios & Consumos (F&B, Spa, Traslados, Extras)

- Menús, órdenes (POS), consumos por habitación, integrable al folio del huésped.
- POS para restaurante/bar con cierre de turno y sincronización con PMS/facturación.

D. Distribución y ventas

9. Channel Manager

- Sincronización en tiempo real con OTAs (Booking, Airbnb, Expedia), evitar doble reserva, reglas de rate parity, control de inventario. Es un módulo complejo pero crítico para maximizar ocupación. www.sirvoy.com+1

10. Direct Booking Engine / E-commerce

- Sitio público por hotel, SEO, pagos directos, upsells (desayuno, traslado).
-

E. Marketing & CRM

11. Clientes / CRM

- Perfiles, historial de estancias, preferencias, segmentación.

12. Programas de fidelidad, cupones y campañas

- Gestión de descuentos, códigos promocionales.
-

F. Inteligencia & Backoffice

13. Reportes y BI

- Ocupación, RevPAR, ADR, ingresos por departamento (habitaciones vs F&B).

14. Integraciones contables

- Exportes para contabilidad, libros fiscales, conectores a sistemas locales.
-

4) Requisitos no funcionales (lo que no se ve, pero es crítico)

- **Seguridad y cumplimiento:** PCI DSS para pagos; cifrado en tránsito/descanso; buenas prácticas OWASP para APIs. (PCI 4.0 obligatorio desde 2025 para ciertos requerimientos). [PCI Security Standards Council+1](#)
 - **Alta disponibilidad / sincronización:** Channel manager y motor de reservas requieren baja latencia y sincronía 2-way.
 - **Multi-tenant:** si pensás SaaS, diseño multi-tenant (tenant isolation, datos por cliente, escalado independiente).
 - **Internacionalización & Fiscalidad:** manejo de monedas, formatos y requisitos fiscales locales (ej. comprobantes fiscales argentinos).
 - **Mobile-first para staff:** housekeeping y mantenimiento con apps/web móviles.
-

5) Integraciones indispensables

- **Pasarelas de pago:** Stripe/Adyen/MercadoPago (según mercado) — atención a tokenización para reducir PCI scope. [Canary Technologies](#)
 - **Channel Managers / OTAs:** conexiones certificadas con Booking.com, Airbnb, Expedia (2-way APIs o iCal cuando no hay otra opción). [Beds24+1](#)
 - **POS / F&B:** integración entre POS y folio de habitación.
 - **Email/SMS/WhatsApp:** notificaciones automáticas.
 - **Contabilidad local:** exportes para sistemas contables o facturación electrónica.
-

6) Arquitectura sugerida (alto nivel)

- **Backend:** API REST/GraphQL (Django/DRF, Node/Express, or Go), microservicios para Channel Manager y Pagos si escala.
- **DB:** PostgreSQL (datos transaccionales), Redis (caching / locks para evitar doble-booking), Elasticsearch (búsqueda/reportes).
- **Frontend:** React (Admin) + React/Tailwind para UI, sitio público SSR se posible (Next.js) para SEO.

- **Mobile/Web para Staff:** PWA o apps ligeras.
 - **Infra:** Cloud (AWS/GCP) con autoscaling, backups y monitorización.
 - **Multi-tenant approach:** esquema por tenant (schemas separados) o fila de tenants en la misma BD con tenant_id — elegir según volumen inicial.
-

7) MVP recomendado (priorizar para salir al mercado rápido)

Objetivo: validar con 1–3 hoteles en 3–6 meses.

Mínimo viable:

1. Gestión genérica de Establecimientos y Habitaciones.
2. Reservas internas + calendario.
3. Motor de reservas público (direct bookings) con pasarela de pago tokenizada.
4. Facturación básica (folio por reserva) y export para contabilidad.
5. Dashboard simple de ocupación e ingresos.

Con eso validás la propuesta de valor y empezás a facturar. Lo de housekeeping, channel manager y POS puede ir en la siguiente fase. (Muchas guías de la industria recomiendan arrancar con core + booking + pagos). [AltexSoft+1](#)

8) Roadmap y priorización (resumen rápido)

- **Phase 0 (planeamiento):** modelos de datos genéricos (**Establishment**, **Unit**, **RatePlan**), wireframes admin + public.
- **Phase 1 (MVP):** core (habitaciones, reservas, pagos, booking engine).
- **Phase 2:** CRM, consumos/servicios (F&B), housekeeping móvil.
- **Phase 3:** Channel Manager, POS integrations, multi-tenant hardening.
- **Phase 4:** BI avanzado, marketplace integraciones, API pública.

9) Costos / tiempo estimado (muy general)

- **Tú solo (full-stack)**: desarrollar MVP realista \approx 4–9 meses según dedicación y si reutilizás libs/boilerplates. Si querés integraciones OTA certificadas, agrega tiempo (puede requerir aprobaciones/contratos).
- **Equipo pequeño (2–3)**: acelera a 2–4 meses.
(Estimación orientativa — depende del alcance y del país)

10) Riesgos y mitigaciones

- **Riesgo**: competir contra players consolidados. → **Mitigar**: nicho (hoteles pequeños regionales), buen soporte local y precios competitivos.
- **Riesgo**: cumplimiento PCI y manejo de datos sensibles. → **Mitigar**: usar tokenización y providers que reduzcan tu scope PCI. [Canary Technologies](#)
- **Riesgo**: integraciones OTA complejas. → **Mitigar**: inicialmente usar iCal + 1-2 integraciones certificadas; luego escalar.

11) Valor comercial y modelo de negocio

- **SaaS mensual por hotel** (ej: ARS/USD variable según mercado y módulos).
- **Módulos premium**: Channel Manager, POS, Channel integrations, Custom reports.
- **Servicios adicionales**: onboarding, soporte premium, integraciones a medida.

12) Fuentes principales consultadas (seleccionadas)

- Guía funcional PMS — AltexSoft / HotelTechReport. [AltexSoft+1](#)
- Channel Manager — Sirvoy / Preno / Beds24 sobre features y 2-way sync.
[www.sirvoy.com+2Preno HQ - Hotel Management Software+2](#)

- Housekeeping modern features — RoomRaccoon / Actabl. [RoomRaccoon+1](#)
- PCI DSS (seguridad pagos) — PCI Security Standards Council y guías. [PCI Security Standards Council+1](#)