# Modelo de datos — Base común del framework

**Objetivo**: unificar configuración, chatbot (auditoría/runtime) y hosting (reservas + planillas) con rutas válidas de Firestore y campos mínimos para operar.

## 1) Colecciones de configuración

/configuracion (colección)  
 /general (doc) → flags y parámetros globales  
 /dashboard\_propiedades (doc) → ancla del dashboard  
 /configuraciones (subcolección)  
 /caracteristicas-departamento (doc) → catálogos/listas comunes  
 /caracteristicas-propiedad (doc)  
 /chatbot (doc)  
 promptsIA: { ... }  
 handlers: { ... }

**Notas** - Subcolecciones siempre cuelgan de un documento (por eso se usa dashboard\_propiedades como ancla). - Usar estructura { slug, label, activo, orden } en listas para estabilidad.

## 2) Chatbot (auditoría + runtime)

/chatbot (colección)  
 /gestion\_reservas (doc)  
 /reservas\_raw (subcolección) ← ventas del bot (auditoría)  
 /reservas\_temporales (subcolección) ← si el flujo lo usa  
 /runtime (doc)  
 /sesiones\_temp (subcolección)

**Notas** - reservas\_raw **no** alimenta Hosting.

## 3) Propiedades (datos reales)

/propiedades (colección)  
 /{propiedadId}  
 api\_key: "<WuBook>"  
 nombre: "..."  
 /departamentos  
 /{idZak} ← suele coincidir con id\_zak\_room  
 nombre: "Depto X"  
 codigo\_depto?: "A-103"  
 wubook\_shortname?: "A103"  
 id\_zak\_room?: 12511

**Notas** - idZak se usa para construir el ID de la reserva canónica.

## 4) Hosting (operativa de anfitriones)

/Hosting (colección)  
 /Reservas (doc)  
 /items (subcolección)  
 /{propId}\_{id\_human}\_{idZak} (doc) ← canónica mínima  
 id\_human: "ABC123"  
 id\_zak\_room: 12511  
 propiedad\_id: "100"  
 propiedad\_nombre: "Arelauquen"  
 codigo\_depto: "A-103"  
 depto\_nombre: "Depto Lago"  
 arrival: "11/09/2025" // dd/mm/yyyy  
 arrival\_iso: "2025-09-11" // yyyy-mm-dd (para filtros)  
 departure: "12/09/2025"  
 departure\_iso: "2025-09-12"  
 nombre\_huesped: "Juan Pérez"  
 adults: 2  
 children: 0  
 source: "WuBook"  
 createdAt, updatedAt  
 /pagos (subcolección)  
 /{pagoId} { monto, moneda, forma, fecha, observacion }  
  
 /Planillas\_diarias (doc)  
 /{YYYY-MM-DD} (subcolección por fecha)  
 /propiedades  
 /{propiedadId}  
 resumen { total\_check\_in, total\_check\_out, last\_updated }  
 /check\_in  
 /{propId}\_{id\_human}\_{idZak}  
 ref\_reserva: <ref /Hosting/Reservas/items/...>  
 snapshot\_min: {  
 codigo\_reserva: "ABC123", // (= id\_human)  
 codigo\_depto: "A-103",  
 nombre\_huesped: "Juan Pérez"  
 }  
 createdAt  
 /check\_out  
 /{propId}\_{id\_human}\_{idZak}  
 ref\_reserva: <ref ...>  
 snapshot\_min: { ... }  
 createdAt

**Notas** - Planillas **no duplican** datos; referencian la reserva y guardan un snapshot\_min para listar rápido. - Formatos de fecha duplicados: humano (dd/mm/yyyy) y filtro (yyyy-mm-dd).

## 5) Índices Firestore sugeridos

* En /Hosting/Reservas/items:
  + (propiedad\_id ASC, arrival\_iso ASC)
  + (propiedad\_id ASC, departure\_iso ASC)

Firestore mostrará un link para crear el índice si hace falta.

## 6) Endpoints previstos

* POST /api/wubookImportRange → llena /Hosting/Reservas/items desde WuBook (día por día).
* POST /api/buildPlanillaDia → genera /Hosting/Planillas\_diarias/{fecha}/propiedades/{propId}/check\_in|check\_out usando arrival\_iso/departure\_iso.

## 7) Reglas de permisos (idea)

* Lectura/escritura Hosting limitada a usuarios con rol host sobre propiedad\_id.
* Auditoría reservas\_raw accesible sólo a admin.
* Catálogos de configuracion en solo lectura para app.

Implementar con Custom Claims de Firebase Auth y reglas por propiedad.

## 8) Consideraciones

* Mantener nombres de colección en **minúsculas** (propiedades, departamentos).
* Evitar subcolecciones profundas salvo casos operativos (pagos, planillas).
* Exportar a BigQuery para reportes históricos si hace falta.