

✓ MINI-FASE 1: Base de Integraciones STR - COMPLETADA

July
17

Fecha de Implementación

Diciembre 6, 2024

🎯 Objetivo

Establecer la infraestructura completa para integraciones con plataformas de alquiler vacacional (STR) como Airbnb, Booking.com, VRBO, etc.

📦 Componentes Implementados

1. Servicio de Integración de Canales ✓

Archivo: `lib/str-channel-integration-service.ts`

Funcionalidades:

- ✓ Conexión/desconexión de canales externos
- ✓ Sincronización de calendario (disponibilidad)
- ✓ Sincronización de precios
- ✓ Importación de reservas desde canales externos
- ✓ Obtención de estado de sincronización
- ✓ Configuración por canal con features soportados

Canales Soportados:

- Airbnb 
- Booking.com 
- VRBO 
- HomeAway 
- Web Propia 
- Expedia 
- TripAdvisor 
- Otros 

2. API Endpoints ✓

POST `/api/str/channels/connect`

- Conecta un listing con un canal externo
- Valida credenciales
- Crea registro de sincronización
- Ejecuta sincronización inicial

POST `/api/str/channels/disconnect`

- Desconecta un canal
- Mantiene el historial

- Marca como inactivo

GET /api/str/channels/[listingId]/status

- Obtiene estado de todos los canales
- Información de última sincronización
- Errores y warnings
- Configuración de cada canal

POST /api/str/channels/[listingId]/sync

- Sincronización manual por tipo:
- calendar : Sincroniza calendario
- bookings : Importa reservas
- prices : Actualiza precios

GET/PUT/DELETE /api/str/listings/[id]

- CRUD completo para listings individuales
- Incluye relaciones (unit, building, channels, bookings, reviews)

3. Interfaz de Usuario

Página de Gestión de Canales

Ruta: /str/listings/[id]/channels

Características:

-  Grid visual de todos los canales
-  Iconos y colores por plataforma
-  Badges de estado (Conectado, Error, Desconectado)
-  Información de última sincronización
-  Botones de acción:
 - Conectar/Desconectar
 - Sincronizar (Calendario, Precios, Reservas)
 - Configurar
 -  Diálogo de configuración con credenciales
 -  Switches para opciones de sincronización
 -  Indicadores de progreso en tiempo real
 -  Notificaciones toast de éxito/error

Página de Detalle de Listing

Ruta: /str/listings/[id]

Características:

-  Vista completa del listing
-  Estadísticas (precio, reservas, rating, ocupación, canales)
-  Tabs de contenido:
 - Información básica
 - Reservas recientes
 - Reseñas
 - Precios y temporadas
 -  Botón destacado “Gestionar Canales”
 -  Navegación fluida

Página de Configuración de Integraciones

Ruta: /str/settings/integrations

Características:

- Dashboard con estadísticas:
- Listings activos
- Canales conectados
- Sincronizaciones activas
- Última sincronización
- 4 tabs de configuración:
 1. **General:** Sincronización automática, intervalos, notificaciones
 2. **Canales:** Configuración de credenciales por canal
 3. **Webhooks:** URLs, eventos, instrucciones
 4. **Avanzado:** Rate limiting, timeouts, reintentos, logs

4. Documentación

Archivo: docs/STR_INTEGRATION_GUIDE.md

Contenido:

- Descripción general del sistema
- Guía de componentes
- Documentación de API endpoints
- Estructura de base de datos
- Modo Demo vs Producción
- Flujos de sincronización
- Manejo de errores
- Estadísticas y monitoreo
- Seguridad
- Roadmap futuro
- FAQ



Modo de Operación: DEMO

Estado Actual: El sistema opera en modo **DEMO/SIMULACIÓN**

Funciona (Simulado):

- Conexión con todos los canales
- Generación de calendario (30 días)
- Creación de reservas demo (2-3 por canal)
- Actualización de precios en calendario local
- Toda la UI y flujos de trabajo
- Gestión de estado y sincronización

No Implementado (Requiere Activación):

- Llamadas reales a APIs externas
- OAuth flows de cada plataforma
- Webhooks reales
- Sincronización bidireccional real

- Rate limiting real de APIs

Para Activar Modo Producción:

1. Obtener credenciales API de cada plataforma
 2. Configurar variables de entorno
 3. Implementar lógica de API real en el servicio
 4. Configurar webhooks en cada plataforma
 5. Implementar manejo de OAuth donde sea necesario
-

Base de Datos

Modelos Existentes Utilizados:

- `STRListing` : Listings de propiedades
- `STRBooking` : Reservas
- `STRCalendar` : Calendario de disponibilidad
- `STRChannelSync` : **Principal** - Configuración de canales
- `STRReview` : Reseñas
- `STRSeasonPricing` : Precios por temporada

Campos Clave en `STRChannelSync`:

```

- canal: ChannelType           // Qué canal (AIRBNB, BOOKING, etc.)
- activo: Boolean              // Si está conectado
- apiKey: String               // Credenciales
- externalId: String          // ID en la plataforma externa
- sincronizarPrecio: Boolean
- sincronizarCalendario: Boolean
- sincronizarReservas: Boolean
- ultimaSync: DateTime          // Última sincronización
- proximaSync: DateTime         // Próxima programada
- estadoSync: String            // conectado, error, sincronizando, etc.
- erroresSync: Int              // Contador de errores
  
```

Características de UI/UX

Diseño Visual

- Cards por canal con colores distintivos
- Iconos emoji para cada plataforma
- Badges de estado con colores semánticos
- Indicadores de carga y sincronización
- Feedback inmediato con toasts
- Grid responsive (2 columnas en desktop)

Interactividad

- Diálogos modales para configuración
- Botones de acción por canal

- Sincronización manual por tipo
- Actualización en tiempo real del estado
- Navegación fluida entre páginas

Información Mostrada

- Estado de conexión
 - Última sincronización (fecha/hora)
 - Próxima sincronización programada
 - Número de errores
 - ID externo del listing
 - Features soportados por canal
-

Seguridad Implementada

1. **Autenticación:** Todos los endpoints requieren sesión de NextAuth
 2. **Autorización:** Validación de pertenencia a companyId
 3. **Logging:** Registro de todas las operaciones con contexto
 4. **Validación:** Tipos y parámetros validados
 5. **Error Handling:** Try-catch en todos los puntos críticos
-

Flujos Principales

Flujo 1: Conectar Canal

1. Usuario va a `/str/listings/[id]/channels`
2. Click en “Conectar” en un canal
3. Ingresa credenciales en diálogo
4. Sistema valida y crea `STRChannelSync`
5. Ejecuta sincronización inicial (30 días)
6. Muestra resultado y actualiza UI

Flujo 2: Sincronización Manual

1. Usuario hace click en botón “Calendario” (o Precios/Reservas)
2. Sistema muestra spinner de carga
3. Se ejecuta sincronización específica
4. Se actualizan datos locales
5. Toast notifica resultado
6. UI se actualiza con nueva información

Flujo 3: Importar Reservas

1. Usuario sincroniza “Reservas” de un canal
2. Sistema genera 2-3 reservas demo (modo demo)
3. Crea `STRBooking` por cada reserva
4. Bloquea fechas en `STRCalendar`

5. Actualiza contador de reservas del listing
 6. Notifica éxito con número de reservas importadas
-

Mejoras y Características Destacadas

1. **Arquitectura Escalable:** Fácil agregar nuevos canales
 2. **Configuración Granular:** Qué sincronizar por canal
 3. **Modo Demo Funcional:** Permite probar sin APIs reales
 4. **UI Intuitiva:** Gestión visual clara
 5. **Documentación Completa:** Guía de 500+ líneas
 6. **Manejo de Errores Robusto:** Con reintentos y logging
 7. **Feedback Inmediato:** Toasts y estados visuales
 8. **Diseño Responsive:** Funciona en móvil y desktop
-

Testing Manual

Para probar la funcionalidad:

1. Ir a `/str/listings` (debe haber listings existentes)
 2. Click en un listing para ver detalle
 3. Click en “Gestionar Canales”
 4. Probar conectar Airbnb:
 - Ingresar cualquier API key (modo demo)
 - Ingresar cualquier Listing ID
 - Click “Conectar”
 - Verificar que se marca como conectado
 5. Probar sincronizaciones:
 - Click en botón “Calendario”
 - Verificar spinner y toast de éxito
 - Click en “Reservas”
 - Verificar importación de reservas demo
 6. Ir a `/str/settings/integrations`
 - Explorar tabs de configuración
 - Verificar estadísticas
-

Notas Técnicas

Dependencias Utilizadas

- `@prisma/client` : ORM para base de datos
- `date-fns` : Manejo de fechas
- `lucide-react` : Iconos
- `react-hot-toast` : Notificaciones
- Componentes de shadcn/ui: UI components

Patrones Implementados

- **Service Layer:** Lógica de negocio separada
 - **API Routes:** Endpoints RESTful
 - **Client Components:** UI interactiva con hooks
 - **Type Safety:** TypeScript en todo el código
 - **Error Boundaries:** Try-catch consistente
-



Próximos Pasos (Mini-Fase 2)

Según el roadmap original:

1. Sincronización Automática

- Background jobs
- Cron programado cada N horas
- Queue de sincronizaciones

2. Webhooks Reales

- Endpoint /api/webhooks/str
- Validación de firmas
- Procesamiento de eventos

3. Gestión de Precios Avanzada

- Pricing dinámico
- Reglas por canal
- Ajustes porcentuales

4. Reportes y Analytics

- Performance por canal
- Gráficos de ocupación
- ROI por plataforma



Conclusión

La **Mini-Fase 1** está **100% completada** con:

- ✓ Servicio de integración completo y documentado
- ✓ 5 API endpoints funcionales
- ✓ 3 páginas de UI completas y responsive
- ✓ Soporte para 8 canales (modo demo)
- ✓ Flujos de trabajo end-to-end
- ✓ Documentación técnica extensa
- ✓ Arquitectura preparada para producción

El sistema está listo para **uso en demo** y tiene la **infraestructura base** para activar integraciones reales cuando se obtengan las credenciales de las plataformas.

Desarrollado por: DeepAgent AI

Proyecto: Inmovea - Plataforma de Gestión Inmobiliaria

Módulo: STR (Short-Term Rental) Integrations