CRM AUTOMATIZADO PARA VIVIENDAS TURÍSTICAS VACACIONALES

Especificación Técnica para Equipo de Desarrollo

1. OBJETIVO Y DESCRIPCIÓN DE LA HERRAMIENTA

PROPÓSITO PRINCIPAL

Desarrollar un sistema CRM que automatice completamente el proceso de gestión de reservas de viviendas turísticas, desde la captación de datos hasta el cumplimiento de las obligaciones legales con el Ministerio del Interior español.

PROBLEMA QUE RESUELVE

Actualmente, los propietarios de viviendas turísticas deben:

• Gestionar manualmente las reservas de múltiples plataformas

• Crear contratos individuales para cada huésped

• Recopilar datos personales de todos los huéspedes

• Comunicar al Ministerio del Interior tanto las reservas como las entradas

• Gestionar documentación y seguimiento de cada proceso

SOLUCIÓN PROPUESTA

Un sistema integral que:

1. Extrae automáticamente datos de reservas desde Airbnb (y otras plataformas)

2. Genera contratos automáticamente personalizados para cada cliente

3. Facilita la recopilación de datos mediante formularios web inteligentes con OCR

4. Comunica automáticamente al Ministerio del Interior

5. Centraliza toda la información en un CRM unificado

6. Traduce contratos automáticamente según la nacionalidad del huésped

USUARIO OBJETIVO

• Primario: Propietarios/gestores de viviendas turísticas

• Secundario: Huéspedes que deben completar formularios

• Futuro: Empresas gestoras con múltiples propiedades

2. PRODUCTO MÍNIMO VIABLE (MVP) - PRIORIDAD MÁXIMA

El objetivo es tener operativo en 3 semanas un sistema que permita:

FUNCIONALIDADES CORE DEL MVP

2.1 Gestión Básica de Reservas

• Dashboard simple para visualizar reservas

• Creación manual de reservas (antes de integrar Airbnb)

• Estados de reserva: Pendiente, En proceso, Completada, Comunicada

2.2 Generación Automática de Contratos

• Plantilla base de contrato configurable

• Variables dinámicas: nombre, fechas, propiedad, precio

• Generación PDF automática

• Envío por email al cliente

2.3 Portal del Cliente con OCR

• Formulario web responsive para datos del huésped

• Subida de documentos: DNI, NIE, Pasaporte

• OCR automático: Extracción de datos de documentos

• Pre-rellenado: Formulario con datos extraídos

• Validación: Verificación de datos antes de envío

2.4 Comunicación con Ministerio del Interior

• Integración API con sistema HOSPEDAJES

• Envío automático de datos de reserva

• Envío automático de datos de entrada (check-in)

• Tracking de estado: Enviado, Confirmado, Error

• Reintento automático en caso de fallo

2.5 Integración Airbnb (Semana 2-3)

• Conexión OAuth con cuenta Airbnb

• Sincronización de reservas existentes

• Webhook para nuevas reservas en tiempo real

• Mapeo automático de datos Airbnb → Sistema interno

FLUJO DE TRABAJO MVP

1. Nueva reserva llega desde Airbnb

2. Sistema genera contrato automáticamente

3. Envía email al huésped con enlace al formulario

4. Huésped completa datos y sube documentos

5. OCR extrae datos automáticamente

6. Sistema valida y almacena información

7. Comunica automáticamente al Ministerio del Interior

8. Notifica confirmación al propietario

3. COMPONENTES Y ESTRUCTURA TÉCNICA

3.1 ARQUITECTURA DEL SISTEMA

BACKEND (API REST)

• /auth - Autenticación y autorización

• /reservations - Gestión de reservas

• /contracts - Generación de contratos

• /clients - CRM de clientes

• /documents - Gestión de documentos

• /ocr - Procesamiento OCR

• /ministry - Comunicación Ministerio

• /airbnb - Integración Airbnb

• /notifications - Emails y notificaciones

• /translation - Traducción de contratos

FRONTEND (DASHBOARD)

• /dashboard - Panel principal

• /reservations - Gestión reservas

• /clients - CRM clientes

• /contracts - Plantillas y generación

• /settings - Configuración

• /reports - Reportes y estados

PORTAL CLIENTE

• /form - Formulario de datos

• /upload - Subida documentos

• /confirmation - Confirmación final

• /status - Estado del proceso

3.2 BASE DE DATOS (PostgreSQL)

TABLAS PRINCIPALES

• users - Usuarios del sistema (id, email, name, role, company\_id)

• properties - Propiedades/viviendas (id, name, address, airbnb\_id, owner\_id)

• reservations - Reservas (id, property\_id, platform, platform\_reservation\_id, check\_in, check\_out, guests\_count, total\_price, status)

• clients - Clientes/huéspedes (id, first\_name, last\_name, email, phone, nationality, document\_type, document\_number, birth\_date)

• contracts - Contratos generados (id, reservation\_id, template\_id, generated\_pdf\_url, language, status, created\_at)

• ministry\_communications - Comunicaciones ministerio (id, reservation\_id, type, status, sent\_at, confirmed\_at, error\_message)

• documents - Documentos subidos (id, client\_id, type, original\_filename, stored\_url, ocr\_extracted\_data, verified)

3.3 INTEGRACIONES EXTERNAS

AIRBNB API

• Autenticación: OAuth 2.0

• Endpoints: Reservations, Calendar, Listings

• Webhooks: Real-time notifications

• Rate Limits: Gestión de límites API

SISTEMA HOSPEDAJES (MINISTERIO)

• Protocolo: SOAP/REST (según especificación oficial)

• Autenticación: Certificado digital

• Endpoints: Crear reserva, Comunicar entrada

• Validaciones: Según normativa vigente

SERVICIOS DE IA/OCR

• OCR Principal: Google Cloud Vision API

• OCR Backup: AWS Textract

• Traducción: OpenAI GPT-4 para contexto legal

• Traducción Backup: Google Translate API

COMUNICACIONES

• Email: SendGrid o AWS SES

• SMS: Twilio (opcional para recordatorios)

• PDF Generation: Puppeteer o jsPDF

3.4 STACK TECNOLÓGICO

BACKEND

• Runtime: Node.js + TypeScript

• Framework: Nest

• ORM: Prisma

• Validación: Zod

• Autenticación: JWT + bcrypt

• Queue: Redis + Bull

FRONTEND

• Framework: React + Vite

• Styling: Tailwind CSS

• Estado: Zustand o React Query

• Forms: React Hook Form + Zod

• UI Components: shadcn/ui

BASE DE DATOS

• Principal: PostgreSQL 15+

• Cache: Redis

• File Storage: AWS S3 o Cloudinary

DEVOPS

• Hosting: Vercel (frontend) + Railway/Render (backend)

• CI/CD: GitHub Actions

• Monitoring: Sentry

• Logging: Winston + CloudWatch

4. FASES DE DESARROLLO Y ESCALADO

FASE 1: MVP CORE (3 semanas)

Objetivo: Sistema funcional básico operativo

SEMANA 1: FUNDAMENTOS

• Backend básico: API REST con autenticación

• Frontend: Dashboard con login

• CRUD básico: Propiedades, reservas manuales

• Generación contratos: 1 plantilla fija

• Portal cliente: Formulario básico de datos

SEMANA 2: OCR Y AUTOMATIZACIÓN

• Integración OCR: Google Cloud Vision

• Subida documentos: Drag & drop con preview

• Extracción automática: DNI/NIE/Pasaporte → formulario

• Email automático: Envío de enlaces a clientes

• Validaciones: Datos obligatorios y formatos

SEMANA 3: INTEGRACIÓN AIRBNB

• OAuth Airbnb: Conexión con cuenta

• Sincronización: Importar reservas existentes

• Webhooks: Nuevas reservas en tiempo real

• Flujo completo: Airbnb → Contrato → Cliente → Datos

ENTREGABLE FASE 1: Sistema operativo que gestiona reservas de Airbnb y genera contratos automáticamente.

FASE 2: CUMPLIMIENTO LEGAL (2-3 semanas)

Objetivo: Automatizar comunicación con Ministerio

SEMANA 4: INVESTIGACIÓN Y PROTOTIPO

• Análisis normativo: Requisitos HOSPEDAJES

• Conexión API: Prototipo con datos de prueba

• Mapeo de datos: Sistema interno → Formato ministerio

• Certificados: Configuración autenticación

SEMANA 5: IMPLEMENTACIÓN

• API completa: Envío automático de reservas y entradas

• Error handling: Reintento automático y alertas

• Validaciones: Verificar datos antes del envío

• Estados tracking: Dashboard de comunicaciones

SEMANA 6: MONITORIZACIÓN (OPCIONAL)

• Dashboard compliance: Estado por reserva

• Reportes: Cumplimiento mensual

• Alertas: Reservas pendientes de comunicar

• Logs auditoría: Trazabilidad completa

ENTREGABLE FASE 2: Cumplimiento normativo automatizado al 100%.

FASE 3: MEJORAS Y TRADUCCIÓN (2-3 semanas)

Objetivo: Optimizar UX y ampliar mercado internacional

SEMANA 7: SISTEMA DE TRADUCCIÓN

• IA Traducción: OpenAI GPT-4 integration

• Detección automática: Nacionalidad → Idioma

• Plantillas multiidioma: 5 idiomas principales

• Personalización: Cláusulas por país

SEMANA 8: REFINAMIENTO UX

• Optimización móvil: Responsive completo

• Mejoras OCR: Mayor precisión y validación

• Bulk actions: Operaciones masivas

• Analytics básicos: Métricas de uso

SEMANA 9: TESTING Y PULIDO (OPCIONAL)

• Testing usuarios: Feedback real

• Performance: Optimización velocidad

• Bug fixes: Resolución de issues

• Documentación: Manual de usuario

ENTREGABLE FASE 3: Sistema pulido y multiidioma listo para producción.

FASE 4: ESCALABILIDAD (4-6 semanas)

Objetivo: Preparar para múltiples clientes y plataformas

MULTI-PLATAFORMA (2-3 semanas)

• Booking.com: Segunda integración

• VRBO: Tercera plataforma

• Dashboard unificado: Todas las reservas

• Configuración: Reglas por plataforma

MULTI-TENANT (2-3 semanas)

• Arquitectura: Separación por empresa

• Configuración: Settings personalizables

• Branding: White-label básico

• Facturación: Sistema de billing

ENTREGABLE FASE 4: Plataforma SaaS escalable para múltiples clientes.

CRONOGRAMA RESUMIDO

SEMANA 1-3: MVP Core - Sistema básico operativo con Airbnb

SEMANA 4-6: Cumplimiento Legal - Comunicación automática Ministerio

SEMANA 7-9: Optimización - Traducción y mejoras UX

SEMANA 10-15: Escalabilidad - Multi-plataforma y multi-tenant

ESTRATEGIA DE LANZAMIENTO

• Semana 3: Beta con cliente piloto

• Semana 6: Producción para empresa inicial

• Semana 9: Optimización basada en feedback

• Semana 15: Apertura a nuevos clientes

CONSIDERACIONES TÉCNICAS ADICIONALES

SEGURIDAD Y CUMPLIMIENTO

• GDPR: Cifrado datos personales, derecho al olvido

• Backup: Automático diario de base de datos

• SSL: HTTPS obligatorio en toda la aplicación

• Logs: Auditoría completa de acciones

PERFORMANCE

• Caching: Redis para consultas frecuentes

• CDN: Assets estáticos optimizados

• Lazy loading: Componentes y datos bajo demanda

• Optimización DB: Índices y queries optimizadas

MONITORIZACIÓN

• Uptime: Alertas caída de servicio

• Errors: Sentry para tracking de errores

• Performance: Métricas de velocidad de carga

• Usage: Analytics de funcionalidades más usadas