



Documentación Técnica - Portal

<https://www.whaly.com>

Objetivo

Desarrollar un portal web inmobiliario tipo **Idealista**, donde las propiedades se cargan automáticamente desde un **CRM externo** (como Inmovilla, Revo, o similares). El frontend debe ofrecer funcionalidades equivalentes a las de Idealista: búsqueda avanzada, filtros dinámicos, mapa, fichas detalladas, favoritos, contacto, etc.

Tecnologías a Utilizar

- **Backend:** Laravel 10+
- **Frontend:** Blade + Tailwind CSS o Vue.js (opcional)
- **Base de Datos:** MySQL / MariaDB
- **Mapas:** Leaflet.js o Google Maps API
- **CRM Externo:** Conexión vía API REST/XML o FTP (dependiendo del proveedor)

Módulos del Sistema

1. Sincronización con CRM

Objetivo:

Importar propiedades del CRM a la base de datos local.

Funcionalidad:

- Conexión diaria o en tiempo real (según API del CRM)
- Mapeo de campos (título, descripción, precio, fotos, etc.)
- Gestión de actualizaciones y bajas
- Guardar log de sincronización

Laravel Componentes:

- Artisan Command: `php artisan sync:properties`
 - Job + Queue para procesar lotes
 - Service Layer: `App\Services\CRM\PropertyImporter.php`
 - API Adapter según CRM: `App\Services\CRM\Adapters\InmovillaAdapter.php`
-

2. Buscador y Filtros Avanzados

Características:

- Búsqueda por ubicación (municipio, zona, código postal)
- Filtros por tipo de propiedad, precio, habitaciones, baños, superficie
- Ordenar por precio, fecha, etc.
- Vista de resultados en listado o en mapa

Laravel Componentes:

- Controlador: `PropertySearchController`
 - Request: `PropertySearchRequest`
 - Eloquent scopes y filtros dinámicos
 - Paginación (15-30 resultados por página)
-

3. Vista de Propiedad (Detalle)

Características:

- Galería de imágenes
- Datos del inmueble

- Mapa con ubicación
- Contacto directo
- Propiedades similares

Laravel Componentes:

- PropertyController@show
 - Componente galería (JS o Vue)
 - Partial para ficha técnica
-

4. Mapa Interactivo

Características:

- Resultados geolocalizados
- Posibilidad de mover el mapa y actualizar resultados
- Markers con miniatura y precio

Tecnologías:

- Google Maps o Leaflet.js
 - API personalizada que devuelve propiedades por lat/lng + radio
 -
 - Geolocalización por zonas
-

5. Gestión de Favoritos

Funcionalidad:

- Guardar propiedades favoritas (requiere login o cookies)

- Ver listado personal de propiedades guardadas

Laravel Componentes:

- Middleware de autenticación
 - Modelo: Favorite
 - Relación User - Properties
 - Opcional: usar LocalStorage + cookies para visitantes no logueados
-

6. Formulario de Contacto

Funcionalidad:

- Envío de mensaje por propiedad
- Validación y protección contra SPAM
- Notificación por correo al agente asignado

Laravel Componentes:

- Form Request + Controller
- Mailables
- Event Logging

7. SEO + OpenGraph

Consideraciones:

- URLs amigables (/pisos/madrid/3-habitaciones)
- Meta tags personalizados por propiedad
- Sitemap dinámico

- Schema.org para inmuebles (RealEstateListing)
- Incorporar el módulo SEO implementado