

API de Sitio Inmuebles para la web propia

Esta API es para la inmobiliaria que **ya tiene su propia página web** y quiere seguir usándola: carga todo en el CRM de Sitio Inmuebles y su web lee de acá las propiedades, los filtros del buscador y el equipo, y manda de vuelta las consultas, las citas y las charlas del chat de IA, que entran al CRM igual que desde nuestro sitio.

- **Dirección base:** `https://sitioinmuebles.net/api/v1`
- **Formato:** JSON, UTF-8.
- **Autenticación:** la clave que le entrega Sitio Inmuebles a la inmobiliaria, en el encabezado `X-API-Key`.

La clave la genera el soporte de Sitio Inmuebles (no se crea desde el CRM). Y solo funciona si el plan contratado incluye la función **API**.

1. Autenticación

Cada pedido lleva la clave en un encabezado:

```
GET /api/v1/propiedades HTTP/1.1
Host: https://sitioinmuebles.net
X-API-Key: si_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Accept: application/json
```

También se acepta `Authorization: Bearer <clave>`. Para **probar** un endpoint desde el navegador se puede pasar `?api_key=<clave>` en la URL, pero no lo usen en producción: las URLs quedan en los logs.

Dónde guardar la clave. Lo recomendado es que la web llame a la API **desde su servidor** (PHP, Node, Python, .NET...) y la clave viva en la configuración del servidor. Si necesitan llamar desde JavaScript en el navegador, pidan que la clave quede restringida a sus dominios: así, aunque alguien la vea, no sirve desde otro sitio.

Respuestas

Todas las respuestas tienen la misma forma:

```
{ "ok": true, "datos": ... }
```

y cuando algo falla:

```
{ "ok": false, "error": "qué pasó" }
```

Código	Significado
200 / 201	Bien.
401	Falta la clave, no existe o está apagada.
403	El plan no incluye la API (o esa función puntual), o el dominio no está autorizado.
404	La propiedad / entrada no existe o no está publicada.
422	Faltan datos o están mal (viene <code>error</code> con el primero y <code>errores</code> con todos).
429	Demasiados pedidos por minuto: esperar lo que dice <code>Retry-After</code> .

Los listados traen además:

```
"paginacion": { "pagina": 1, "por_pagina": 20, "total": 143, "paginas": 8 }
```

Se pide la página con `?pagina=N` y el tamaño con `?por_pagina=N` (máximo 100).

Tope de pedidos

Cada clave tiene un tope por minuto (120 por defecto). Como la web llama desde su servidor, conviene **guardar en caché** el listado y las fichas durante unos minutos: las propiedades no cambian cada segundo.

2. Endpoints de lectura

GET /ping

Para probar la clave. Devuelve el id de la inmobiliaria y la hora del servidor.

GET /inmobiliaria

La marca y el contacto: nombre, logo, email, teléfono, WhatsApp, dirección, redes sociales, textos del pie y SEO del sitio, cuántas propiedades hay publicadas por operación, y qué funciones tiene habilitadas el plan (`funciones.citas`, `funciones.chat_ia`, `funciones.blog`, `funciones.reservas`).

GET /agentes

El equipo (nombre, email, teléfono, foto, matrícula). Si la inmobiliaria tiene apagado "mostrar agentes" en su configuración, viene vacío con un `aviso`.

GET /filtros

Todo lo que necesita el buscador: tipos de propiedad (con subtipos), operaciones, ambientes, dormitorios, baños, monedas (con cotización), características agrupadas (Servicios, Amenities, Seguridad...), ciudades y barrios **que tienen propiedades publicadas**, las casillas (apto crédito, mascotas...) y los órdenes disponibles.

GET /ubicaciones

Provincias → ciudades → barrios, en árbol, solo las que tienen propiedades.

GET /propiedades

El listado, paginado. Acepta **exactamente los mismos filtros** que el buscador de nuestro sitio, por nombre:

Parámetro	Qué filtra	Ejemplo
<code>operacion</code>	id de operación (1 alquiler, 2 venta, 3 temporal, 4 permuta)	<code>operacion=2</code>
<code>tipo</code>	id de tipo de propiedad	<code>tipo=3</code>
<code>provincias[]</code> , <code>ciudades[]</code> , <code>barrios[]</code>	ids de ubicación; admiten varios	<code>ciudades[]=12&ciudades[]=15</code>
<code>ambientes[]</code>	1..5 o <code>mas5</code> ; admite varios (suman)	<code>ambientes[]=2&ambientes[]=3</code>
<code>dormitorios[]</code>	ídem	<code>dormitorios[]=mas5</code>
<code>banos</code>	cantidad exacta	<code>banos=2</code>
<code>estacionamiento</code>	Y	<code>estacionamiento=Y</code>

Parámetro	Qué filtra	Ejemplo
apto_credito, permite_mascotas, apto_profesional, barrio_cerrado, accesible	1 para exigirlo	apto_credito=1
caracteristicas[]	ids de características; se exigen todas	caracteristicas[]=4&caracteristicas[]=9
moneda, minimo, maximo	rango de precio, en la moneda elegida (id)	moneda=2&minimo=50000&maximo=120000
codigo	código de la propiedad (el nuestro o el propio de la inmobiliaria)	codigo=AB12CD
q	texto libre en título y descripción	q=pileta
sucursal	id de sucursal, para ver solo su cartera	sucursal=45
orden	creado (más nuevas primero, default), menor, mayor (precio)	orden=menor
pagina, por_pagina	paginación	pagina=2&por_pagina=24

Cada propiedad del listado trae lo que necesita una tarjeta:

```
{
  "codigo": "AB12CD",
  "codigo_propio": "V-102",
  "slug": "casa-3-ambientes-en-lomas",
  "titulo": "Casa 3 ambientes en Lomas",
  "tipo": "Casa", "tipo_id": 3,
  "operacion": "Venta", "operacion_id": 2,
  "destacada": true,
  "precio": { "operacion": "Venta", "precio": 120000, "moneda": "USD", "moneda_simbolo": "U$", "a_consultar": false,
  "expensas": null },
  "ambientes": 3, "dormitorios": 2, "banos": 1, "cocheras": 1,
  "superficie_cubierta": 85.5, "superficie_total": 300,
  "ubicacion": { "provincia": "Buenos Aires", "ciudad": "Lomas de Zamora", "barrio": "Centro", "direccion": "Av. Meeks 1200", "latitud": -34.76, "longitud": -58.40, "mostrar_direccion": true },
  "foto": { "orden": 1, "url": "https://.../uploads/...jpg", "pie": null },
  "fotos_cantidad": 14,
  "actualizada": "2026-09-13T10:22:00-03:00"
}
```

Cuando la inmobiliaria eligió **no mostrar el precio**, `precio.precio` viene `null` y `a_consultar` en `true`: mostrar "Consultar". Cuando marcó **ocultar la dirección**, `direccion`, `latitud` y `longitud` vienen `null` y hay `latitud_aproximada` / `longitud_aproximada` (a unos 100 m) para el mapa.

GET /propiedades/destacadas

Las de la portada: hasta `cantidad` (3 por defecto, 12 máximo) de alquiler, de venta y de temporal, al azar.

GET /propiedades/{codigo}

La ficha completa. `{codigo}` es el código (AB12CD) o el slug. Trae todo lo del listado más:

- `descripcion` (con el aviso legal vigente al pie, como en nuestra web) y `descripcion_sin_legal` (para la meta description);

- `operaciones[]` : cada operación publicada (una propiedad puede estar en venta y en alquiler) con precio, moneda, expensas, comisión y, en temporal, las `tarifas` (diaria / semanal / mensual);
- `superficies` (cubierta, semicubierta, descubierta, terreno...), `condiciones` (apto crédito, mascotas...), `caracteristicas[]`;
- `fotos[]`, `planos[]`, `videos[]` (URLs de YouTube/Vimeo), `tours_virtuales[]`;
- `agente` (nombre, email, teléfono, foto);
- `enlaces.ficha_publica` (una ficha sin marca para pasar por WhatsApp) y `enlaces.reserva` (la página de reserva alojada en Sitio Inmuebles, si la propiedad la tiene habilitada; ver 4).

Abrir la ficha por la API **cuenta como visita** en las estadísticas del CRM.

GET /propiedades/{codigo}/similares

Hasta `cantidad` (6 por defecto) parecidas: misma operación y tipo, primero de la misma ciudad y después de la misma provincia.

GET /blog y GET /blog/{slug}

Las entradas publicadas del blog (título, resumen, imagen, autor, fecha) y cada una con su `contenido` en HTML y el video de YouTube si lo tiene. Solo si el plan incluye el blog; si no, 403.

3. Endpoints de escritura

Todos son `POST` con el cuerpo en JSON (`Content-Type: application/json`) o como formulario. Llevan un campo honeypot opcional `sitio_web`: si viene con algo adentro se descarta el pedido en silencio (contestando ok). Déjenlo oculto en el formulario para frenar robots.

POST /consultas

Una consulta del formulario de la web. Entra a la bandeja del CRM, se asigna al agente de la propiedad, crea o actualiza el contacto, arranca el túnel de ventas y avisa por correo, igual que desde nuestro sitio.

Campo		
<code>nombre</code>	obligatorio	Puede venir "Nombre Apellido" junto.
<code>apellido</code>	opcional	
<code>email</code>	obligatorio si no hay teléfono	
<code>telefono</code>	obligatorio si no hay email	Se guarda con <code>pais</code> y <code>area</code> adelante si los mandan.
<code>pais</code> , <code>area</code>	opcional	Código de país y de área, solo números.
<code>mensaje</code>	opcional	Hasta 3000 caracteres.
<code>propiedad</code>	opcional	Código o slug. Con esto es una consulta por esa propiedad ; sin esto, general.
<code>motivo</code>	opcional	Para las generales: <code>consulta</code> (default), <code>tasacion</code> , <code>venta</code> , <code>alquiler</code> ...

```
curl -X POST https://sitioinmuebles.net/api/v1/consultas \
-H "X-API-Key: si_XXXX" -H "Content-Type: application/json" \
-d '{"nombre":"Ana Pérez","email":"ana@mail.com","telefono":"1155551234","pais":"54","area":"11","mensaje":"Quiero visitarla","propiedad":"AB12CD"}'
```

Respuesta 201: `{ "ok": true, "datos": { "recibida": true, "consulta_id": 8812, "contacto_id": 2201 } }`.

POST /citas

Pedir una cita de visita. La cita queda **pendiente** en la agenda del CRM, le avisa al agente por WhatsApp, manda la invitación de calendario al interesado y los recordatorios, exactamente como el botón "Solicitar cita" de nuestra ficha. Requiere que el plan incluya las citas desde la web.

Campo	
propiedad	obligatorio: código o slug
nombre, telefono, email	obligatorios
fecha	AAAA-MM-DD, de hoy en adelante
hora	HH:MM
mensaje	opcional

Respuesta: { "ok": true, "mensaje": "Recibimos tu solicitud de cita..." }.

POST /chat

El agente de IA de la web. Requiere que el plan lo incluya.

```
{ "mensaje": "busco un 2 ambientes en alquiler en Lanús",  
  "historial": [ { "rol": "usuario", "texto": "hola" }, { "rol": "asistente", "texto": "¡Hola! ¿Qué buscás?" } ] }
```

Devuelve { "respuesta": "...", "propiedades": [...], "cita": "..." }. Cada propiedad trae título, precio, foto y código: con el código armen el enlace a la ficha de **su** web. Guarden el `historial` (hasta 10 turnos) y mándenlo en cada mensaje para que la charla tenga memoria.

POST /eventos/whatsapp-click

Cuenta un clic en el botón de WhatsApp de la web, para las estadísticas del CRM. Campos opcionales: `propiedad` (código o slug) y `origen` (`ficha`, `header`, `footer` o `web`).

4. Reservas

La reserva de una propiedad (seña) lleva firma, verificación por códigos que van al correo y al celular, y pago por Mercado Pago o PayPal. Eso no se replica en otra web: se usa la página alojada en Sitio Inmuebles. La ficha trae `enlaces.reserva` cuando la propiedad tiene la reserva habilitada; basta con poner el botón "Reservar" apuntando ahí. El interesado vuelve a la web al terminar.

5. CORS (llamadas desde el navegador)

La API contesta el preflight `OPTIONS` y manda `Access-Control-Allow-Origin`. Sin dominios cargados en la clave, vale `*`; con dominios cargados, solo esos. Si van a llamar desde JavaScript, pidan que la clave quede restringida a sus dominios.

6. Ejemplo mínimo en PHP

```
<?php  
function si_api(string $ruta, array $params = []): array {
```

```

$url = 'https://' . 'https://sitioinmuebles.net' . '/api/v1' . $ruta . ($params ? '?' . http_build_query($params) :
'');
$ch = curl_init($url);
curl_setopt_array($ch, [
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_HTTPHEADER => ['X-API-Key: ' . getenv('SI_API_KEY'), 'Accept: application/json'],
    CURLOPT_TIMEOUT => 15,
]);
$json = json_decode((string) curl_exec($ch), true);
curl_close($ch);
return $json ?: ['ok' => false, 'error' => 'sin respuesta'];
}

$listado = si_api('/propiedades', ['operacion' => 2, 'por_pagina' => 12, 'pagina' => 1]);
foreach ($listado['datos'] as $p) {
    echo '<a href="/propiedad/' . $p['slug'] . '">' . htmlspecialchars($p['titulo']) . '</a>';
}

```

(Si `https://sitioinmuebles.net` ya incluye `https://`, quiten el prefijo del ejemplo.)

7. Preguntas frecuentes

¿Las fotos las tengo que copiar? No: las URLs de `fotos[].url` se pueden usar directo en `<img>`. Si quieren, cópienlas a su CDN.

¿Cada cuánto cambia el catálogo? Cuando la inmobiliaria carga o edita en el CRM. Una caché de 5 a 15 minutos en su servidor es razonable.

¿Qué pasa si la inmobiliaria cambia de plan? Si el plan nuevo no incluye la API, todos los endpoints contestan 403 hasta que vuelva a tenerla. Los datos no se pierden.

¿Puedo cargar propiedades por la API? Por ahora no: la carga se hace en el CRM, que tiene el editor con IA, la publicación en portales y la validación de cada portal. Si hace falta, pídanlo a soporte.