GUÍA DE USO

API Compra Ágil V2

PARA DESARROLLADORES Y CIUDADANÍA

VERSIÓN 2.0 – MAYO 2026

ChileCompra

MÁS EFICIENCIA Y PROBIDAD
EN COMPRAS PÚBLICAS

Historial de versiones

Versión	Fecha	Descripción de cambios
2.0	Abril 2026	Versión inicial de la guía de uso API Compra Ágil v2.
3.0	Mayo 2026	Correcciones basadas en observaciones de usuarios: documentación del comportamiento real del estado oc_emitida; corrección de campos orden_compra; incorporación de id_oc; corrección de estructura proveedores_cotizando[]; nuevos campos no documentados (id_cotizacion, codigo_empresa, etc.); documentación de limitación de filtro por codigo_organismo; completar proceso de obtención de ticket; nuevo ejemplo 8.6.

1. Introducción

La API de Compra Ágil permite obtener, de forma estructurada y paginada, la información publicada en el mecanismo Compra Ágil de Mercado Público. Con ella es posible consultar, filtrar y descargar datos de procesos de compra realizados por organismos públicos del Estado de Chile.

¿Qué es Compra Ágil?

Compra Ágil es un mecanismo de contratación simplificada de Mercado Público que permite a los organismos del Estado adquirir bienes y servicios de forma rápida, convocando directamente a proveedores registrados. Cada proceso tiene un código único con formato como 1057539-228-COT26.

Casos de uso de la API

- Monitoreo de oportunidades: detectar nuevas Compras Ágiles publicadas en tiempo real.
- Sincronización incremental: mantener una base de datos local actualizada con los últimos cambios.
- Búsqueda y análisis: filtrar por región, estado, palabras clave o rango de fechas.
- Transparencia pública: acceder a proveedores, montos y resultados de cada proceso.

URL base de la API

```
https://api2.mercadopublico.cl
```

2. Primeros pasos

2.1 Requisitos previos

- Contar con un ticket de acceso válido (ver sección 2.2).
- Tener acceso a internet desde el entorno donde se ejecutará el cliente.
- Para los ejemplos de código: Python 3.8+ con la librería requests instalada, o acceso a curl.

2.2 Cómo obtener el ticket de acceso

El acceso a la API requiere un ticket que identifica al cliente y controla el consumo. Para obtenerlo, siga estos pasos:

- Ingrese a https://www.chilecompra.cl/api/ en su navegador.
- Haga clic en el botón «Pide tu ticket».
- Acepte los términos y condiciones de uso e inicie sesión con su Clave Única.
- Complete el formulario de solicitud y seleccione la opción «Solicitar ticket».
- Recibirá el ticket automáticamente por correo electrónico. Si no lo encuentra, revise la carpeta de correo no deseado (spam).

Una vez obtenido el ticket, guárdelo de forma segura. No lo comparta públicamente ni lo incluya en repositorios de código.

3. Autenticación

Toda solicitud a la API debe incluir el ticket de acceso en el header HTTP. Sin él, la API responde con error 401 (No autorizado).

3.1 Header requerido

Header	Tipo	Requerido	Descripción
ticket	string	Sí	Ticket de acceso que identifica al cliente.

3.2 Ejemplo de autenticación

curl:

```
curl -H "ticket: TU_TICKET_AQUI" \
     "https://api2.mercadopublico.cl/v2/compra-agil?ttl_cambio_ms=300000"
```

Python:

```
import requests

TICKET = 'TU_TICKET_AQUI'
BASE_URL = 'https://api2.mercadopublico.cl'

headers = {'ticket': TICKET}
response = requests.get(f'{BASE_URL}/v2/compra-agil', headers=headers,
                        params={'ttl_cambio_ms': 300000})
print(response.json())
```

4. Control de cuota (límite de uso)

La API implementa un control de cuota diaria por ticket para proteger el servicio de usos excesivos y garantizar disponibilidad para todos los clientes.

4.1 Cómo funciona

Cada ticket tiene asociado un límite de solicitudes diarias determinado por su tipo de ticket. El sistema registra el uso diario de cada ticket y verifica si se ha alcanzado el límite antes de procesar cada solicitud.

Concepto	Descripción
Cuota diaria	Número máximo de solicitudes permitidas por día calendario.
Contador de uso	Registra cuántas solicitudes se han realizado en el día actual.
Reset automático	El contador vuelve a cero al cambiar la fecha (nuevo día calendario).
Error 429	Se devuelve cuando se alcanza el límite diario.
Cuota ilimitada	Un valor de -1 en la cuota indica uso ilimitado.

Importante: el límite es por día calendario, no por período de 24 horas. La cuota se restablece cuando cambia la fecha, independientemente de cuándo se hizo la primera solicitud del día.

4.2 Comportamiento cuando se alcanza el límite

Cuando se agota la cuota diaria:

- La API responde con código HTTP 429 (Too Many Requests).
- El mensaje de error indica: "Se ha alcanzado el límite de solicitudes permitido. Intente nuevamente más tarde."


```
{
  "success": "NOK",
  "trace": null,
  "payload": null,
  "errors": [
    {
      "codigo": "429",
      "mensaje": "Se ha alcanzado el límite de solicitudes permitido. Intente
nuevamente más tarde.",
      "detalle": null
    }
  ]
}
```


4.3 Recomendaciones

- Implemente manejo del error 429 en su código antes de pasar a producción.
- Si recibe un 429, espere hasta el inicio del siguiente día calendario para reintentar.
- Use el parámetro ttl_cambio_ms o rangos de fechas para sincronización incremental, evitando consultas innecesarias.
- Si requiere una cuota mayor, contacte a ChileCompra para solicitar un ticket con mayor límite.

Ejemplo de manejo del error 429 en Python:

```
import requests
from datetime import datetime, timezone, timedelta

def get_compra_agil(url, ticket, params):
    resp = requests.get(url, headers={'ticket': ticket}, params=params)
    if resp.status_code == 429:
        # La cuota se restablece al inicio del próximo día calendario
        ahora = datetime.now(timezone.utc)
        manana = (ahora + timedelta(days=1)).replace(hour=0, minute=1, second=0)
        espera_seg = int((manana - ahora).total_seconds())
        print(f'Cuota diaria agotada. Se restablece en {espera_seg // 3600}h ')
        raise Exception('Cuota diaria agotada. Intente mañana.')
    resp.raise_for_status()
    return resp.json()
```

5. Endpoints disponibles

La API expone dos endpoints principales:

Endpoint	Método	Descripción
/v2/compra-agil	GET	Listado y búsqueda de Compras Ágiles con filtros y paginación.
/v2/compra-agil/{codigo}	GET	Detalle completo de una Compra Ágil específica.

5.1 Listado y búsqueda

GET https://api2.mercadopublico.cl/v2/compra-agil

Permite consultar Compras Ágiles aplicando distintos filtros. Retorna resultados paginados. Útil para monitorear nuevas publicaciones, sincronizar bases de datos y realizar búsquedas operativas.

Parámetros de consulta (query parameters)

Grupo 1: Ventana de cambios

Define qué registros retornar según cambios recientes. Use opción A o B (no ambas a la vez).

Parámetro	Tipo	Descripción	Ejemplo
ttl_cambio_ms	int (ms)	Opción A: cambios en los últimos X milisegundos.	300000 (5 min)
cambio_desde	datetime ISO-8601	Opción B: fecha/hora de inicio del rango de cambios.	2026-01-19T00:00:00Z
cambio_hasta	datetime ISO-8601	Opción B: fecha/hora de fin del rango de cambios.	2026-01-20T00:00:00Z

Grupo 2: Fecha de publicación

Parámetro	Tipo	Descripción	Ejemplo
publicado_desde	datetime ISO-8601	Fecha/hora mínima de publicación.	2026-01-01T00:00:00Z
publicado_hasta	datetime ISO-8601	Fecha/hora máxima de publicación.	2026-01-31T23:59:59Z

Grupo 3: Estado

Filtra por estado del proceso. Admite múltiples valores separados por coma.

Valor	Descripción
publicada	La Compra Ágil está abierta y recibiendo cotizaciones.
cerrada	El plazo de cotización finalizó.
desierta	No se recibieron ofertas válidas.
cancelada	El proceso fue cancelado por el organismo comprador.
proveedor_seleccionado	Se seleccionó al proveedor ganador. Incluye también las Compras Ágiles con OC ya emitida (ver nota más abajo).
oc_emitida	Se emitió una Orden de Compra. ⚠ Ver nota de comportamiento real más abajo.

Ejemplo: estado=publicada,proveedor_seleccionado

Limitación conocida — Filtro por organismo: A diferencia de las APIs de Licitaciones y Órdenes de Compra, la API de Compra Ágil no dispone del parámetro codigo_organismo. Para obtener datos de un organismo específico, utilice el filtro region= y luego filtre por institucion.rut o institucion.organismo_comprador en su propio sistema.

Grupo 4: Región

Filtra por región del organismo comprador (entero entre 1 y 16, repetible).

Código	Región	Código	Región
1	Tarapacá	9	Araucanía
2	Antofagasta	10	Los Lagos
3	Atacama	11	Aysén
4	Coquimbo	12	Magallanes y Antártica
5	Valparaíso	13	Metropolitana
6	O'Higgins	14	Los Ríos
7	Maule	15	Arica y Parinacota
8	Biobío	16	Ñuble

Ejemplo: region=13,5 (Metropolitana y Valparaíso)

Grupo 5: Búsqueda por código o palabras clave

id y q son mutuamente excluyentes (solo uno de los dos puede enviarse).

Parámetro	Tipo	Descripción	Ejemplo
id	string	Código exacto de la Compra Ágil.	507428-142-COT25
q	string (URL-encoded)	Palabras clave para búsqueda en nombre/descripción.	materiales%20electricos

Grupo 6: Paginación

Parámetro	Tipo	Default	Máximo	Descripción
tamano_pagina	int	15	50	Número de resultados por página.
numero_pagina	int	1	—	Número de la página a consultar (comienza en 1).

Grupo 7: Orden

Valor	Descripción
FechaUltimaModificacion (default)	Ordena por fecha del último cambio, descendente.
FechaPublicacion	Ordena por fecha de publicación, descendente.

Uso: &ordenar_por=FechaPublicacion

5.2 Detalle de una Compra Ágil

GET https://api2.mercadopublico.cl/v2/compra-agil/{codigo}

Retorna el detalle completo de una Compra Ágil específica identificada por su código externo. Incluye productos solicitados, proveedores que cotizaron, montos, estado y documentos adjuntos.

Parámetro de ruta (path parameter)

Parámetro	Ubicación	Tipo	Requerido	Descripción
codigo	path (URL)	string	Sí	Código externo de la Compra Ágil. Ej: 1057539-228-COT26

Reglas del modelo

Estados y convocatoria:

- Estado publicada + estado_convocatoria = 1 → primer llamado abierto.
- Estado publicada + estado_convocatoria = 2 → segundo llamado abierto.
- Estado cerrada + estado_convocatoria = 1 → primer llamado cerrado.
- Estado cerrada + estado_convocatoria = 2 → segundo llamado cerrado.

Proveedores cotizando:

- Siempre retorna los proveedores del llamado en que está actualmente la Compra Ágil.
- Si está en Primer Llamado → cotizaciones del Primer Llamado.
- Si está en Segundo Llamado → solo cotizaciones del Segundo Llamado.
- Si no hay ofertas, el arreglo es vacío [] (no se omite el campo).
- El detalle completo de cotizaciones se muestra desde estado Cerrada en segundo llamado en adelante.

6. Referencia de campos

Todas las respuestas exitosas tienen la estructura: {"success": "OK", "trace": null, "payload": {...}, "errors": null}

6.1 Campos — Listado (payload.items[])

Campo	Tipo	Descripción
codigo	string	Código único de la Compra Ágil. Ej: 1057539-228-COT26
nombre	string	Título o nombre del proceso de compra.
estado.id_estado	int	Identificador numérico del estado.
estado.codigo	string (enum)	Código del estado: publicada, cerrada, desierta, cancelada, proveedor_seleccionado. (oc_emitida está definido en el modelo pero no se usa en la práctica; ver nota en §5.1 Grupo 3.)
estado.glosa	string	Descripción legible del estado. Ej: 'OC Emitida'.
convocatoria.estado_convocatoria	int	Etapa del llamado: 1 = primer llamado, 2 = segundo llamado.
convocatoria.descripcion	string	Glosa del llamado. Ej: 'Primer llamado'.
documentos[].id	string (UUID)	Identificador del documento adjunto.
documentos[].nombre	string	Nombre del archivo adjunto.
fechas.fecha_publicacion	datetime ISO-8601	Fecha y hora de publicación.
fechas.fecha_cierre	datetime ISO-8601	Fecha y hora de cierre del llamado vigente.
fechas.fecha_ultimo_cambio	datetime ISO-8601	Timestamp del último cambio (usar para sincronización).
fechas.fecha_cancelacion	datetime|null	Fecha de cancelación, si aplica.
montos.moneda	string	Moneda del presupuesto. Ej: CLP.
montos.monto_disponible	number	Monto disponible informado por el comprador.
montos.monto_disponible_clp	number	Monto disponible normalizado a CLP.
institucion.organismo_comprador	string	Nombre del organismo público comprador.
institucion.rut	string	RUT del organismo comprador.
institucion.unidad_compra	string	Unidad o división que realiza la compra.
institucion.region	int|null	Código de región (1 a 16).
institucion.nombre_region	string|null	Nombre de la región.
resumen.total_ofertas_recibidas	int	Total de cotizaciones recibidas en el llamado actual.
motivos.motivo_cancelacion	string|null	Motivo de cancelación (si aplica).
motivos.motivo_desierta	string|null	Motivo de declaración desierta (si aplica).
motivos.motivo_seleccion	string|null	Motivo de selección del proveedor (si aplica).
links.detalle	string	URL relativa para obtener el detalle. Ej: /v2/compra-agil/1057539-228-COT26

6.2 Campos — Paginación (payload.paginacion)

Campo	Tipo	Descripción
total_paginas	int	Total de páginas disponibles.
numero_pagina	int	Número de la página actual.
tamano_pagina	int	Cantidad de resultados por página.
total_resultados	int	Total de resultados que coinciden con los filtros.

6.3 Campos — Detalle (payload)

Campo	Tipo	Descripción
codigo	string	Código externo de la Compra Ágil.
nombre	string	Título del proceso.
descripcion	string	Descripción detallada de la necesidad publicada.
estado.id_estado	int	Identificador numérico del estado.
estado.codigo	string	Código normalizado del estado.

estado.glosa	string	Glosa legible del estado.
convocatoria.estado_convocatoria	int	1 = primer llamado, 2 = segundo llamado.
convocatoria.descripcion	string	Glosa del llamado.
convocatoria.fecha_cierre_primer_llamado	datetime|null	Fecha/hora de cierre del primer llamado.
convocatoria.fecha_cierre_segundo_llamado	datetime|null	Fecha/hora de cierre del segundo llamado (null si no aplica).
fechas.fecha_publicacion	datetime	Fecha de publicación.
fechas.fecha_cierre	datetime	Fecha de cierre del llamado vigente.
fechas.fecha_ultimo_cambio	datetime	Timestamp del último cambio (sincronización incremental).
fechas.fecha_cancelacion	datetime|null	Fecha de cancelación, si aplica.
entrega.direccion_entrega	string	Dirección de entrega de los bienes/servicios.
entrega.plazo_entrega_dias	int|null	Plazo de entrega en días.
documentos[].id	string	Identificador del documento adjunto.
documentos[].nombre	string	Nombre del archivo adjunto.
presupuesto.tipo_presupuesto	string	Tipo de presupuesto: Disponible o Estimado.
presupuesto.moneda	string	Moneda. Ej: CLP.
presupuesto.presupuesto_estimado	number|null	Presupuesto estimado.
presupuesto.monto_disponible	number|null	Monto disponible informado.
presupuesto.monto_disponible_clp	number|null	Monto normalizado a CLP.
presupuesto.valor_cambio_moneda	number|null	Tipo de cambio (si moneda ≠ CLP).
presupuesto.fecha_cambio_moneda	datetime|null	Fecha del tipo de cambio.
orden_compra.id_orden_compra	int|null	ID interno de la Orden de Compra. Es el indicador confiable de que se emitió una OC (distinto de null cuando hay OC).
orden_compra.id_oc	int|null	ID interno de la OC emitida. Campo adicional presente en la respuesta real. Úselo para relacionar con la API de Órdenes de Compra.
orden_compra.codigo_orden_compra	string|null	Código externo de la OC (ej: 1057532-156-AG26). ⚠ En la práctica retorna null incluso cuando se emitió la OC; use id_orden_compra como indicador.
orden_compra.estado_orden_compra	string|null	Estado textual de la OC. ⚠ En la práctica retorna null en estado proveedor_seleccionado aunque la OC exista.
institucion.organismo_comprador	string	Nombre del organismo comprador.
institucion.rut	string	RUT del organismo comprador.
institucion.unidad_compra	string	Unidad/división de compra.
institucion.region	int|null	Código de región.
institucion.nombre_region	string|null	Nombre de la región.

Cómo relacionar una Compra Ágil con su Orden de Compra: Cuando orden_compra.id_orden_compra es distinto de null, significa que la OC fue emitida. Utilice ese ID para consultar el detalle de la OC en la API de Órdenes de Compra de Mercado Público. El campo codigo_orden_compra (ej: "1057532-156-AG26") puede obtenerse a través de dicha API usando id_orden_compra como clave de búsqueda. Vea el Ejemplo 8.6 para una implementación paso a paso.

6.4 Campos — Productos solicitados (payload.productos_solicitados[])

Campo	Tipo	Descripción
codigo_producto	int|string	Código del producto/servicio en el catálogo.
nombre	string	Nombre del producto/servicio.
descripcion	string|null	Descripción de la línea solicitada.
cantidad	number	Cantidad solicitada.

unidad_medida	string	Unidad de medida (ej: EA = unidad, KG = kilogramo).

6.5 Campos — Proveedores cotizando (payload.proveedores_cotizando[])

Campo	Tipo	Descripción
rut_proveedor	string	RUT del proveedor.
razon_social	string	Razón social del proveedor.
es_emt	boolean	true si el proveedor es Empresa de Menor Tamaño (EMT).
id_cotizacion	int	ID interno de la cotización. (Campo existente en la API, no estaba documentado.)
codigo_empresa	string	Código de empresa del proveedor. (Campo existente en la API, no estaba documentado.)
codigo_sucursal_empresa	string	Código de sucursal de la empresa. (Campo existente en la API, no estaba documentado.)
estado_cotizacion.id	int	⚠ Documentado pero no confirmado en la respuesta real. Puede no existir con esta estructura.
estado_cotizacion.glosa	string	⚠ Documentado pero no confirmado en la respuesta real. Estado legible: ENVIADA, INADMISIBLE, etc.
estado_por_comprador	string|null	Estado de la cotización según la perspectiva del comprador. (Campo existente en la API, no estaba documentado.)
seleccion.proveedor_seleccionado	boolean	⚠ Documentado pero no confirmado en la respuesta real.
seleccion.motivo_seleccion	string|null	⚠ Documentado pero no confirmado en la respuesta real.
seleccion.criterio_seleccion	string|null	⚠ Documentado pero no confirmado en la respuesta real.
activo	boolean	Indica si la cotización está activa. (Campo existente en la API, no estaba documentado.)
fecha_creacion	datetime	Fecha/hora de registro de la cotización. (En la API real este campo está al nivel raíz, no dentro de un objeto fechas.)
fecha_vigencia	datetime|null	Fecha hasta la cual la cotización es válida. (Al nivel raíz en la respuesta real.)
valor_neto	number|null	Valor neto de la cotización (sin impuestos). (Al nivel raíz en la respuesta real, no dentro de montos.)
total_impuesto	number|null	Total de impuestos aplicados. (Al nivel raíz en la respuesta real.)
monto_despacho	number|null	Costo de despacho/envío. (Al nivel raíz en la respuesta real.)
monto_total	number|null	Total final (neto + impuestos + despacho). (Al nivel raíz en la respuesta real.)
nombre_impuesto	string|null	Nombre del impuesto. Ej: IVA. (Al nivel raíz en la respuesta real.)
porcentaje_impuesto	number|null	Porcentaje del impuesto. Ej: 19. (Al nivel raíz en la respuesta real.)
descripcion_cotizacion	string|null	Descripción libre de la oferta del proveedor.
descripcion	string|null	Descripción adicional distinta de descripcion_cotizacion. (Campo existente en la API, no estaba documentado.)
justificacion_inadmisibilidad	string|null	Justificación si la cotización fue declarada inadmisible.

productos_cotizados[].codigo_producto	int|string	Código del producto cotizado.
productos_cotizados[].nombre_producto	string	Nombre del producto cotizado.
productos_cotizados[].descripcion	string|null	Descripción de la línea cotizada.
productos_cotizados[].cantidad	number	Cantidad cotizada.
productos_cotizados[].precio_unitario	number|null	Precio unitario neto.
productos_cotizados[].monto_total_producto	number|null	Total de la línea cotizada.

6.6 Campos — Resumen, Motivos y Flags

Campo	Tipo	Descripción
resumen.multa_sancion	number|null	Monto de multa o sanción, si aplica.
resumen.total_ofertas_recibidas	int	Total de cotizaciones recibidas.
resumen.total_demandas	int	Total de demandas asociadas.
motivos.motivo_cancelacion	string|null	Motivo de cancelación, si aplica.
motivos.motivo_desierta	string|null	Motivo de declaración desierta, si aplica.
flags.considera_requisitos_medioambientales	boolean	Indica si el proceso considera requisitos medioambientales.
flags.considera_requisitos_impacto_social_economico	boolean	Indica si considera requisitos de impacto social o económico.

7. Manejo de errores

Cuando ocurre un error, la API responde con success: "NOK", payload: null y un arreglo errors[] con el código, mensaje y detalle del problema.

```
{
  "success": "NOK",
  "trace": null,
  "payload": null,
  "errors": [
    {
      "codigo": "401",
      "mensaje": "El ticket no existe, es inválido o no tiene permisos.",
      "detalle": null
    }
  ]
}
```

Código HTTP	Nombre	Causa típica	Qué hacer
400	Bad Request	Parámetros inválidos: formato incorrecto, rango fuera de límite, combinaciones no permitidas.	Revise los parámetros enviados según la referencia de este documento.
401	Unauthorized	Falta el header ticket en la solicitud.	Agregue el header ticket: TU_TICKET a su request.
403	Forbidden	El ticket no existe, está inactivo o fue bloqueado.	Verifique que su ticket sea correcto y esté activo.
404	Not Found	No existe Compra Ágil con el código indicado (endpoint detalle).	Verifique el código de la Compra Ágil. Puede que no sea público o no exista.
429	Too Many Requests	Se agotaron los tokens del bucket (cuota diaria).	Espere el tiempo indicado en el header Retry-After antes de reintentar.
500	Internal Server Error	Error inesperado en el servidor.	Intente nuevamente en unos minutos. Si persiste, contacte a soporte.
503	Service Unavailable	El servicio está temporalmente no disponible (mantenimiento).	Espere e intente más tarde.

8. Ejemplos de uso

8.1 Ver compras publicadas en la última hora

Ideal para monitorear nuevas Compras Ágiles en tiempo real. Retorna todos los registros que tuvieron algún cambio en los últimos 60 minutos (3.600.000 ms).

curl:

```
curl -H "ticket: TU_TICKET_AQUI" \
     "https://api2.mercadopublico.cl/v2/compra-agil?ttl_cambio_ms=3600000"
```

Python:

```
import requests

TICKET  = 'TU_TICKET_AQUI'
BASE    = 'https://api2.mercadopublico.cl'

resp = requests.get(
    f'{BASE}/v2/compra-agil',
    headers={'ticket': TICKET},
    params={'ttl_cambio_ms': 3_600_000}
)
data = resp.json()
for item in data['payload']['items']:
    print(item['codigo'], item['nombre'], item['estado']['glosa'])
```

8.2 Sincronización incremental entre dos fechas

Útil para mantener una base de datos local actualizada. Se consulta el rango de tiempo desde la última sincronización hasta ahora.

curl:

```
curl -H "ticket: TU_TICKET_AQUI" \
     "https://api2.mercadopublico.cl/v2/compra-agil?cambio_desde=2026-04-
01T00:00:00Z&cambio_hasta=2026-04-02T00:00:00Z&ordenar_por=FechaUltimaModificacion"
```

Python:

```
import requests
from datetime import datetime, timezone, timedelta

TICKET = 'TU_TICKET_AQUI'
BASE   = 'https://api2.mercadopublico.cl'

ahora       = datetime.now(timezone.utc)
hace_un_dia = ahora - timedelta(days=1)

resp = requests.get(
    f'{BASE}/v2/compra-agil',
```

```
    headers={'ticket': TICKET},
    params={
        'cambio_desde': hace_un_dia.strftime('%Y-%m-%dT%H:%M:%SZ'),
        'cambio_hasta': ahora.strftime('%Y-%m-%dT%H:%M:%SZ'),
        'ordenar_por':  'FechaUltimaModificacion'
    }
)
print(resp.json()['payload']['paginacion'])
```

8.3 Buscar por palabras clave y región

Busca Compras Ágiles que contengan 'materiales electricos' en la Región Metropolitana, filtrando solo procesos publicados o con proveedor seleccionado.

curl:

```
curl -H "ticket: TU_TICKET_AQUI" \
     "https://api2.mercadopublico.cl/v2/compra-
agil?q=materiales%20electricos&region=13&estado=publicada,proveedor_seleccionado"
```

Python:

```
import requests

TICKET = 'TU_TICKET_AQUI'
BASE   = 'https://api2.mercadopublico.cl'

resp = requests.get(
    f'{BASE}/v2/compra-agil',
    headers={'ticket': TICKET},
    params={
        'q':      'materiales electricos',
        'region': 13,
        'estado': 'publicada,proveedor_seleccionado'
    }
)
items = resp.json()['payload']['items']
print(f'Resultados: {len(items)}')
for item in items:
    print(f"  {item['codigo']} | {item['nombre']} |
${item['montos']['monto_disponible_clp']:,}")
```

8.4 Obtener el detalle de una Compra Ágil

Consulta el detalle completo de un proceso, incluyendo productos, proveedores y montos.

curl:

```
curl -H "ticket: TU_TICKET_AQUI" \
     "https://api2.mercadopublico.cl/v2/compra-agil/1195-39-COT26"
```

Python:

```
import requests

TICKET = 'TU_TICKET_AQUI'
BASE   = 'https://api2.mercadopublico.cl'

codigo = '1195-39-COT26'

resp = requests.get(
    f'{BASE}/v2/compra-agil/{codigo}',
    headers={'ticket': TICKET}
)
ca = resp.json()['payload']
print(f"Nombre     : {ca['nombre']}")
print(f"Estado     : {ca['estado']['glosa']}")
print(f"Institución: {ca['institucion']['organismo_comprador']}")
print(f"Presupuesto: ${ca['presupuesto']['monto_disponible_clp']:,}
{ca['presupuesto']['moneda']}")
print(f"Productos  : {len(ca['productos_solicitados'])}")
print(f"Proveedores cotizando: {len(ca['proveedores_cotizando'])}")
```

8.5 Recorrer todas las páginas de resultados

Cuando hay más resultados que los que caben en una página, se itera incrementando numero_pagina hasta alcanzar total_paginas.

Python:

```
import requests

TICKET = 'TU_TICKET_AQUI'
BASE   = 'https://api2.mercadopublico.cl'

params = {
    'publicado_desde': '2026-04-01T00:00:00Z',
    'publicado_hasta': '2026-04-02T23:59:59Z',
    'tamano_pagina':   50,
    'numero_pagina':   1
}

todos_los_items = []

while True:
    resp = requests.get(
        f'{BASE}/v2/compra-agil',
        headers={'ticket': TICKET},
        params=params
    )
    payload    = resp.json()['payload']
    paginacion = payload['paginacion']
    todos_los_items.extend(payload['items'])

    print(f"Página {paginacion['numero_pagina']} de {paginacion['total_paginas']}")

    if paginacion['numero_pagina'] >= paginacion['total_paginas']:
        break
```

```
    params['numero_pagina'] += 1

print(f'Total descargado: {len(todos_los_items)} registros')
```

8.6 Detectar emisión de OC en Compras Ágiles con proveedor seleccionado

El estado oc_emitida no aparece en la práctica. Para identificar qué Compras Ágiles ya tienen una Orden de Compra emitida, se consultan las que están en estado proveedor_seleccionado y se verifica que id_orden_compra sea distinto de null en el detalle.

Python:

```
import requests

TICKET = 'TU_TICKET_AQUI'
BASE   = 'https://api2.mercadopublico.cl'

# 1. Obtener todas las Compras Ágiles en estado proveedor_seleccionado
params = {
    'estado':        'proveedor_seleccionado',
    'tamano_pagina': 50,
    'numero_pagina': 1,
}

con_oc   = []  # tienen OC emitida
sin_oc   = []  # proveedor seleccionado, OC aún no emitida

while True:
    resp      = requests.get(f'{BASE}/v2/compra-agil', headers={'ticket': TICKET},
params=params)
    payload   = resp.json()['payload']
    paginacion = payload['paginacion']

    for item in payload['items']:
        # 2. Consultar el detalle de cada una
        det = requests.get(
            f"{BASE}/v2/compra-agil/{item['codigo']}",
            headers={'ticket': TICKET}
        ).json()['payload']

        id_oc = det.get('orden_compra', {}).get('id_orden_compra')
        if id_oc is not None:
            con_oc.append({'codigo': item['codigo'], 'id_orden_compra': id_oc})
        else:
            sin_oc.append(item['codigo'])

    if paginacion['numero_pagina'] >= paginacion['total_paginas']:
        break
    params['numero_pagina'] += 1

print(f'Con OC emitida : {len(con_oc)}')
print(f'Sin OC emitida : {len(sin_oc)}')
for ca in con_oc:
    print(f"  {ca['codigo']} → id_orden_compra={ca['id_orden_compra']}")
```

9. Glosario

Término	Definición
API	Interfaz de Programación de Aplicaciones. Permite que sistemas informáticos se comuniquen entre sí de forma estructurada.
Compra Ágil	Mecanismo de contratación simplificada de Mercado Público para adquisición rápida de bienes y servicios por organismos del Estado.
Ticket	Credencial de acceso a la API. Identifica al cliente y controla su cuota de uso.
Token Bucket	Algoritmo de control de tasa de solicitudes. Un 'balde' de tokens se consume con cada request y se recarga automáticamente.
Header HTTP	Información adicional que se envía en el encabezado de una solicitud HTTP, como credenciales o metadatos.
Endpoint	URL específica de la API que expone una funcionalidad concreta.
Payload	Contenido principal de la respuesta de la API que contiene los datos solicitados.
Paginación	Mecanismo que divide un conjunto grande de resultados en páginas de tamaño fijo para facilitar su procesamiento.
ISO-8601	Estándar internacional para representar fechas y horas. Ej: 2026-04-01T12:00:00Z
CLP	Código de moneda del Peso Chileno según el estándar ISO 4217.
EMT	Empresa de Menor Tamaño. Clasificación del Registro de Proveedores de ChileCompra.
Convocatoria	Llamado a proveedores para cotizar en una Compra Ágil. Puede haber primer y segundo llamado.
OC / Orden de Compra	Documento oficial que formaliza la compra al proveedor seleccionado en el proceso.
429 Too Many Requests	Código HTTP que indica que se superó el límite de solicitudes permitidas.
Retry-After	Header HTTP que indica cuántos segundos esperar antes de volver a intentar una solicitud rechazada.
DCCP	Dirección de Compras y Contratación Pública. Organismo público que administra Mercado Público.
Sincronización incremental	Técnica para obtener solo los datos que han cambiado desde la última consulta, evitando descargas completas.