



API Taller Mecánico 1.0 OAS 3.0

Documentación de endpoints para el Taller

Authorize



🔒 Autenticación



POST /api/auth/login Iniciar sesión y obtener token JWT



POST /api/auth/register Registrar nuevo usuario (Solo ADMIN)



Users



GET /api/users



Parameters

[Try it out](#)

No parameters

Responses

Code Description

Links

200

No links

PATCH /api/users/change-password



Parameters

[Try it out](#)

No parameters

Request body required

[application/json](#)

Example Value Schema

```
{}
```

Responses

Code	Description	Links
------	-------------	-------

200		<i>No links</i>
-----	--	-----------------

DELETE /api/users/{id} ^

Parameters

[Try it out](#)

Name	Description
------	-------------

id * required	
----------------------	--

string	
--------	--

(path)	
--------	--

Responses

Code	Description	Links
------	-------------	-------

200		<i>No links</i>
-----	--	-----------------

Products



POST /api/products Crear un nuevo producto

Parameters

[Try it out](#)

No parameters

Request body required

application/json

Example Value Schema

```
{  
    "sku": "F-001",  
    "nombre": "Pastilla de Freno Delantera",  
    "marca": "Bosch",  
    "calidad": "Cerámica",  
    "precio_venta": 28000,  
    "stock_actual": 10,  
    "stock_minimo": 5,  
    "categoriaId": "uuid-categoría",  
    "modelosCompatiblesIds": [  
        "uuid-modelo-1",  
        "uuid-modelo-2"  
    ]  
}
```

Responses

Code	Description	Links
201	Producto creado exitosamente	No links
409	El SKU ya existe	No links

GET /api/products Obtener todos los productos

Parameters

[Try it out](#)

No parameters

Responses

Code	Description	Links
200	Lista de productos	No links

GET /api/products/{id} Obtener un producto por ID ^

Parameters

Try it out

Name	Description
id * required	

string
(path)

Responses

Code	Description	Links
200	Producto encontrado	No links
404	Producto no encontrado	No links

PATCH /api/products/{id} Actualizar un producto ^

Parameters

Try it out

Name	Description
id * required	

string
(path)

Request body required

application/json

[Example Value](#) [Schema](#)

{}

Responses

Code	Description	Links
200	Producto actualizado	No links
404	Producto no encontrado	No links

[**DELETE** /api/products/{id} Eliminar un producto](#) [^](#)

Parameters

[Try it out](#)

Name	Description
id * required	string (path)

Responses

Code	Description	Links
200	Producto eliminado	No links
404	Producto no encontrado	No links

Categories

[^](#)[**POST** /api/categories Crear una nueva categoría](#) [^](#)

Parameters

No parameters

Request body required

application/json

Example Value Schema

```
{  
  "nombre": "Frenos",  
  "descripcion": "Repuestos relacionados con el sistema de frenos"  
}
```

Responses

Code	Description	Links
201	Categoría creada exitosamente	<i>No links</i>
409	La categoría ya existe	<i>No links</i>

GET /api/categories Obtener todas las categorías

Parameters

No parameters

Responses

Code	Description	Links
200	Lista de categorías	<i>No links</i>

GET /api/categories/{id} Obtener una categoría por ID

Parameters

Name	Description
id * required string (path)	<input type="text" value="id"/>

Responses		
Code	Description	Links
200	Categoría encontrada	No links
404	Categoría no encontrada	No links

PATCH	/api/categories/{id}	Actualizar una categoría	^
Parameters			Try it out
Name	Description		
id * required string (path)	<input type="text" value="id"/>		

Request body	required	application/json
Example Value		Schema
{}		

Responses	
-----------	--

Code	Description	Links
200	Categoría actualizada	No links
404	Categoría no encontrada	No links

DELETE /api/categories/{id} Eliminar una categoría ^

Parameters **Try it out**

Name	Description
id * required string (path)	<input type="text" value="id"/>

Responses

Code	Description	Links
200	Categoría eliminada	No links
404	Categoría no encontrada	No links

Clients

^

POST /api/clients Crear nuevo cliente 🔒 ^

Parameters **Try it out**

No parameters

Request body required application/json

Example Value Schema

{}

Responses

Code	Description	Links
201	Cliente creado exitosamente	No links
400	Datos inválidos	No links

GET /api/clients Listar clientes (ADMIN y WORKER ven solo datos básicos)  

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Lista de clientes	No links

Proveedores



POST /api/providers  

Parameters

Try it out

No parameters

Request body required

application/json

Example Value Schema

```
{  
  "nombre": "Repuestos Don Gato",  
  "telefono": "+56912345678",  
  "email": "contacto@repuestos.cl"  
}
```

Responses

Code	Description	Links
201		No links

GET /api/providers  

Parameters



No parameters

Responses

Code	Description	Links
200		No links

GET /api/providers/{id} Obtener un proveedor por ID  

Parameters



Name	Description
id * required string (path)	<input type="text" value="id"/>

Responses

Code	Description	Links
200	Proveedor encontrado	No links
404	Proveedor no encontrado	No links

PATCH /api/providers/{id} Actualizar un proveedor  

Parameters

[Try it out](#)

Name	Description
id * required	string (path)

Request body required

application/json

Example Value Schema

{}

Responses

Code	Description	Links
200	Proveedor actualizado	No links
404	Proveedor no encontrado	No links

DELETE /api/providers/{id} Eliminar un proveedor  

Parameters

[Try it out](#)

Name	Description
id * required string (path)	id

Responses		
Code	Description	Links
200	Proveedor eliminado	No links
404	Proveedor no encontrado	No links

Compras

^

POST	/api/purchases	Registrar compra a proveedor (solo ADMIN)	
Parameters			
No parameters			
Request body required			
Example Value Schema			
<pre>{ "proveedor_nombre": "Repuestos Chile", "numero_documento": "F-12345", "tipo_documento": "FACTURA", "items": [{ "sku": "F-001", "nombre": "Pastilla de Freno", "marca": "Bosch", "calidad": "Cerámica", "modelos_compatibles_ids": ["uuid-modelo-1", "uuid-modelo-2"], "cantidad": 10, "precio_costo": 15000, "precio_venta_sugerido": 28000 }] }</pre>			

}]
}

Responses

Code	Description	Links
201	Compra registrada, stock actualizado	No links
400	Datos inválidos	No links
401	No autenticado	No links
403	Solo ADMIN puede crear compras	No links

GET /api/purchases Listar todas las compras (solo ADMIN)

Parameters

Try it out

No parameters

Responses

Code	Description	Links
403	Solo ADMIN puede ver compras	No links

DELETE /api/purchases/{id} Eliminar compra y revertir stock (solo ADMIN)

Parameters

Try it out

Name	Description
------	-------------

id * requiredstring
(path)

id

Responses		
Code	Description	Links
200		No links

Vehicles



POST	/api/vehicles	Crear un nuevo vehículo	
Parameters			
No parameters			
Request body <small>required</small>			
Example Value <small>Schema</small>			
{ "patente": "ABCD12", "marca": "Toyota", "modelo": "Yaris", "anio": 2020, "kilometraje": 50000 }			
Responses			
Code	Description	Links	
201	Vehículo creado exitosamente	No links	
409	La patente ya existe	No links	

GET	/api/vehicles	Obtener todos los vehículos	

Parameters

[Try it out](#)

No parameters

Responses

Code	Description	Links
------	-------------	-------

200	Lista de vehículos	<i>No links</i>
-----	--------------------	-----------------

GET /api/vehicles/{id} Obtener un vehículo por ID [^](#)

Parameters

[Try it out](#)

Name	Description
------	-------------

id * required

string
(path)

Responses

Code	Description	Links
------	-------------	-------

200	Vehículo encontrado	<i>No links</i>
-----	---------------------	-----------------

404	Vehículo no encontrado	<i>No links</i>
-----	------------------------	-----------------

PATCH /api/vehicles/{id} Actualizar un vehículo [^](#)

Parameters

[Try it out](#)

Name	Description
id * required string (path)	<input type="text" value="id"/>

Request body	required
	<input type="text" value="application/json"/>

Example Value	Schema
{}	

Responses		
Code	Description	Links
200	Vehículo actualizado	No links
404	Vehículo no encontrado	No links

DELETE /api/vehicles/{id} Eliminar un vehículo	
Try it out	

Name	Description
id * required string (path)	<input type="text" value="id"/>

Responses	
-----------	--

Code	Description	Links
200	Vehículo eliminado	No links
404	Vehículo no encontrado	No links

Vehicle Models



POST /api/vehicle-models Crear un nuevo modelo de vehículo

Parameters

No parameters

Request body

Example Value Schema

```
{  
  "marca": "Toyota",  
  "modelo": "Yaris",  
  "anio": 2020,  
  "motor": "1.5L"  
}
```

Responses

Code **Description** **Links**

201 Modelo de vehículo creado exitosamente

Media type

Controls Accept header.

Example Value Schema

```
{}
```

Code	Description	Links
409	El modelo ya existe	No links

GET /api/vehicle-models Obtener todos los modelos de vehículos

Parameters
[Try it out](#)

No parameters

Responses

Code	Description	Links
200	Lista de modelos de vehículos	No links

Media type

application/json

Controls Accept header.

Example Value Schema

```
[  
  {}  
]
```

GET /api/vehicle-models/search Buscar modelos por texto (autocompleteado)

Parameters
[Try it out](#)

Name	Description
q * required string (query)	Texto de búsqueda (mínimo 2 caracteres)

Responses

Code	Description	Links
200	Modelos encontrados	No links
<p>Media type</p> <div style="border: 2px solid #00AEEF; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p>		
<p>Example Value Schema</p> <pre>[{}]</pre>		
<p>GET /api/vehicle-models/marcas Obtener lista de marcas únicas</p> <p> </p>		
<p>Parameters</p> <p>Try it out</p>		
<p>No parameters</p>		
<p>Responses</p>		
Code	Description	Links
200	Lista de marcas	No links
<p>Media type</p> <div style="border: 2px solid #00AEEF; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p>		
<p>Example Value Schema</p> <pre>["string"]</pre>		

GET	/api/vehicle-models/marcas/{marca}/modelos Obtener modelos de una marca específica	 
<p>Parameters</p> <p>Try it out</p>		

Name	Description
marca * required string (path)	Nombre de la marca <input type="text" value="marca"/>

Responses

Code	Description	Links
200	Lista de modelos de la marca Media type application/json Controls Accept header. Example Value Schema ["string"]	No links

GET /api/vehicle-models/{id} Obtener un modelo de vehículo por ID 🔒 ⌂

Parameters **Try it out**

Name	Description
id * required string (path)	UUID del modelo <input type="text" value="id"/>

Responses

Code	Description	Links						
200	Modelo encontrado	No links						
	<p>Media type</p> <p>application/json</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <p>{}</p>							
404	Modelo no encontrado	No links						
PATCH /api/vehicle-models/{id} Actualizar un modelo de vehículo		 						
Parameters		Try it out						
<table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>id * required</td><td>UUID del modelo</td></tr><tr><td>string (path)</td><td>id</td></tr></tbody></table>			Name	Description	id * required	UUID del modelo	string (path)	id
Name	Description							
id * required	UUID del modelo							
string (path)	id							
Request body required		application/json						
<p>Example Value Schema</p> <p>{}</p>								
Responses								
<table border="1"><thead><tr><th>Code</th><th>Description</th><th>Links</th></tr></thead><tbody><tr><td>200</td><td>Modelo actualizado</td><td>No links</td></tr></tbody></table>			Code	Description	Links	200	Modelo actualizado	No links
Code	Description	Links						
200	Modelo actualizado	No links						

Code	Description	Links						
	<p>Media type</p> <div style="border: 2px solid green; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{}</pre>							
404	Modelo no encontrado	No links						
DELETE /api/vehicle-models/{id} Eliminar un modelo de vehículo 								
Parameters		Try it out						
<table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>id * required</td><td>UUID del modelo</td></tr> <tr> <td>string (path)</td><td><input type="text" value="id"/></td></tr> </tbody> </table>			Name	Description	id * required	UUID del modelo	string (path)	<input type="text" value="id"/>
Name	Description							
id * required	UUID del modelo							
string (path)	<input type="text" value="id"/>							
Responses								
Code	Description	Links						
200	Modelo eliminado	No links						
404	Modelo no encontrado	No links						

Órdenes de Trabajo

GET	/api/work-orders/services-catalog Obtener catálogo de servicios disponibles	
<p>Retorna la lista completa de servicios que se pueden seleccionar en el formulario de órdenes. Útil para poblar selectores y checkboxes en el frontend.</p>		

Parameters
Try it out

No parameters

Responses

Code	Description	Links
200	<p>Lista de servicios disponibles retornada exitosamente</p> <p>Media type</p> <div style="border: 2px solid green; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre style="font-family: monospace; margin: 10px 0;">["Cambio Pastillas", "Cambio Discos", "Rectificado", "Cambio Líquido Frenos", "Revisión Sistema Completo", "Cambio Zapatas Traseras", "Purga Sistema Frenos", "Revisión ABS", "Otros"]</pre>	<i>No links</i>
401	Token JWT no proporcionado o inválido	<i>No links</i>

POST
/api/work-orders Crear una nueva orden de trabajo

Crea una nueva orden de trabajo con cliente, vehículo y servicios realizados.

Lógica automática:

- Si el cliente (RUT) ya existe, se reutiliza y actualiza sus datos
- Si el vehículo (patente) ya existe, se reutiliza y actualiza kilometraje
- Si un ítem tiene product_sku, descuenta stock automáticamente del inventario
- Calcula total_cobrado sumando todos los precios de ítems

Validaciones:

- numero_order_papel debe ser único (no repetir números de talón)
- Si usa producto, valida que exista y tenga stock suficiente
- Los precios no pueden ser negativos

Parameters
Try it out

No parameters

Request body required

application/json

Example Value Schema

```
{
  "numero_orden_papel": 1547,
  "realizado_por": "Carlos González",
  "revisado_por": "Pedro Supervisor",
  "cliente": {
    "nombre": "Juan Pérez González",
    "rut": "12.345.678-9",
    "email": "juan.perez@gmail.com",
    "telefono": "+56912345678"
  },
  "vehiculo": {
    "patente": "ABCD12",
    "marca": "Toyota",
    "modelo": "Yaris Sport",
    "kilometraje": 85000
  },
  "items": [
    {
      "servicio_nombre": "Cambio Pastillas",
      "descripcion": "Pastillas delanteras cerámicas",
      "precio": 45000,
      "product_sku": "F-001",
      "cantidad_producto": 1
    }
  ]
}
```

Responses

Code	Description	Links
201	Orden de trabajo creada exitosamente con descuento de stock aplicado	No links

Media type

application/json

Controls Accept header.

Example Value Schema

```
{
  "message": "Orden de trabajo creada exitosamente",
  "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "numero_orden_papel": 1547,
  "total_cobrado": 125000,
  "cliente": "Juan Pérez",
  "vehiculo": "ABCD12",
```

Code	Description	Links
	<pre> "items_procesados": 3 } </pre>	
400	Error de validación: stock insuficiente, producto no existe, número de orden duplicado o datos inválidos	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div>	
	<p>Example Value Schema</p> <pre> { "statusCode": 400, "message": "Stock insuficiente para Disco Ventilado. Disponible: 2, So llicitado: 5", "error": "Bad Request" } </pre>	
401	Token JWT no proporcionado o inválido	No links
403	Usuario no tiene permisos para esta operación	No links

GET	/api/work-orders Listar todas las órdenes de trabajo	
Retorna todas las órdenes de trabajo con información del cliente, vehículo y detalles de servicios. Las órdenes se ordenan por fecha de creación descendente (más recientes primero).		
Parameters		Try it out
No parameters		
Responses		

Code	Description	Links
200	Lista de órdenes de trabajo retornada exitosamente	No links
	<p>Media type</p> <div style="border: 2px solid #00AEEF; padding: 2px; display: inline-block;">application/json</div>	
	Controls Accept header.	
	<p>Example Value Schema</p>	

Code	Description	Links
	<pre>[{ "id": "uuid", "numero_orden_papel": 1547, "total_cobrado": 125000, "realizado_por": "Carlos González", "fecha_creacion": "2026-01-24T10:30:00.000Z", "cliente": { "nombre": "Juan Pérez", "rut": "12345678-9" }, "vehiculo": { "patente": "ABCD12", "marca": "Toyota" }, "items": [] }]</pre>	
401	Token JWT no proporcionado o inválido	No links

Ventas Mostrador ^

POST /api/counter-sales Registrar movimiento de inventario (venta, pérdida o uso interno)  

Registra salidas de inventario que NO son órdenes de trabajo.

Tipos de movimiento:

- **VENTA:** Cliente compra producto sin servicio de instalación. Requiere: comprador, precio_venta por item.
- **PERDIDA:** Producto dañado, vencido o robado. Solo registra el costo perdido (NO suma a caja).
- **USO_INTERNO:** Consumo del taller (ej: aceite para herramientas).

Lógica automática:

- Descuenta stock del inventario para todos los tipos
- Calcula total_venta (para VENTA) o costo_perdida (para PERDIDA)
- Solo las VENTAS suman al reporte de caja diaria

Parameters Try it out

No parameters

Request body required application/json

Example Value Schema

```
{
  "tipo_movimiento": "VENTA",
  "items": [
    {
      "id": "uuid",
      "numero_orden_papel": 1547,
      "total_cobrado": 125000,
      "realizado_por": "Carlos González",
      "fecha_creacion": "2026-01-24T10:30:00.000Z",
      "cliente": {
        "nombre": "Juan Pérez",
        "rut": "12345678-9"
      },
      "vehiculo": {
        "patente": "ABCD12",
        "marca": "Toyota"
      },
      "items": []
    }
]
```

```

        "sku": "F-001",
        "cantidad": 2,
        "precio_venta": 28000
    }
],
"comentario": "Cliente compró sin instalación",
"comprador": "Juan Pérez (walk-in)"
}

```

Responses

Code	Description	Links
201	Movimiento de inventario registrado exitosamente	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "message": "Movimiento registrado exitosamente", "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890", "tipo": "VENTA", "total_venta": 56000, "costo_perdida": 0, "items_procesados": 2 }</pre>	
400	Error de validación: stock insuficiente, producto no existe, cantidad inválida o datos faltantes	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>Example Value Schema</p> <pre>{ "statusCode": 400, "message": "Stock insuficiente para Pastilla Freno. Disponible: 1, Sol icitado: 2", "error": "Bad Request" }</pre>	
401	Token JWT no proporcionado o inválido	No links
403	Usuario no tiene permisos para esta operación	No links

GET

/api/counter-sales Listar todos los movimientos de inventario



Retorna todos los movimientos registrados (ventas, pérdidas, uso interno). Se puede filtrar por tipo usando el query param "tipo".

Parameters

[Try it out](#)

Name	Description
tipo string (query)	Filtrar por tipo de movimiento: VENTA, PERDIDA, USO_INTERNO. Si no se especifica, retorna todos. <i>Available values : VENTA, PERDIDA, USO_INTERNO</i>

Responses

Code	Description	Links
200	<p>Lista de movimientos de inventario retornada exitosamente</p> <p>Media type</p> <p>application/json</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>[{"id": "uuid", "tipo": "VENTA", "total_venta": 56000, "costo_perdida": 0, "comprador": "Juan Pérez", "comentario": "Cliente compró sin instalación", "fecha_creacion": "2026-01-24T10:30:00.000Z", "items": []}]</pre>	No links
401	Token JWT no proporcionado o inválido	No links

```
[{"id": "uuid", "tipo": "VENTA", "total_venta": 56000, "costo_perdida": 0, "comprador": "Juan Pérez", "comentario": "Cliente compró sin instalación", "fecha_creacion": "2026-01-24T10:30:00.000Z", "items": []}]
```

GET /api/reports/low-stock Obtener productos con stock bajo (alerta de recompra)

Retorna productos donde stock_actual <= stock_minimo. Útil para generar alertas de recompra y mantener inventario óptimo.

Parameters

No parameters

Responses

Code	Description	Links
200	<p>Lista de productos con stock bajo retornada exitosamente</p> <p>Media type</p> <p>application/json</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "total_alertas": 2, "fecha_consulta": "2026-01-24T10:30:00.000Z", "productos": [{ "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890", "sku": "F-002", "nombre": "Disco Ventilado", "marca": "Brembo", "stock_actual": 2, "stock_minimo": 5, "diferencia": 3, "categoria": "Frenos", "precio_venta": 45000 }] }</pre>	<i>No links</i>
401	Token JWT no proporcionado o inválido	<i>No links</i>

GET /api/reports/daily-cash Obtener resumen de caja diaria

Retorna el resumen de ingresos del día especificado o del día actual.

Cálculo de totales:

- **total_taller:** Suma de todas las WorkOrders del día
- **total_meson:** Suma de CounterSales tipo VENTA únicamente
- **total_final:** total_taller + total_meson

Importante: Las PERDIDAS y USO_INTERNO no suman a la caja, solo las VENTAS.

Parameters

[Try it out](#)

Name	Description
fecha string (query)	Fecha en formato YYYY-MM-DD. Si no se envía, usa la fecha actual del servidor. 2026-01-24

Responses

Code	Description	Links
200	Resumen de caja del día retornado exitosamente	No links

Media type

application/json

Controls Accept header.

Example Value Schema

```
{
  "fecha": "2026-01-24",
  "total_taller": 350000,
  "cantidad_ordenes": 5,
  "total_meson": 85000,
  "cantidad_ventas_meson": 3,
  "total_final": 435000
}
```

400	Formato de fecha inválido	No links
-----	---------------------------	----------

Media type

application/json

Example Value Schema

```
{
  "statusCode": 400,
  "message": "Formato de fecha inválido. Use YYYY-MM-DD",
  "error": "Bad Request"
}
```

Code	Description	Links																		
401	Token JWT no proporcionado o inválido	No links																		
GET	/api/reports/search Buscador global de clientes, vehículos y órdenes	 																		
<p>Busca en múltiples entidades simultáneamente:</p> <ul style="list-style-type: none"> • Clientes: por nombre o RUT • Vehículos: por patente • Órdenes: por patente del vehículo asociado <p>Útil para acceso rápido al historial de un cliente o vehículo.</p>																				
Parameters		Try it out																		
<table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>q * required string (query)</td><td>Texto a buscar (mínimo 2 caracteres). Busca en nombres, RUT y patentes. Juan</td><td></td></tr> </tbody> </table>			Name	Description	q * required string (query)	Texto a buscar (mínimo 2 caracteres). Busca en nombres, RUT y patentes. Juan														
Name	Description																			
q * required string (query)	Texto a buscar (mínimo 2 caracteres). Busca en nombres, RUT y patentes. Juan																			
Responses																				
<table border="1"> <thead> <tr> <th>Code</th><th>Description</th><th>Links</th></tr> </thead> <tbody> <tr> <td>200</td><td>Resultados de búsqueda agrupados por entidad</td><td>No links</td></tr> <tr> <td colspan="2"> Media type <div style="border: 2px solid green; padding: 2px; display: inline-block;">application/json</div> </td><td></td></tr> <tr> <td colspan="3"> Controls Accept header. </td></tr> <tr> <td colspan="2"> Example Value </td><td>Schema</td></tr> <tr> <td colspan="3"> <pre>{ "busqueda": "Juan", "total_resultados": 5, "clientes": [{ "id": "uuid", "nombre": "Juan Pérez", "rut": "12345678-9", "telefono": "+56912345678" }], }</pre> </td></tr> </tbody> </table>			Code	Description	Links	200	Resultados de búsqueda agrupados por entidad	No links	Media type <div style="border: 2px solid green; padding: 2px; display: inline-block;">application/json</div>			Controls Accept header.			Example Value		Schema	<pre>{ "busqueda": "Juan", "total_resultados": 5, "clientes": [{ "id": "uuid", "nombre": "Juan Pérez", "rut": "12345678-9", "telefono": "+56912345678" }], }</pre>		
Code	Description	Links																		
200	Resultados de búsqueda agrupados por entidad	No links																		
Media type <div style="border: 2px solid green; padding: 2px; display: inline-block;">application/json</div>																				
Controls Accept header.																				
Example Value		Schema																		
<pre>{ "busqueda": "Juan", "total_resultados": 5, "clientes": [{ "id": "uuid", "nombre": "Juan Pérez", "rut": "12345678-9", "telefono": "+56912345678" }], }</pre>																				

Code	Description	Links
	<pre>"vehiculos": [], "ordenes_recientes": [{ "id": "uuid", "numero_orden_papel": 1234, "total_cobrado": 85000, "fecha": "2026-01-20" }]</pre>	
400	Query muy corta o vacía	No links

Media type

application/json

Example Value Schema

```
{
  "statusCode": 400,
  "message": "La búsqueda debe tener al menos 2 caracteres",
  "error": "Bad Request"
}
```

401	Token JWT no proporcionado o inválido	No links
-----	---------------------------------------	----------

Schemas



LoginDto

CreateUserDto

ChangePasswordDto

CreateProductDto

UpdateProductDto

CreateCategoryDto

UpdateCategoryDto

CreateClientDto

CreateProviderDto

UpdateProviderDto

PurchaseItemDto

CreatePurchaseDto

CreateVehicleDto

UpdateVehicleDto

CreateVehicleModelDto

VehicleModel

UpdateVehicleModelDto

ClientDto

VehicleDto

WorkOrderItemDto

CreateWorkOrderDto

CounterSaleItemDto

CreateCounterSaleDto