

# Definiciones API - Danich Greenex

*Nombre del archivo*

000499r3-Definiciones APIREST Paletizado – Sistema Danich Greenex

*Autor*

Ángel Alcañiz

*Código de documento*

000499r3

*Proyecto*

Danich - Greenex

Fecha	Revisión	Autor	Descripción
18/08/2025	1	Ángel Alcañiz	Creación documento
01/09/2025	2	Ángel Alcañiz	Cambio estructura documento
13/09/2025	3	Ángel Alcañiz	Endpoints

# 1 Instrucciones mediante APIREST

A continuación, se muestran los *endpoints* que se han establecido:

<b>GET</b>	/Mesa/Estado	Obtener el estado de una mesa	▼
<b>POST</b>	/Mesa/NuevaCaja	Lectura de una nueva caja	▼
<b>POST</b>	/Mesa/Vaciar	Vaciar una mesa	▼
<b>POST</b>	/Mesa	Insertar una nueva orden de fabricación en una mesa	▼

## 1.1 Uso de los parámetros de consulta (query parameters)

En esta API se hará uso de **parámetros de consulta (query parameters)** en lugar de parámetros de ruta (path parameters, ej. Mesa/{id}/Vaciar). Hemos optado por este enfoque porque los parámetros de consulta aportan mayor flexibilidad, permitiendo incluir varios valores sin necesidad de modificar la estructura del endpoint. Dentro del ecosistema .NET, esta forma de trabajo resulta especialmente cómoda gracias al modelo de binding, que enlaza automáticamente los parámetros de consulta con los argumentos del controlador. Esto nos permite **mantener rutas más consistentes y limpias**, permitiendo que un mismo recurso tenga distintas combinaciones de parámetros sin multiplicar las rutas posibles ni hacerlas más complejas.

### 1.1.1 ¿Cómo utilizar los parámetros de consulta?

Los parámetros de consulta se añaden al final del endpoint tras un signo de interrogación "?", y, si hay **más de uno**, se separan con el carácter &. Cada parámetro sigue la forma **nombre=valor**.

Un ejemplo dentro de este proyecto sería vaciar una mesa, donde hay que indicar tanto el identificador de la mesa como el modo de vaciado: /Mesa/Vaciar?id=1&modo=2. De esta manera, en lugar de incrustar los valores en la ruta, se especifican explícitamente como parte de la consulta.

## 2 Consulta de estado mesa paletizado

Podrá consultarse el estado de una mesa o de todas ellas. Devolverá si la mesa o mesas están ocupadas o no, los detalles de la orden activa (si la hay), y la configuración actual de la mesa.

El endpoint para consultar el estado de las mesas es el siguiente:

**GET**

**[[baseURL]] /Mesa/Estado?id=0**

- \* Para consultar el estado de todas las mesas, se debe indicar “0” como id (como se ve en el ejemplo de arriba), o no introducir ningún parámetro.
- \* Para consultar el estado de una mesa concreta, se debe introducir en su lugar el id de la misma.

**GET /Mesa/Estado** Obtener el estado de una mesa

**Parameters**

Name	Description
<b>id</b> <b>integer</b> (query)	ID de la mesa id

**Responses**

Code	Description	Links
200	Estado de la mesa	No links
	Media type	
	<b>application/json</b>	
	Controls Accept header.	
	<b>Example Value</b>   Schema	
	<pre>{     "idMesa": 0,     "estado": 0,     "descripcionEstado": "string",     "estadoPLC": 0,     "descripcionEstadoPLC": "string",     "numeroCajasPale": 0,     "numeroPaleActual": 0,     "codigoTipoCaja": "string",     "codigoTipoPalé": "string",     "cajasPorCapa": 0,     "numeroPales": 0 }</pre>	
202	Mesa NO tiene OF activa.	No links
404	Mesa no encontrada	No links
500	Ha ocurrido un error intentando obtener el estado de la mesa.	No links

Los **datos** que se proporcionan al devolver el **estado de la mesa** son los siguientes:

```
EstadoMesa ▼ {  
    idMesa           ▼ integer  
                    ID de la mesa  
  
    estado          ▼ integer  
                    Indica si hay una orden activa en la mesa: 1 = libre 2 = bloqueado (hay orden activa)  
  
    descripcionEstado  ▼ string  
                    Texto descriptivo del estado de la mesa  
  
    estadoPLC        ▼ integer  
                    Estado del PLC: 1 = Parado 2 = Manual 3 = Automático 4 = Avisos 5 = Alarmas  
  
    descripcionEstadoPLC  ▼ string  
                    Texto descriptivo del estado del PLC  
  
    numeroCajasPale  ▼ integer  
                    Cuántas cajas irán en cada palé  
  
    numeroPaleActual  ▼ integer  
                    Número de palé actual (0 si ninguno)  
  
    codigoTipoCaja   ▼ string  
                    Código de tipo de caja  
  
    codigoTipoPalé   ▼ string  
                    Código de tipo de palé  
  
    cajasPorCapa     ▼ integer  
                    Número de cajas que habrá por cada capa  
  
    numeroPales      ▼ integer  
                    Cantidad de palés totales para hacer en esta orden  
}
```

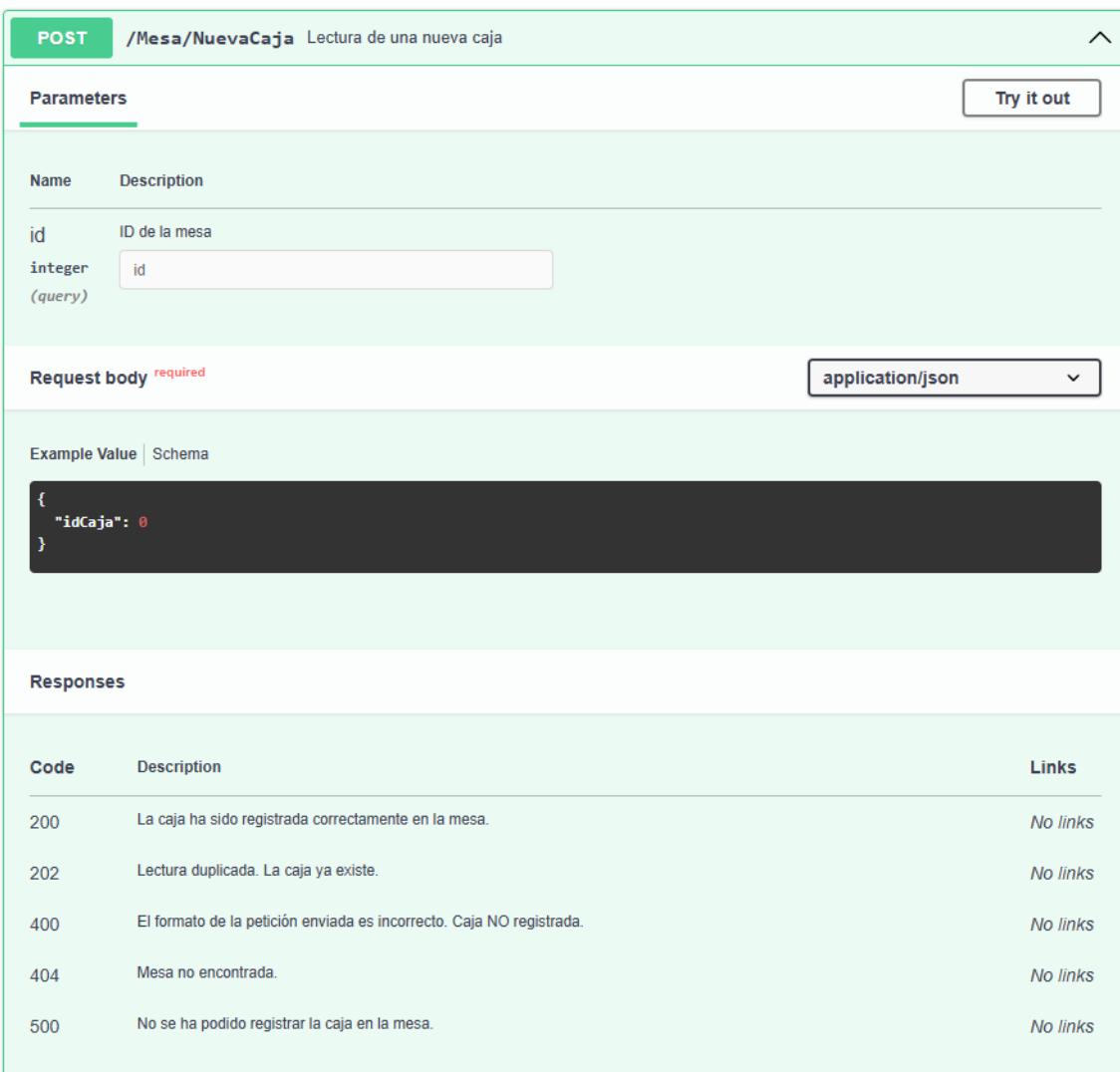
### 3 Lectura de una nueva caja

Danich enviará los datos de la lectura de cada caja en la mesa. Si la mesa no tiene orden activa, o si se realiza una lectura duplicada, no se podrá enviar y devolverá error.

El endpoint para registrar una nueva lectura es el siguiente:

**POST** | {{baseUrl}} /Mesa/NuevaCaja?idMesa=1

\* El ID (código) de la caja leída se enviará en el “body”.



The screenshot shows a detailed view of a REST API endpoint for reading a new box from a table. The endpoint is `POST /Mesa/NuevaCaja` with the description "Lectura de una nueva caja".

**Parameters:**

Name	Description
<code>id</code>	ID de la mesa
<code>integer (query)</code>	<code>id</code>

**Request body (required):** application/json

**Example Value:**

```
{ "idCaja": 0 }
```

**Responses:**

Code	Description	Links
200	La caja ha sido registrada correctamente en la mesa.	No links
202	Lectura duplicada. La caja ya existe.	No links
400	El formato de la petición enviada es incorrecto. Caja NO registrada.	No links
404	Mesa no encontrada.	No links
500	No se ha podido registrar la caja en la mesa.	No links

## 4 Envío de orden de fabricación (OF).

Danich podrá enviar una orden de fabricación a una mesa de paletizado. Si la mesa está ocupada, no se podrá enviar y devolverá error al aplicar la orden.

El endpoint para insertar una nueva OF es el siguiente:

**POST** | `{{baseUrl}} /Mesa?id=1`

**POST** `/Mesa` Insertar una nueva orden de fabricación en una mesa ^

**Parameters** **Try it out**

Name	Description
<b>id</b> * required	ID de la mesa integer (query)

**Request body** required application/json

**Example Value** | Schema

```
{
  "numeroPales": 0,
  "cajasPorPale": 0,
  "cajasPorCapa": 0,
  "codigoEnvase": "string",
  "codigoPale": "string",
  "idProgramaFlejado": 0
}
```

**Responses**

Code	Description	Links
200	Orden creada exitosamente.	No links
209	La mesa no está disponible para iniciar una orden.	No links
400	El formato de la orden enviada es incorrecto.	No links
404	Mesa, envase o palé no encontrados.	No links
500	Ha ocurrido un error en el proceso de crear la orden.	No links

Los **parámetros** que se deberán proporcionar para crear la orden son los siguientes:

```
DatosOrdenFabricacion {
    numeroPales*           ↴ integer
                            La cantidad de palés a hacer en esta orden

    cajasPorPale*          ↴ integer
                            El número de cajas que habrá en cada palé

    cajasPorCapa*          ↴ integer
                            Cuántas cajas habrá en cada capa

    codigoEnvase*          ↴ string
                            El código del envase que se utilizará en esta orden

    codigoPale*             ↴ string
                            El código del tipo de palé que se utilizará en esta orden

    idProgramaFlejado*     ↴ integer
                            El id del programa de flejado

}
```

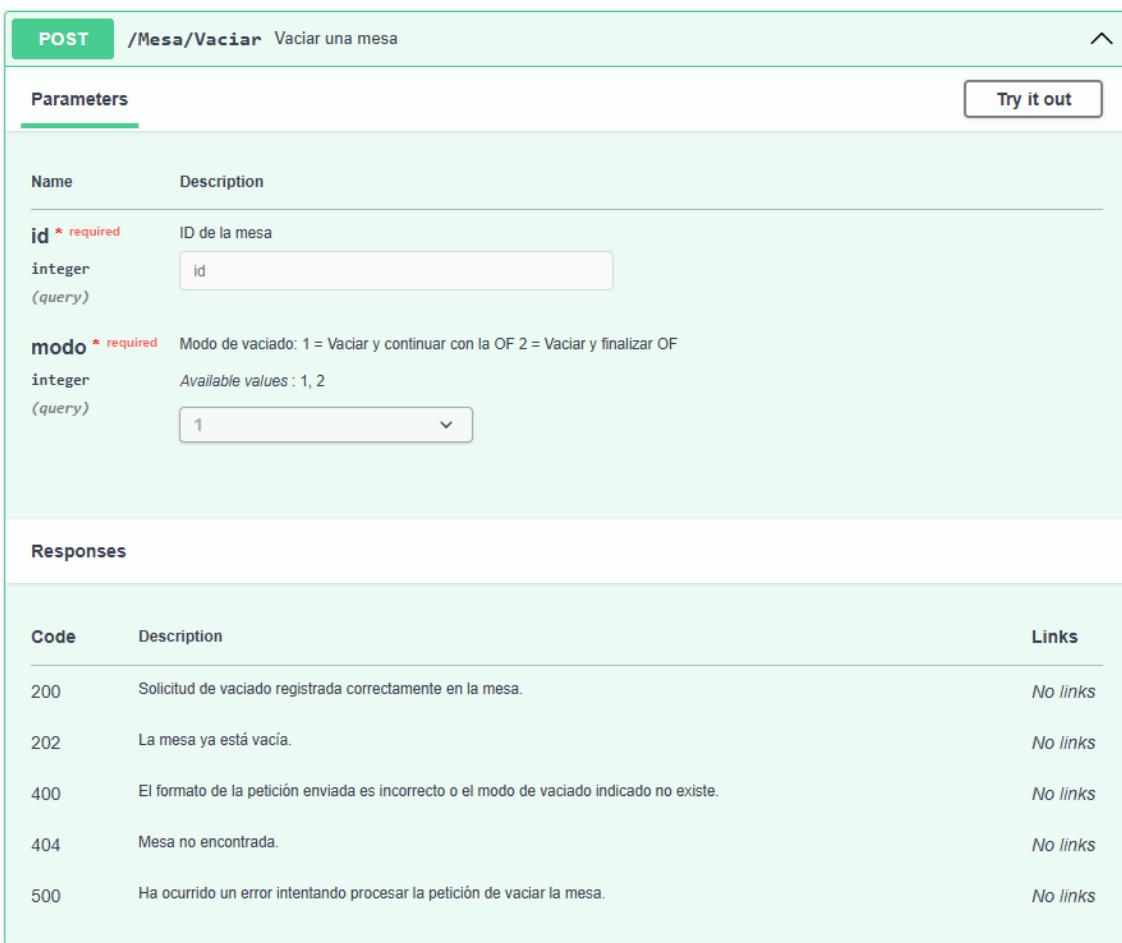
## 5 Vaciado de mesa de paletizado.

Danich podrá enviar una orden para vaciar una mesa de paletizado. Se podrá indicar que se quiere vaciar al acabar la capa o al acabar el palé.

Los **endpoints para vaciar una mesa** son los siguientes:

**POST**       `{{baseUrl}} /Mesa/Vaciar?id=1&modo=1`

**POST**       `{{baseUrl}} /Mesa/Vaciar?id=2&modo=2`



The screenshot shows a detailed view of a REST API endpoint for 'Vaciar una mesa'. At the top, it specifies a **POST** method and the URL `/Mesa/Vaciar`. The operation is described as 'Vaciar una mesa'. Below this, there's a 'Parameters' section with two fields: 'id' (integer, query) and 'modo' (integer, query). The 'modo' dropdown is set to '1'. To the right of the parameters is a 'Try it out' button. The bottom section is titled 'Responses' and lists five possible HTTP status codes with their descriptions and links:

Code	Description	Links
200	Solicitud de vaciado registrada correctamente en la mesa.	No links
202	La mesa ya está vacía.	No links
400	El formato de la petición enviada es incorrecto o el modo de vaciado indicado no existe.	No links
404	Mesa no encontrada.	No links
500	Ha ocurrido un error intentando procesar la petición de vaciar la mesa.	No links

\* **Modo 1:** Vaciar y continuar con la OF. Se vaciará la mesa con el palé actual, pero al finalizar se continuará con la misma orden.

\* **Modo 2:** Vaciar y finalizar OF. Se vaciará la mesa con el palé actual y al finalizar se borrará la orden del PLC. La mesa quedará libre para insertar una nueva orden.

---

#### **Información en protección de datos.**

Según el RGPD y LOPDGDD los datos facilitados serán tratados por SERFRUIT S.A. con la finalidad de gestionar la compra del producto adquirido, con fines de facturación, realizar gestiones administrativas y/o contables, gestión de consultas, así como, cumplir con obligaciones legales impuestas. Podrá ejercer los derechos reconocidos en la normativa citada enviando la solicitud al correo electrónico [protecciondedatos@serfruit.com](mailto:protecciondedatos@serfruit.com). Para el ejercicio de sus derechos, en caso de que sea necesario, se le solicitará documento que acredite su identidad. Puede solicitar información adicional acerca de cómo tratamos sus datos al correo electrónico indicado.