

# Control/carguero/listar - (1,2)

## [indexCooler]

### ? Descripción

Este servicio valida un **código de cooler (BINS, COOLER o PALLET)** y retorna la **lista de guías (OSE)** que tienen paquetes dentro del cooler y aún no están embarcadas.

Realiza consultas combinadas en **CentOS** y **Empresarial**, arma las guías agrupando cantidades y devuelve un resumen estructurado.

### ? Endpoint

POST /index-cooler

### ? Request Body

```
{
  "cooler": "string",
  "usuario": "string",
  "terminal": "string",
  "cap_id": "string"
}
```

### Parámetros requeridos

Campo	Tipo	Obligatorio	Descripción
cooler	string	Sí	Código escaneado del cooler (debe contener BINS, COOLER o PALLET).

Campo	Tipo	Obligatorio	Descripción
usuario	string	Sí	Usuario que ejecuta el proceso.
terminal	string	No	Terminal desde donde opera.
cap_id	string	No	ID de capacidad.

## ? Seguridad

Generalmente protegido por token o cabecera de sesión (dependiendo del middleware definido). No se valida en la función directamente.

## ? Flujo del Servicio (resumen real)

### 1. Validaciones iniciales

- Verifica que `cooler` y `usuario` existan.
- Valida que `cooler` contenga: `BINS` o `COOLER` o `PALLET`.

### 2. Verifica existencia del cooler

Busca en `emp_cooler`:

```
SELECT id, codigo_barra, cap_id
FROM emp_cooler
WHERE codigo_barra = <cooler>
AND estado = 1
ORDER BY id DESC
```

Si no existe → “Aún no hay paquetes en el cooler”

### 3. Obtiene paquetes del cooler aún no embarcados

```
SELECT ose_id, codigo_barra AS cooler, numero, usuario, fecha
FROM emp_cooler_guias
WHERE codigo_barra = <cooler>
AND estado_embarque = 0
AND estado = 1
ORDER BY fecha DESC
```

### 4. Agrupa paquetes por OSE

- Extrae los primeros dígitos del campo `numero`.

- Agrupa cantidades por OSE.

## 5. Obtiene datos adicionales de la OSE

Consulta empresarial:

```
SELECT osd.osd_osid AS ose_id,  
       os.ose_nroguiacliente AS guia,  
       SUM(osd.osd_unidad) AS cantidad  
FROM emp_os_detalle osd  
LEFT JOIN emp_ordenservicio os ON os.ose_id = osd.osd_osid  
WHERE osd.osd_eliminado = 0  
      AND osd.osd_osid IN <ose_ids>  
GROUP BY osd.osd_osid
```

## 6. Construye respuesta final

Por cada OSE:

- código del cooler
- usuario y fecha del último registro
- cantidades únicas encontradas
- cantidad total desde la OS
- número de guía del cliente

# ? Response 200 – Ejemplo

```
{  
  "success": true,  
  "message": "lista guias embarcadas.",  
  "data": [  
    {  
      "ose_id": 12345,  
      "cooler": "COOLER-001",  
      "usuario": "jlopez",  
      "fecha": "2025-01-20 10:15:00",  
      "cantidades": "10,12",  
      "cantidad": 22,  
      "guia": "GUIA-998877"  
    }  
  ]  
}
```

# ? Posibles Errores

## 1. Cooler no enviado

```
{ "success": false, "message": "No se encontró el cooler vuelva a intentarlo" }
```

## 2. Usuario no enviado

```
{ "success": false, "message": "No se encontró el usuario vuelva a intentarlo" }
```

## 3. Código de cooler inválido

```
{  
  "success": false,  
  "message": "Cooler inválido. Debe contener la palabra BINS o COOLER"  
}
```

*(Tu código acepta PALLET también, pero el mensaje no lo menciona.)*

## 4. Cooler no tiene registros activos

```
{ "success": false, "message": "Aun no hay paquetes en el cooler" }
```

## 5. No hay paquetes no embarcados

```
{  
  "success": false,  
  "message": "Aun no hay paquetes en el cooler",  
  "data": []  
}
```

---

# ? Schemas (estructuras usadas)

## emp\_cooler

```
{  
  "id": "number",  
  "codigo_barra": "string",  
  "cap_id": "number"  
}
```

## emp\_cooler\_guias

```
{  
  "ose_id": "number",  
  "cooler": "string",  
  "numero": "string",  
  "usuario": "string",  
  "fecha": "datetime"  
}
```

## emp\_os\_detalle + emp\_ordenservicio

```
{  
  "ose_id": "number",  
  "guia": "string",  
  "cantidad": "number"  
}
```

---

## ? Lógica en pseudo-código

```
if !cooler -> error
if !usuario -> error

if cooler not contains ["BINS", "COOLER", "PALLET"] -> error

coolerData = SELECT from emp_cooler where codigo=cooler and estado=1
if empty -> error

paquetes = SELECT from emp_cooler_guias where cooler=cooler and estado_embarque=0
if empty -> error

agrupar paquetes por ose_id:
  extract cantidades from numero

detalles = SELECT cantidades y guía por cada ose_id

combinar datos:
  return lista por cada ose_id con cantidades únicas + total OS
```

---

Revisión #2

Creado 2025-11-27 09:41:04 -05 por Elian

Actualizado 2025-11-27 11:13:48 -05 por Elian