

control/carguero/grupo - [listarPorGrupo]

Path: /var/www/html/qaservicesapp/app/Http/Controllers/ShalomControl/CargueroController.php

? Descripción

Este servicio recibe un **ID de grupo de cargueros** (`grp_id`) y retorna la **lista de cargueros asociados al grupo**, incluyendo información enriquecida:

- Datos del carguero (placas, choferes, ruta, origen/destino)
- Estado de QR generado
- Datos de desembarque (paquetes desembarcados y guías desembarcadas)
- Escalas generadas dinámicamente según origen y destino

El servicio consulta varias tablas del sistema Empresarial y CentOS y consolida la información en una sola respuesta.

? Endpoint

POST /listar-por-grupo

? Request Body

```
{
  "grp_id": "number"
}
```

Parámetros

Campo	Tipo	Obligatorio	Descripción
-------	------	-------------	-------------

grp_id	number	Sí	ID del grupo registrado en la tabla emp_grupo_cargueros_desembarque.
--------	--------	----	--

? Seguridad

Protegido generalmente por autenticación de la API (token/middleware definido).
El método no valida credenciales directamente.

? Flujo del Servicio (resumen real)

1. Validación de entrada

- Verifica que `grp_id` exista; si no, retorna error.
- Busca el grupo en `emp_grupo_cargueros_desembarque`.
- Verifica que el grupo tenga cargueros registrados.

2. Obtiene cargueros asociados

Consulta `emp_cargprogramado` para los `cap_id` del grupo:

```
SELECT fecha, cap_id, chofer1, chofer2, ruta,  
       origen, destino, placal, placa2, cerrado  
FROM emp_cargprogramado  
WHERE eliminado = 0 AND cap_id IN (<cap_ids>)  
ORDER BY fecha DESC
```

3. Obtiene catálogo de terminales

- Lista nombres y abreviaturas de todas las terminales.
- Se usa para convertir IDs de terminal en texto legible.

4. Procesamiento por cada carguero

Para cada `cap_id`:

- Decodifica JSON de `origen` y `destino`.
- Traduce IDs a abreviaturas de terminal:

```
ORIGEN → "LIM - CHI - ARE"  
DESTINO → "ARE - JUL"
```

- Verifica si tiene QR generado (consulta en CentOS).
- Obtiene información de desembarque por guía:
 - Paquetes desembarcados
 - Total de paquetes

- Estado de guía (DESEMBARCADA, SIN GUIA, FALTA DESEMBARCAR GUIA)
 - Calcula:
 - Total de paquetes desembarcados
 - Total de guías desembarcadas
 - Construye lista de **escalas** combinando origen + destino.
5. **Devuelve lista final de cargueros procesados**

? Response 200 – Ejemplo

```
{
  "success": true,
  "message": "Lista de cargueros",
  "data": [
    {
      "fecha": "2025-01-10 08:00:00",
      "cap_id": 98765,
      "chofer1": 1122,
      "chofer2": 3344,
      "ruta": "RUTA-01",
      "origen": "LIM - ARE",
      "destino": "ARE - JUL",
      "placa1": "ABC-123",
      "placa2": "XYZ-789",
      "cerrado": 0,
      "qr_generado": true,
      "escalas": [
        "1 Terminal Lima",
        "2 Terminal Arequipa",
        "3 Terminal Juliaca"
      ],
      "paquetes": 150,
      "guias": 12
    }
  ]
}
```

? Posibles Errores

1. No se envió `grp_id`

```
{
  "success": false,
  "message": "Envie grp_id"
}
```

2. Grupo no existe

```
{
  "success": false,
  "message": "No se encontró grupo"
}
```

3. Grupo sin cargueros

```
{
  "success": false,
  "message": "Agregue cargueros al grupo"
}
```

4. No hay cargueros coincidentes

```
{
  "success": false,
  "message": "Agregue cargueros al grupo"
}
```

? Schemas (estructuras utilizadas)

emp_grupo_cargueros_desembarque

```
{
  "id": number,
  "grupo": "string (JSON array)"
}
```

emp_cargprogramado

```
{
  "cap_id": number,
  "cap_fhsalidaprogramada": datetime,
  "cap_chofer1id": number,
  "cap_chofer2id": number|null,
  "cap_rutacarguero": string,
  "cap_origen": "JSON array",
  "cap_destino": "JSON array",
  "cap_carguero": string,
  "cap_carguero_id2": string|null,
  "cap_cerrado": number
}
```

emp_terminal

```
{
  "ter_id": number,
  "ter_nombre": string,
  "ter_abrebiatura": string
}
```

emp_desembarque_historial + emp_ordenservicio

```
{
  "des_ose_id": number,
  "desembarcados": "comma separated string",
  "total": number,
  "estadoguia": "FALTA DESEMBARCAR GUIA | SIN GUIA | GUIA DESEMBARCADA"
}
```

? Lógica en pseudo-código

```
if !grp_id -> error

grupo = SELECT grupo WHERE id=grp_id
if !grupo -> error

cap_ids = json_decode(grupo.grupo)
if empty -> error

cargueros = SELECT cargueros WHERE cap_id IN cap_ids
if empty -> error

terminales = SELECT ter_id, ter_nombre, abreviatura

for each carguero:
  origen_ids = json_decode(carguero.origen)
  destino_ids = json_decode(carguero.destino)

  traducir ids a abreviaturas
  verificar QR generado
  consultar historial de desembarque

  calcular paquetes desembarcados y guías desembarcadas
  generar escalas a partir de origen + destino

return cargueros procesados
```

Revisión #1

Creado 2025-11-27 15:40:37 -05 por Elian

Actualizado 2025-11-27 15:46:49 -05 por Elian