

control/embarque/eliminar - [destroy]

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

? Descripción

Este servicio elimina el **embarque** asociado a una orden de servicio (**OSE**) y un carguero (**cap_id**). Revierte todos los estados de embarque relacionados y registra el historial de eliminación del proceso.

Si aplica, también elimina el embarque en **Moova** (integración externa).

Es un servicio crítico dentro del flujo de **desembarque / reversión de embarque**.

? Endpoint

POST /embarque/destroy

? Request Body

```
{
  "ose_id": "number",
  "usuario": "string",
  "terminal": "string",
  "cap_id": "number"
}
```

Parámetros requeridos

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

ose_id	number	Sí	ID de la orden de servicio a revertir.
usuario	string	Sí	Usuario que ejecuta la eliminación del embarque.
terminal	string	Sí	Terminal desde donde se realiza el proceso.
cap_id	number	Sí	ID del carguero asociado al embarque.

? Seguridad

Protegido por autenticación estándar de la API (middleware Laravel).

La función asume que ya hay un usuario validado.

? Flujo del Servicio (resumen real)

1. Validaciones iniciales

Verifica que existan los campos obligatorios: `ose_id`, `usuario`, `terminal`, `cap_id`.

Si falta alguno → responde con error inmediato.

2. Obtiene información del carguero

Consulta en `emp_cargprogramado` para asegurarse de que el carguero existe.

```
SELECT cap_origen, cap_destino, cap_rutacarguero, cap_restricciones, ruta_aerea
FROM emp_cargprogramado
LEFT JOIN emp_grupos
WHERE cap_id = <cap_id>
```

Si no existe → error: "No se encontró el carguero."

3. Define estructura de reversión de OSE

Se arma un arreglo con todos los campos de embarque en estado inicial:

```
{
  "ose_usuario_embarque": null,
  "ose_estadoembarque": 0,
  "ose_ultimo_carprog_embarque": null,
  "ose_ultima_fecha_embarque": null,
  "ose_fecha_paquete_embarque": null,
  "ose_fecha_guia_embarque": null,
  "ose_estado_guia_embarque": 0,
  "ose_usuario_guia_embarque": null,
  "ose_usuario_paquete_embarque": null,
  "ose_estado_paquete_embarque": 0
}
```

4. Valida si la OSE tuvo desembarque anterior

Consulta:

```
SELECT des_id FROM emp_desembarque_historial
WHERE des_ose_id = <ose_id> AND des_carg_id = <cap_id>
```

Si existen → marca `estado_os = "Desembarque"`

5. Revierte el embarque en historial empresarial

Actualiza `emp_embarque_historial`:

```
WHERE emb_carg_id = cap_id AND emb_ose_id = ose_id
UPDATE status = 0
```

6. Resetea información de embarque en la OSE

```
UPDATE emp_ordenservicio SET <campos de $OS> WHERE ose_id = <ose_id>
```

7. Revierte procesos en historial de CentOS

Busca procesos activos en:

```
emp_procesos_historial_app
WHERE idose = ose_id
AND idcarguero = cap_id
AND proceso = 'embarque'
AND nombre_metodo IN ('embarque_paquete', 'embarque_guiia')
UPDATE status = 0
```

8. Registra un nuevo historial de eliminación

Inserta un registro indicando el proceso "eliminar_embarque".

9. Integración con Moova (si aplica)

Si el carguero contiene destino "523", verifica si el terminal está habilitado para Moova:

```
SELECT id FROM emp_ter_moova
WHERE ter_id = terminal AND status = 1
```

Si existe → Ejecuta: `MoovaController::destroy($ose_id)`

10. Respuesta final

Devuelve mensaje de éxito: {

```
"success": true,
"message": "Embarque eliminado"
}
```

? Response 200 – Ejemplo

```
{  
  "success": true,  
  "message": "Embarque eliminado"  
}
```

? Posibles Errores

1. Falta ose_id

```
{ "success": false, "message": "Falta ose_id." }
```

2. Falta usuario

```
{ "success": false, "message": "Falta usuario." }
```

3. Falta terminal

```
{ "success": false, "message": "Falta terminal." }
```

4. Falta cap_id

```
{ "success": false, "message": "Falta cap_id." }
```

5. Carguero no encontrado

```
{ "success": false, "message": "No se encontró el carguero." }
```

? Schemas (estructuras usadas)

emp_cargprogramado

```
{  
  "cap_origen": "string",  
  "cap_destino": "json/string",  
  "cap_rutacarguero": "string",  
  "cap_restricciones": "json/string"  
}
```

emp_embarque_historial

```
{  
  "emb_ose_id": "number",  
  "emb_carg_id": "number",  
  "status": "number"  
}
```

emp_ordenservicio

```
{  
  "ose_id": "number",  
  "ose_estadoembarque": "number",  
  "ose_usuario_embarque": "string|null"  
}
```

emp_procesos_historial_app

```
{
  "idose": "string",
  "idcarguero": "string",
  "proceso": "string",
  "nombre_metodo": "string",
  "status": "number"
}
```

? Lógica en pseudo-código

```
validar campos obligatorios: ose_id, usuario, terminal, cap_id

carguero = obtener carguero
if !carguero -> error

OS = campos en estado inicial

desembarque = buscar desembarque previo
if existe -> OS.estado_os = 'Desembarque'

actualizar emp_embarque_historial SET status = 0

actualizar emp_ordenservicio SET OS

actualizar procesos_historial_app SET status = 0

insertar registro eliminar_embarque

si carguero.cap_destino contiene '523':
  si terminal tiene configuración moova:
    llamar MoovaController.destroy(ose_id)

return success
```

Revisión #2

Creado 2025-11-27 15:17:56 -05 por Elian

Actualizado 2025-11-27 15:25:27 -05 por Elian