

# control/embarque/cerrar - [cerrar]

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

## ? Descripción

Este servicio realiza el **cierre de embarque** para un carguero (cap\_id).  
Procesa todas las **órdenes de servicio asociadas al carguero**, registra el embarque individual de cada una, actualiza sus estados y marca el carguero como embarcado.

Además:

- Actualiza el historial de embarque.
- Procesa posibles transbordos.
- Actualiza el detalle del carguero.
- Registra un reporte de embarque.
- Retorna confirmación del cierre.

## ? Endpoint

POST /cerrar

## ? Request Body

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

### Parámetros requeridos

Campo	Tipo	Obligatorio	Descripción
cap_id	number	Sí	ID del carguero programado a cerrar.
terminal	string	Sí	Terminal donde se realiza el cierre.
usuario	string	Sí	Usuario que ejecuta el proceso.

## ? Seguridad

Controlada por autenticación global (middleware).  
El servicio no valida tokens directamente.

## ? Flujo del Servicio (resumen real)

### 1. Obtiene órdenes asociadas al cap\_id

Consulta `emp_ordenservicio`:

- No eliminadas
- Asignadas al carguero (`ose_ultimo_carprog_embarque = cap_id`)

Si:

- No hay órdenes, o
- Todas ya están cerradas (`ose_estadoembarque = '1'`)  
→ Retorna "Carguero cerrado".

### 2. Marca historial de embarque como cerrado

Actualiza `emp_embarque_historial`:

- Donde status = 1
  - Y emb\_estado\_cierre = 0  
→ Lo cambia a 1.
- 

### 3. Iteración por cada orden

Para cada orden:

#### a) Obtiene placa del carguero

Desde `emp_cargprogramado`.

#### b) Si la orden está en estado de reparto `T`, registra transbordo

Actualiza `emp_transbordo`:

- tra\_estado = "SI"
- registra usuario, fecha y placa del carguero

#### c) Registra el embarque en `emp_embarque`

Guarda:

- ose\_id
- cap\_id
- terminal
- fecha de embarque
- usuario creador
- observaciones temporales almacenadas en la orden

#### d) Actualiza estado de la orden en `emp_ordenservicio`:

```
ose_estadoembarque = 1
ose_terminalactual = terminal
ose_estadodesembarque = 0
ose_reparto_estado = 0
ose_movilidad_reparto = null
ose_estado_paquete_embarque = 0
ose_estado_guia_embarque = 0
estado_os = "Embarque"
```

---

### 4. Actualiza detalle del carguero (`emp_cargprogramado_detalle`)

Si existe registro:

- Agrega el terminal a la lista `id_terminal`
- Cambia `cap_estado_embarque` a 1

Si no existe:

- Inserta nuevo registro con ese terminal y estado de embarque = 1.
- 

## 5. Calcula monto total embarcado

Suma `ose_montofinal` de todas las OS:

- No eliminadas
  - De ese terminal
  - Con estado embarque E o 1
  - Y asignadas al carguero
- 

## 6. Registra un reporte de embarque

Inserta en `emp_cargprogramado_reporte`:

- `cap_id`
  - `terminal`
  - `usuario`
  - `fecha`
  - `tipo_estado = "Embarque"`
  - `flete = monto total`
- 

## 7. Retorna respuesta

Devuelve:

```
{ "success": true, "message": "Se grabó el Embarque correctamente." }
```

---

# ? Response 200 – Ejemplo

```
{
  "success": true,
  "message": "Se grabó el Embarque correctamente."
}
```

## ? Posibles Errores

1. No hay órdenes para embarcar o ya están cerradas

```
{
  "success": false,
  "message": "Carguero cerrado."
}
```

*Es el único error explícito que retorna el servicio.*

## ? Schemas (estructuras usadas)

*(Basado exclusivamente en los campos usados en el código)*

### emp\_ordenservicio

```
{
  "ose_id": "number",
  "ose_ultima_fecha_embarque": "datetime",
  "ose_reparto_estado": "string",
  "ose_emb_obs_tipo_temp": "string",
  "ose_emb_obs_mensaje_temp": "string",
  "ose_emb_obs_usuario_temp": "string",
  "ose_emb_obs_fecha_temp": "datetime",
  "ose_emb_obs_urling_temp": "string",
  "ose_estadoembarque": "string"
}
```

### emp\_embarque\_historial

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

## emp\_transbordo

```
{
  "tra_osid": "number",
  "tra_estado": "string",
  "tra_salidaid2": "number",
  "tra_user2": "string",
  "tra_fecha2": "datetime",
  "tra_carguero2": "string"
}
```

## emp\_embarque

Campos insertados:

```
{
  "emb_ordenservicioid": "number",
  "emb_cargprograid": "number",
  "emb_horaembarque": "datetime",
  "emb_terminalembarque": "string",
  "eliminado": "number",
  "fhcrea": "datetime",
  "usercreaid": "string",
  "emb_obs_tipo": "string",
  "emb_obs_mensaje": "string",
  "emb_obs_usuario": "string",
  "emb_obs_fecha": "datetime",
  "emb_obs_urlimg": "string"
}
```

## emp\_cargprogramado\_detalle

```
{
  "cap_id": "number",
  "id_terminal": "string",
  "cap_estado_embarque": "number"
}
```

## emp\_cargprogramado\_reporte

```
{
  "cap_id": "number",
  "id_terminal": "string",
  "usuario_reg": "string",
  "fechareg": "datetime",
  "tipo_estado": "string",
  "flete": "number"
}
```

# ? Lógica en pseudo-código

```
ordenes = SELECT OS where cap_id

si ordenes vacío o todas cerradas:
  return "Carguero cerrado"

actualizar emp_embarque_historial: emb_estado_cierre = 1

para cada orden:
  cargar placa del carguero

  si reparto_estado == T:
    actualizar transbordo

  insertar en emp_embarque (datos de la OS)

  actualizar emp_ordenservicio:
    estadoembarque = 1
    terminalactual = terminal
    otros estados...

actualizar o insertar emp_cargprogramado_detalle

monto = suma montofinal de ordenes embarcadas del terminal y cap_id

insertar reporte en emp_cargprogramado_reporte

return success
```

Revisión #1

Creado 2025-11-27 11:26:50 -05 por Elian

Actualizado 2025-11-27 11:32:53 -05 por Elian