

Leer QR (1) - [leerCodigoQR]

? Descripción

Este servicio valida un **código QR de acceso a buses internos** y determina si un empleado puede registrar un nuevo escaneo.

El flujo consulta los registros previos, controla tiempos de reescaneo (mínimo 4 horas) y registra un nuevo ingreso si corresponde.

El servicio interactúa con:

- **Tabla “Registro de Escaneres de Bus”** (documento principal con el QR)
- **Tabla “Tabla Lista Blanca”** (registros de escaneo por empleado)
- **ERP vía dbErp() y ServiceErp()**
- **Carbon** para manejo de fechas

? Endpoint

POST /leer-codigo-qr

? Request Body

```
{
  "qr": "string",
  "employee": "string"
}
```

Parámetros

Campo	Tipo	Obligatorio	Descripción
qr	string	✓	Código QR escaneado
employee	string	✓	Código del empleado que intenta registrar el escaneo

? Seguridad

Requiere autenticación interna del ERP (vía `dbErp()` y `ServiceErp()`).

? Flujo del Servicio (resumen real)

1?? Validaciones iniciales

- El QR no puede venir vacío
- El código de empleado tampoco

Si falla, retorna error inmediato.

2?? Buscar el documento del QR

Consulta el ERP:

```
GET Registro de Escaneres de Bus
JOIN Tabla Lista Blanca
WHERE qr = <qr>
ORDER BY fecha DESC
```

Campos devueltos:

- `name` (documento padre)
- `empleado`
- `fecha` del escaneo

Si no hay resultados: **QR inválido**.

3?? Filtrar último registro por empleado

Construye una lista de empleados únicos y toma el último escaneo para cada uno.

Si el empleado actual tiene un registro:

- Calcula diferencia de horas entre el último registro y ahora.
- Si es **menor a 4 horas**, se retorna:

```
{
  "status": true,
  "msn": "denegado"
}
```

4?? Registrar nuevo escaneo

Si cumple la validación, inserta un registro en la tabla:

```
POST Tabla Lista Blanca
{
  fecha,
  parent,
  empleado,
  parentlink: Registro de Escaneres de Bus
}
```

Si falla el registro → error

Si se registra correctamente → "permitido"

? Response 200 – Ejemplos

?? Permitido

```
{
  "status": true,
  "msn": "permitido"
}
```

? Denegado por escaneo reciente

```
{
  "status": true,
  "msn": "denegado"
}
```

? QR sin registros

```
{
  "status": false,
  "msn": "No se encontraron registros para el QR escaneado"
}
```

? Error interno

```
{
  "status": false,
  "msn": "Error interno: <mensaje>"
}
```

? Posibles Errores

1. QR vacío

```
{
  "status": false,
  "msn": "El QR no puede estar vacío"
}
```

2. Empleado vacío

```
{
  "status": false,
  "msn": "El empleado no puede estar vacío"
}
```

3. QR inexistente en ERP

```
{
  "status": false,
  "msn": "No se encontraron registros para el QR escaneado"
}
```

4. Error al insertar registro

```
{
  "status": false,
  "msn": "No se pudo registrar el escaner de bus"
}
```

5. Error interno (try-catch)

```
{
  "status": false,
  "msn": "Error interno: <mensaje>"
}
```

? Tablas / Schemas utilizados

Registro de Escaneres de Bus (tpr)

Campos usados:

```
{
  "name": "string",
  "qr": "string"
}
```

Tabla Lista Blanca (teas)

Campos usados:

```
{
  "empleado": "string",
  "fecha": "datetime",
  "parent": "string"
}
```

Registro Insertado

```
{
  "fecha": "datetime",
  "parent": "string",
  "empleado": "string",
  "parentfield": "table_1",
  "parenttype": "Registro de Escaneres de Bus"
}
```

? Pseudocódigo del flujo

```
if qr is empty → error
if employee is empty → error

// Buscar registros del QR
dataQR = SELECT * FROM RegistroEscaneresBus JOIN TablaListaBlanca WHERE qr = qr ORDER BY fecha DESC

if dataQR is empty → error

// Obtener últimos registros por empleado
empleadosUnicos = agrupar_por_empleado(dataQR)

if empleadosUnicos[employee] existe:
    diffHoras = horas_entre(último registro, ahora)
    if diffHoras < 4:
        return "denegado"

// Registrar nuevo escaneo
insert = POST TablaListaBlanca {
    fecha: now,
    empleado,
    parent: documentoPrincipal
}

if insert.error:
    return error

return "permitido"
```

Revisión #1

Creado 2025-11-26 17:18:25 -05 por Elian

Actualizado 2025-11-26 17:23:40 -05 por Elian