

Curso: Cloud Infrastructure

Catedráticos: Joseph Arreola y Alma Orrego



Documentación de Endpoints

Mayco Castellanos

Isaac Cyrman

Christian Barrios

Documentación de Endpoints

Pruebas y Validación del Sistema

1. Flujo Principal de Cobro

Endpoint: POST /webhook/toll

Objetivo: Simular el paso de un vehículo por un peaje y activar el flujo completo de cobro mediante Step Functions: cálculo de tarifa, registro de transacción, generación de factura y envío de notificación al usuario.

Prueba 1 – Usuario con Tag (Modalidad 3)

JSON

```
{  
    "placa": "P-456DEF",  
    "peaje_id": "PEAJE_01010",  
    "tag_id": "TAG-001",  
    "timestamp": "2025-11-11T13:00:00Z"  
}
```

Prueba 2 – Usuario Registrado sin Tag (Modalidad 2)

JSON

```
{  
    "placa": "P-123ABC",  
    "peaje_id": "PEAJE_01010",  
    "tag_id": null,  
    "timestamp": "2025-11-11T13:01:00Z"  
}
```

Prueba 3 – Usuario No Registrado (Modalidad 1)

JSON

```
{  
    "placa": "P-TEST-987",  
    "peaje_id": "PEAJE_01010",  
    "tag_id": null,  
    "timestamp": "2025-11-11T13:02:00Z"  
}
```

2. Consulta de Usuarios

Endpoint: **GET /users/{placa}**

Objetivo: Obtener la información completa de un usuario.

| Caso | URL | Resultado Esperado |
|-------|-------------------------------|------------------------------|
| Éxito | {{baseUrl}}/users/P-123ABC | 200 OK con datos del usuario |
| Error | {{baseUrl}}/users/P-NO-EXISTE | 404 Not Found |

Endpoint: **GET /users/{placa}/tag**

Objetivo: Obtener el Tag asociado a un usuario.

| Caso | URL | Resultado Esperado |
|---------|--------------------------------|--|
| Con Tag | {{baseUrl}}/users/P-456DEF/tag | 200 OK con "tag_id": "TAG-001" |
| Sin Tag | {{baseUrl}}/users/P-123ABC/tag | 404 Not Found o mensaje indicando que no tiene Tag |

3. Gestión de Tags (CRUD)

Endpoint : **GET /users/{placa}/tag**

Objetivo: Obtener información del Tag

| Caso | URL | Resultado Esperado |
|-------|---|--|
| Éxito | <code>{{baseUrl}}/users/P-111JKL/tag</code> | <code>200 OK</code> con datos del usuario asociado |

Endpoint: **POST /users/P-111JKL/tag**

Objetivo: Asociar Tag a usuario

```
JSON
{
    "tag_id": "TAG-01016",
    "tag_status": "active"
}
```

Validación:

- `200 OK` y actualización de **tiene_tag: true** en DynamoDB.

Endpoint: **PUT /users/P-111JKL/tag**

Objetivo: Actualizar tag

```
JSON
{
    "tag_status": "inactive"
}
```

Validación:

- **200 OK** y tag_status = "inactive".

Endpoint: **DELETE /users/P-111JKL/tag**

Objetivo: Eliminar tag

Validación:

- **200 OK.**
- En DynamoDB: tiene_tag = false y tag_deleted_at = <timestamp>.

4. Consulta de Historial

Endpoint: **GET /history/payments/{placa}**

Objetivo: Obtener información de transacciones

| Caso | URL | Resultado Esperado |
|---------|---|--|
| General | {{baseUrl}}/history/payments/P-456DEF | 200 OK con lista de transacciones |
| Filtro | {{baseUrl}}/history/payments/P-456DEF?limit=1 | 200 OK con un único elemento |

Endpoint: **GET /history/invoices/{placa}**

Objetivo: Obtener información de Facturas

| Caso | URL | Resultado Esperado |
|---------|--|--|
| General | {{baseUrl}}/history/invoices/P-TEST-987 | 200 OK con lista de facturas |
| Filtro | {{baseUrl}}/history/invoices/P-TEST-987?status=pendiente | 200 OK con facturas en estado pendiente |