

## **ANEXO 3. API PARA INTEROPERABILIDAD SIITP-RCC**

**Proyecto: SIITP**

## ÍNDICE GENERAL

Capítulo	Descripción	Página
<b>1.</b>	<b>INTRODUCCIÓN</b>	<b>6</b>
<b>2.</b>	<b>GENERALIDADES</b>	<b>7</b>
2.1	SERVICIO REST	7
2.1.1	Autenticación	7
2.1.2	Errores Generales	7
2.2	SERVICIO GRAPHQL	7
2.2.1	Autenticación	8
2.2.2	Errores Generales	9
<b>3.</b>	<b>DESCARGA DE CONFIGURACIONES</b>	<b>11</b>
3.1	MODOS DE TRANSPORTE	11
3.2	RUTAS ASIGNADAS	11
3.3	ITINERARIOS DE UNA RUTA	12
3.4	CUPO DE VEHÍCULOS	13
3.5	PERFILES DE USUARIO	14
3.6	MEDIOS DE PAGO	14
3.7	POLÍTICAS TARIFARIAS	15
3.8	TARIFAS DE UNA POLÍTICA TARIFARIA	17
<b>4.</b>	<b>CARGAR DE CONFIGURACIONES</b>	<b>19</b>
4.1	VEHÍCULO MATRICULADO	19
4.2	PLAN DE SERVICIOS	28
<b>5.</b>	<b>REPORTAR OPERACIÓN DE RECAUDO</b>	<b>36</b>
5.1	MEDIOS DE SEGURIDAD INICIALIZADOS	36
5.2	MEDIOS DE PAGO GENERADOS	38
5.3	PAGO EN TRANSPORTE	45
5.4	RECARGA DE MEDIO DE PAGO	52
<b>6.</b>	<b>REPORTE OPERACIÓN CONTROL DE FLOTA</b>	<b>59</b>
6.1	REPORTE EN LÍNEA	59
6.1.1	Inicio de Servicio	59
6.1.2	Fin de Servicio	64
6.1.3	Reportar Localización	66
6.1.4	Apertura/Cierre de Puertas	70
6.1.5	Reportar Ingreso/Salida de Pasajeros	75
6.2	REPORTE EN BATCH	79
6.2.1	Reporte Batch	79
6.2.2	Consulta Procesamiento Batch	85

## **1. INTRODUCCIÓN**

Este documento presenta el esquema de comunicaciones para la integración de las diferentes soluciones de recaudo y control de flota en el Área Metropolitana de Barranquilla.

Esta API admite métodos REST para acceder a una serie de recursos, u objetos, a través de un conjunto fijo de operaciones. Las llamadas a la API se realizan enviando una solicitud a un recurso de API y especificando un método y argumentos. Una respuesta con formato detalla el resultado de la operación.

REST hace fácil crear aplicaciones y servicios que puedan usar diferentes clientes y dispositivos. El único requisito es que entiendan HTTP, algo muy común, por ejemplo, para cualquier desarrollador web.

## 2. GENERALIDADES

La plataforma implementa dos canales de comunicación: REST y GraphQL, el operador de transporte puede implementar cualquiera de los canales disponibles.

### 2.1 SERVICIO REST

EndPoint: <https://siitp.ambq.gov.co/api/external-system-gateway/rest/...>

#### 2.1.1 Autenticación

El canal de REST se autentica usando http-basic-access-authentication

#### 2.1.2 Errores Generales

A continuación, se describen los posibles errores en el proceso de autenticación

- Credenciales Inválidas

```
{
  "code": 403,
  "msg": "Credentials has been rejected",
  "name": "AUTH"
}
```

- Permiso Denegado

```
{
  "name": String - nombre del servicio que ha denegado el recurso,
  "code": 2,
  "msg": "Permission denied",
  "method": String - nombre del recurso/método denegado,
}
```

- Timeout Servidor

```
{
  "code": 1,
  "name": "TimeoutError",
  "msg": "Timeout has occurred"
}
```

### 2.2 SERVICIO GRAPHQL

HTTP EndPoint: <https://siitp.ambq.gov.co/api/emi-gateway/graphql/http>

**WebSocket EndPoint:** <wss://siitp.ambq.gov.co/api/emi-gateway/graphql/ws>

**Web Playground:** <https://siitp.ambq.gov.co/api/external-system-gateway/graphql/http>

### 2.2.1 Autenticación

El canal de GraphQL se autentica usando un token JWT que es adquirido mediante el servidor de autenticación.

#### Autenticación y obtención del token JWT

##### EndPoint REST:

<https://siitp.ambq.com.co/auth/realms/FLEET/protocol/openid-connect/token>

**Header:** Content-Type:application/x-www-form-urlencoded

**Body** (x-www-form-urlencoded):

- username: [Nombre de usuario]
- password: [Contraseña de usuario]
- grant\_type:password
- client\_id:emi
- scope:offline\_access

#### Respuesta:

```
{
  "access_token": String - JWT token,
  "expires_in": 900,
  "refresh_expires_in": 0,
  "refresh_token": String - Token pde refresco JWT,
  "token_type": "bearer",
  "not-before-policy": 0,
  "scope": "offline_access email profile"
}
```

#### Uso del token JWT

El token JWT (access\_token) debe ser enviado siempre como parte del HEADER de la petición:

- **Authorization:** Bearer <token>  
Para autenticarse en el Playground usando el token JWT se debe ingresar el siguiente JSON en la parte inferior "HTTP HEADER":
- {"Authorization": "Bearer <token> "}

### 2.2.2 Errores Generales

A continuación, se describen los posibles errores en el proceso de autenticación

- Credenciales Inválidas

```
{
  ... ,
  "errors": [
    {
      ... ,
      "message": {
        "code": 2001,
        "msg": "Invalid token.",
        "name": "Token"
      },
      ... ,
    }
  ]
}
```

- Permiso denegado

```
{
  ... ,
  "errors": [
    {
      ... ,
      "message": {
        "code": 2,
        "method": String - nombre del recurso denegado,
        "msg": "Permission denied",
        "name": String - nombre del servicio que ha denegado el recurso
      },
      ... ,
    }
  ]
}
```

- Timeout Servidor

```
{  
  ... ,  
  "errors": [  
    {  
      ... ,  
      "message": {  
        "code": 1,  
        "msg": "Timeout has occurred",  
        "name": "TimeoutError"  
      },  
      ... ,  
    }  
  ]  
}
```

### 3. DESCARGA DE CONFIGURACIONES

A continuación, se describen los métodos disponibles para la descarga de configuraciones por parte de los operadores de transporte.

#### 3.1 MODOS DE TRANSPORTE

- **REST**  
**Url:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/modes>  
**Método:** GET  
**Parámetros:** N/A
- **GraphQL**  
**Query:** Modes  
**Parámetros:** N/A

**Respuesta:**

```
[
  {
    "id": String - Id único del Modo,
    "active": Boolean - indicador si el modo está activo,
    "name": String - Nombre del Modo,
    "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
    "description": String - Nota descriptiva,
    "metadata": {
      "createdBy": String - Creado por,
      "createdAt": Number - Fecha creación en milisegundos UTC,
      "updatedBy": String - Editado por,
      "updatedAt": Number - Fecha edición en milisegundos UTC
    },
  },
]
```

#### 3.2 RUTAS ASIGNADAS

- **REST**  
**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/routes>  
**Método:** GET  
**Parámetros** : N/A  
**Cuerpo:** N/A
- **GraphQL**  
**Query:** Routes  
**Parámetros:** N/A



## Respuesta exitosa

```
[
  {
    "id": String - Id único del Modo,
    "active": Boolean - indicador si el modo está activo,
    "name": String - Nombre del Modo,
    "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
    "code": String - Código ruta según la autoridad,
    "system": String - (opc) sistema según la autoridad,
    "type": String - (opc) tipo según la autoridad,
    "description": String - Nota descriptiva,
    "metadata": {
      "createdBy": String - Creado por,
      "createdAt": Number - Fecha creación en milisegundos UTC,
      "updatedBy": String - Editado por,
      "updatedAt": Number - Fecha edición en milisegundos UTC
    },
    "divipolCode": String - código divipol
    "modelId": String - ID Modo a la que pertenece,
    "companyId": String - ID compañía (Operador de transporte) a la que pertenece,
  }
]
```

### 3.3 ITINERARIOS DE UNA RUTA

- **REST**

**Url:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/itineraries>

**Método:** GET

**Parámetros :** N/A

**Cuerpo:** N/A | JSON

```
{
  "routeId": String - Id único de la ruta de la que se quieren obtener los itinerarios
}
```

- **GraphQL**

**Query:** Itineraries

**Parámetros:**

```
{
  "routeId": String - Id único de la ruta de la que se quieren obtener los itinerarios
}
```

```

{
  "id": String - Id único del Itinerario,
  "active": Boolean - Indicador si el itinerario está activo,
  "name": String - Nombre del itinerario,
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "companyId": String - Id del operador de transporte a la cual pertenece,
  "routeId": String - Id de la ruta a la cual pertenece,
  "direction": String - Dirección del itinerario,
  "path": {
    "type": String - Tipo de punto de control (Point | Control | Stop)
    "name": String - Nombre del punto de control
    "radius": Number - Radio del punto de control
    "eta": Number - Tiempo estimado de llegada al punto de control
    "maxSpeed": Number - Vel max permitido al pasar por el punto de control
    "checkPointId": String - id único del punto de control
    "coords": [
      Number - Lista de dos elementos que contiene las coordenadas del punto de control [lat, lng]
    ]
  },
  "metadata": {
    "createdBy": String - Creado por,
    "createdAt": Number - Fecha creación en milisegundos UTC,
    "updatedBy": String - Editado por,
    "updatedAt": Number - Fecha edición en milisegundos UTC
  }
}

```

### 3.4 CUPO DE VEHÍCULOS

- **REST**  
**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/vehicle-quota>  
**Método:** GET  
**Parámetros :** N/A  
**Cuerpo:** N/A
- **GraphQL**  
**Query:** VehicleQuota  
**Parámetros:** N/A

#### Respuesta exitosa

```

{
  "id": String - Id único del operador de transporte,
  "vehicleQuota": String - Cupo de vehículos asignado al operador de transporte,
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
}

```

**Respuesta exitosa**  
**3.5 PERFILES DE USUARIO**

- **REST**  
**Url:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/profiles>  
**Método:** GET  
**Parámetros :** N/A  
**Cuerpo:** N/A
- **GraphQL**  
**Query:** Profiles  
**Parámetros:** N/A

**Respuesta exitosa**

```
{  
  "id": String - Id único del perfil,  
  "active": Boolean - indicador si el perfil está activo,  
  "name": String - Nombre del perfil,  
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,  
  "description": String - Nota descriptiva,  
  "minRechargeValue": Number - Valor mínimo de recarga asignado al perfil,  
  "maxBalanceValue": Number - Valor máximo de recarga asignado al perfil,  
  "minTimeBetweenValidations": Number - Valor mínimo en minutos que transcurre entre validaciones para el perfil,  
  "prm": Boolean - indicador si el perfil está habilitado para personas con movilidad reducida,  
}
```

**3.6 MEDIOS DE PAGO**

- **REST**  
**Url:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/payment-medium-types>  
**Método:** GET  
**Parámetros :** N/A
- **GraphQL**  
**Query:** PaymentMediumTypes  
**Parámetros:** N/A

```

{
  "id": String - Id único del tipo de medio de pago,
  "active": Boolean - indicador si el tipo de medio de pago está activo,
  "name": String - Nombre del tipo de medio de pago,
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "Mappings": [
    {
      "version": Number - Versión del mapping,
      "active": Boolean - indicador si el mapping está activo,
      "readOnly": Boolean - indicador si el mapping está marcado como solo lectura,
      "mapping": JSON - estructura (body) del mapping,
      "description": String - Nota descriptiva,
      "changelog": String - Notas de información de los cambios realizados en la versión,
      "metadata":
        {
          "createdBy": String - Creado por,
          "createdAt": Number - Fecha creación en milisegundos UTC,
          "updatedBy": String - Editado por,
          "updatedAt": Number - Fecha edición en milisegundos UTC
        }
    }
  ]
  "metadata": {
    "createdBy": String - Creado por,
    "createdAt": Number - Fecha creación en milisegundos UTC,
    "updatedBy": String - Editado por,
    "updatedAt": Number - Fecha edición en milisegundos UTC
  }
}

```

### 3.7 POLÍTICAS TARIFARIAS

- **REST**  
**Url:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/farePolicies>  
**Método:** GET  
**Parámetros :** N/A  
**Cuerpo:** N/A
- **GraphQL**  
**Query:** FarePolicies  
**Parámetros:** N/A

**Respuesta exitosa**  
**Respuesta exitosa**

```
{
  "id": String - Id único de la política tarifaria,
  "active": Boolean - indicador si la política tarifaria está activa,
  "name": String - Nombre de la política tarifaria,
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "description": String - Nota descriptiva,
  "validTo": Number - Fecha en milisegundos en la que deja de ser válida la política tarifaria,
  "validFrom": Number - Fecha en milisegundos en la que empieza a ser válida la política
tarifaria,
  "version": String - Versión de la política tarifaria,
  "publishedVersion": String - Versión publicada de la política tarifaria,
  "graph":{
    "edges": [
      "data": {
        "fareId": String - Id único de la tarifa,
        "label": String - Etiqueta de la arista,
        "source": String - Id del nodo fuente,
        "target": String - Id del nodo destino,
        "type": String - tipo de arista,
      }
    ],
    "nodes": [
      "data": {
        "active": Boolean - indicador si el nodo está activo,
        "id": String - Id único del nodo,
        "label": String - Etiqueta del nodo,
        "name": String - Nombre del nodo,
        "type": String - Tipo del nodo,
      }
    ]
  },
  "integratedFares":[
    {
      "active": Boolean - indicador si la integración está activa,
      "id": String - Id único de la integración ,
      "name": String - Nombre de la integración,
      "integrations": {
        "fareId": String - Id de la tarifa integrada
        "farePolicyId": String - Id de la política tarifa integrada
        "nodeId": String - Id del nodo asociado a la integración
        "nodeType": String - Tipo del nodo asociado a la integración
        "parentId": String - Id de la integración padre
        "parentNodeId": String - Id del nodo padre
        "parentNodeType": String - Tipo de nodo padre
      }
    }
  ]
}
```

### 3.8 TARIFAS DE UNA POLÍTICA TARIFARIA

- **REST**

**Url:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/fares>

**Método:** GET

**Parámetros :** N/A

**Cuerpo:** JSON

```
{
  "farePolicyId": String - Id de la política tarifaria
  "paginationInput": {
    "page": Number - Página actual en el paginador
    "count": Number - Cantidad de elementos a retornar por página
  }
}
```

JSON SCHEMA:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "farePolicyId": {
      "type": "string"
    },
    "paginationInput": {
      "type": "object",
      "properties": {
        "page": {
          "type": "number"
        },
        "count": {
          "type": "number"
        }
      }
    }
  }
}
```

- **GraphQL**  
**Query:** Fares  
**Parámetros:**

```

{
  "farePolicyId": String - Id de la política tarifaria
  "paginationInput": {
    "page": Number - Página actual en el paginador
    "count": Number - Cantidad de elementos a retornar por página
  }
}

GQL SCHEMA:
input FarePaginationInput {
  "Page"
  page: Int!
  "Count"
  count: Int!
}

```

### Respuesta exitosa

```

{
  "id": String - Id único de la tarifa,
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "farePolicyId": String - Id de la política tarifaria a la cual pertenece,
  "type": String - Tipo de tarifa (Plain or Integrated ),
  "elementId": String - El id del elemento al que pertenece la tarifa,
  "Fares":{
    "profileId": String - Id del perfil asociado a la tarifa
    "fare": Number -Valor de la tarifa
    "specialTimeRanges": {
      "Id": String - id del horario especial
      "name": String - nombre del horario especial
      "days": [
        Number - Lista de días habilitados para el horario especial, estos días (1:L, 2:M, 3:MI, 4:J, 5:V, 6:S,
0:D)
      ]
      "initTime": Number - Hora del día en la que empieza a regir el horario especial (el formato es en minutos
y empieza desde la hora 00:00, ej: 08:00 am seria 480)
      "endTime": Number - Hora del día en la que finaliza el horario especial (el formato es en minutos y
empieza
desde la hora 00:00, ej: 08:00 am seria 480)
      "fare": Number - Valor de la tarifa en el horario especial
      "type": String - Tipo de horario especial
    }
  }
}
}

```

#### 4. CARGAR DE CONFIGURACIONES

A continuación, se describen los métodos disponibles para carga de configuraciones por parte de los operadores de transporte.

##### 4.1 VEHÍCULO MATRICULADO

- **REST**

**Url:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/vehicle>

**Método:** POST

**Parámetros :** N/A

**Cuerpo:** N/A | JSON

```
{
  "vehicleId": String - id único del vehículo
  "ownership": String -Tipo de propiedad del vehículo
  "type": String - Tipo de vehículo
  "plate": String - Placa de vehículo
  "manufacturer": String - Nombre del fabricante del vehículo
  "model": String - Modelo del vehículo
  "year": String - Año de salida del vehículo
  "active": Boolean - indicador si el vehículo está activo
  "emissionsStandard": String - Tipo de emisión
  "schemaType": String - Tipo de esquema
  "colors": [
    String - Colores del vehículo
  ]
  "chassisNumber": String - id del chasis
  "engineNumber": String - id del motor
  "internalNumber": String - id del motor
  "fuelRangeWithFullTank": String - rango de combustible con tanque lleno
  "companyId": String - Id del operador de transporte al cual pertenece,
  "deviceId": String - Id del dispositivo asociado al vehículo,
  "capacity": {
    "seated": Number - Capacidad de pasajeros en asientos
    "standing": Number - Capacidad pasajeros de pie
  },
  "regulatoryCompliance": {
    "operationCardNumber": String - Número de tarjeta de operación
    "operationCardExpeditionDate": Number - Fecha de expedición de la tarjeta de operación en milisegundos
    "operationCardExpirationDate": Number - Fecha de expiración de la tarjeta de operación en milisegundos
    "operationCardCompany": String - Numero de tarjeta de operación de la compañía
    "operationCardInternalNumber": String - Numero de tarjeta de operación de la interno
    "mandatoryInsuranceNumber": String - Numero de seguro obligarorio
    "mandatoryInsuranceExpeditionDate": String - Fecha de expedición del numero de seguro obligatorio
    "mandatoryInsuranceExpirationDate": String - Fecha de expiración del seguro obligatorio
    "mandatoryInsuranceIssuer": String - Nombre del seguro
    "technomechanicalNumber": String - Numero de tecnico mecanica
    "technomechanicalExpeditionDate": String - Fecha de expedición de la técnico mecánica
    "technomechanicalExpirationDate": String - Fecha de expiración de la tecnico mecánica
    "technomechanicalIssuer": String - Nombre del emisor de la técnico mecánica
  }
}
```



```
}  
}  
  
JSON SCHEMA:
```

```
{  
  "type": "object",  
  "properties": {  
    "active": {  
      "type": "boolean"  
    },  
    "organizationId": {  
      "type": "string"  
    },  
    "ownership": {  
      "type": "string",  
      "enum": [  
        "OWN",  
        "AGREEMENT",  
        "AFFILIATE"  
      ]  
    },  
    "type": {  
      "type": "string",  
      "enum": [  
        "OTHER",  
        "BUS",  
        "BUSETA",  
        "MICRO-BUS"  
      ]  
    },  
    "plate": {  
      "type": "string"  
    },  
    "manufacturer": {  
      "type": "string",  
      "enum": [  
        "OTHER",  
        "AGRALE",  
        "ALFA ROMEO",  
        "ASTON MARTIN",  
        "AUDI",  
        "BENTLEY",  
        "BENZ",  
        "BMW",  
        "BUGATTI",  
        "CADILLAC",  
        "CHEVROLET",  
        "CHRYSLER",  
        "CITROEN",  
        "CORVETTE",  
        "DAF",
```

"DACIA",  
"DAEWOO",  
"DAIHATSU",  
"DATSUN",  
"DE LOREAN",  
"DINO",  
"DODGE",  
"FARBOUD",  
"FERRARI",  
"FIAT",  
"FORD",  
"HINO",  
"HONDA",  
"HUMMER",  
"HYUNDAI",  
"JAGUAR",  
"JEEP",  
"KIA",  
"KOENIGSEGG",  
"LADA",  
"LAMBORGHINI",  
"LANCIA",  
"LAND ROVER",  
"LEXUS",  
"LIGIER",  
"LINCOLN",  
"LOTUS",  
"MARTINI",  
"MASERATI",  
"MAXUS",  
"MAYBACH",  
"MAZDA",  
"MCLAREN",  
"MERCEDES",  
"MERCEDES-BENZ",  
"MINI",  
"MITSUBISHI",  
"NISSAN",  
"NOBLE",  
"OPEL",  
"PEUGEOT",  
"PONTIAC",  
"PORSCHE",  
"RENAULT",  
"ROLLS-ROYCE",  
"ROVER",  
"SAAB",  
"SEAT",  
"SKODA",  
"SMART",  
"SPYKER",  
"SUBARU",  
"SUZUKI",

```
"TOYOTA",
"VAUXHALL",
"VOLKSWAGEN",
"VOLVO",
"YUTONG"
]
},
"model": {
  "type": "string"
},
"year": {
  "type": "integer"
},
"emissionsStandard": {
  "type": "string",
  "enum": [
    "OTHER",
    "EURO 1",
    "EURO 2",
    "EURO 3",
    "EURO 4",
    "EURO 5"
  ]
},
"schemaType": {
  "type": "string",
  "enum": [
    "NONE",
    "CLASSIC",
    "FLOWER",
    "OTHER"
  ]
},
"colors": {
  "type": "array",
  "items": [
    {
      "type": "string",
      "enum": ["OTHER",
        "ALUMINUM",
        "BEIGE",
        "BLACK",
        "BLUE",
        "BROWN",
        "BRONZE",
        "CLARET",
        "COPPER",
        "CREAM",
        "GOLD",
        "GRAY",
        "GREEN",
        "MAROON",
        "METALLIC",
```

```
"NAVY",
"ORANGE",
"PINK",
"PURPLE",
"RED",
"ROSE",
"RUST",
"SILVER",
"TAN",
"TURQUOISE",
"WHITE",
"YELLOW"]
}
]
},
"chassisNumber": {
  "type": "string"
},
"engineNumber": {
  "type": "string"
},
"internalNumber": {
  "type": "string"
},
"capacity": {
  "type": "object",
  "properties": {
    "seated": {
      "type": "integer"
    },
    "standing": {
      "type": "integer"
    }
  }
},
"regulatoryCompliance": {
  "type": "object",
  "properties": {
    "operationCardNumber": {
      "type": "string"
    },
    "operationCardExpeditionDate": {
      "type": "integer"
    },
    "operationCardExpirationDate": {
      "type": "integer"
    },
    "operationCardCompany": {
      "type": "string"
    },
    "operationCardInternalNumber": {
      "type": "string"
    }
  }
},
```

```

"mandatoryInsuranceNumber": {
  "type": "string"
},
"mandatoryInsuranceExpeditionDate": {
  "type": "integer"
},
"mandatoryInsuranceExpirationDate": {
  "type": "integer"
},
"mandatoryInsuranceIssuer": {
  "type": "string"
},
"technomechanicalNumber": {
  "type": "string"
},
"technomechanicalExpeditionDate": {
  "type": "integer"
},
"technomechanicalExpirationDate": {
  "type": "integer"
},
"technomechanicalIssuer": {
  "type": "string"
}
}
},
"deviceId": {
  "type": "string"
}
}
}

```

- **GraphQL**  
**Mutation:** PersistVehicle  
**Parámetros:**

```

{
  "vehicleId": String - id único del vehículo,
  "input" {
    "ownership": String - Tipo de propiedad del vehículo
    "type": String - Tipo de vehículo
    "plate": String - Placa de vehículo
    "manufacturer": String - Nombre del fabricante del vehículo
    "model": String - Modelo del vehículo
    "year": String - Año de salida del vehículo
    "active": Boolean - indicador si el vehículo está activo
    "emissionsStandard": String - Tipo de emisión
    "schemaType": String - Tipo de esquema
    "colors": [

```

```

    String - Colores del vehículo
  ]
  "chassisNumber": String - id del chasis
  "engineNumber": String - id del motor
  "internalNumber": String - id del motor
  "fuelRangeWithFullTank": String - rango de combustible con tanque lleno
  "companyId": String - Id del operador de transporte al cual pertenece,
  "deviceId": String - Id del dispositivo asociado al vehículo,
  "capacity": {
    "seated": Number - Capacidad de pasajeros en asientos
    "standing": Number - Capacidad pasajeros de pie
  },
  "regulatoryCompliance": {
    "operationCardNumber": String - Número de tarjeta de operación
    "operationCardExpeditionDate": Number - Fecha de expedición de la tarjeta de operación en milisegundos
    "operationCardExpirationDate": Number - Fecha de expiración de la tarjeta de operación en milisegundos
    "operationCardCompany": String - Numero de tarjeta de operación de la compañía
    "operationCardInternalNumber": String - Número de tarjeta de operación de la interno
    "mandatoryInsuranceNumber": String - Número de seguro obligarorio
    "mandatoryInsuranceExpeditionDate": String - Fecha de expedición del numero de seguro obligatorio
    "mandatoryInsuranceExpirationDate": String - Fecha de expiración del seguro obligatorio
    "mandatoryInsuranceIssuer": String - Nombre del seguro
    "technomechanicalNumber": String - Número de tecnico mecanica
    "technomechanicalExpeditionDate": String - Fecha de expedición de la tecnico mecanica
    "technomechanicalExpirationDate": String - Fecha de expiración de la tecnico mecanica
    "technomechanicalIssuer": String - Nombre del emisor de la técnico mecánica
  }
}
}
}

```

**GraphQL SCHEMA:**

```
input OrganizationMngVehicleInput {
```

```

  "ownership type"
  ownership: String
  "Vehicle Type"
  type: String
  "License palte"
  plate: String!
  "automotive manufacturer"
  manufacturer: String
  "model"
  model: String
  "year"
  year: Int
  "Vehicle active flag"
  active: Boolean!
  "emissions standard"
  emissionsStandard: String
  "schema type"
  schemaType: String

```

```

"Vehicle Colors"
colors: [String]
"chassis number"
chassisNumber: String
"engine number"
engineNumber: String
"Organization's internal number"
internalNumber: String
fuelRangeWithFullTank: Int
"Company id"
companyId: String
"Assigned Device Id"
deviceId: String
"Vehicle passenger capacity"
capacity: OrganizationMngVehicleCapacityInput
"Law regulatory compliance docs"
regulatoryCompliance: OrganizationMngVehicleRegulatoryComplianceInput
}

"Vehicle passenger capacity"
input OrganizationMngVehicleCapacityInput{
  seated: Int
  standing: Int
}

"Law regulatory compliance docs"
input OrganizationMngVehicleRegulatoryComplianceInput{
  operationCardNumber: String,
  operationCardExpeditionDate: BigInt
  operationCardExpirationDate: BigInt
  operationCardCompany: String
  operationCardInternalNumber: String

  mandatoryInsuranceNumber: String
  mandatoryInsuranceExpeditionDate: BigInt
  mandatoryInsuranceExpirationDate: BigInt
  mandatoryInsuranceIssuer: String

  technomechanicalNumber: String
  technomechanicalExpeditionDate: BigInt
  technomechanicalExpirationDate: BigInt
  technomechanicalIssuer: String
}

```

## Respuesta exitosa

```
{
  "vehicleId": String - id único del vehículo
  "ownership": String -Tipo de propiedad del vehíoculo
  "type": String - Tipo de vehículo
  "plate": String - Placa de vehículo
  "manufacturer": String - Nombre del fabricante del vehículo
  "model": String - Modelo del vehículo
  "year": String - Año de salida del vehículo
  "active": Boolean - indicador si el vehículo está activo
  "emissionsStandard": String - Tipo de emisión
  "schemaType": String - Tipo de esquema
  "colors": [
    String - Colores del vehículo
  ]
  "chassisNumber": String - id del chasis
  "engineNumber": String - id del motor
  "internalNumber": String - id del motor
  "fuelRangeWithFullTank": String - rango de combustible con tanque lleno
  "companyId": String - Id del operador de transporte al cual pertenece,
  "deviceId": String - Id del dispositivo asociado al vehículo,
  "capacity": {
    "seated": Number - Capacidad de pasajeros en asientos
    "standing": Number - Capacidad pasajeros de pie
  },
  "regulatoryCompliance": {
    "operationCardNumber": String - Numero de tarjeta de operación
    "operationCardExpeditionDate": Number - Fecha de expedición de la tarjeta de operación en milisegundos
    "operationCardExpirationDate": Number - Fecha de expiración de la tarjeta de operación en milisegundos
    "operationCardCompany": String - Numero de tarjeta de operación de la compañía
    "operationCardInternalNumber": String - Numero de tarjeta de operación de la interno
    "mandatoryInsuranceNumber": String - Numero de seguro obligarorio
    "mandatoryInsuranceExpeditionDate": String - Fecha de expedición del numero de seguro obligatorio
    "mandatoryInsuranceExpirationDate": String - Fecha de expiración del seguro obligatorio
    "mandatoryInsuranceIssuer": String - Nombre del seguro
    "technomechanicalNumber": String - Numero de tecnico mecanica
    "technomechanicalExpeditionDate": String - Fecha de expedición de la tecnico mecanica
    "technomechanicalExpirationDate": String - Fecha de expiración de la tecnico mecanica
    "technomechanicalIssuer": String - Nombre del emisor de la técnico mecánica
  }
}
```



## 4.2 PLAN DE SERVICIOS

- **REST**

**Url:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/service-planning>

**Método:** POST

**Parámetros :** N/A

**Cuerpo:** JSON

```
{
  "id":String - id único de la planificación,
  "name": String - Nombre de la planificación,
  "active":Boolean - indicador si la planificación está activa,
  "description": String - Descripción de la planificación,
  "routes": [
    {
      "routeId":String - id de la ruta que se está planificando ,
      "itineraries": [
        {
          "itineraryId": String - id del itinerario que se está planificando y al cual se le generan los servicios (Debe pertenecer a la ruta)
          "timetables": [
            {
              "days": Array(String) - días de la semana a los que aplica esta planificación de servicios (["0","1","2","3","4","5","6"])
              "times": Array(Array(Int)) - tiempo de minutos que se configura para el lanzamiento de servicios y el tiempo debe tomar entre puntos del itinerario
            }
          ]
        }
      ]
    }
  ]
}
```

**JSON SCHEMA:**

```
{
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "name": {
      "type": "string"
    },
    "active": {
      "type": "boolean"
    },
    "description": {
      "type": "string"
    },
    "routes": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "routeId": {
              "type": "string"
            },
            "itineraries": {
              "type": "array",

```

```
"items": [  
  {  
    "type": "object",  
    "properties": {  
      "itineraryId": {  
        "type": "string"  
      },  
      "timetables": {  
        "type": "array",  
        "items": [  
          {  
            "type": "object",  
            "properties": {  
              "days": {  
                "type": "array",  
                "items": [  
                  {  
                    "type": "string"  
                  }  
                ]  
              },  
              "times": {  
                "type": "array",  
                "items": [  
                  {  
                    "type": "array",  
                    "items": [  
                      {  
                        "type": "integer"  
                      }  
                    ]  
                  }  
                ]  
              }  
            }  
          }  
        ]  
      }  
    }  
  }  
],  
"additionalProperties": false  
}
```

- **GraphQL**  
**Mutation:** PersistServicePlanning  
**Parámetros:**

```

{
  servicePlanningId:String - id único de la planificación
  Input:{
    {
      "name": String - Nombre de la planificación,
      "active":Boolean - indicador si la planificación está activa,
      "description": String - Descripción de la planificación,
      "routes": [
        {
          "routeId":String - id de la ruta que se está planificando ,
          "itineraries": [
            {
              "itineraryId": String - id del itinerario que se está planificando y al cual se le generan los servicios
              (Debe pertenecer a la ruta)
              "timetables": [
                {
                  "days": Array(String) - días de la semana a los que aplica esta planificación de servicios
                  ([“0”,“1”,“2”,“3”,“4”,“5”, “6”])
                  "times": Array(Array(Int)) - tiempo de minutos que se configura para el lanzamiento de
                  servicios y el tiempo debe tomar entre puntos del itinerario
                }
              ]
            }
          ]
        }
      ]
    }
  }
}

```

GQL SCHEMA:

```

type ServiceMngServicePlanning {
  "ServicePlanning ID"
  id: ID!
  "Company id"
  companyId: String
  "name"
  name: String
  "organization id"
  organizationId: String!
  "ServicePlanning description"
  description: String
  "ServicePlanning active flag"
  active: Boolean
  "ServicePlanning active flag"
  routes: [JSON]
  "ServicePlanning version"
}

```

```

version:String
}

input ServiceMngServicePlanningInput {
  "name"
  name: String!
  "ServicePlanning description"
  description: String
  "ServicePlanning active flag"
  active: Boolean
  "ServicePlanning active flag"
  routes: [JSON]
}
type Mutation {
  "Get the vehicle quota of an company"
  PersistServicePlanning(servicePlanningId: String!, input: ServiceMngServicePlanningInput!):
  ServiceMngServicePlanning
}

```

### Respuesta exitosa

```

{
  "id": String - id único de la planificación
  "name":String - Nombre de la planificación,
  "active": Boolean - indicador si la planificación está activa,
  "description":String - Descripción de la planificación,
  "routes": [
    {
      "routeId": String - id de la ruta que se está planificando ,
      "routeName": String - Nombre de la ruta que se está planificando ,
      "routeCode": String - Código de la ruta que se está planificando ,
      "itineraries": [
        {
          "itineraryId": "String - id del itinerario que se está planificando y al cual se le generan los servicios (Debe pertenecer a la ruta)
          "itineraryName":String - nombre del itinerario que se está planificando y al cual se le generan los servicios (Debe pertenecer a la ruta)
          "checkPoints": [ {
            "type": Tipo de punto (Control o Parada),
            "name":Nombre,
            "eta": Int estimado tiempo de llegada a este punto,
            "checkPointId": String - id unico del punto
          }
        ],
        "timetables": [
          {
            "days": Array(String) - días de la semana a los que aplica esta planificación de servicios
            ["0","1","2","3","4","5", "6"]
            "times": Array(Array(Int)) - tiempo de minutos que se configura para el lanzamiento de servicios y el tiempo debe tomar entre puntos del itinerario
          }
        ]
      }
    ]
  }
}

```

```
    ]  
  }  
} ]  
}  
},  
"organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece  
"companyId":String - Id del operador de transporte al cual pertenece,  
"version": String - Version en la que se cargó la planificación,  
"metadata": {  
  "createdBy": String - Creado por,  
  "createdAt": Number - Fecha creación en milisegundos UTC,  
  "updatedBy": String - Editado por,  
  "updatedAt": Number - Fecha edición en milisegundos UTC  
}  
}
```

**Respuestas con errores:**

- Planificación está siendo editada por otro usuario

```
{  
  "code": 17020  
  "msg": Service planning cannot be edited while it is in edit mode on the platform  
  "name": ServicePlanningCannotBeEdited  
}
```

- Planificación está siendo editada por otro usuario

```
{  
  "code": 17021  
  "msg": Service planning cannot be created/updated because an empty name was sent,  
  "name": ServicePlanningTooShortName  
}
```

- Nombre de la planificación muy corto

```
{  
  "code": 17022  
  "msg": Service planning cannot be created/updated because name doesnt have at least 4 characters,  
  "name": ServicePlanningTooShortName  
}
```

- Nombre de la planificación muy largo

```
{  
  "code": 17023,  
  "msg": "Service planning cannot be created/updated because name have more than 100 characters,  
  "name": "ServicePlanningTooLongName  
}
```

- Id de ruta inválido

```
{  
  "code": 17024  
  "msg": "Service planning cannot be created/updated because couldn't find routes with Id: id de la ruta enviada,  
  "name": "ServicePlanningCouldNotFindRoute  
}
```

- Id de itinerario inválido

```
{  
  "code": 17025  
  "msg": "Service planning cannot be created/updated because couldn't find itineraries with Id: Id del itinerario enviado,  
  "name": "ServicePlanningCouldNotFindItinerary  
}
```

- Id de itinerario no pertenece a la ruta

```
{  
  "code": 17026  
  "msg": "Service planning cannot be created/updated because selected itinerary doesn't belong to route Id: Id del itinerario enviado,  
  "name": "ServicePlanningDoesNotBelongToRoute  
}
```

- Arreglo de días nulo

```
{  
  "code": 17027  
  "msg": "Service planning cannot be created/updated because day cannot be null arreglo de dias enviado  
  "name": "ServicePlanningDayCannotBeNull  
}
```

- Día invalido (NaN) dentro del arreglo de días

```
{  
  "code": 17028  
  "msg": Service planning cannot be created/updated because day has not a valid value 0-6: arreglo de días enviados  
  "name": ServicePlanningDaylsNotValid  
}
```

- Día invalido (fuera del rango 0-6) dentro del arreglo de días

```
{  
  "code": 17029  
  "msg": Service planning cannot be created/updated because day has not a valid value 0-6: arreglo de dias enviados  
  "name": ServicePlanningDaylsNotValid  
}
```

- Hora, en minutos, fuera del rango (0-1439)

```
{  
  "code": 17030  
  "msg": Service planning cannot be created/updated because times is not inside the time range (0 - 1439)(minutes):  
arreglo de tiempos enviados  
  "name": ServicePlanningDaylsNotValid  
}
```

- Planificación sin asignación de rutas

```
{  
  "code": 17031  
  "msg": Service planning cannot be created/updated because an empty routes property was sent,  
  "name": ServicePlanningEmptyRoutes  
}
```

- Ruta de planificación sin asignación de itinerarios

```
{  
  "code": 17032  
  "msg":Service planning cannot be created/updated because an empty itinerary property was sent. Información de la  
ruta enviada,  
  "name": ServicePlanningEmptyRoutes  
}
```



## 5. REPORTAR OPERACIÓN DE RECAUDO

A continuación, se describen los métodos disponibles para reporte de las configuraciones y transacciones de recaudo por parte de los operadores de recaudo.

### 5.1 MEDIOS DE SEGURIDAD INICIALIZADOS

- **REST**  
**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/security-medium>  
**Método:** POST  
**Parámetros :** N/A  
**Cuerpo:** JSON

```
{
  "active": Boolean - indicador si el medio de seguridad está activo,
  "description":String - Descripción del medio de seguridad,
  "location": String - Descripción en palabras de donde se encuentra el medio de seguridad ,
  "state": String - Indica el estado del medio de seguridad (Adquirido, pre-inicializado, inicializado),
  "mediumId": String - Id único del medio de seguridad,
  "purpose": String - >Describe el uso que se le va dar el medio de seguridad
}
JSON SCHEMA:
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "active": {
      "type": "boolean"
    },
    "description": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "state": {
      "type": "string",
      "enum": ["ACQUIRED",
        "PRE_INITIALIZED",
        "INITIALIZED"]
    },
    "mediumId": {
      "type": "string"
    },
    "purpose": {
      "type": "string"
    }
  },
  "required": [
    "active",
```



```

"SecurityMedium description"
description: String
"SecurityMedium active flag"
active: Boolean!
}
type Mutation {
  "Persist PaymentMedium info"
  PersistSecurityMedium(
    mediumId: String!
    input: PaymentMediumSecurityMediumInput!
  ): PaymentMediumSecurityMedium @requireAuth
}

```

### Respuesta exitosa

```

{
  "mediumId": String - Id único del medio de seguridad,
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "active": Boolean - indicador si el medio de seguridad está activo,
  "description":String - Descripción del medio de seguridad,
  "location": String - Descripción en palabras de donde se encuentra el medio de seguridad ,
  "state": String - Indica el estado del medio de seguridad (Adquirido, pre-inicializado, inicializado),
  "purpose": String - >Describe el uso que se le va dar el medio de seguridad
  "id": String - Id único del medio de seguridad, en el sistema
  "metadata": {
    "createdBy": String - Creado por,
    "createdAt": Number - Fecha creación en milisegundos UTC,
    "updatedBy": String - Editado por,
    "updatedAt": Number - Fecha edición en milisegundos UTC
  },
}

```

## 5.2 MEDIOS DE PAGO GENERADOS

- REST**  
**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/payment-medium>  
**Método:** POST  
**Parámetros :** N/A  
**Cuerpo:** JSON

```

{
  "id": String - Id único del medio de seguridad, en el sistema,
}

```

```

"paymentMediumTypeld": String - Id de tipo de medio de pago,
"active": Boolean - indicador si el modo está activo,,
"state": String- Estado del medio de pago,
"currentData": {
  "properties": {
    "timestamp":,
    "blocked": Boolean - indicador si el modo está bloqueado,
    "data": JSON - (opc) - datos binarios internos del medio de pago,
    "version": String - Version en la que se encuentra el medio de pago
  }
},
"stateHistory": [
  {
    "state": String - Estado en el que se encontraba el medio de pago,
    "timestamp": Number - Fecha en milisegundos UTC,,
    "responsibleUserFullname": String - Nombre del responsable,
    "responsibleUserId": String - id único del responsable del cambio
  }
],
"endUser": {
  "id": String - id unico de usuario final,
  "name": String - Nombre del Modo,
  "documentType": String- Tipo de documento del usuario final del sistema
  "documentId":String - Documento de identidad del usuario final
}
}
JSON SCHEMA:
{
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "paymentMediumTypeld": {
      "type": "string"
    },
    "active": {
      "type": "boolean"
    },
    "state": {
      "type": "string",
      "enum": ["ACQUIRED",
        "PRE_INITIALIZED",
        "INITIALIZED",
        "PERSONALIZED",
        "EMITED",
        "BLOCKED"]
    },
    "currentData": {
      "type": "object",
      "properties": {
        "timestamp": {
          "type": "integer"
        }
      }
    }
  }
}

```

```
    },
    "blocked": {
      "type": "boolean"
    },
    "data": {
      "type": "object"
    },
    "version": {
      "type": "string"
    }
  }
},
"stateHistory": {
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "state": {
          "type": "string"
        },
        "timestamp": {
          "type": "integer"
        },
        "responsibleUserFullName": {
          "type": "string"
        },
        "responsibleUserId": {
          "type": "string"
        }
      }
    }
  ]
},
"endUser": {
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "name": {
      "type": "string"
    },
    "documentType": {
      "type": "string",
      "enum": ["CITENSHIP_CARD",
        "FOREIGNER",
        "PASSPORT",
        "IDENTITY_CARD"]
    },
    "documentId": {
      "type": "string"
    }
  }
}
```

```

    },
    "required": [
      "id",
      "name",
      "documentType",
      "documentId"
    ]
  }
},
"required": [
  "id",
  "paymentMediumTypeId",
  "active",
  "state",
  "endUser"
]
}

```

- **GraphQL**  
**Mutation:** PersistPaymentMedium  
**Parámetros:**

```

{
  "paymentMediumId": String - Id único del medio de seguridad, en el sistema,
  input: {
    {
      "paymentMediumTypeId": String - Id de tipo de medio de pago,
      "active": Boolean - indicador si el modo está activo,,
      "state": String- Estado del medio de pago,
      "currentData": {
        "properties": {
          "timestamp":1596122733000,
          "blocked": Boolean - indicador si el modo está bloqueado,
          "data": JSON - (opc) - datos binarios internos del medio de pago,
          "version": String - Version en la que se encuentra el medio de pago
        }
      },
      "stateHistory": [
        {
          "state": String - Estado en el que se encontraba el medio de pago,
          "timestamp": Number - Fecha en milisegundos UTC,,
          "responsibleUserFullname": "TEST TEST",
          "responsibleUserId": String - id unico del responsable del cambio
        }
      ],
      "endUser": {
        "id": String - id unico de usuario final,
        "name": String - Nombre del Modo,
        "documentType": String- Tipo de documento del usuario final del sistema
      }
    }
  }
}

```

```

        "documentId":String - Documento de identidad del usuario final
    }
}
}

```

**GQL SCHEMA:**

```

type PaymentMediumCurrentData {
  timestamp: BigInt
  blocked: Boolean
  data: JSON
  version: String
}

```

```

type PaymentMediumStateHistory {
  timestamp: BigInt
  responsibleUserId: String
  responsibleUserFullname: String
  state: String
}

```

```

type PaymentMediumEndUser {
  id: String
  name: String
  documentType: String
  documentId: String
}

```

```

type PaymentMediumPaymentMedium {
  "PaymentMedium ID"
  id: ID!
  "name"
  name: String!
  "organization id"
  organizationId: String!
  "Payment medium type"
  paymentMediumTypeId: String
  "State history"
  stateHistory: [PaymentMediumStateHistory]
  "End user basic info"
  endUser: PaymentMediumEndUser
  "Payment external system id"
  externalSystemId: String
  "state"
  state: String
  currentData: PaymentMediumCurrentData
  "PaymentMedium active flag"
  active: Boolean!
}

```

```

type PaymentMediumPaymentMediumMetadata {
  createdBy: String
  createdAt: BigInt
  updatedBy: String
}

```

```
    updatedAt: BigInt
  }

type PaymentMediumPaymentMediumListing {
  "Listing"
  listing: [PaymentMediumPaymentMedium]
  "query total result count"
  queryTotalResultCount: Int
}

input PaymentMediumCurrentDataInput {
  timestamp: BigInt
  blocked: Boolean
  data: JSON
  version: String
}

input PaymentMediumStateHistoryInput {
  timestamp: BigInt
  responsibleUserId: String
  responsibleUserFullname: String
  state: String
}

input PaymentMediumEndUserInput {
  id: String!
  name: String!
  documentType: PaymentMediumDocumentTypeEnum!
  documentId: String!
}

enum PaymentMediumStateEnum {
  ACQUIRED
  PRE_INITIALIZED
  INITIALIZED
  PERSONALIZED
  EMITED
  BLOCKED
}

enum PaymentMediumDocumentTypeEnum {
  CITENSHIP_CARD
  FOREIGNER
  PASSPORT
  IDENTITY_CARD
}

input PaymentMediumPaymentMediumInput {
  "Payment medium type"
  paymentMediumTypeId: String!
  "State history"
  stateHistory: [PaymentMediumStateHistoryInput]
  "End user basic info"
```



```

endUser: PaymentMediumEndUserInput!
"state"
state: PaymentMediumStateEnum!
currentData: PaymentMediumCurrentDataInput
"PaymentMedium active flag"
active: Boolean!
}

type Mutation {
  "Persist PaymentMedium info"
  PersistPaymentMedium(
    paymentMediumId: String!
    input: PaymentMediumPaymentMediumInput!
  ): PaymentMediumPaymentMedium @requireAuth
}

```

### Respuesta exitosa

```

{
  "externalSystemId": String - id único enviado en el request,
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "active": Boolean - indicador si el modo está activo,,
  "currentData": {
    "properties": {
      "timestamp":,
      "blocked": Boolean - indicador si el modo está bloqueado,
      "data": JSON - (opc) - datos binarios internos del medio de pago,
      "version": String - Version en la que se encuentra el medio de pago
    }
  }
  "endUser": {
    "id": String - id unico de usuario final,
    "name": String - Nombre del Modo,
    "documentType": String- Tipo de documento del usuario final del sistema
    "documentId":String - Documento de identidad del usuario final
  },
  "metadata": {
    "createdBy": String - Creado por,
    "createdAt": Number - Fecha creación en milisegundos UTC,
    "updatedBy": String - Editado por,
    "updatedAt": Number - Fecha edición en milisegundos UTC
  },
  "paymentMediumTypeId": String - Id de tipo de medio de pago,
  "state": String- Estado del medio de pago,
  "stateHistory": [
    {
      "state": String - Estado en el que se encontraba el medio de pago,
      "timestamp": Number - Fecha en milisegundos UTC,,
      "responsibleUserFullName": String - Nombre del responsable,

```

```

        "responsibleUserId": String - id único del responsable del cambio
    }
],
"Id": String - id único registrado en el sistema
}

```

**Respuesta con error:**

Tipo de medio de pago inválido

```

{
  "name": "Invalid PaymentMediumType",
  "code": 1,
  "msg": ""
}

```

**5.3 PAGO EN TRANSPORTE**

- **REST**  
**URL:** <http://siiip.ambq.gov.co/api/external-system-gateway/rest/collected-fare>  
**Método:** POST  
**Parámetros :** N/A  
**Cuerpo:** JSON

```

{
  "id":String - id único del pago en el transporte con medio de pago ,
  "endUser": {
    "id": String - id unico de usuario final,
    "name": String - Nombre del Modo,
    "documentType": String- Tipo de documento del usuario final del sistema
    "documentId":String - Documento de identidad del usuario final
  },
  "fare": {
    "farePolicyId": String - id único de la política tarifaria aplicada,
    "fareId": String - id único de la tarifa aplicada,,
    "fareType":String - tipo de tarifa aplicada "PLAIN",
    "value":Int - Valor descontado del saldo por la tarifa
  },
  "paymentMedium": {
    "typeId": String - id único de tipo de medio de pago,
    "dataPreState": JSON - (opc) - Estado previo del medio de pago de los datos binarios internos del medio de pago,
    "dataPostState":JSON - (opc) - Estado posterior del medio de pago de los datos binarios internos del medio de
    pago,,
  }
}

```

```

    "externalSystemId":String - id único enviado en el request del medio de pago,
  },
  "terminal": {
    "id":String - id unico del terminal usado para el pago,
    "desc":String - String - Nota descriptiva del terminal usado para el pago ,
    "location": {
      "type":String - tipo de localización enviada en la terminal,
      "coordinates": Array[Number,Number] - ubicación en coordenadas de la terminal usada para el pago,
      "timestamp": Number - Fecha de la ubicación en milisegundos UTC
    }
  },
  "timestamp": Number - Fecha de la terminal en milisegundos UTC
}
JSON SCHEMA:
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "endUser": {
      "type": "object",
      "properties": {
        "id": {
          "type": "string"
        },
        "name": {
          "type": "string"
        },
        "documentType": {
          "type": "string",
          "enum": ["CITENSHIP_CARD",
            "FOREIGNER",
            "PASSPORT",
            "IDENTITY_CARD"]
        },
        "documentId": {
          "type": "string"
        }
      }
    },
    "fare": {
      "type": "object",
      "properties": {
        "farePolicyId": {
          "type": "string"
        },
        "farePolicyVersion": {
          "type": "integer"
        }
      }
    }
  }
}

```

```
"fareType": {
  "type": "string",
  "enum": ["PLAIN",
    "INTEGRATION"]
},
"value": {
  "type": "integer"
}
},
"paymentMedium": {
  "type": "object",
  "properties": {
    "typeId": {
      "type": "string"
    },
    "dataPreState": {
      "type": "object"
    },
    "dataPostState": {
      "type": "object"
    },
    "externalSystemId": {
      "type": "string"
    }
  }
},
"terminal": {
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "desc": {
      "type": "string"
    },
    "location": {
      "type": "object",
      "properties": {
        "type": {
          "type": "string"
        },
        "coordinates": {
          "type": "array",
          "items": [
            {
              "type": "number"
            },
            {
              "type": "number"
            }
          ]
        }
      }
    }
  }
}
}
```



```
}
```

GQL SCHEMA:

```
type CollectedFareEndUser {  
  id: String!  
  name: String!  
  documentType: String!  
  documentId: String!  
}
```

```
type CollectedFarePaymentMedium {  
  typeId: String!  
  dataPreState: JSON  
  dataPostState: JSON  
  externalSystemId: String!  
}
```

```
type CollectedFarePaymentFare {  
  farePolicyId: String!  
  farePolicyVersion: Int!  
  fareType: String!  
  value: Int!  
}
```

```
type CollectedFareLocation {  
  type: String!  
  coordinates: [Float]!  
  timestamp: BigInt!  
}
```

```
type CollectedFarePaymentTerminal {  
  id: String!  
  desc: String  
  location: CollectedFareLocation!  
}
```

```
type PaymentMediumCollectedFare {  
  "timestamp"  
  timestamp: BigInt!  
  "end user"  
  endUser: CollectedFareEndUser!  
  "payment medium info"  
  paymentMedium: CollectedFarePaymentMedium!  
  "terminal"  
  terminal: CollectedFarePaymentTerminal!  
  "Fare"  
  fare: CollectedFarePaymentFare!  
}
```

```
enum CollectedFareDocumentTypeEnum {  
  CITENSHIP_CARD  
  FOREIGNER  
  PASSPORT
```

```

IDENTITY_CARD
}

enum CollectedFareType {
  PLAIN
  INTEGRATION
}

input CollectedFareEndUserInput {
  id: String!
  name: String!
  documentType: CollectedFareDocumentTypeEnum!
  documentId: String!
}
input CollectedFarePaymentMediumInput {
  "payment medium type id"
  typeId: String!
  dataPreState: JSON
  dataPostState: JSON
  externalSystemId: String!
}
input CollectedFarePaymentFareInput {
  farePolicyId: String!
  fareId: String!
  fareType: CollectedFareType!
  value: Int!
}
input CollectedFareLocationInput {
  type: String!
  coordinates: [Float!]
  timestamp: BigInt!
}
input CollectedFarePaymentTerminalInput {
  id: String!
  desc: String
  location: CollectedFareLocationInput!
}
input PaymentMediumCollectedFareInput {
  "timestamp"
  timestamp: BigInt!
  "end user"
  endUser: CollectedFareEndUserInput!
  "payment medium info"
  paymentMedium: CollectedFarePaymentMediumInput!
  "terminal"
  terminal: CollectedFarePaymentTerminalInput!
  "Fare"
  fare: CollectedFarePaymentFareInput!
}
type Mutation {

```

```
"Persist PaymentMedium info"
PersistCollectedFare(
  collectedFareId: String!
  input: PaymentMediumCollectedFareInput!
): PaymentMediumCollectedFare @requireAuth
}
```

Respuesta exitosa

```
{
  "externalSystemId": String - id unico enviado en el request,,
  "organizationId":String - Id de la organización (operador de recaudo) a la cual pertenece,
  "endUser": {
    "id": String - id unico de usuario final,
    "name": String - Nombre del Modo,
    "documentType": String- Tipo de documento del usuario final del sistema
    "documentId":String - Documento de identidad del usuario final
  },
  "fare": {
    "farePolicyId": String - id único de la política tarifaria aplicada,
    "fareId": String - id único de la tarifa aplicada,,
    "fareType":String - tipo de tarifa aplicada "PLAIN",
    "value":Int - Valor descontado del saldo por la tarifa
  },
  "metadata": {
    "createdBy": String - Creado por,
    "createdAt": Number - Fecha creación en milisegundos UTC,
    "updatedBy": String - Editado por,
    "updatedAt": Number - Fecha edición en milisegundos UTC
  },
  "paymentMedium": {
    "typeId": String - id único de tipo de medio de pago,
    "dataPreState": JSON - (opc) - Estado previo del medio de pago de los datos binarios internos del medio de
    pago,
    "dataPostState":JSON - (opc) - Estado posterior del medio de pago de los datos binarios internos del medio de
    pago,,
    "externalSystemId":String - id único enviado en el request del medio de pago,

  },
  "terminal": {
    "id":String - id unico del terminal usado para el pago,
    "desc":String - String - Nota descriptiva del terminal usado para el pago ,
    "location": {
```



```

    "type":String - tipo de localización enviada en la terminal,
    "coordinates": Array[Number,Number] - ubicación en coordenadas de la terminal usada para el pago,
    "timestamp": Number - Fecha de la ubicación en milisegundos UTC
  }
},
"timestamp": Number - Fecha de la terminal en milisegundos UTC
"id":String - id unico del pago en el transporte con medio de pago ,
}

```

#### 5.4 RECARGA DE MEDIO DE PAGO

- **REST**  
**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/payment-medium-recharge>  
**Método:** POST  
**Parámetros :** N/A  
**Cuerpo:** JSON

```

{
  "id":String - id único de la recarga,
  "rechargeValue":Number - Valor de la recarga ,
  "endUser": {
    "id": String - id unico de usuario final,
    "name": String - Nombre del Modo,
    "documentType": String- Tipo de documento del usuario final del sistema
    "documentId":String - Documento de identidad del usuario final
  },
  "paymentMedium": {
    "typeId": String - id único de tipo de medio de pago,
    "dataPreState": JSON - (opc) - Estado previo del medio de pago de los datos binarios internos del medio de
    pago,
    "dataPostState":JSON - (opc) - Estado posterior del medio de pago de los datos binarios internos del medio de
    pago,,
    "externalSystemId":String - id único enviado en el request del medio de pago,
  },
  "terminal": {
    "id":String - id unico del terminal usado para el pago,
    "desc":String - String - Nota descriptiva del terminal usado para el pago ,
    "location": {
      "type":String - tipo de localización enviada en la terminal,
      "coordinates": Array[Number,Number] - ubicación en coordenadas de la terminal usada para el pago,
      "timestamp": Number - Fecha de la ubicación en milisegundos UTC
    }
  },
}

```

```
"timestamp": Number - Fecha de la terminal en milisegundos UTC
}
JSON SCHEMA:
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "rechargeValue": {
      "type": "number"
    },
    "endUser": {
      "type": "object",
      "properties": {
        "id": {
          "type": "string"
        },
        "name": {
          "type": "string"
        },
        "documentType": {
          "type": "string",
          "enum": ["CITENSHIP_CARD",
            "FOREIGNER",
            "PASSPORT",
            "IDENTITY_CARD"]
        },
        "documentId": {
          "type": "string"
        }
      }
    },
    "paymentMedium": {
      "type": "object",
      "properties": {
        "typeId": {
          "type": "string"
        },
        "dataPreState": {
          "type": "object"
        },
        "dataPostState": {
          "type": "object"
        },
        "externalSystemId": {
          "type": "string"
        }
      }
    },
    "terminal": {
      "type": "object",
```



```

    "paymentMedium": {
      "typeId": String - id único de tipo de medio de pago,
      "dataPreState": JSON - (opc) - Estado previo del medio de pago de los datos binarios internos del medio de
      pago,
      "dataPostState":JSON - (opc) - Estado posterior del medio de pago de los datos binarios internos del medio de
      pago,,
      "externalSystemId":String - id único enviado en el request del medio de pago,
    },
    "terminal": {
      "id":String - id unico del terminal usado para el pago,
      "desc":String - String - Nota descriptiva del terminal usado para el pago ,
      "location": {
        "type":String - tipo de localización enviada en la terminal,
        "coordinates": Array[Number,Number] - ubicación en coordenadas de la terminal usada para el pago,
        "timestamp": Number - Fecha de la ubicación en milisegundos UTC
      }
    },
    "timestamp": Number - Fecha de la terminal en milisegundos UTC
  }
}

```

#### GQL SCHEMA:

```

type PaymentMediumRechargeEndUser {
  id: String!
  name: String!
  documentType: String!
  documentId: String!
}

type PaymentMediumRechargePaymentMedium {
  typeId: String!
  dataPreState: JSON
  datePostState: JSON
  externalSystemId: String!
}

type PaymentMediumRechargeLocation {
  type: String!
  coordinates: [Float!]
  timestamp: BigInt!
}

type PaymentMediumRechargePaymentTerminal {
  id: String!
  desc: String
  location: PaymentMediumRechargeLocation!
}

type PaymentMediumPaymentMediumRecharge {
  "timestamp"
  timestamp: BigInt!
}

```

```
"recharge value"
rechargeValue: Int!
"end user"
endUser: PaymentMediumRechargeEndUser!
"payment medium info"
paymentMedium: PaymentMediumRechargePaymentMedium!
"terminal"
terminal: PaymentMediumRechargePaymentTerminal!
}
enum PaymentMediumRechargeDocumentTypeEnum {
  CITENSHIP_CARD
  FOREIGNER
  PASSPORT
  IDENTITY_CARD
}

input PaymentMediumRechargeEndUserInput {
  id: String!
  name: String!
  documentType: PaymentMediumRechargeDocumentTypeEnum!
  documentId: String!
}

input PaymentMediumRechargePaymentMediumInput {
  typeId: String!
  dataPreState: JSON
  dataPostState: JSON
  externalSystemId: String!
}

input PaymentMediumRechargeLocationInput {
  type: String!
  coordinates: [Float!]
  timestamp: BigInt!
}

input PaymentMediumRechargePaymentTerminalInput {
  id: String!
  desc: String!
  location: PaymentMediumRechargeLocationInput!
}

input PaymentMediumPaymentMediumRechargeInput {
  "timestamp"
  timestamp: BigInt!
  "recharge value"
  rechargeValue: Int
  "end user"
  endUser: PaymentMediumRechargeEndUserInput!
  "payment medium info"
  paymentMedium: PaymentMediumRechargePaymentMediumInput!
  "terminal"
```

```

terminal: PaymentMediumRechargePaymentTerminalInput!
}

type Mutation {
  "Persist PaymentMedium info"
  PersistPaymentMediumRecharge(
    paymentMediumRechargeId: String!
    input: PaymentMediumPaymentMediumRechargeInput!
  ): PaymentMediumPaymentMediumRecharge @requireAuth
}

```

### Respuesta exitosa

```

{
  "externalSystemId": String - id único enviado en el request,
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "endUser": {
    "id": String - id unico de usuario final,
    "name": String - Nombre del Modo,
    "documentType": String- Tipo de documento del usuario final del sistema
    "documentId":String - Documento de identidad del usuario final
  },
  "metadata": {
    "createdBy": String - Creado por,
    "createdAt": Number - Fecha creación en milisegundos UTC,
    "updatedBy": String - Editado por,
    "updatedAt": Number - Fecha edición en milisegundos UTC
  },
  "paymentMedium": {
    "typeId": String - id único de tipo de medio de pago,
    "dataPreState": JSON - (opc) - Estado previo del medio de pago de los datos binarios internos del medio de pago,
    "dataPostState":JSON - (opc) - Estado posterior del medio de pago de los datos binarios internos del medio de
    pago,,
    "externalSystemId":String - id único enviado en el request del medio de pago,
  },
  "rechargeValue":Number - Valor de la recarga ,
  "terminal": {
    "id":String - id único del terminal usado para el pago,
    "desc":String - String - Nota descriptiva del terminal usado para el pago ,
    "location": {
      "type":String - tipo de localización enviada en la terminal,
      "coordinates": Array[Number,Number] - ubicación en coordenadas de la terminal usada para el pago,
      "timestamp": Number - Fecha de la ubicación en milisegundos UTC
    }
  }
}

```

```
},  
"timestamp": Number - Fecha de la terminal en milisegundos UTC  
"id":String - id único del pago en el transporte con medio de pago ,  
}
```

## 6. REPORTE OPERACIÓN CONTROL DE FLOTA

Reportes de operación se refiere a las acciones o eventos realizados por cada uno de los vehículos de los operadores de transporte.

**Los eventos para reportar se reducen a:**

- Ubicación del vehículo
- Ingreso o salida de pasajeros en un vehículo
- Apertura o cierre de puertas de un vehículo
- Inicio de servicio o recorrido de un vehículo
- Finalización de un servicio o recorrido de un vehículo

Los eventos pueden ser transmitidos de dos formas, en línea o en batch.

Transmisión en línea hace referencia a la transmisión del evento en el momento de la ocurrencia, mientras la transmisión en batch, se refiere a el envío de eventos como un conjunto el cual se procesa por lotes.

### 6.1 REPORTE EN LÍNEA

#### 6.1.1 Inicio de Servicio

- **REST**  
**URL:** <https://siitp.ambq.gov.co/api/external-system-gateway/rest/service-started>  
**Método:** POST  
**Parámetros :** N/A  
**Cuerpo:** JSON

```
{
  "serviceld": String - Identificador único del servicio o recorrido. Este id será usado en la petición de finalización ,
  "routeAuthorityCode": String - Código de la ruta dado por la autoridad
  "itineraryAuthorityCode": String - Código del itinerario dado por la autoridad,
  "vehiclePlate": String - Placa del vehículo",
  "driverDocumentId": String - Número de documento del conductor,,
  "scheduleDateTime": String - FechaHora de inicio del servicio/recorrido en milisegundos UTC,
  "Latitude": Float - Latitud actual del vehículo,
  "Longitude": Float - Altitud actual del vehículo,
  "in": Number - Número total de ingresos acumulados de pasajeros en el vehículo,
  "out": Number - Número total de salidas acumuladas de pasajeros en el vehículo,
}
```

JSON SCHEMA:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
```



```
},
"routeAuthorityId": {
  "type": "string"
},
"itineraryAuthorityId": {
  "type": "string"
},
"vehiclePlate": {
  "type": "string"
},
"driverDocumentId": {
  "type": "string"
},
"scheduleDateTime": {
  "type": "integer"
},
"latitude": {
  "type": "number"
},
"longitude": {
  "type": "number"
},
"in": {
  "type": "number"
},
"out": {
  "type": "number"
}
},
"required": [
  "id",
  "routeAuthorityId",
  "itineraryAuthorityId",
  "vehiclePlate",
  "driverDocumentId",
  "scheduleDateTime",
  "latitude",
  "longitude",
  "in",
  "out"
],
"additionalProperties": false
}
```

- **GraphQL**  
**Mutation:** ServiceStarted  
**Parámetros:**

```
{
  "serviceld": String - Identificador único del servicio o recorrido. Este id será usado en la petición de finalización ,
  "Input" : {
    "routeAuthorityCode": String - Código de la ruta dado por la autoridad
    "itineraryAuthorityCode": String - Código del itinerario dado por la autoridad,
    "vehiclePlate": String - Placa del vehículo",
    "driverDocumentId": String - Número de documento del conductor,,
    "scheduleDateTime": String - FechaHora de inicio del servicio/recorrido en milisegundos UTC,
    "Latitude": Float - Latitud actual del vehículo,
    "Longitude": Float - Altitud actual del vehículo,
    "in": Number - Número total de ingresos acumulados de pasajeros en el vehículo,
    "out": Number - Número total de salidas acumuladas de pasajeros en el vehículo,
  }
}
```

**GQL SCHEMA:**

```
### ServiceMngLiveExecutedService
type ServiceMngLiveExecutedService {
  "LiveExecutedService ID"
  id: ID
  organizationId: String
  state: String
  companyId: String
  itinerary: ServiceMngLiveExecutedServiceItinerary
  route: ServiceMngLiveExecutedServiceRoute
  driver: ServiceMngLiveExecutedServiceDriver
  vehicle: ServiceMngLiveExecutedServiceVehicle
  scheduleDateTime: BigInt
}

type ServiceMngLiveExecutedServiceLocation {
  type: String
  coordinates: [Float]
  timestamp: BigInt
}

type ServiceMngLiveExecutedServiceItinerary {
  id: String
  name: String
}

type ServiceMngLiveExecutedServiceDispatch {
  type: String
  userId: String
  userFullName: String
  userLogin: String
}

type ServiceMngLiveExecutedServiceVehicle {
```

```
manufacturer: String
model: String
organizationId: String
ownership: String
plate: String
id: String
internalNumber: String
}

type ServiceMngLiveExecutedServiceRoute {
  code: String
  name: String
  organizationId: String
  id: String
}

input ServiceMngLiveExecutedServiceCreateInput {
  "route id"
  routeAuthorityId: String!
  "itinerary id"
  itineraryAuthorityId: String!
  "vehicle id"
  vehiclePlate: String!
  "driver id"
  driverDocumentId: String!
  "schedule DateTime"
  scheduleDateTime: BigInt!
  "latitude"
  latitude: Float!
  "longitude"
  longitude: Float!
  "Passenger in"
  in: Int!
  "Passenger out"
  out: Int!
}

type Mutation {
  "Persist changes on liveExecutedService"
  ServiceStarted(
    serviceId: String!
    input: ServiceMngLiveExecutedServiceCreateInput!
  ): ServiceMngLiveExecutedService @requireAuth
}
```

### Respuesta exitosa

```
{
  "id": String - id único del servicio en el sistema SIITP,
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "scheduleDateTime": Number - FechaHora de inicio del servicio o recorrido en milisegundos UTC,,
  "companyId": String - Id de la compañía (operador de transporte) a la cual pertenece,
  "vehicle": {
    "internalNumber": String número interno del vehículo,
    "manufacturer": String - marca del vehículo,
    "model": String - Modelo del vehículo,
    "Ownership": String - tipo de tenencia,
    "plate": String - placa del vehículo,
    "location": {
      "type": "Point",
      "coordinates": [
        Number - longitud actual del vehículo,
        Number - Altitud actual del vehículo
      ],
      "timestamp": Number - FechaHora localización en milisegundos UTC,
    },
    "id":String - Id único del vehículo en el servidor SIITP,
  },
  "route": {
    "code": String - código de ruta según la autoridad,
    "name": String - Nombre de la ruta
    "id": String - Id de la ruta en el servidor SIITP
  },
  "itinerary": {
    "id": String - Id del itinerario en el servidor SIITP
    "name": String - Nombre del itinerario
  }
}
```

### Respuestas de errores:

- **Organización (operador de recaudo) no encontrado**

```
{
  "code": 10002
  "msg": "Organization does not exists, or is not active"
  "name": "OrganizationNotFound"
}
```

- **Ruta inactiva o inexistente**

```
{
  "code": 10003,
  "msg": "Route does not exists, or is not active",
}
```

```
"name": "RouteNotFound"
}
```

- **Itinerario Inactivo o inexistente**

```
{
  "code": 10004,
  "msg": "Itinerary does not exists, or is not active",
  "name": "ItineraryNotFound"
}
```

- **Vehículo inactivo o inexistente**

```
{
  "code": 10001,
  "msg": "Vehicle not found",
  "name": "VehicleNotFound"
}
```

- **Id del servicio ya existe (duplicado)**

```
{
  "code": 17050,
  "msg": "Service already exists",
  "name": "ServiceAlreadyExists"
}
```

## 6.1.2 Fin de Servicio

- **REST**

**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/service>

**Método:** DELETE

**Parámetros :** N/A

**Cuerpo:** JSON

```
{
  "id":String - id único del servicio
  "message":String - mensaje informativo que explica la razón de finalización del servicio
  "serviceEndTimestamp":Number - Fecha de finalización del servicio en milisegundos
  "in":String - Cantidad de contador de pasajeros entrada
  "out":String - Cantidad de contador de pasajeros salida
}
JSON SCHEMA:
{
  "$schema": "http://json-schema.org/draft-04/schema#",
```

```

"type": "object",
"properties": {
  "id": {
    "type": "string"
  },
  "message": {
    "type": "string"
  },
  "serviceEndTimestamp": {
    "type": "number"
  },
  "in": {
    "type": "number"
  },
  "out": {
    "type": "number"
  }
},
"required": [
  "id",
  "message",
  "serviceEndTimestamp",
  "in",
  "out"
],
"additionalProperties": false
}

```

- **GraphQL**  
**Mutation:** ServiceFinished  
**Parámetros:**

```

{
  "serviceId": String - Identificador único del servicio o recorrido.
  "input": {
    "message":String - mensaje informativo que explica la razón de finalización del servicio
    "serviceEndTimestamp":Number - Fecha de finalización del servicio en milisegundos
    "in":String - Cantidad de contador de pasajeros entrada
    "out":String - Cantidad de contador de pasajeros salida
  }
}

```

GQL SCHEMA:

```

input ServiceMngServiceFinishedInput {
  message: String
  serviceEndTimestamp: BigInt
  "Passenger in"
}

```

```
in: Int!  
"Passenger out"  
out: Int!  
}
```

### Respuesta exitosa

```
{  
  "code": Number - Si el código es 200 es porque la operación se finalizó exitosamente,  
  "message": String - Mensaje informativo de la respuesta  
}
```

### Respuestas de errores:

- **Servicio no encontrado**

```
{  
  "code": 17051  
  "msg": "LiveService not found"  
  "name": "LiveServiceNotFound"  
}
```

### 6.1.3 Reportar Localización

- **REST**  
**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/vehicle-report>  
**Método:** POST  
**Parámetros :** N/A  
**Cuerpo:** JSON

```
{  
  "vehiclePlate": String - placa del vehículo  
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,  
  "companyId": String - ID compañía (Operador de transporte) a la que pertenece,  
  "evts": [  
    "type": String - Indica el tipo de mensaje que se está enviando (en este caso sería "LOC")  
    "ts": Number - Fecha en milisegundos del reporte de localización  
    "lat": Number - latitud de las coordenadas  
    "lng": Number - longitud de las coordenadas  
    "brg": Number - dirección reportada por el gps (bearing)  ]  
}
```

```

    "spd": Number - velocidad del vehículo
  ]
}
JSON SCHEMA:
{
  "type": "object",
  "properties": {
    "vehiclePlate": {
      "type": "string"
    },
    "organizationId": {
      "type": "string"
    },
    "companyId": {
      "type": "string"
    },
    "evts": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "anyOf": [
            {
              "properties": {
                "type": {
                  "type": "string"
                },
                "ts": {
                  "type": "integer"
                },
                "open": {
                  "type": "boolean"
                },
                "front": {
                  "type": "boolean"
                },
                "in": {
                  "type": "integer"
                },
                "out": {
                  "type": "integer"
                },
                "lat": {
                  "type": "number"
                },
                "lng": {
                  "type": "number"
                },
                "spd": {
                  "type": "integer"
                }
              }
            }
          ],
          "required": [

```



```
"type",
"ts",
"in",
"out",
"open",
"front",
"lat",
"lng",
"spd"
]
},
{
  "properties": {
    "type": {
      "type": "string"
    },
    "ts": {
      "type": "integer"
    },
    "open": {
      "type": "boolean"
    },
    "front": {
      "type": "boolean"
    },
    "lat": {
      "type": "number"
    },
    "lng": {
      "type": "number"
    },
    "brg": {
      "type": "integer"
    },
    "spd": {
      "type": "integer"
    }
  },
  "required": [
    "type",
    "ts",
    "open",
    "front",
    "lat",
    "lng",
    "spd"
  ]
},
{
  "properties": {
    "type": {
      "type": "string"
    }
  }
},
}
```

```

    "ts": {
      "type": "integer"
    },
    "lat": {
      "type": "number"
    },
    "lng": {
      "type": "number"
    },
    "brg": {
      "type": "integer"
    },
    "spd": {
      "type": "integer"
    }
  },
  "required": [
    "type",
    "ts",
    "lat",
    "lng",
    "brg",
    "spd"
  ]
}
]
}
]
}
},
"required": [
  "vehiclePlate",
  "evts"
],
"additionalProperties": false
}

```

- **GraphQL**  
**Mutation:** VehicleEvent  
**Parámetros:**

```

{
  "vehiclePlate": String - placa del vehículo
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "companyId": String - ID compañía (Operador de transporte) a la que pertenece,
  "evts": [
    "type": String - Indica el tipo de mensaje que se está enviando (en este caso sería "LOC")
    "ts": Number - Fecha en milisegundos del reporte de localización
    "lat": Number - latitud de las coordenadas
  ]
}

```

```
    "lng": Number - longitud de las coordenadas
    "brg": Number - dirección reportada por el gps (bearing)
    "spd": Number - velocidad del vehículo

  ]
}
}
GQL SCHEMA:
input VehicleReportInput {
  "license plate"
  vehiclePlate: String
  "organization id"
  organizationId: String
  "company id"
  companyId: String
  "evts"
  evts: [JSON]
}
```

### Respuesta exitosa

```
{
  "code": Number - Si el código es 200 es porque la operación se finalizó exitosamente,
  "message": String - Mensaje informativo de la respuesta
}
```

### Respuestas de errores:

- **Vehículo no encontrado**

```
{
  "code": 10001
  "msg": "Vehicle not found",
  "name": "VehicleNotFound"
}
```

#### 6.1.4 Apertura/Cierre de Puertas

- **REST**  
**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/vehicle-report>  
**Método:** POST  
**Parámetros :** N/A  
**Cuerpo:** JSON

```
{
  "vehiclePlate": String - placa del vehículo
}
```

```

"organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
"companyId": String - ID compañía (Operador de transporte) a la que pertenece,
"evts": [
  "type": String - Indica el tipo de mensaje que se está enviando (en este caso seria "DOOR")
  "ts": Number - Fecha en milisegundos del reporte de apertura y cierre de puertas
  "open": Boolean - bandera que identifica si las puertas están abiertas o no
  "front": Boolean - bandera que identifica si la puerta reportada es la frontal o no
  "lat": Number - latitud de las coordenadas
  "lng": Number - longitud de las coordenadas
  "brg": Number - dirección reportada por el gps (bearing)
  "spd": Number - velocidad del vehículo
]
}

JSON SCHEMA:
{
  "type": "object",
  "properties": {
    "vehiclePlate": {
      "type": "string"
    },
    "organizationId": {
      "type": "string"
    },
    "companyId": {
      "type": "string"
    },
    "evts": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "anyOf": [
            {
              "properties": {
                "type": {
                  "type": "string"
                },
                "ts": {
                  "type": "integer"
                },
                "open": {
                  "type": "boolean"
                },
                "front": {
                  "type": "boolean"
                },
                "in": {
                  "type": "integer"
                },
                "out": {
                  "type": "integer"
                }
              }
            }
          ]
        }
      ]
    }
  }
}

```

```
    },  
    "lat": {  
      "type": "number"  
    },  
    "lng": {  
      "type": "number"  
    },  
    "spd": {  
      "type": "integer"  
    }  
  },  
  "required": [  
    "type",  
    "ts",  
    "in",  
    "out",  
    "open",  
    "front",  
    "lat",  
    "lng",  
    "spd"  
  ]  
},  
{  
  "properties": {  
    "type": {  
      "type": "string"  
    },  
    "ts": {  
      "type": "integer"  
    },  
    "open": {  
      "type": "boolean"  
    },  
    "front": {  
      "type": "boolean"  
    },  
    "lat": {  
      "type": "number"  
    },  
    "lng": {  
      "type": "number"  
    },  
    "brg": {  
      "type": "integer"  
    },  
    "spd": {  
      "type": "integer"  
    }  
  },  
  "required": [  
    "type",  
    "ts",
```

```
        "open",
        "front",
        "lat",
        "lng",
        "spd"
    ]
},
{
    "properties": {
        "type": {
            "type": "string"
        },
        "ts": {
            "type": "integer"
        },
        "lat": {
            "type": "number"
        },
        "lng": {
            "type": "number"
        },
        "brg": {
            "type": "integer"
        },
        "spd": {
            "type": "integer"
        }
    },
    "required": [
        "type",
        "ts",
        "lat",
        "lng",
        "brg",
        "spd"
    ]
}
]
}
},
"required": [
    "vehiclePlate",
    "evts"
],
"additionalProperties": false
}
```

- **GraphQL**  
**Mutation:** VehicleEvent  
**Parámetros:**

```
{
  "vehiclePlate": String - placa del vehículo
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "companyId": String - ID compañía (Operador de transporte) a la que pertenece,
  "evts": [
    "type": String - Indica el tipo de mensaje que se está enviando (en este caso sería "DOOR")
    "ts": Number - Fecha en milisegundos del reporte de apertura y cierre de puertas
    "open": Boolean - bandera que identifica si las puertas están abiertas o no
    "front": Boolean - bandera que identifica si la puerta reportada es la frontal o no
    "lat": Number - latitud de las coordenadas
    "lng": Number - longitud de las coordenadas
    "brg": Number - dirección reportada por el gps (bearing)
    "spd": Number - velocidad del vehículo
  ]
}
GQL SCHEMA:
input VehicleReportInput {
  "license plate"
  vehiclePlate: String
  "organization id"
  organizationId: String
  "company id"
  companyId: String
  "evts"
  evts: [JSON]
}
```

### Respuesta exitosa

```
{
  "code": Number - Si el código es 200 es porque la operación se finalizó exitosamente,
  "message": String - Mensaje informativo de la respuesta
}
```

### Respuestas de errores:

- **Vehículo no encontrado**

```
{
  "code": 10001
  "msg": "Vehicle not found",
  "name": "VehicleNotFound"
}
```

```
}

```

### 6.1.5 Reportar Ingreso/Salida de Pasajeros

- **REST**

**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/vehicle-report>

**Método:** POST

**Parámetros :** N/A

**Cuerpo:** JSON

```
{
  "vehiclePlate": String - placa del vehículo
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "companyId": String - ID compañía (Operador de transporte) a la que pertenece,
  "evts": [
    "type": String - Indica el tipo de mensaje que se está enviando (en este caso seria
    "PAX")
    "ts": Number - Fecha en milisegundos del reporte de pasajeros
    "in": Number - cantidad de pasajeros que han ingresado
    "out": Number - cantidad de pasajeros que han salido
    "front Boolean - bandera que identifica si la puerta reportada es la frontal o no
    "lat": Number - latitud de las coordenadas
    "lng": Number - longitud de las coordenadas
    "brg": Number - dirección reportada por el gps (bearing)
    "spd": Number - velocidad del vehículo
  ]
}
JSON SCHEMA:
{
  "type": "object",
  "properties": {
    "vehiclePlate": {
      "type": "string"
    },
    "organizationId": {
      "type": "string"
    },
    "companyId": {
      "type": "string"
    },
    "evts": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "anyOf": [
            {
              "properties": {
                "type": {
                  "type": "string"
                }
              }
            }
          ]
        }
      ]
    }
  }
}
```



```
"ts": {
  "type": "integer"
},
"open": {
  "type": "boolean"
},
"front": {
  "type": "boolean"
},
"in": {
  "type": "integer"
},
"out": {
  "type": "integer"
},
"lat": {
  "type": "number"
},
"lng": {
  "type": "number"
},
"spd": {
  "type": "integer"
}
},
"required": [
  "type",
  "ts",
  "in",
  "out",
  "open",
  "front",
  "lat",
  "lng",
  "spd"
]
},
{
  "properties": {
    "type": {
      "type": "string"
    },
    "ts": {
      "type": "integer"
    },
    "open": {
      "type": "boolean"
    },
    "front": {
      "type": "boolean"
    },
    "lat": {
      "type": "number"
    }
  }
}
```

```
    },
    "lng": {
      "type": "number"
    },
    "brg": {
      "type": "integer"
    },
    "spd": {
      "type": "integer"
    }
  },
  "required": [
    "type",
    "ts",
    "open",
    "front",
    "lat",
    "lng",
    "spd"
  ]
},
{
  "properties": {
    "type": {
      "type": "string"
    },
    "ts": {
      "type": "integer"
    },
    "lat": {
      "type": "number"
    },
    "lng": {
      "type": "number"
    },
    "brg": {
      "type": "integer"
    },
    "spd": {
      "type": "integer"
    }
  },
  "required": [
    "type",
    "ts",
    "lat",
    "lng",
    "brg",
    "spd"
  ]
}
]
```

```

    ]
  }
},
"required": [
  "vehiclePlate",
  "evts"
],
"additionalProperties": false
}

```

- **GraphQL**  
**Mutation:** VehicleEvent  
**Parámetros:**

```

{
  "vehiclePlate": String - placa del vehículo
  "organizationId": String - Id de la organización (operador de recaudo) a la cual pertenece,
  "companyId": String - ID compañía (Operador de transporte) a la que pertenece,
  "evts": [
    "type": String - Indica el tipo de mensaje que se está enviando (en este caso seria "PAX")
    "ts": Number - Fecha en milisegundos del reporte de pasajeros
    "in": Number - cantidad de pasajeros que han ingresado
    "out": Number - cantidad de pasajeros que han salido
    "front Boolean - bandera que identifica si la puerta reportada es la frontal o no
    "lat": Number - latitud de las coordenadas
    "lng": Number - longitud de las coordenadas
    "brg": Number - dirección reportada por el gps (bearing)
    "spd": Number - velocidad del vehículo
  ]
}

```

GQL SCHEMA:

```

input VehicleReportInput {
  "license plate"
  vehiclePlate: String
  "organization id"
  organizationId: String
  "company id"
  companyId: String
  "evts"
  evts: [JSON]
}

```

### Respuesta exitosa

```

{
  "code": Number - Si el código es 200 es porque la operación se finalizó exitosamente,

```

```
  "message": String - Mensaje informativo de la respuesta  
}
```

#### Respuestas de errores:

- Vehículo no encontrado

```
{  
  "code": 10001  
  "msg": "Vehicle not found",  
  "name": "VehicleNotFound"  
}
```

## 6.2 REPORTE EN BATCH

El vehículo recopila todos los eventos dentro de un rango de tiempo para luego ser enviados como un conjunto al servidor, el cual procesa en lotes los eventos.

El procesamiento en batch soporta los siguientes eventos:

- Ubicación de vehículo
- Ingreso o salida de pasajeros en un vehículo
- Apertura o cierre de puertas de un vehículo
- Inicio de servicio o recorrido de un vehículo
- Finalización de un servicio o recorrido de un vehículo

### 6.2.1 Reporte Batch

- **REST**  
**URL:** <http://siitp.ambq.gov.co/api/external-system-gateway/rest/batch-event>  
**Método:** POST  
**Parámetros:** N/A  
**Cuerpo:** JSON

```
{  
  "vehiclePlate": String - placa del vehículo  
  "evts": [  
    "Type": String - identifica el tipo de mensaje que se está enviando (Valores válidos "LOC | DOOR |  
    PAX | SERVICE_STARTED | SERVICE_FINISHED")  
    ...evt  
  ]  
}
```

JSON SCHEMA:

```
{
  "type": "object",
  "properties": {
    "vehiclePlate": {
      "type": "string"
    },
    "evts": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "anyOf": [
            {
              "properties": {
                "type": {
                  "type": "string",
                  "enum": ["PAX"]
                },
                "ts": {
                  "type": "integer"
                },
                "open": {
                  "type": "boolean"
                },
                "front": {
                  "type": "boolean"
                },
                "in": {
                  "type": "integer"
                },
                "out": {
                  "type": "integer"
                },
                "lat": {
                  "type": "number"
                },
                "lng": {
                  "type": "number"
                },
                "brg": {
                  "type": "integer"
                },
                "spd": {
                  "type": "integer"
                }
              }
            },
            {
              "required": [
                "type",
                "ts",
                "open",
                "front",
                "in",

```

```
        "out",
        "lat",
        "lng",
        "brg",
        "spd"
    ],
    "additionalProperties": false
  },
  {
    "properties": {
      "type": {
        "type": "string",
        "enum": ["DOOR"]
      },
      "ts": {
        "type": "integer"
      },
      "open": {
        "type": "boolean"
      },
      "front": {
        "type": "boolean"
      },
      "lat": {
        "type": "number"
      },
      "lng": {
        "type": "number"
      },
      "brg": {
        "type": "integer"
      },
      "spd": {
        "type": "integer"
      }
    },
    "required": [
      "type",
      "ts",
      "open",
      "front",
      "lat",
      "lng",
      "brg",
      "spd"
    ],
    "additionalProperties": false
  },
  {
    "properties": {
      "type": {
```

```
        "type": "string",
        "enum": ["LOC"]
    },
    "ts": {
        "type": "integer"
    },
    "lat": {
        "type": "number"
    },
    "lng": {
        "type": "number"
    },
    "brg": {
        "type": "integer"
    },
    "spd": {
        "type": "integer"
    }
},
"required": [
    "type",
    "ts",
    "lat",
    "lng",
    "brg",
    "spd"
],
"additionalProperties": false
},
{
    "properties": {
        "type": {
            "type": "string",
            "enum": ["SERVICE_STARTED"]
        },
        "id": {
            "type": "string"
        },
        "routeAuthorityId": {
            "type": "string"
        },
        "itineraryAuthorityId": {
            "type": "string"
        },
        "vehiclePlate": {
            "type": "string"
        },
        "driverDocumentId": {
            "type": "string"
        },
        "scheduleDateTime": {
            "type": "integer"
        }
    }
},
```

```
"latitude": {
  "type": "number"
},
"longitude": {
  "type": "number"
},
"in": {
  "type": "number"
},
"out": {
  "type": "number"
}
},
"required": [
  "type",
  "id",
  "routeAuthorityId",
  "itineraryAuthorityId",
  "vehiclePlate",
  "driverDocumentId",
  "scheduleDateTime",
  "latitude",
  "longitude",
  "in",
  "out"
],
"additionalProperties": false
},
{
  "properties": {
    "type": {
      "type": "string",
      "enum": ["SERVICE_FINISHED"]
    },
    "id": {
      "type": "string"
    },
    "message": {
      "type": "string"
    },
    "serviceEndTimestamp": {
      "type": "number"
    },
    "in": {
      "type": "number"
    },
    "out": {
      "type": "number"
    }
  },
  "required": [
    "type",
    "id",
```



```

        "message",
        "serviceEndTimestamp",
        "in",
        "out"
      ],
      "additionalProperties": false
    }
  ]
}
},
"required": [
  "vehiclePlate",
  "evts"
],
"additionalProperties": false
}

```

- **GraphQL**  
**Mutation:** BatchEvents  
**Parámetros:**

```

{
  "vehiclePlate": String - placa del vehículo
  "evts": [
    "Type": String - identifica el tipo de mensaje que se está enviando (Valores validos "LOC | DOOR | PAX | SERVICE_STARTED | SERVICE_FINISHED")
    ...evt
  ]
}

```

GQL SCHEMA:

```

input BatchEventInput {
  vehiclePlate: String
  evts:[JSON]
}

```

**Respuesta exitosa**

```

{
  "ack": Boolean - Bandera que identifica si el servidor recibió o no correctamente la información
  "errCode": Number - Código del error generado
  "errDesc": String - Información del mensaje generado
}

```

```
"batchProcessId": String - Id generado en el momento de procesar el batch, con este id se puede consultar el estado del batch  
}
```

#### Respuestas de errores:

- Límite de número de eventos excedido

```
{  
  "code": 10010  
  "msg": "List limit exceeded",  
  "name": "ListLimitExceeded"  
}
```

#### 6.2.2 Consulta Procesamiento Batch

- **REST**  
**URL:** <http://localhost:3001/api/external-system-gateway/rest/batch-process>  
**Método:** GET  
**Parámetros :**  
**id:** Identificador del procesamiento en batch  
**Cuerpo:** N/A
- **GraphQL**  
**Query:** BatchEvent  
**Parámetros:**

```
{  
  "id": String - Identificador del procesamiento en batch  
}
```

#### Respuesta exitosa

```
{  
  "id": String - Identificador del procesamiento en batch  
  "vehiclePlate": String - Placa del vehículo al que le pertenecen los eventos en el batch  
  "status": String - Estado actual del procesamiento del batch (Posibles estados "PROCESSED_OK |  
  PROCESSED_PARTIAL | PROCESSED_WITH_ERRORS")  
  "evts": [  
    Evento original reportado en el batch adicionando las siguientes variables
```

```
    "state": String - identifica el estado actual del evento (Posibles estados "QUEUED |  
    PROCESSED_OK | PROCESSED_PARTIAL | PROCESSED_WITH_ERRORS")  
    "  
    "syncTries": Number - Cantidad de intentos realizados por el servidor para procesar el mensaje  
    "lastSyncTimestamp": Number - Fecha del último intento de procesamiento del mensaje  
  ]  
}
```