

# Estructura de la API y la Base de Datos

## Contents

<b>Introducción</b>	<b>2</b>
<b>Tablas de la Base de Datos</b>	<b>2</b>
source . . . . .	2
state . . . . .	3
city . . . . .	3
sector . . . . .	3
street . . . . .	4
tribunal . . . . .	4
alert_type . . . . .	4
alert_subtype . . . . .	4
alert_direction . . . . .	5
alert_cause . . . . .	5
route_name . . . . .	5
route_type . . . . .	5
waze_route . . . . .	6
route_point . . . . .	6
alert . . . . .	6
injury_level . . . . .	7
injury_place . . . . .	7
injury . . . . .	7
alert_street . . . . .	8
tracking_folder . . . . .	8
route . . . . .	8
vehicle . . . . .	9
direction . . . . .	9
speed . . . . .	9
brand . . . . .	10
model . . . . .	10

link_type . . . . .	10
provider . . . . .	10
camera . . . . .	11
<b>Endpoints de la API</b>	<b>11</b>
GET /api/data/route . . . . .	11
GET /api/data/route/:id . . . . .	11
GET /api/data/alert . . . . .	12
GET /api/data/alert/:id . . . . .	12
GET /api/data/speed . . . . .	13
GET /api/data/speed/:id . . . . .	13
GET /api/data/camera . . . . .	13
GET /api/data/camera/:id . . . . .	14
<b>Anexos</b>	<b>14</b>
Modelo Entidad Relación . . . . .	14
Estructura de la API . . . . .	19
Tipos de Typescript . . . . .	19
Visualización de tipos . . . . .	23
<b>Notas Finales</b>	<b>28</b>

## Introducción

Este documento describe la estructura de la base de datos y los endpoints de la API desarrollada para la gestión y consulta de datos de movilidad urbana, incluyendo accidentes, alertas de tráfico, mediciones de velocidad, rutas y cámaras de vigilancia.

## Tablas de la Base de Datos

A continuación se presentan las tablas que componen la base de datos, indicando sus fuentes de datos y las columnas que las integran.

El Modelo Entidad Relación se encuentra en los anexos.

### source

#### Fuente(s):

- Siniestros buses interurbanos(1).xlsx
- GH026929.xlsx
- DTPR

- Data accidentes de carabineros.xlsx
- Incidentes de tráfico radio.xlsx
- Med velo CHIGUAYANTE.xlsx
- Med velo LA VEGA.xlsx
- Alertas de Tráfico.csv
- Copia de Accidentes.csv
- Waze for Cities Data Key Alerts Dashboard\_Traffic Irregularities\_Tabla(1).csv
- Red de Waze

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(255)

**state**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx
- Data accidentes de carabineros.xlsx
- Inventario CCTV Biobío(1).xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(20)

**city**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx
- Data accidentes de carabineros.xlsx
- Inventario CCTV Biobío(1).xlsx
- Alertas de Tráfico.csv
- Copia de Accidentes.csv

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(20)
- state\_id: INTEGER, FOREIGN KEY (id - state)

**sector**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx
- Data accidentes de carabineros.xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(20)

## **street**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx
- Data accidentes de carabineros.xlsx
- Alertas de Tráfico.csv
- Copia de Accidentes.csv
- Waze for Cities Data Key Alerts Dashboard\_Traffic Irregularities\_Tabla(1).csv
- Red de Waze

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(255)
- city\_id: INTEGER, FOREIGN KEY (id - city)

## **tribunal**

**Fuente(s):**

- Data accidentes de carabineros.xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(255)

## **alert\_type**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx
- Data accidentes de carabineros.xlsx
- Incidentes de tráfico radio.xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(255)

## **alert\_subtype**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx
- Data accidentes de carabineros.xlsx
- Incidentes de tráfico radio.xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(255)
- type\_id: INTEGER, FOREIGN KEY (id - alert\_type)

## **alert\_direction**

**Fuente(s):**

- Incidentes de tráfico radio.xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(10)

## **alert\_cause**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx
- Data accidentes de carabineros.xlsx
- Waze for Cities Data Key Alerts Dashboard\_Traffic Irregularities\_Tabla(1).csv

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(255)

## **route\_name**

**Fuente(s):**

- Red de Waze

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(255)

## **route\_type**

**Fuente(s):**

- Red de Waze

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(10)

## waze\_route

Fuente(s):

- Red de Waze

Columnas:

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- src\_id: INTEGER, FOREIGN KEY (id - source)
- date: DATETIME
- historic\_time: FLOAT
- from\_id: INTEGER, FOREIGN KEY (id - street)
- name\_id: INTEGER, FOREIGN KEY (id - route\_name)
- to\_id: INTEGER, FOREIGN KEY (id - street)
- length: FLOAT
- time: FLOAT
- jam\_level: INTEGER
- type\_id: INTEGER, FOREIGN KEY (id - route\_type)

## route\_point

Fuente(s):

- Red de Waze

Columnas:

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- route\_id: INTEGER, FOREIGN KEY (id - waze\_route)
- latitude: FLOAT
- longitude: FLOAT
- index: INTEGER

## alert

Fuente(s):

- Siniestros buses interurbanos(1).xlsx
- Data accidentes de carabineros.xlsx
- Incidentes de tráfico radio.xlsx
- Alertas de Tráfico.csv
- Copia de Accidentes.csv
- Waze for Cities Data Key Alerts Dashboard\_Traffic Irregularities\_Tabla(1).csv
- Red de Waze

Columnas:

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- src\_id: INTEGER, FOREIGN KEY (id - source)

- date: DATETIME
- city\_id: INTEGER, FOREIGN KEY (id - city)
- subtype\_id: INTEGER, FOREIGN KEY (id - alert\_subtype)
- cause\_id: INTEGER, FOREIGN KEY (id - alert\_cause)
- km: FLOAT
- fine: FLOAT
- sector\_id: INTEGER, FOREIGN KEY (id - sector)
- tribunal\_id: INTEGER, FOREIGN KEY (id - tribunal)
- complete\_report: BOOLEAN
- traffic\_incident: BOOLEAN
- latitude: FLOAT
- longitude: FLOAT
- reliability: FLOAT
- length: FLOAT
- delay: FLOAT
- dir\_id: INTEGER, FOREIGN KEY (id - alert\_direction)
- waze\_route\_id: INTEGER, FOREIGN KEY (id - waze\_route)

## **injury\_level**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- level: VARCHAR(15)

## **injury\_place**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- place: VARCHAR(5)

## **injury**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx
- Waze for Cities Data Key Alerts Dashboard\_Traffic Irregularities\_Tabla(1).csv

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT

- quantity: INTEGER
- alert\_id: INTEGER, FOREIGN KEY (id - alert)
- level\_id: INTEGER, FOREIGN KEY (id - injury\_level)
- place\_id: INTEGER, FOREIGN KEY (id - injury\_place)

## **alert\_street**

**Fuente(s):**

- Siniestros buses interurbanos(1).xlsx
- Data accidentes de carabineros.xlsx
- Alertas de Tráfico.csv
- Copia de Accidentes.csv
- Waze for Cities Data Key Alerts Dashboard\_Traffic Irregularities\_Tabla(1).csv
- Red de Waze

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- alert\_id: INTEGER, FOREIGN KEY (id - alert)
- street\_id: INTEGER, FOREIGN KEY (id - street)

## **tracking\_folder**

**Fuente(s):**

- DTPR

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(10)

## **route**

**Fuente(s):**

- DTPR

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(5)
- folder\_id: INTEGER, FOREIGN KEY (id - tracking\_folder)

## **vehicle**

**Fuente(s):**

- DTPR
- Med velo CHIGUAYANTE.xlsx
- Med velo LA VEGA.xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- plate: VARCHAR(6)
- name: VARCHAR(20)

## **direction**

**Fuente(s):**

- Med velo CHIGUAYANTE.xlsx
- Med velo LA VEGA.xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(20)

## **speed**

**Fuente(s):**

- GH026929.xlsx
- DTPR
- Med velo CHIGUAYANTE.xlsx
- Med velo LA VEGA.xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- src\_id: INTEGER, FOREIGN KEY (id - source)
- date: DATETIME
- latitude: FLOAT
- longitude: FLOAT
- speed: FLOAT
- fix: FLOAT
- precision: FLOAT
- city\_id: INTEGER, FOREIGN KEY (id - city)
- route\_id: INTEGER, FOREIGN KEY (id - route)
- vehicle\_id: INTEGER, FOREIGN KEY (id - vehicle)
- direction: INTEGER
- dop: FLOAT
- comment: VARCHAR(255)
- dir\_id: INTEGER, FOREIGN KEY (id - direction)

## **brand**

**Fuente(s):**

- Inventario CCTV Biobío(1).xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(20)

## **model**

**Fuente(s):**

- Inventario CCTV Biobío(1).xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(25)
- brand\_id: INTEGER, FOREIGN KEY (id - brand)

## **link\_type**

**Fuente(s):**

- Inventario CCTV Biobío(1).xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(10)

## **provider**

**Fuente(s):**

- Inventario CCTV Biobío(1).xlsx

**Columnas:**

- id: INTEGER, PRIMARY KEY, AUTOINCREMENT
- name: VARCHAR(30)

## **camera**

**Fuente(s):**

- Inventario CCTV Biobío(1).xlsx

**Columnas:**

- `id`: INTEGER, PRIMARY KEY, AUTOINCREMENT
- `src_id`: INTEGER, FOREIGN KEY (`id` - source)
- `name`: VARCHAR(50)
- `city_id`: INTEGER, FOREIGN KEY (`id` - city)
- `integration`: DATETIME
- `online`: BOOLEAN
- `link_id`: INTEGER, FOREIGN KEY (`id` - link\_type)
- `provider_id`: INTEGER, FOREIGN KEY (`id` - provider)
- `camera_id`: VARCHAR(20)
- `encoder_id`: VARCHAR(20)
- `model_id`: INTEGER, FOREIGN KEY (`id` - model)
- `encoder_model_id`: INTEGER, FOREIGN KEY (`id` - model)
- `latitude`: FLOAT
- `longitude`: FLOAT

## **Endpoints de la API**

La API está organizada en cuatro recursos principales: rutas, alertas, velocidades y cámaras. Cada recurso soporta consultas con filtros avanzados y devuelve datos estructurados en formato JSON.

Anexos

### **GET /api/data/route**

**Descripción:** Obtiene un conjunto de rutas de Waze.

**Parámetros de consulta:**

- `token` (requerido)
- `src` (fuente)
- `min_date`, `max_date` (rango temporal)
- `min_lat`, `max_lat`, `min_lng`, `max_lng` (rango geográfico)
- `disc_spc`, `disc_temp` (discretización espacial y temporal en metros y segundos)

**Respuesta:** Objeto WazeRouteGroup.

### **GET /api/data/route/:id**

**Descripción:** Obtiene una ruta específica por ID.

**Parámetros de consulta:**

- `token` (requerido)

**Parámetros de URL:**

- **id**: ID de la ruta de waze

**Respuesta:** Objeto WazeRoute individual.

**GET /api/data/alert**

**Descripción:** Consulta alertas y accidentes con múltiples filtros.

**Parámetros de consulta:**

- **token** (requerido)
- **src** (fuente)
- **min\_date, max\_date** (rango temporal)
- **min\_lat, max\_lat, min\_lng, max\_lng** (rango geográfico)
- **disc\_spc, disc\_temp** (discretización espacial y temporal en metros y segundos)
- **state, city** (comuna y región)
- **type, subtype** (clasificación de alerta/accidente)
- **min\_km, max\_km** (rango de kilómetros)
- **min\_fine, max\_fine** (rango de partes)
- **sector** (sector)
- **tribunal** (tribunal)
- **complete\_report** (0: no es un reporte completo; 1: es un reporte completo)
- **traffic\_incident** (0: no es accidente de tráfico; 1: es accidente de tráfico)
- **min\_rel, max\_rel** (rango de confiabilidades)
- **min\_len, max\_len** (rango de largos)
- **min\_del, max\_del** (rango de desfases)
- **dir** (dirección: N-S, S-N, etc)

**Respuesta:** Objeto AlertGroup.

**GET /api/data/alert/:id**

**Descripción:** Obtiene una alerta específica por ID.

**Parámetros de consulta:**

- **token** (requerido)

**Parámetros de URL:**

- **id**: ID de la alerta o accidente

**Respuesta:** Objeto Alert individual.

## **GET /api/data/speed**

**Descripción:** Consulta medidas de velocidad con múltiples filtros.

**Parámetros de consulta:**

- `token` (requerido)
- `src` (fuente)
- `min_date, max_date` (rango temporal)
- `min_lat, max_lat, min_lng, max_lng` (rango geográfico)
- `disc_spc, disc_temp` (discretización espacial y temporal en metros y segundos)
- `min_spd, max_spd` (rango de velocidades)
- `min_fix, max_fix` (rango de fixes)
- `min_pres, max_pres` (rango de presiones)
- `state, city` (comuna y región)
- `folder, route` (carpeta y recorrido)
- `vh_type, plate` (tipo y patente del vehículo)
- `direction` (dirección: -1, 0 o 1)
- `min_dop, max_dop`: (rango de presiones geométricas)

**Respuesta:** Objeto SpeedGroup.

## **GET /api/data/speed/:id**

**Descripción:** Obtiene una medida específica por ID.

**Parámetros de consulta:**

- `token` (requerido)

**Parámetros de URL:**

- `id`: ID de la medida

**Respuesta:** Objeto Speed individual.

## **GET /api/data/camera**

**Descripción:** Consulta cámaras con múltiples filtros.

**Parámetros de consulta:**

- `token` (requerido)
- `src` (fuente)
- `min_int, max_int` (fechas de integración)
- `min_lat, max_lat, min_lng, max_lng` (rango geográfico)
- `state, city` (comuna y región)
- `online` (0: la cámara no está online o 1: la cámara está online)
- `link` (tipo de enlace)
- `prov` (proveedor)
- `brand, model, enc_brand, enc_model` (marca y modelo de la cámara y el codificador de video)

**Respuesta:** Objeto CameraGroup.

**GET /api/data/camera/:id**

**Descripción:** Obtiene una cámara específica por ID.

**Parámetros de consulta:**

- token (requerido)

**Parámetros de URL:**

- id: ID de la cámara

**Respuesta:** Objeto Camera individual.

## Anexos

### Modelo Entidad Relación

El diagrama completo del modelo de base de datos está disponible en formato Mermaid:

Ver modelo ER

```
erDiagram
    state {
        INTEGER id "PK AI"
        VARCHAR(20) name
    }

    city {
        INTEGER id "PK AI"
        VARCHAR(20) name
        INTEGER state_id "FK"
    }

    sector {
        INTEGER id "PK AI"
        VARCHAR(20) name
    }

    street {
        INTEGER id "PK AI"
        VARCHAR(255) name
        INTEGER city_id "FK"
    }

    tribunal {
        INTEGER id "PK AI"
        VARCHAR(255) name
    }

    alert_type {
        INTEGER id "PK AI"
    }
```

```

        VARCHAR(255) name
    }

alert_subtype {
    INTEGER id "PK AI"
    VARCHAR(255) name
    INTEGER type_id "FK"
}

alert_direction {
    INTEGER id "PK AI"
    VARCHAR(10) name
}

alert_cause {
    INTEGER id "PK AI"
    VARCHAR(255) name
}

route_name {
    INTEGER id "PK AI"
    VARCHAR(255) name
}

route_type {
    INTEGER id "PK AI"
    VARCHAR(10) name
}

waze_route {
    INTEGER id "PK AI"
    INTEGER src_id "FK"
    DATETIME date
    FLOAT historic_time
    INTEGER from_id "FK"
    INTEGER name_id "FK"
    INTEGER to_id "FK"
    FLOAT length
    FLOAT time
    INTEGER jam_level
    INTEGER type_id "FK"
}

route_point {
    INTEGER id "PK AI"
    INTEGER route_id "FK"
    FLOAT latitude
    FLOAT longitude
    INTEGER index
}

alert {
    INTEGER id "PK AI"

```

```

    INTEGER src_id "FK"
    DATETIME date
    INTEGER city_id "FK"
    INTEGER subtype_id "FK"
    INTEGER cause_id "FK"
    FLOAT km
    FLOAT fine
    INTEGER sector_id "FK"
    INTEGER tribunal_id "FK"
    BOOLEAN complete_report
    BOOLEAN traffic_incident
    FLOAT latitude
    FLOAT longitude
    FLOAT reliability
    FLOAT length
    FLOAT delay
    INTEGER dir_id "FK"
    INTEGER waze_route_id "FK"
}

injury_level {
    INTEGER id "PK AI"
    VARCHAR(15) level
}

injury_place {
    INTEGER id "PK AI"
    VARCHAR(5) place
}

injury {
    INTEGER id "PK AI"
    INTEGER quantity
    INTEGER alert_id "FK"
    INTEGER level_id "FK"
    INTEGER place_id "FK"
}

alert_street {
    INTEGER id "PK AI"
    INTEGER alert_id "FK"
    INTEGER street_id "FK"
}

tracking_folder {
    INTEGER id "PK AI"
    VARCHAR(10) name
}

route {
    INTEGER id "PK AI"
    VARCHAR(5) name
    INTEGER folder_id "FK"
}

```

```

}

vehicle {
    INTEGER id "PK AI"
    VARCHAR(6) plate
    VARCHAR(20) name
}

direction {
    INTEGER id "PK AI"
    VARCHAR(20) name
}

speed {
    INTEGER id "PK AI"
    INTEGER src_id "FK"
    DATETIME date
    FLOAT latitude
    FLOAT longitude
    FLOAT speed
    FLOAT fix
    FLOAT precision
    INTEGER city_id "FK"
    INTEGER route_id "FK"
    INTEGER vehicle_id "FK"
    INTEGER direction
    FLOAT dop
    VARCHAR(255) comment
    INTEGER dir_id "FK"
}

brand {
    INTEGER id "PK AI"
    VARCHAR(20) name
}

model {
    INTEGER id "PK AI"
    VARCHAR(25) name
    INTEGER brand_id "FK"
}

link_type {
    INTEGER id "PK AI"
    VARCHAR(10) name
}

provider {
    INTEGER id "PK AI"
    VARCHAR(30) name
}

camera {

```

```

        INTEGER id "PK AI"
        INTEGER src_id "FK"
        VARCHAR(50) name
        INTEGER city_id "FK"
        DATETIME integration
        BOOLEAN online
        INTEGER link_id "FK"
        INTEGER provider_id "FK"
        VARCHAR(20) camera_id
        VARCHAR(20) encoder_id
        INTEGER model_id "FK"
        INTEGER encoder_model_id "FK"
        FLOAT latitude
        FLOAT longitude
    }

    source {
        INTEGER id "PK AI"
        VARCHAR(255) name
    }

model ||--{| camera : model_id
model ||--{| camera : encoder_model_id
source ||--{| waze_route: src_id
speed }|--|| city : city_id
state ||--{| city : state_id
city ||--{| street : city_id
alert_type ||--{| alert_subtype : type_id
alert_subtype ||--{| alert : subtype_id
alert_cause ||--{| alert : cause_id
alert_direction ||--{| alert : dir_ids
city ||--{| alert : city_id
sector ||--{| alert : sector_id
tribunal ||--{| alert : tribunal_id
source ||--{| alert: src_id
street ||--{| alert_street : street_id
street ||--{| waze_route : from_id
street ||--{| waze_route : to_id
alert_street }|--|| alert : alert_id
alert ||--{| injury : alert_id
injury_level ||--{| injury : level_id
injury_place ||--{| injury : place_id
route_name ||--{| waze_route : name_id
route_type ||--{| waze_route : type_id
route_point }|--|| waze_route : route_id
waze_route ||--{| alert : waze_route_id
camera }|--|| city : city_id
tracking_folder ||--{| route : folder_id
route ||--{| speed : route_id
vehicle ||--{| speed : vehicle_id
direction ||--{| speed : dir_id
speed }|--|| source: src_id
brand ||--{| model : brand_id

```

```

link_type ||--|{ camera : link_id
provider ||--|{ camera : provider_id
camera }|--|| source: src_id

```

## Estructura de la API

Diagrama de flujo de los endpoints disponibles:

preview

```

flowchart LR
    base{/api/data} -->
    use0{/route} & use1{/alert} & use2{/speed} & use3{/camera}
    use0 --> get00[GET /] & get01[GET /:id]
    use1 --> get10[GET /] & get11[GET /:id]
    use2 --> get20[GET /] & get21[GET /:id]
    use3 --> get30[GET /] & get31[GET /:id]

```

## Tipos de Typescript

Interfaces completas para los objetos devueltos por la API, incluyendo:

- WazeRoute, Alert, Speed, Camera
- Estructuras de agrupación: WazeRouteGroup, AlertGroup, etc.
- Estadísticas y categorías

```

export interface IdName {
    id: number;
    name?: string;
}

export interface Position {
    latitude: number;
    longitude: number;
}

export interface City extends IdName {
    state: IdName;
}

export interface Street extends IdName {
    city: City;
}

export interface WazePoint extends Position {
    id: number;
    index: number;
}

export interface WazeRoute {
    id: number;
}

```

```

source: IdName;
date: Date;
historicTime: number;
from: Street;
name: IdName;
to: Street;
length: number;
time: number;
jamLevel: number;
type: IdName;
points: WazePoint[] ;
}

export interface AlertSubtype extends IdName {
  type: IdName;
}

export interface Injury {
  level: IdName;
  place: IdName;
}

export interface Alert extends Partial<Position> {
  id: number;
  source: IdName;
  date: Date;
  city?: City;
  subtype?: AlertSubtype;
  cause?: IdName;
  km?: number;
  fine?: number;
  sector?: IdName;
  tribunal?: IdName;
  completeReport?: boolean;
  trafficIncident?: boolean;
  reliability?: number;
  length?: number;
  delay?: number;
  dir?: IdName;
  wazeRoute?: WazeRoute;
  injuries?: Injury[];
}

export interface Route extends IdName {
  folder: IdName;
}

export interface Vehicle extends IdName {
  plate: string;
}

export interface Speed extends Position {
  id: number;
}

```

```

source: IdName;
date: Date;
speed: number;
fix?: number;
precision?: number;
city?: City;
route?: Route;
vehicle?: Vehicle;
direction?: number;
dop?: number;
comment?: string;
dir?: IdName;
}

export interface Model extends IdName {
  brand: IdName;
}

export interface Camera extends IdName, Position {
  source: IdName;
  city: City;
  integration: Date;
  online: boolean;
  link: IdName;
  provider: IdName;
  camId: string;
  encoderId?: string;
  model: Model;
  encoderModel?: Model;
}

export interface Stats<T> {
  count: number;
  min: T;
  mean: T;
  logMean: T;
  median: T;
  max: T;
  stdDev: T;
  logStdDev: T;
  NAs: number;
}

export interface CatStats<T> {
  total: number;
  categories: [T, number][] ;
  NAs: number;
}

export interface PositionGroup {
  latitude: Stats<number>;
  longitude: Stats<number>;
}

```

```

export interface WazeRouteGroup extends PositionGroup {
  source: CatStats<IdName>;
  date: Stats<Date>;
  historicTime: Stats<number>;
  from: CatStats<Street>;
  name: CatStats<IdName>;
  to: CatStats<Street>;
  length: Stats<number>;
  time: Stats<number>;
  jamLevel: Stats<number>;
  type: CatStats<IdName>;
  discretizedWazeRoutes?: Omit<WazeRouteGroup, "discretizedWazeRoutes">[] ;
  wazeRoutes?: WazeRoute[] ;
}

export interface AlertGroup extends PositionGroup {
  source: CatStats<IdName>;
  date: Stats<Date>;
  city: CatStats<City>;
  state: CatStats<IdName>;
  subtype: CatStats<AlertSubtype>;
  type: CatStats<IdName>;
  cause: CatStats<IdName>;
  km: Stats<number>;
  fine: Stats<number>;
  sector: CatStats<IdName>;
  tribunal: CatStats<IdName>;
  completeReport: CatStats<boolean>;
  trafficIncident: CatStats<boolean>;
  reliability: Stats<number>;
  length: Stats<number>;
  delay: Stats<number>;
  dir: CatStats<IdName>;
  discretizedAlerts?: Omit<AlertGroup, "discretizedAlerts">[] ;
  alerts?: Alert[] ;
}

export interface SpeedGroup extends PositionGroup {
  source: CatStats<IdName>;
  date: Stats<Date>;
  speed: Stats<number>;
  fix: Stats<number>;
  precision: Stats<number>;
  city: CatStats<City>;
  state: CatStats<IdName>;
  route: CatStats<Route>;
  folder: CatStats<IdName>;
  vehicle: CatStats<Vehicle>;
  direction: CatStats<number>;
  dop: Stats<number>;
  dir: CatStats<IdName>;
  discretizedSpeeds?: Omit<SpeedGroup, "discretizedSpeeds">[] ;
  speeds?: Speed[] ;
}

```

```

}

export interface CameraGroup extends PositionGroup {
  source: CatStats<IdName>;
  city: CatStats<City>;
  state: CatStats<IdName>;
  integration: Stats<Date>;
  online: CatStats<boolean>;
  link: CatStats<IdName>;
  provider: CatStats<IdName>;
  model: CatStats<Model>;
  brand: CatStats<IdName>;
  encoderModel: CatStats<Model>;
  encoderBrand: CatStats<IdName>;
  cameras: Camera[] ;
}

```

## Visualización de tipos

Diagrama de clases interactivo que muestra las relaciones entre interfaces:

Ver diagrama de clases

```

classDiagram
  class IdName {
    number id
    string? name
  }

  class Position {
    number latitude
    number longitude
  }

  class City {
    IdName state
  }

  class Street {
    City city
  }

  class WazePoint {
    number id
    number index
  }

  class WazeRoute {
    number id
    IdName source
    Date date
    number historicTime
    Street from
  }

```

```

    IdName name
    Street to
    number length
    number time
    number jamLevel
    IdName type
    WazePoint[] points
}

class AlertSubtype {
    IdName type
}

class Injury {
    IdName level
    IdName place
}

class Alert {
    number id
    IdName source
    Date date
    City? city
    AlertSubtype? subtype
    IdName? cause
    number? km
    number? fine
    IdName? sector
    IdName? tribunal
    boolean? completeReport
    boolean? trafficIncident
    number? reliability
    number? length
    number? delay
    IdName? dir
    WazeRoute? wazeRoute
    Injury[] injuries
}

class Route {
    IdName folder
}

class Vehicle {
    string plate
}

class Speed {
    number id
    IdName source
    Date date
    number speed
    number? fix
}

```

```

number? precision
City? city
Route? route
Vehicle? vehicle
number? direction
number? dop
string? comment
IdName dir
}

class Model {
    IdName brand
}

class Camera {
    IdName source
    City city
    Date integration
    boolean online
    IdName link
    IdName provider
    string camId
    string? encoderId
    Model model
    Model? encoderModel
}

class Stats~T~{
    number count
    T min
    T mean
    T logMean
    T median
    T stdDev
    T logStdDev
    T max
    number NAs
}

class CatStats~T~{
    number total
    [T, number] [] categories
}

class PositionGroup {
    Stats~number~ latitude
    Stats~number~ longitude
}

class WazeRouteGroup {
    CatStats~IdName~ source
    Stats~Date~ date
    Stats~number~ historicTime
}

```

```

CatStats~Street~ from
CatStats~IdName~ name
CatStats~Street~ to
Stats~number~ length
Stats~number~ time
Stats~number~ jamLevel
Omit~WazeRouteGroup, "discretizedWazeRoutes"~[]? discretizedWazeRoutes
WazeRoute[]? wazeRoutes
}

class AlertGroup {
    CatStats~IdName~ source
    Stats~Date~ date
    CatStats~City~ city
    CatStats~IdName~ state
    CatStats~AlertSubtype~ subtype
    CatStats~IdName~ type
    CatStats~IdName~ cause
    Stats~number~ km
    Stats~number~ fine
    CatStats~IdName~ sector
    CatStats~IdName~ tribunal
    CatStats~boolean~ completeReport
    CatStats~boolean~ trafficIncident
    Stats~number~ reliability
    Stats~number~ length
    Stats~number~ delay
    CatStats~IdName~ dir
    Omit~AlertGroup, "discretizedAlerts"~[]? discretizedAlerts
    Alert[]? alerts
}

class SpeedGroup {
    CatStats~IdName~ source
    Stats~Date~ date
    Stats~number~ speed
    Stats~number~ fix
    Stats~number~ precision
    CatStats~City~ city
    CatStats~IdName~ state
    CatStats~Route~ route
    CatStats~IdName~ folder
    CatStats~Vehicle~ vehicle
    CatStats~number~ direction
    Stats~number~ dop
    CatStats~IdName~ dir
    Omit~SpeedGroup, "discretizedSpeeds"~[]? discretizedSpeeds
    Speed[]? speed
}

class CameraGroup {
    CatStats~IdName~ source
    CatStats~City~ city

```

```

CatStats~IdName~ state
Stats~Date~ integration
CatStats~boolean~ online
CatStats~IdName~ link
CatStats~IdName~ provider
CatStats~Model~ model
CatStats~IdName~ brand
CatStats~Model~ encoderModel
CatStats~IdName~ encoderBrand
Camera[] camera
}

CatStats --- Model
CatStats --- IdName
CatStats --- Vehicle
CatStats --- Route
CatStats --- City
CatStats --- AlertSubtype
CatStats --- Street
PositionGroup --o Stats
PositionGroup <|-- CameraGroup
PositionGroup <|-- SpeedGroup
PositionGroup <|-- WazeRouteGroup
PositionGroup <|-- AlertGroup
Stats o-- CameraGroup
Stats o-- SpeedGroup
Stats o-- WazeRouteGroup
Stats o-- AlertGroup
CameraGroup --o Camera
CameraGroup --o CatStats
AlertGroup --o CatStats
WazeRouteGroup --o CatStats
SpeedGroup --o Speed
SpeedGroup --o CatStats
AlertGroup --o Alert
WazeRouteGroup --o WazeRoute
Camera --|> Position
Model --|> IdName
Camera --|> IdName
Camera ---o IdName
Camera ---o Model
Position <|-- Alert: Partial
Position <|-- WazePoint
Position <|-- Speed
City o-- Speed
City o-- Alert
City o-- Street
Street o-- WazeRoute
Route o-- Speed
Vehicle o-- Speed
IdName o-- Injury
IdName o-- WazeRoute
IdName o-- Speed

```

```

WazePoint o-- WazeRoute
Injury o-- Alert
WazeRoute o-- Alert
IdName <|-- Route
IdName o-- Route
IdName <|-- Vehicle
IdName <|-- City
IdName o-- City
IdName <|-- AlertSubtype
IdName o-- AlertSubtype
IdName <|-- Street
AlertSubtype o-- Alert

<<interface>> IdName
<<interface>> Position
<<interface>> City
<<interface>> Street
<<interface>> WazePoint
<<interface>> WazeRoute
<<interface>> AlertSubtype
<<interface>> Injury
<<interface>> Alert
<<interface>> Route
<<interface>> Vehicle
<<interface>> Speed
<<interface>> Model
<<interface>> Camera
<<interface>> Stats
<<interface>> CatStats
<<interface>> PositionGroup
<<interface>> WazeRouteGroup
<<interface>> AlertGroup
<<interface>> SpeedGroup

```

## Notas Finales

- Todas las consultas requieren autenticación mediante `token`.
- Los filtros son opcionales y permiten refinamiento por fecha, ubicación, tipo de dato y otros atributos.
- Las respuestas incluyen metadatos estadísticos y estructuras discretizadas para análisis espaciotemporales.
- La API sigue un diseño RESTful y devuelve JSON en todas las respuestas.