

Estructura de la API y la Base de Datos

Contents

Tablas de la Base de Datos	2
state	2
city	3
sector	3
street	3
tribunal	4
alert_type	4
alert_subtype	4
alert_direction	4
alert_cause	5
route_name	5
route_type	5
waze_route	5
route_point	6
alert	6
injury_level	6
injury_place	7
injury	7
alert_street	7
tracking_folder	8
route	8
vehicle	8
direction	8
speed	9
brand	9
model	9
link_type	10
provider	10
camera	10

Endpoints de la API	10
USE /route	11
GET /	11
GET /:id	11
USE /alert	11
GET /	11
GET /:id	12
USE /speed	12
GET /	12
GET /:id	13
USE /camera	13
GET /	13
GET /:id	14
Anexos	14
Modelo Entidad Relación	14
Estructura de la API	19
Tipos de Typescript	19

Tablas de la Base de Datos

A continuación se muestran las diferentes tablas de la Base de Datos, donde se indican:

- Fuentes: Tablas de donde se obtendrían los datos
- Columnas: Nombre y tipo de dato

El Modelo Entidad Relación se encuentra en los anexos.

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
- 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
- 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
- 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
- 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

A continuación, se muestran los diferentes endpoints que podría tener la API, la estructura está en los anexos

USE /route

GET /

Obtener un conjunto de rutas de Waze

Queries:

- ?token: Token
- ?min_date: Fecha mínima (por defecto = Ahora - 1 día)
- ?max_date: Fecha máxima (por defecto = Ahora)
- ?min_lat: Latitud mínima
- ?max_lat: Latitud máxima
- ?min_lng: Longitud mínima
- ?max_lng: Longitud máxima

Response:

- WazeRouteGroup

GET /:id

Obtener una ruta de Waze en específico

Queries:

- ?token: Token

Params:

- /:id: ID de la ruta de waze

Response:

- WazeRoute

USE /alert

GET /

Obtener un conjunto de alertas o accidentes

Queries:

- ?token: Token
- ?min_date: Fecha mínima (por defecto = Ahora - 1 día)
- ?max_date: Fecha máxima (por defecto = Ahora)
- ?min_lat: Latitud mínima
- ?max_lat: Latitud máxima
- ?min_lng: Longitud mínima
- ?max_lng: Longitud máxima
- ?state: Región

- ?city: Ciudad
- ?type: Tipo de alerta/accidente
- ?subtype: Subtipo de alerta/accidente
- ?min_km: Kilómetro mínimo
- ?max_km: Kilómetro máximo
- ?min_fine: Parte mínimo
- ?max_fine: Parte máximo
- ?sector: Sector
- ?tribunal: Tribunal
- ?complete_report: Reporte Completo (0 o 1)
- ?traffic_incident: Accidente de tráfico (0 o 1)
- ?min_rel: Confiabilidad Mínima (entre 5 y 10)
- ?max_rel: Confiabilidad Máxima (entre 5 y 10)
- ?min_len: Largo mínimo
- ?max_len: Largo máximo
- ?min_del: Desfase mínimo
- ?max_del: Desfase máximo
- ?dir: Dirección (N-S, S-N, etc)

Response:

- AlertGroup

GET /:id

Obtener una alerta o accidente en específico

Queries:

- ?token: Token

Params:

- /:id: ID de la alerta o accidente

Response:

- Alert

USE /speed

GET /

Obtener un conjunto de medidas de velocidad

Queries:

- ?token: Token
- ?min_date: Fecha mínima (por defecto = Ahora - 1 día)
- ?max_date: Fecha máxima (por defecto = Ahora)
- ?min_lat: Latitud mínima

- ?max_lat: Latitud máxima
- ?min_lng: Longitud mínima
- ?max_lng: Longitud máxima
- ?min_spd: Velocidad mínima
- ?max_spd: Velocidad máxima
- ?min_fix: Fix mínimo
- ?max_fix: Fix máximo
- ?min_pres: Presición mínima
- ?max_pres: Presición máxima
- ?state: Región
- ?city: Ciudad
- ?folder: Carpeta de la ruta
- ?route: Ruta
- ?vh_type: Tipo de vehículo
- ?plate: Patente del vehículo
- ?direction: Dirección (-1, 0 o 1)
- ?min_dop: Presión geométrica mínima
- ?max_dop: Presión geométrica máxima

Response:

- SpeedGroup

GET /:id

Obtener una medida en específico

Queries:

- ?token: Token

Params:

- /:id: ID de la medida

Response:

- Speed

USE /camera

GET /

Obtener un conjunto de cámaras

Queries:

- ?token: Token
- ?min_int: Fecha de integración mínima
- ?max_int: Fecha de integración máxima
- ?min_lat: Latitud mínima

- `?max_lat`: Latitud máxima
- `?min_lng`: Longitud mínima
- `?max_lng`: Longitud máxima
- `?state`: Región
- `?city`: Ciudad
- `?online`: Estado de la cámara (0 o 1)
- `?link`: Tipo de enlace
- `?prov`: Proveedor
- `?brand`: Marca de la cámara
- `?model`: Modelo de la cámara
- `?enc_brand`: Marca del codificador de video
- `?enc_model`: Modelo del codificador de video

Response:

- CameraGroup

GET `/:id`

Obtener una cámara en específico

Queries:

- `?token`: Token

Params:

- `/:id`: ID de la medida

Response:

- Camera

Anexos

Modelo Entidad Relación

preview

```
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"
    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"
    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"
    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"
        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
    }

    state ||--|{ city : state_id
    city ||--|{ street : city_id
    speed }|--|| city : 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

    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
brand |--|{ model : brand_id
model |--|{ camera : model_id
model |--|{ camera : encoder_model_id

link_type |--|{ camera : link_id
provider |--|{ camera : provider_id

```

Estructura de la API

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

```

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;
    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;
    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;
  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 {
  city: City;
  integration: Date;
  online: boolean;
  link: IdName;
  provider: IdName;
  camId: string;
  encoderId?: string;
  model: Model;
  encoderModel?: Model;
}

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

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

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

```

```

export interface WazeRouteGroup {
  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>;
  wazeRoutes: WazeRoute[];
}

export interface AlertGroup extends Partial<PositionGroup> {
  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>;
  alerts: Alert[];
}

export interface SpeedGroup extends PositionGroup {
  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>;
  speeds: Speed[];
}

export interface CameraGroup extends PositionGroup {
  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[];
}
```