**Apuntes en sucio memoria**

Proceso de desarrollo de modelo relacional y de interfaz de usuario:  
El diseño del **modelo relacional** de la base de datos comenzó con una fase conceptual en la que se esbozaron las entidades y relaciones clave. Una vez consolidado el diseño, se empleó **MySQL Workbench** para su implementación definitiva. Esta herramienta facilitó la creación de un modelo relacional detallado y la generación automática de la base de datos, incluyendo la implementación de **claves foráneas** para mantener la integridad de los datos.

En cuanto a la interfaz de usuario, se realizaron prototipos iniciales utilizando **Balsamiq**, lo que permitió visualizar y ajustar la distribución de los elementos en las pantallas de manera eficiente. Este enfoque iterativo ayudó a definir la experiencia del usuario antes de proceder con el desarrollo de la interfaz final.

Conjuntos de datos:

* Accidentes de tráfico de la ciudad de Madrid (accidentalidad.csv) - **Última actualización:**29/01/2024 | **Actualización:**Mensual
* Accidentes de tráfico con implicación de bicicletas (accidentes\_bicicletas\_2023.csv) - **Última actualización:**29/01/2024 | **Actualización:**Mensual
* Servicio de Estacionamiento Regulado (SER). Calles y número de plazas (estacionamiento.csv) - **Última actualización:**30/01/2024 | **Actualización:**Trimestral
* Plazas de estacionamiento para personas de movilidad reducida (estacionamiento\_movilidad\_reducida.csv) - **Última actualización:**18/12/2023 | **Actualización:**Anual
* Moto. Reservas moto (reservas\_moto.csv) - **Actualización:**Semestral | **Última actualización:**06/10/2023
* Carga y descarga. Reserva de estacionamiento para carga y descarga de mercancías en vía pública (Carga\_descarga\_2023.csv) - **Última actualización:**17/07/2023 | **Actualización:**Anual
* Ubicación de los radares (radares.csv)
* Aforos de tráfico en la ciudad de Madrid permanentes:

Para estos datos necesitamos 3 archivos:

AforoPermanenteNoviembre2023.csv 🡪 Aquí tenemos la fecha de la toma de datos, la estación, el sentido y las diferentes horas con el aforo/trafico

ExplicacionAforoPermanente.pdf 🡪 PDF que explica los datos del archivo de noviembre

UbicacionEstacionesPermanentesSentidos.csv 🡪 Con las ubicaciones de las estaciones

* Aparcamientos públicos (Rotacionales). Datos de ocupación en tiempo real. ¿Como funciona la API?
* Aparcamientos públicos municipales (Rotacionales). Histórico de ocupación (aparcamientos\_rotacionales.csv) - **Última actualización:**16/01/2024
* Multas de circulación: detalle (va por meses)

ACCIDENTALIDAD:  
Las coordenadas vienen en utm tengo que usar alguna api (OSM, OpenStreetMap) para pasar de utm a geográficas.

Puedo filtrar por fecha y hora, por el estado meteorológico en el momento del accidente, por el rango de edad del afectado, por la gravedad de la lesión, el tipo de vehículo, el distrito y si es positivo en alcohol/drogas.

Puedo mostrar un mapa con marcadores en los lugares de los accidentes y filtrar por los criterios mencionados anteriormente.

APARCAMIENTOS PÚBLICOS. HISTORICO DE OCUPACIÓN.

Me muestra los resultados de ocupación media de cada parking por cada mes del año. No se si me servirá mucho esto

Zonas públicas estacionamiento.

Coordenadas UTM, distrito, barrio, calle, nº finca (nº de policía de la via según catastro), color (tipo de plaza de estacionamiento), batería/línea, nº de plazas. También hay datasets de parquímetros, tiques de aparcamiento,…

AFORO APARCAMIENTOS EN TIEMPO REAL.

Se hace a través de una API. Muy útil. Como puedo usarla?

RADARES.

Algunas coordenadas vienen en WGS84. Todas vienen con longitud y latitud pero hay que investigar que tipo de coordenadas porque no parecen geográficas.

ACCIDENTES BICIS.

Las coordenadas vienen en utm. Similar a los accidentes de vehículos.

Ideas para el TFG:

1. App de rutas que muestre los caminos más seguros (evitando las zonas con mayor tasa de accidentes)
2. App con rutas y horarios de los autobuses
3. **App con mapa que tenga múltiples funcionalidades: mostrar radares, zonas con mayores accidentes, aparcamientos libres en tiempo real (API),…**

**TECNOLOGÍAS**

Base de datos SQL 🡪 Mysql, SqlWorkbench

Express 🡪 Backend

React 🡪 Frontend

**REACT-LEAFLET 🡪 FUNCIONALIDADES QUE PUEDO IMPLEMENTAR   
Documentación:** [**https://react-leaflet.js.org/**](https://react-leaflet.js.org/)

* Mostrar la ubicación del usuario en el mapa 🡪 Events
* Mostrar un Popup al poner el ratón encima de un Polygon, Marker,… 🡪 Tooltips
* **Hacer el mostrar todas las zonas o markers concretos de otra manera 🡪 Layers Control**
* **Colocar un marcador en el mapa. Usos: Colocar un marcador en el mapa y mostrar los datos en un radio definido (accidentes, estacionamientos, radares,…) 🡪 Draggable marker**
* **Viajar por el mapa dando click en lugar de arrastrándose 🡪 Animated panning**
* Añadir un botón que vuelva a centrar el mapa 🡪 External state

**RECHARTS 🡪 FUNCIONALIDADES QUE PUEDO IMPLEMENTAR   
Documentación:** [**https://recharts.org/en-US/guide**](https://recharts.org/en-US/guide) **Blog sobre como crear un mapa de calor:** [**https://medium.com/swlh/creating-a-football-heat-map-react-component-with-recharts-ff310162cee0**](https://medium.com/swlh/creating-a-football-heat-map-react-component-with-recharts-ff310162cee0)

* SameDataComposedChart: Muestra barCharts y una línea que muestra la progresión de una a la otra.   
  Usos:
  + Tráfico: Puedo usarlo para mostrar el flujo de tráfico de una estación/distrito en un periodo de tiempo determinado   
    (MOSTRARLO EN PÁGINA APARTE?/MOSTRARLO COMO VENTANA MODAL?)
* BrushBarChart: Permite cargar un BarChart con muchísimas barras y tener debajo un elemento con el que hacer zoom y desplazarse por el grid.  
  Usos:
  + Tráfico: Para mostrar el trafico de cada distrito/estación (Puedo mostrarlo ordenado de mayor a menos o viceversa). Al mostrarlo debo de deshabilitar el ControlLayer.Overlay (al darle a un botón desaparece el Chart y vuelve a aparecer el ControlLayer). Al clickar un Bar que se haga referencia al distrito/estación
* CustomActiveShapePieChart: Muestra un PieChart (círculo con datos) y al poner el ratón sobre cada zona muestra datos específicos.  
  Usos:
  + Accidentes: Mostrar estadísticas sobre una característica concreta (accidentes por el tipo de vehículo, tipo de accidente, tipo de lesión por significado?)

MOSTRAR CHARTS EN LOS ESTACIONAMIENTOS (TIPO QUE ZONAS TIENEN MAS ESTACIONAMIENTOS)

**Bootstrap 🡪 Mejoras en la interfaz  
Documentación:** [**https://getbootstrap.com/docs/5.3/getting-started/introduction/**](https://getbootstrap.com/docs/5.3/getting-started/introduction/)

* Forms 🡪 Validation
* Range 🡪 Para el radio de la ubicación en accidentes
* Toggle button, Input group,…
* Tooltips 🡪 Para mostrar mensajes en el formulario que notifiquen (hace falta Popper) + Mostrar mensajes al poner el ratón encima de un botón o cualquier otra cosa
* Placeholders 🡪 Mostrarlos cuando este cargando la pagina?
* Spinners 🡪 Cambiar el loading por un spinner?
* Shadow detrás de los mapas?
* Iconos Bootstrap. 2 usos: svg (copiar y pegar) | <i> (hay que meter el cdn o enlace en el proyecto)
* Agregar fuentes con Google fonts: meter los links en el index.html
* Usar posicionamiento sticky en el navbar para fijarlo arriba
* Alerts para el manejo de errores (Dismissing)
* **¿¿Hacer formulario con Offcanvas??**
* Mostrar multas en tabla de react-bootstrap?

**REDUX Toolkit 🡪 Mejor manejo de variables de estado  
Documentación:** [**https://redux-toolkit.js.org/**](https://redux-toolkit.js.org/)

* Redux Thunk permite que las acciones de Redux puedan no solo devolver objetos sino también funciones. Puedo usarlo para el manejo de funciones asíncronas y hacer que unas funciones esperen a que otras se ejecuten. Thunk es un middleware.

**Entidad Accidente (vehículo o bici)**

**LOS ACCIDENTES DE VEHICULO NO SE INSERTAN TODOS**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Descripción | DataType MySQL | DataType Sequelize |
| Fecha (dd/mm/yyyy) |  | DATE | DATEONLY |
| Hora (hh/mm/ss) |  | TIME | TIME |
| Localización | calle del accidente (a veces aparece el numero) | VARCHAR(255) | STRING |
| Distrito | Nombre del distrito | VARCHAR(255) | STRING |
| Tipo\_accidente \*1 | Accidente tipificado | VARCHAR(255) | STRING |
| Estado\_meteorologico \*2 |  | VARCHAR(255) | STRING |
| Tipo\_vehiculo \*3 | Tipo vehículo afectado | VARCHAR(255) | STRING |
| Tipo\_persona \*4 |  | VARCHAR(255) | STRING |
| Tramo\_edad | De 0-4 ó 5-9 \*6 | INT | INTEGER |
| Sexo | Hombre, Mujer o no asignado | VARCHAR(255) | STRING |
| Lesividad \*5 | Tipo de la lesión | INT | INTEGER |
| Lat | Latitud | FLOAT | FLOAT |
| Lon | Longitud | FLOAT | FLOAT |
| Drogas | Positivo en drogas | TINYINT(1) | BOOLEAN |
| alcohol | Positivo en alcohol | TINYINT(1) | BOOLEAN |

\*1 Tipos de accidentes:

- Colisión doble: accidente de tráfico entre 2 vehículos (colisión frontal, colisión fronto-lateral, colisión lateral)

- Colisión múltiple: accidente de tráfico entre más de 2 vehículos

- Alcance: Accidente que se produce cuando un vehículo circulando o detenido por las circunstancias del tráfico es golpeado en su parte posterior por otro vehículo.

- Choque contra obstáculo fijo: Accidente ocurrido entre un vehículo en movimiento con conductor y un objeto inmóvil que ocupa la vía o zona apartada de la misma.

- Atropello a persona: Accidente ocurrido ente un vehículo y un peatón.

- Vuelco: Accidente sufrido por un vehículo con más de dos ruedas y que por alguna circunstancia sus neumáticos pierden el contacto con la calzada quedando apoyado sobre un costado o sobre el techo.

- Caída: Se agrupan todas las caídas relacionadas con el desarrollo y las circunstancias del tráfico, (motocicleta, ciclomotor, bicicleta, viajero bus, etc.,).

- Otro: Recoge accidentes por atropello a animal, despeñamiento, salida de la vía, y otros.

\*2 Estado meteorológico:

- Despejado.

- Nublado.

- Lluvia débil.

- Se desconoce.

- Lluvia intensa.

- Granizando.

**??NO SE SI HABRA MAS?? HAY QUE COMPROBARLO**

\*3 Tipo de vehículo: Todo terreno, Turismo, Motocicleta hasta 125cc, Furgoneta, Vehículo articulado, Autobús, Camión rígido, Ciclomotor, Tractocamión, Motocicleta > 125cc, Bicicleta, Otros vehículos con motor, Bicicleta EPAC (pedaleo asistido), Maquinaria de obras, VMU eléctrico.

**??NO SE SI HABRA MAS?? HAY QUE COMPROBARLO**

\*4 Tipo de persona: Conductor, Pasajero, Peatón

\*5 Lesividad:

- 01: Atención en urgencias sin posterior ingreso. - LEVE

- 02: Ingreso inferior o igual a 24 horas – INTERMEDIO

- 03: Ingreso superior a 24 horas. - GRAVE

- 04 Fallecido 24 horas - FALLECIDO

- 05 Asistencia sanitaria ambulatoria con posterioridad - LEVE

- 06 Asistencia sanitaria inmediata en centro de salud o mutua - LEVE

- 07 Asistencia sanitaria sólo en el lugar del accidente - LEVE

- 14 Sin asistencia sanitaria Ninguna

- 77 Se desconoce Ninguna

- En blanco – Sin asistencia sanitaria

\*6 En el csv el rango de edad viene en este formato:

- “Menor de 5 años”, “De 6 a 9 años”, “De 10 a 14 años”, “De 15 a 17 años”, “De 18 a 20 años”, “De 21 a 24 años”, “De 25 a 29 años”, “De 30 a 34 años” y el resto sigue este patrón: 0 a 4, 5 a 9.

- En la base de datos almacenaré el número mínimo del rango de manera que si tengo un 21 se que el rango va de 21 a 24.

**Posibles consultas**

Inicialmente se mostrará el mapa con polígonos de colores que representen el “peligro” de la zona según la cantidad de accidentes que han ocurrido y su gravedad.

* Clasificar zonas por la cantidad y la gravedad de los accidentes. Podemos hacer una consulta que devuelva una lista de objetos. En la que cada objeto contenga las coordenadas de la zona para marcarla (un cuadrado) y un número que indique el “peligro” de la zona.

Podemos asignar un número a cada tipo de lesión:

1 🡪 Sin asistencia sanitaria / Se desconoce | AZUL

2 🡪 Leve | AMARILLO

3 🡪 Grave | NARANJA

5 🡪 Fallecido | ROJO

El peligro de la zona será la media de la suma del valor de sus lesiones.

* Mostrar una tabla con estadísticas de accidentes agrupados por diferentes características:
* Por rango de edad
* Por tipo de vehículo
* Por distrito
* Por gravedad de la lesión
* Por tipo de accidente
* Por positivo en drogas
* Por positivo en alcohol

Podremos buscar accidentes concretos filtrando por distintos parámetros:

* Buscar por fecha:
  + Buscar por fecha concreta.
  + Buscar entre 2 fechas concretas.
  + Buscar por año y mes.
* Buscar por hora (sin fecha):
  + Buscar por hora concreta
  + Buscar entre 2 horas concretas.
* Buscar por rango de edad: busco dentro de un rango dado por 2 números. Todo accidente que esté dentro de ese rango lo filtro.
* Buscar por sexo
* Buscar por distrito
* Buscar por positivo en drogas
* Buscar por positivo en alcohol
* Buscar por tipo de vehículo
* Buscar por estado meteorológico
* Buscar por tipo de accidente
* Buscar por lesividad – Gravedad (LEVE, GRAVE, FALLECIDO o Sin asistencia sanitaria)
* Buscar por lesividad concreta (Ingreso inferior o igual a 24 horas, ingreso superior a 24 horas, Asistencia sanitaria ambulatoria con posterioridad,…) – Busco por codigo

Consultas mezcladas

**ENTIDAD ESTACIONAMIENTO (incluimos los minusválidos, las reservas de motos y las zonas de carga y descarga)**

estacionamiento\_movilidad\_reducida.csv 🡪 Plazas de estacionamiento para personas de movilidad reducida

estacionamiento.csv 🡪 Servicio de Estacionamiento Regulado (SER). Calles y número de plazas

reservas\_moto.csv 🡪 Moto. Reservas moto

Carga\_descarga\_2023.csv 🡪 Carga y descarga. Reserva de estacionamiento para carga y descarga de mercancías en vía pública

Los csv son distintos por lo que tenemos que adaptar los datos.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Descripción | DataType MySQL | DataType Sequelize | CSV SER | CSV MINUSV | CSV  Motos |
| Lat | Longitud | FLOAT | FLOAT | gis\_x (UTM) | Latitud | Latitud |
| Lon | Latitud | FLOAT | FLOAT | gis\_y (UTM) | Longitud | Longitud |
| Distrito | Nombre del distrito | VARCHAR(255) | STRING | distrito | Distrito | Distrito |
| Barrio | Nombre del barrio | VARCHAR(255) | INTEGER | barrio | Barrio | Barrio |
| Color \*7 | Tipo de plaza | VARCHAR(255) | STRING | color | NO HAY / Minusvalido | NO HAY/ Motos |
| tipo | Línea o bateria | VARCHAR(255) | STRING | batería\_linea | Línea / Batería | Línea / Batería |
| plazas | Número de plazas | INT | INTEGER | num\_plazas | Número de Plazas | Número de Plazas |

\*7 Color.

Con el color indicamos el tipo de aparcamiento:

* Verde: uso residencial.
* Azul: uso rotacional.
* Naranja: Uso larga estancia.
* Rojo: Uso ámbito sanitario.
* Amarillo: Minusválidos.
* Negro: Reserva de motos
* Morado: Carga y descarga

**Posibles consultas**

* Mostrar todas las plazas de estacionamiento diferenciadas por su tipo (color)
* Buscar por distrito
* Buscar por barrio
* Buscar por tipo de aparcamiento
* Buscar por color

Toda zona de estacionamiento aparecerá con un marcador cuyo color dependerá del tipo de estacionamiento. Al pinchar en el marcador aparecerá el número de plazas y si el aparcamiento es en batería o en línea.

A un lado aparecerá una leyenda que explique el tipo de aparcamiento que representa cada color.

**Si la consulta devuelve demasiados resultados podría marcar las zonas y al darle click a una zona muestro solo los estacionamientos de esa zona.**

**Zonas: [**

**{ zona: [limite\_superior\_izq, limite\_superior\_der, limite\_inferior\_izq, limite\_inferior\_der],  
 [estacionamientos]}**

**]**

**Entidad Radar**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Descripción | DataType MySQL | DataType Sequelize |
| Lat |  | FLOAT | FLOAT |
| Lon |  | FLOAT | FLOAT |
| Sentido |  | VARCHAR(255) | STRING |
| Tipo | carril | VARCHAR(255) | STRING |
|  |  |  |  |

**Entidad Multa**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo BD | Nombre en el CV | Descripción | DataType MySQL | DataType Sequelize |
| Calificación | = | Leve, Grave o Muy grave | Enum | Enum |
| Lugar | = | Direccion del lugar | VARCHAR(255) | STRING |
| Mes | = |  | INT | INTEGER |
| Año | Anio |  | INT | INTEGER |
| Hora | = |  |  |  |
| Coste | Imp. Bol. | Importe en euros |  |  |
| Descuento | = | Si/No 🡪 -50% del costo | Boolean |  |
| Puntos | = | Detracción de puntos en el permiso de conducir | INT | INTEGER |
| Infracción | Hecho-Bol | Descripción de la infracción cometida | VARCHAR(255) | STRING |
| Vel\_Limite | Vel. Límite | Limite de circulación en caso de que la infracción sea exceso | INT | INTEGER |
| Vel\_Circula | Vel. Circula | Limite al que circula el individuo en caso de que la infracción sea exceso | INT | INTEGER |
| Lat | Coordenada X |  | FLOAT | FLOAT |
| Lon | Coordenada Y |  | FLOAT | FLOAT |
| Denunciante |  |  |  |  |

Análsis Asociación radares – barrio – zona

|  |  |  |  |
| --- | --- | --- | --- |
| Radar ID | Nº Radar | ID barrio (nombre Barrio) | Lugar (multas.csv) |
| 61 | 3 | 423 (Hispanoamérica) | P. CASTELLANA\_300 |
| 83 + 84 | 18 + 19 | 446 (Argüeles) | M-30 FT K 04,150 CR1 |
| 78 | 13 | 471 (Pradolongo) | RADAR P. SM.CABEZA\_N115 |
| 67 + 68 | 8 + 9 | 409 (Pacífico) +  475 (Palomeras de Sureste) | M-30 XC K 10,300 CR1 |
| 63 + 64 | 4 + 5 | 486 (Ventas) | M-30 C-1 KM 6,700 CR1 |
| 79 | 20 | 455 (Campamento) | A5 KM 4.0 SALIDA |
| 72 | 16 |  | M-30 XC K 16,530 CR2 |
| 80 | 23 |  | M-30 C-1 KM 19,800 CRAL |
| 65 + 66 | 6 + 7 |  | M-30 C-2 KM 7,800 CR-IZDO |
| 59 + 60 | 1 + 2 |  | M-30 C-1 KM 0,500 CR-CRAL |
| 76 + 86 | 21 + 22 |  | M-30 XC K 19,06 CR4 |
| 67 + 68 | 8 + 9 |  | M-30 XC K 10,300 CR3 |
| 81 | 17 |  | M-30 NC K 17,170 CR1 |
| 85 | 24 |  | M-30 CALZADA 1 KM 25,700 |
| 70 | 11 |  | M-30 NL K 13,860 CR1 |

ANALIZAR QUE RADAR ES

* **M-30 CALZADA 2 KM 27.000 => PARECE SER UN RADAR NUEVO. NO LO TENGO EN LA BD??**
* **N015 MONASTERIOS SUSO Y Y | N021 MONASTERIOS SUSO Y Y |**
* **N168 HERMANOS GARCIA NOBL**

**Posibles consultas**

**RADARES**

* Mostrar todos los radares en el mapa
* Al hacer click en un radar mostrar una tabla con las multas asociadas a ese radar
* Filtrar las multas por diferentes criterios como hora, puntos perdidos, calificación de gravedad, coste.
* Mostrar grafica de radares ordenada de mayores multas a menos.

**MULTAS**

* Mostrar una tabla con x multas (100, 500 ?)
* Filtrar la tabla por: calificación, mes concreto, entre meses, hora concreta, entre horas, coste (concreto, rango, mínimo, máximo), puntos (concreto, rango, mínimo, máximo), ¿denunciante?, …
* Ordenar por: fecha, hora, coste o puntos. Solo se podrá ordenar por un parametro

**Entidad Trafico**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Descripción | DataType MySQL | DataType Sequelize |
| fecha | Fecha de la toma de datos | DATE | DATEONLY |
| estacion | Numero de la estación | INT | INTEGER |
| fsen | Sentido y hora \*1 | VARCHAR(255) | STRING |
| orient | Orientación \*2 | VARCHAR(255) | STRING |
| Hor1 | Trafico en la hora 1 (puede ser 1:00 o 13:00 depende de fsen) | INTEGER | INTEGER |
| Hor2 | Tráfico en la hora 2 | INT | INTEGER |
| Hor3 | … | INT | INTEGER |
| Hor4 | … | INT | INTEGER |
| Hor5 | … | INT | INTEGER |
| Hor6 | … | INT | INTEGER |
| Hor7 | … | INT | INTEGER |
| Hor8 | … | INT | INTEGER |
| Hor9 | … | INT | INTEGER |
| Hor10 | … | INT | INTEGER |
| Hor11 | … | INT | INTEGER |
| Hor12 | … | INT | INTEGER |
| Lat | Latitud | FLOAT | FLOAT |
| Lon | Longitud | FLOAT | FLOAT |
| nombre | Nombre de la estacion | VARCHAR(255) | STRING |

\*1 Formato de fsen:

1- Sentido 1. Datos tomados de 1:00 a 12:00

1= Sentido 1. Datos tomados de 13:00 a 24:00

2- Sentido 2. Datos tomados de 1:00 a 12:00

2= Sentido 2. Datos tomados de 13:00 a 24:00

\*2 Formato de orient:

S-N 🡪 De sur a norte

N-S 🡪 De norte a sur

O-E 🡪 De oeste a este

E-O 🡪 De este a oeste

Si fsen = 1- | orient = S-N 🡪 Significa que va de sur a norte la carretera y que las horas son de 1:00 a 12:00

**\*ESTACIONES QUE NO TENEMOS INFO: 35, 34 (sentido E-O), 9 (O-E), 10 (E-O), 12 (E-O),   
17 (E-O), 19 (S-N), 25 (N-S), 44 (S-N), 45 (N-S),**

**Posibles consultas**

* Mostrar mapa con las estaciones marcadas
* Al pinchar sobre una estación podremos ver información sobre el tráfico medio en función del filtro que hayamos aplicado
* Podremos filtrar para ver el tráfico función de:
  + 1 mes concreto
  + 1 día concreto
  + Un periodo de tiempo concreto (especificando 2 dias e incluso 2 horas y ambas combinadas)
  + La calle

(Podremos combinar la calle con la fecha u hora)

* Para una estación concreta podremos ver el tráfico medio en la dirección que especifiquemos

**LO QUE VIENE A CONTINUACIÓN NO ES SEGURO**

* Si seleccionamos 1 zona podremos seleccionar que filtro aplicar sobre esa zona concreta. De manera que el resto de zonas desaparezcan y solo observemos la información de esa zona concreta.

Por ejemplo, estamos visualizando todo el mapa con la información del tráfico en un mes concreto. Por ejemplo, Noviembre de 2023. Si pinchamos sobre una zona veremos que podemos aplicar más filtros sobre la información que ya nos aparece (tráfico en Noviembre de 2023). De manera que podemos filtrar sobre eso para ver el tráfico en una dirección. Por ejemplo, el tráfico norte. De esta manera veremos solo sobre esta zona el tráfico que hay de media en el mes de Noviembre de 2023 hacia la dirección norte.

* También al seleccionar sobre 1 zona podremos visualizar una tabla con el tráfico concreto de cada hora en función del filtro que ya se haya aplicado sobre esa zona.

**INFO EXTRA**

Para sacar estos datos hemos usado 2 csv distintos:  
- Aforos de tráfico en la ciudad de Madrid permanentes  
- Ubicación de estaciones permanentes y sentidos calles

La info del aforo va por meses por lo que tenemos que recopilar distintos csv

Hay en total 60 estaciones

**Actualmente en la bd están insertados los meses:  
Octubre, Noviembre y Diciembre de 2023 | Enero de 2024**

**Distrito**

Datos de las delimitaciones:  
<https://geoportal.madrid.es/IDEAM_WBGEOPORTAL/dataset.iam?id=541f4ef6-762b-11e9-861d-ecb1d753f6e8>

**Barrio**

Datos de los barrios:

<https://geoportal.madrid.es/IDEAM_WBGEOPORTAL/dataset.iam?id=422fa235-762b-11e9-861d-ecb1d753f6e8>