

Práctica 4: Aeropuertos

Abrir en el entorno de desarrollo el archivo `index.html` correspondiente a la práctica 2 y completar el código. El archivo ya contiene parte del código y el objetivo es editar los comentarios que comiencen por la palabra “TODO”.

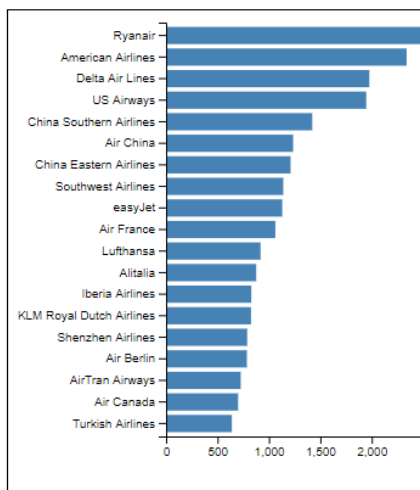
Es importante mantener el resto del código sin modificar, así como los nombres de las funciones, clases CSS o IDs de los elementos.

En esta práctica el objetivo es añadir al mapa un punto por cada uno de los aeropuertos que hay en los datos. Así se puede visualizar en el mapa los principales aeropuertos desde los que las compañías operan.

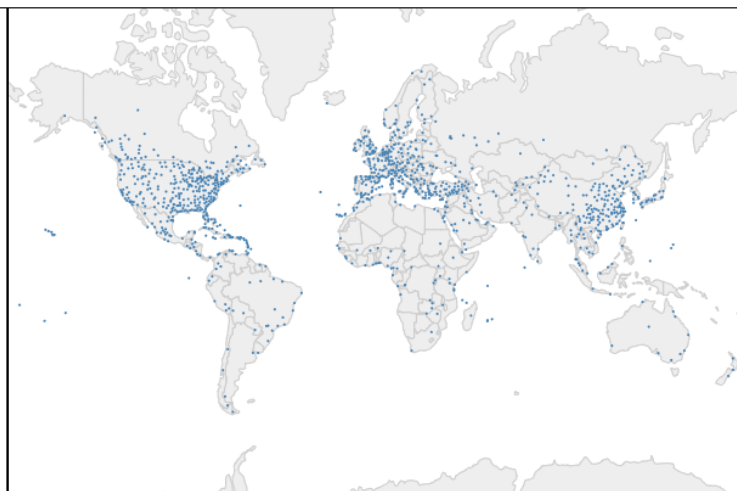
Al final de la práctica debería obtener algo similar a esto:

Rutas Aéreas

Compañías



Aeropuertos



Tareas:

1) Agrupando por aeropuertos.

De la misma forma que para hacer el diagrama de barras hubo que agrupar los datos por compañías, ahora hay que hacerlo por aeropuertos. La diferencia fundamental está en que cada ruta tiene 2 aeropuertos y hay que contar con los 2 para aparezcan todos. La función que se encarga de hacer esta tarea es “*agruparPorAeropuerto*”.

```

function agruparPorAeropuerto ( datos ) {
    //se utiliza reduce para transformar una lista en un objeto
    //donde cada entrada apunta a un aeropuerto usando como clave un
    ID.
    //De esta forma comprobamos si es la primera vez que encontramos
    un aeropuerto
    let resultado = datos.reduce((resultado, d) => {

        //el operador || comprueba si el objeto es false (incluye
        undefined, null 0 0),
        //y en ese caso se usa la 2ª opción.

        let aeropuertoDestino = resultado [d.DestAirportID] || {
            "ID":          d.DestAirportID,
            "Aeropuerto": d.DestAirport,
            "Latitud":    +d.DestLatitude,
            "Longitud":  +d.DestLongitude,
            "Ciudad":     d.DestCity,
            "Pais":       d.DestCountry,
            "Contador":   0
        };

        aeropuertoDestino.Contador += 1;
        resultado[ aeropuertoDestino.ID ] = aeropuertoDestino;

        //Hacemos lo mismo para el aeropuerto de partida.
        let aeropuertoOrigen = resultado [d.SourceAirportID] || {
            "ID":          d.SourceAirportID,
            "Aeropuerto": d.SourceAirport,
            "Latitud":    +d.SourceLatitude,
            "Longitud":  +d.SourceLongitude,
            "Ciudad":     d.SourceCity,
            "Pais":       d.SourceCountry,
            "Contador":   0
        };

        aeropuertoOrigen.Contador += 1;
        resultado[aeropuertoOrigen.ID] = aeropuertoOrigen;

        return resultado;
    }, {});

    //Convertir el disccionario de aeropuertos en una lista
    resultado = Object.keys(resultado).map( id => resultado[id]);
    return resultado;
}

```

Como resultado de esta función obtenemos una lista de 1168 aeropuertos con sus coordenadas geográficas para poder situarlos en el mapa.

¿Cuántos aeropuertos de todos esos están en España?

¿Qué habría que hacer para ver la lista de aeropuertos españoles?

¿Está en la lista el de Valladolid?

2) Dibujar los aeropuertos.

Como ya tenemos la proyección y la configuración del mapa de anteriores prácticas, sólo hay que completar la función “*dibujarAeropuertos*” para pintar en el mapa un círculo por cada uno de los aeropuertos geo-localizados.

```
function dibujarAeropuertos (aeropuertos) {  
    let configuracion = configuracionMapa(); //obtener la  
    configuracion  
    let proyeccion = almacen.proyeccionMapa; //obtener la  
    proyeccion  
    let contenedor = configuracion.contenedor; //obtener el  
    contenedor  
  
    let circulos = contenedor.selectAll("circle");  
    circulos. //TODO 1: enlazar los aeropuertos a los  
    circulos con el método .data  
  
    . //TODO 2a: para cada aeropuerto  
    .append("circle")  
        .attr("r", ) //TODO 2b : Asignar el radio a 1  
        .attr("cx", ) // TODO 2c: asignar el centro del  
    circulo usando la proyeccion para convertir longitud y latitud a posicion  
    x.  
        .attr("cy", ) // TODO 2d: asignar el centro del  
    circulo usando la proyeccion para convertir longitud y latitud a posicion  
    y.  
        .attr("fill", ); //TODO 2e: asignar el color a  
    "steelblue"  
}
```

3) Añadir llamada en mostrarDatos.

```
function mostrarDatos() {  
  
    almacen.companias = agruparPorCompania ( almacen.rutas );  
  
    //console.log (almacen.companias );  
  
    dibujarDiagramaCompanias (almacen.companias);  
  
    dibujarMapa ( almacen.geoJSON );  
  
    almacen.aeropuertos = ;//TODO 3: añadir llamada a función para  
agrupar por aeropuerto  
    dibujarAeropuertos (almacen.aeropuertos);  
}
```