

## Práctica 5: Rutas

Abrir en el entorno de desarrollo el archivo `index.html` correspondiente a la práctica 5 y completar el código. El archivo ya contiene parte del código y el objetivo es editar los comentarios que comiencen por la palabra “TODO”.

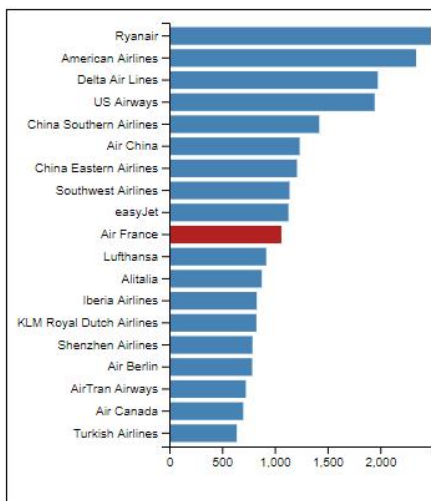
Es importante mantener el resto del código sin modificar, así como los nombres de las funciones, clases CSS o IDs de los elementos.

El objetivo en esta parte es añadir cierta interacción al proyecto, permitiendo que se visualicen las rutas para una determinada compañía.

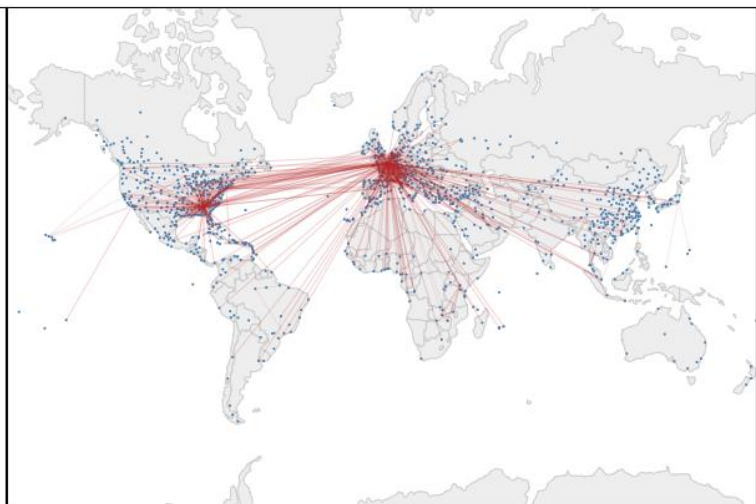
Al final de la práctica debería obtener algo similar a esto:

### Rutas Aéreas

#### Compañías



#### Aeropuertos



## Tareas:

### 1) Dibujando las rutas.

Hay una función que se encarga de dibujar las rutas. Según los datos, los aeropuertos están conectados por una ruta con un origen y un destino.

La función recibe como parámetro un identificador de compañía y dibujará todas las rutas de esa compañía. Para ello hay que utilizar la proyección ya creada en prácticas anteriores y filtrar los datos cargados para obtener las rutas de la compañía seleccionada.

```

function dibujarRutas ( companiaID ) {
    let rutas =;          //TODO 1: obtener las rutas del almacen
    let proyeccion =; //TODO 2: obtener la proyeccion del almacen
    let contenedor =; //TODO 3: seleccionar el svg con el id del mapa
    let rutasSeleccionadas = rutas.filter ( ); //TODO 4: filtrar las
rutas para obtener las pertenecientes a la companiaID

    let datosEnlazados = contenedor.selectAll("line")
        .data(rutasSeleccionadas, d => d.ID) //Este 2º parámetro
indica a D3 como identificar las rutas, y facilita para añadir o quitar.
        . //TODO 5: Usar enter para añadir una línea por cada ruta.
        .append("line") //TODO 6: en cada línea asignar el punto de
origen (x1 and y1) del aeropuerto origen (SourceLongitude and
SourceLatitude)
                                // usando la proyección para
convertir longitud y latitud a x e y.
        .attr("x1", )
        .attr("y1", )
        .attr("x2", ) //TODO 7: en cada línea asignar el punto
final (x2 and y2) en la posición del aeropuerto destino (DestLongitude
and DestLongitude)
        .attr("y2", )
        .attr("stroke" , ) //TODO 8: asignar el color de la línea
(stroke) a "#firebrick"
        .attr("opacity", ); //TODO 9: asignar la opacidad a 0.1

    contenedor.selectAll("line")
        .data(rutasSeleccionadas, d => d.ID)
        . //TODO 10: usar la función exit para eliminar las
rutas que ya no hay que dibujar.
        .remove();
}

```

## 2) Probando la función.

Probar que la función funciona como se espera añadiendo una llamada a la función anterior desde “mostrarDatos”.

```
function mostrarDatos() {

    almacen.companias = agruparPorCompania ( almacen.rutas );

    //console.log (almacen.companias );

    dibujarDiagramaCompanias (almacen.companias);

    dibujarMapa ( almacen.geoJSON );

    almacen.aeropuertos = agruparPorAeropuerto ( almacen.rutas );
    dibujarAeropuertos (almacen.aeropuertos);


    dibujarRutas ("2822"); // ← añadir esta línea
}
```

Una vez comprobado que funciona bien, borrar la línea añadida.

### 3) Añadir interacción.

Vamos a añadir interacción con el usuario haciendo que cuando sobre alguna de las barras del diagrama de barras aparezcan las rutas de la compañía correspondiente.

Para ello hay que modificar la función “*dibujarDiagramaBarras*” que completamos en una práctica anterior, añadiendo respuestas a eventos. El primer evento es ‘mouseenter’ que se usa para saber cuándo el cursor entra en una de las barras y , cuando esto suceda, que se visualicen las rutas. El segundo evento es ‘mouseleave’ que es cuando el cursor abandona la barra y, en este caso, hay que eliminar las rutas.

Al mismo tiempo también cambiará el color de la barra para indicar que el cursor del ratón está encima.

```

function dibujarDiagramaBarras ( companias, escalas, configuracion ) {
    let {margenes, contenedor} = configuracion; // equivalente a 'let
margenes = configuracion.margenes; ...
    let {escalaX, escalaY} = escalas;
    let cuerpo = contenedor.append("g").style("transform",
`translate(${margenes.left}px,${margenes.top}px)`); //coloca una grupo
para las barras

    let barras = cuerpo.selectAll(".barra")
        .data (companias);

    //Añadir un rectángulo por cada compañía
    barras.enter().append("rect")
        .attr("height", escalaY.bandwidth())
        .attr("y", (d) => escalaY( d.Nombre ))
        .attr("width", (d) => escalaX( d.Rutas))
        .attr("fill", "steelblue")
        .on("mouseenter", function(c) { // <- a partir de aquí, nuevo
código

        ;//TODO 11: llamar a dibujarRutas con el ID de la
compañía

        d3.select(this)
            .attr("fill", ); //TODO 12: cambiar el color de la
barra a 'firebrick' para marcar la barra
        })
        .on("mouseleave", function(c) {

        ;//TODO 13:llamar a dibujarRutas con null para eliminar
las rutas en el mapa

        d3.select(this)
            .attr("fill", );//TODO 14: volver a poner el color de
la barra a 'steelblue'

        });

    }
}

```

Una vez completada es posible ver las rutas de cada una de las compañías.

**¿Qué compañías es tiene centradas todas sus rutas en Europa principalmente?**

**¿En qué país Shenzhen Airlines tiene mayor presencia?**

**¿Qué compañías no tienen rutas entre Europa y Norteamérica?**

**¿Qué compañías tienen rutas hasta Australia?**

**¿Dónde tiene Turkish Airlines su base de operaciones?**

#### 4) Añadir más interacción (opcional)

Esta parte es completamente opcional y consiste en añadir gestores de eventos para zoom y arrastrar (drag) con el ratón. Para ello hay que modificar la función “dibujarMapaBase”.

```
function dibujarMapaBase (contenedor , paises , proyeccion ){
    let path = d3.geoPath()
        .projection(proyeccion);

    contenedor.selectAll("path")
        .data(paises)
        .enter()
            .append("g")
            .append("path")
                .attr("d", path )
                .attr("stroke", "#ccc")
                .attr("fill", "#eee");

    contenedor.on("wheel.zoom",function(){

    })
        .call( d3.drag().on("drag", function(){

    }) );
}
```

#### 5) Añadir una llamada que informe de datos del aeropuerto (opcional).

Esta parte es opcional y consiste en añadir información de los aeropuertos mediante una llamada. Para ello hay que modificar la función “dibujarAeropuertos”.

```
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.data (aeropuertos)  
        .enter()  
        .append("circle")  
            .attr("r", "2px")  
            .attr("cx", d => proyeccion ( [d.Longitud ,  
d.Latitud]))[0] )  
            .attr("cy", d => proyeccion ( [d.Longitud ,  
d.Latitud]))[1] )  
            .attr("fill" , "steelblue" )  
            .on ("mouseenter", (d) => { })  
  
            .on ("mouseleave", d => { });  
  
}
```