

# Tutorial:

## Interfaz Web Modelación Lagrangiana

### 1. Propósito

En este tutorial revisaremos las instrucciones básicas para usar la interfaz web de modelación de deriva lagrangiana

1 <http://modelacion-biofisica.ifop.cl:8081/IFOPv6.0.0/>

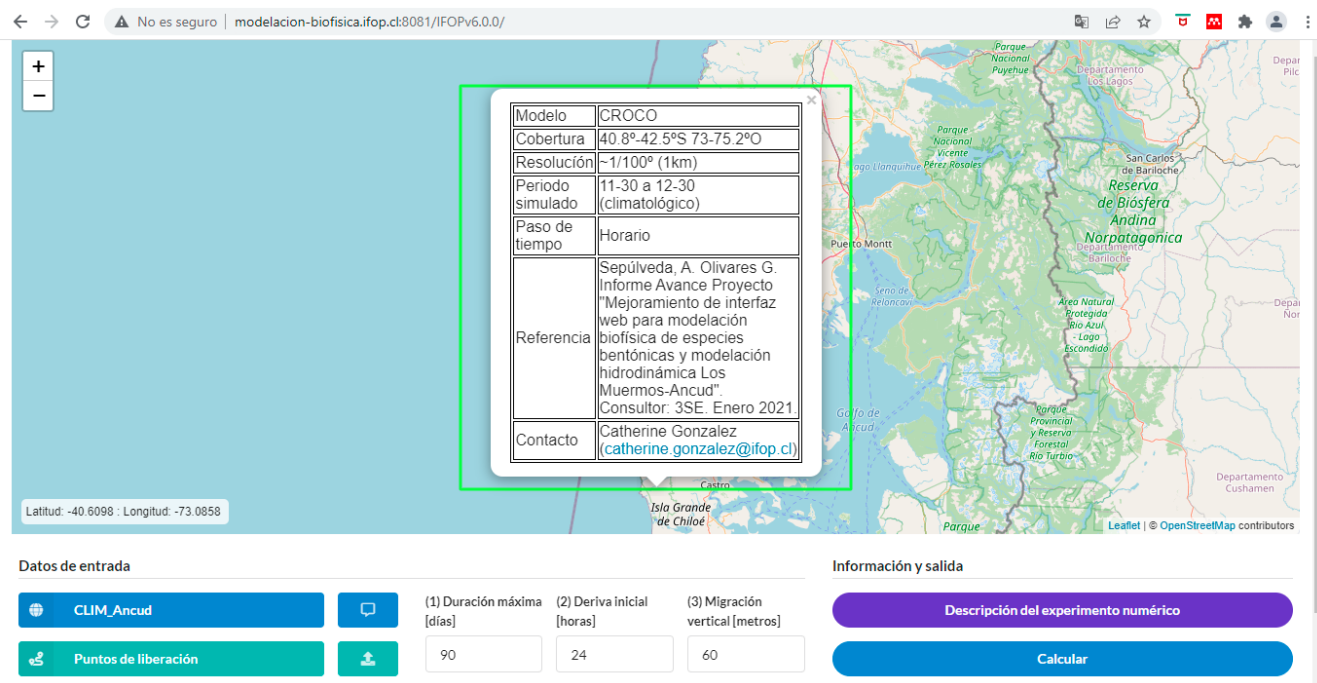


Figura 1: Interfaz Web IFOPv6.0.0

Los archivos que usaremos están disponibles en

1 <https://github.com/AndresSepulveda/CapacitacionInterfazIFOP>

## 2. Elementos Fundamentales

Los elementos fundamentales que se usan para un estudio de deriva lagrangiana son:

1. Resultados de un modelo hidrodinámico
2. Un listado de las AMERB a estudiar
3. Modelo IBM lagrangiano
4. Códigos computacionales de Teoría de Grafos
5. Códigos computacionales de análisis de trayectorias

El modelo hidrodinámico es usado para representar la circulación oceánica en una determinada zona y este es calculado *a priori*.

El modelo lagrangiano incorpora elementos como el momento de la liberación, la profundidad inicial, el rango de migración vertical, el tiempo de vida, y la respuesta al medio ambiente (temperatura y salinidad) de las larvas. Algunos de estos elementos pueden ser definidos al usar la interfaz y otros están fijos, si son característicos de alguna especie.

### 2.1. Modelo hidrodinámico

Un elemento clave en los estudios de modelación lagrangiana son los modelos numéricos de circulación oceánica. Los modelos numéricos entregan una representación de variables físicas relevantes para la dispersión de la fase pelágica de las larvas, principalmente las corrientes, la temperatura, y salinidad del agua.

Es importante que las simulaciones numéricas usadas representen los elementos claves en la zona de estudio, tales como la marea, la descarga de ríos, la batimetría, etc. En el caso de la marea es necesario que las salidas de los modelos estén grabadas cada una hora.

La interfaz web del IFOP incorpora en este momento salidas de modelos numéricos ROMS AGRIF, CROCO, y FVCOM. El que un cierto modelo numérico pueda ser usado en esta interfaz depende principalmente del modelo lagrangiano que usa. Es necesario un cierto nivel de preprocesamiento de las salidas de los modelos para que estos puedan ser usados por el modelo lagrangiano.

### 2.2. Modelo lagrangiano

Los modelos lagrangianos son códigos numéricos que se usan para estudiar el transporte virtual de larvas producto de las corrientes calculadas por un modelo numérico del océano.

El modelo lagrangiano más básico asume un comportamiento pasivo de la larva, sin movimiento vertical propio, cambios en la densidad o respuesta al medio. Este tipo de simulación base es útil para describir los patrones de circulación existentes en una zona e identificar estructuras físicas relevantes como filamentos, remolinos, o corrientes costeras.

La interfaz web del IFOP usa el código lagrangiano **OpenDrift** puesto que este está desarrollado en el lenguaje de programación Python y que además tiene un grupo de apoyo bastante activo, por lo que las dudas que surgen al implementar cambios los códigos son, en general, rápidamente resueltas.

Esta interfaz entrega la posición y profundidad de las partículas cada hora. Esta información es útil para entender la trayectoria que estas recorren, así como la temperatura y salinidad a la que se ven expuestas en su recorrido. Sin embargo, la mayoría de los análisis se basan únicamente en la posición inicial y la posición final de las partículas, si existir hasta ahora un análisis profundo de lo vivido por las partículas durante su trayectoria.

Dado que estamos usando corrientes ya calculadas por un modelo, la trayectoria de una partícula liberada en un determinado momento del espacio-tiempo (x,y,z,t) será siempre la misma. Es posible incluir el código OpenDrift "perturbaciones" a la posición de liberación, por ejemplo liberando un número determinado de partículas alrededor del centroide definido para la AMERB.

También es posible liberar partículas en toda el área de la AMERB si esta es especificada a través de un archivo tipo *Shapefile*, y no de una posición latitud/longitud. Esta opción no está aún implementada.

El término de vida de las partículas es definido a través del parámetro de tiempo de vida. Sin embargo se puede definir el tiempo de vida (y el crecimiento) a través de el número de **grados C** de la masa de agua que transporta la larva. Así una larva en aguas más frías duraría más tiempo (medido en horas o días) que una larva que se desplace por aguas de mayor temperatura. Esta opción no está aún implementada.

La definición de punto final de la trayectoria de una partícula presenta muchos desafíos. Por ejemplo el código OpenDrift permite definir diversos resultados en el caso que una partícula sea desplazada en dirección a la costa, terminado tierra adentro, después de un paso de tiempo. Se puede definir que la partícula se queda en tierra (opción *stranded*) o que regresa a su posición anterior (opción *previous*). En este último caso la posición final de la partícula es la latitud/longitud del último paso de tiempo en que estuvo activa.

Si guardamos no sólo las posiciones iniciales y finales, sino toda la trayectoria de cada partícula, es posible usar criterios como el de Garavelli et al (2014) en el cual se consideró que las partículas habían llegado a una cierta AMERB si se encontraban cerca de esta en los últimos 20 días de su trayectoria.

## 2.3. Teoría de Grafos

Los códigos R desarrollados por el Dr. Andrés Ospina-Alvarez representan un análisis avanzado de la relación entre los distintos sitios (AMERB) de liberación de partículas.

Este tipo de análisis considera no solo la conexión entre dos sitios sino la importancia de una AMERB en el contexto de la red de conexiones.

## 2.4. Análisis de Trayectorias

El análisis de trayectorias está basado principalmente en el estudio de la posición inicial y la posición final de las partículas.

Primero se calcula la distancia entre estos dos puntos, analizando la distribución de esos valores y si existe una relación con la latitud de inicio. El mismo cálculo se puede hacer para la duración del viaje.

En todo caso se suele sólo considera las trayectorias *“exitosas”*, es decir aquellas que llegaron *“cerca”*, es decir dentro de una distancia predefinida, de los puntos de interés. Comúnmente se consideran los mismos puntos de liberación como posibles puntos de llegada. Para los casos en que las partículas llegan al mismo punto de salida, se habla de **autoreclutamiento**.

Mediante el conteo de cuantas partículas de una cierta AMERB llegaron a otra de las AMERB, se construye la llamada **Matriz de Conectividad Potencial**. La diagonal de esta matriz es nuestro indicador de autoreclutamiento.

## 3. Detalles avanzados

Es importante comparar simulaciones considerando

1. Modelos de circulación oceánica climatológicos vs interanuales
2. Variar los rangos de migración vertical o la profundidad inicial de liberación
3. Usar simulaciones con distinta resolución (tamaño de pixel)
4. Estudiar el impacto del inicio de la liberación de larvas
5. Los archivos de salida de OpenDrift están, por defecto, en formato NetCDF.
6. Se está modificando la interfaz para que los archivos de salida sean en formato CSV.
7. Es muy importante estudiar la consistencia o convergencia de los resultados a la cantidad de partículas liberadas desde cada punto. Liberar 8760 partículas desde un punto, equivale a libera una partícula cada hora, por un año.
8. Sin embargo si consideramos del orden de cientos de puntos de liberación, es fácil llegar a calcular la trayectoria de millones de partículas, lo cual puede tener un costo computacional alto, en términos de tiempo de cálculo y de archivos de salida generados (del orden de decenas de GB), lo cual dificulta la realización de experimentos y su posterior análisis.

## 4. Conclusión

En este tutorial revisamos los elementos que componen un estudio de deriva lagrangiana a través de la interfaz web desarrollada para el IFOP por 3SE.

Para más información:  
Andrés Sepúlveda ([andres.sepulveda@gmail.com](mailto:andres.sepulveda@gmail.com))

## 5. Referencias

Dagestad, K.-F., Röhrs, J., Breivik, Ø., and Ådlandsvik, B.: OpenDrift v1.0: a generic framework for trajectory modelling, *Geosci. Model Dev.*, 11, 1405-1420, <https://doi.org/10.5194/gmd-11-1405-2018>, 2018.

Debreu, L., P. Marchesiello, P. Penven, and G. Cambon, 2012: Two-way nesting in split-explicit ocean models: algorithms, implementation and validation. *Ocean Modelling*, 49-50, 1-21.

Garavelli, L., Kaplan, D. M., Colas, F., Stotz, W., Yannicelli, B., & Lett, C. (2014). Identifying appropriate spatial scales for marine conservation and management using a larval dispersal model: the case of *Concholepas concholepas* (loco) in Chile. *Progress in Oceanography*, 124, 42-53.

Ospina-Alvarez, Andrés, S. de Juan, Josep. Alós, G. Basterretxea, A. Alonso-Fernández, G. Follana-Berná, M. Palmer, I A. Catalán. MPA network design based on graph theory and emergent properties of larval dispersal *MARINE ECOLOGY PROGRESS SERIES* , Vol. 650. 309-326 (2020) <https://doi.org/10.3354/meps13399>

## 6. Enlaces útiles

### 6.1. Modelo Oceánico

<https://www.croco-ocean.org/>  
<http://fvcom.smast.umassd.edu/>

### 6.2. Modelo Lagrangiano

<https://github.com/OpenDrift/opendrift>  
<https://opendrift.github.io/index.html>

## Apéndice A: Pasos Resumidos

1. Escoger modelo y área de estudio
2. Escoger puntos de liberación
  - a) Usar puntos preseleccionados
  - b) Subir un archivo de puntos
3. Definir fechas de liberación
4. Configurar IBM
  - a) Definir duración máxima
  - b) Definir periodo de deriva inicial
  - c) Definir rango de migración vertical
  - d) Definir duración de fase bentónica
5. Apretar botón “Calcular”
6. Esperar....
  - a) Bajar archivo NetCDF IFOPv6.0.0
  - b) Bajar archivo CSV IFOPv6.1.0
7. Extraer puntos iniciales y finales (.CSV)
  - a) R  
`https://github.com/AndresSepulveda/CapacitacionInterfazIFOP/blob/main/NetCDF\_InicialFinal.R`
  - b) Octave  
`https://github.com/AndresSepulveda/CapacitacionInterfazIFOP/blob/main/NetCDF\_InicialFinal\_MatrizConectividad\_TablaConectividad.m`
8. Graficar histogramas, calcular distancias, dirección de viaje, crear tablas de conectividad, matrices de conectividad potencial, histogramas de autoconectividad, que son producidos por el siguiente archivo:  
`https://github.com/AndresSepulveda/CapacitacionInterfazIFOP/blob/main/NetCDF\_InicialFinal\_MatrizConectividad\_TablaConectividad.m`
9. Analizar mediante Teoría de Grafos
  - a) Crear gráficos
  - b) Listar Nodos y Conexiones prioritarias  
`https://github.com/AndresSepulveda/CapacitacionInterfazIFOP/tree/main/output\_analysis\_graph\_theory`
10. Crear Diagrama de Sankey/Alluvialflow a partir de la matriz de conectividad potencial.  
`https://github.com/AndresSepulveda/CapacitacionInterfazIFOP/blob/main/do\_sankey\_diagram.m`  
`https://github.com/AndresSepulveda/CapacitacionInterfazIFOP/blob/main/test\_alluvialflow.m`

## Apéndice B: Tipos de Gráficos

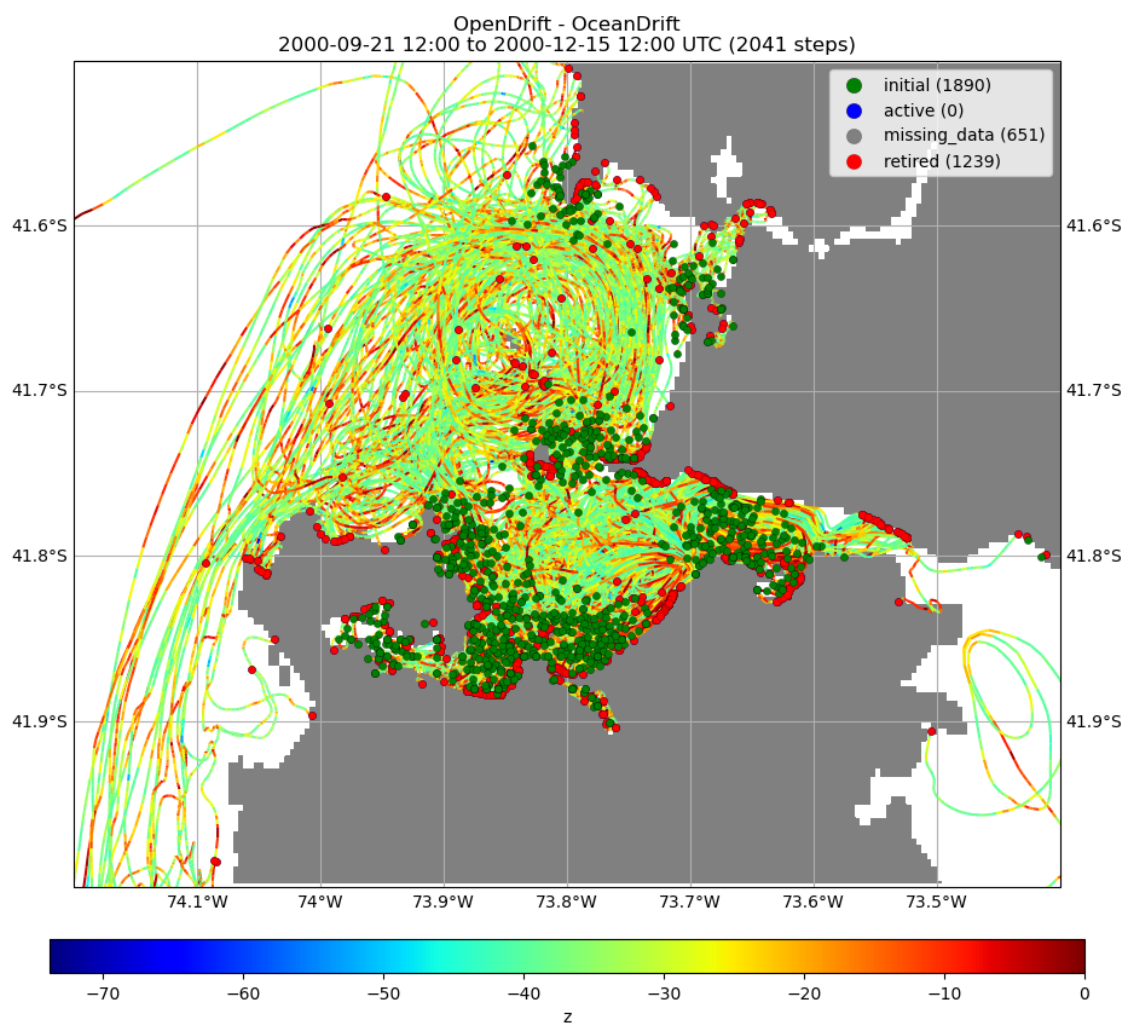


Figura 2: Trayectoria de Partículas

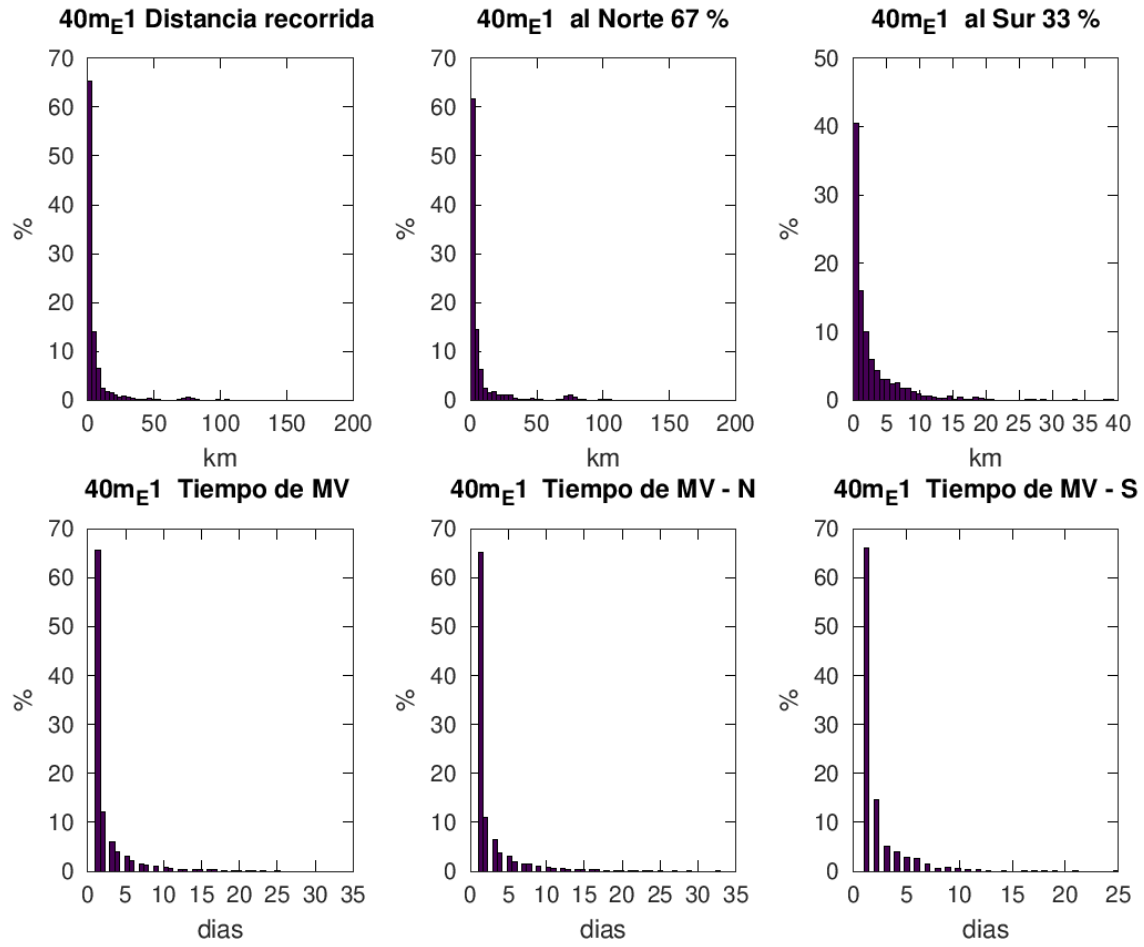


Figura 3: Distancia, tiempo, y dirección de viaje



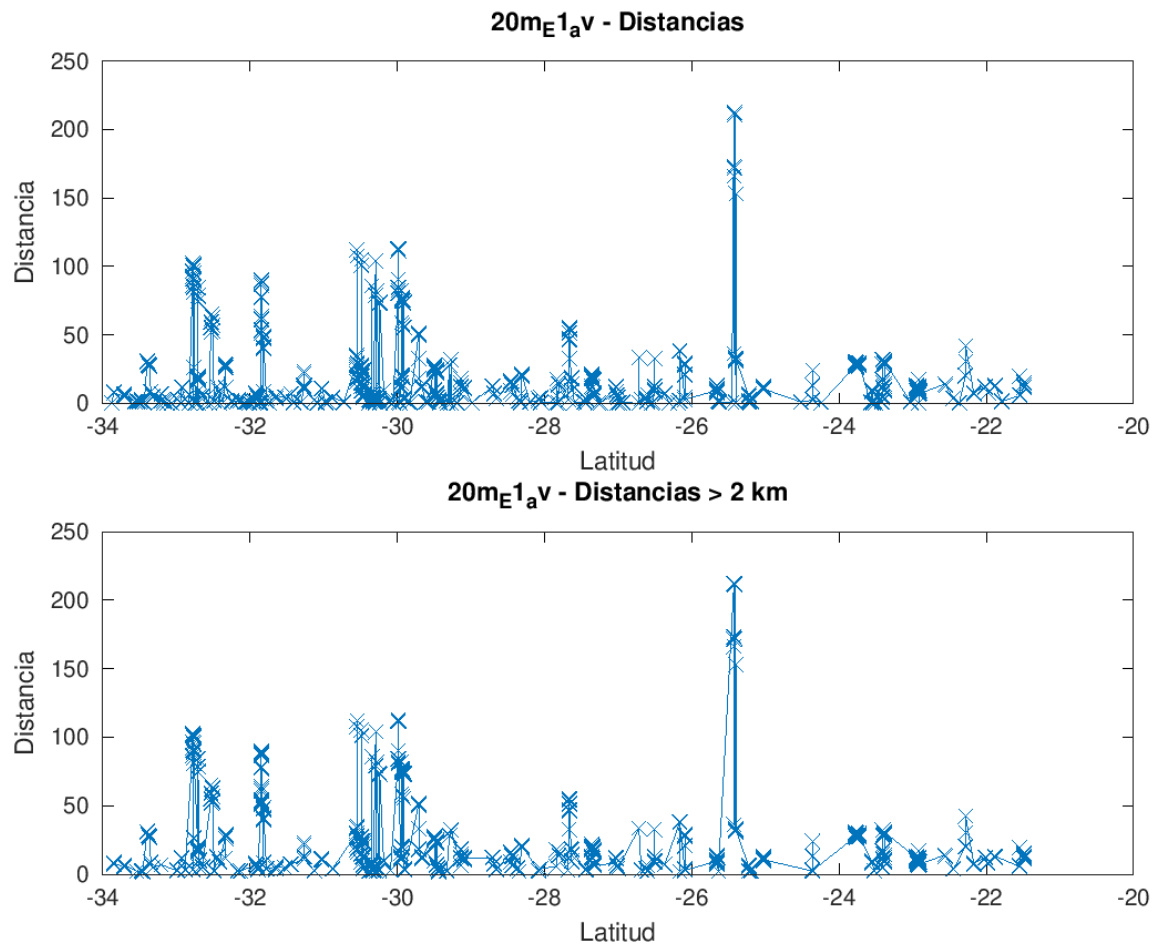


Figura 4: Dependencia latitudinal de la distancia de viaje

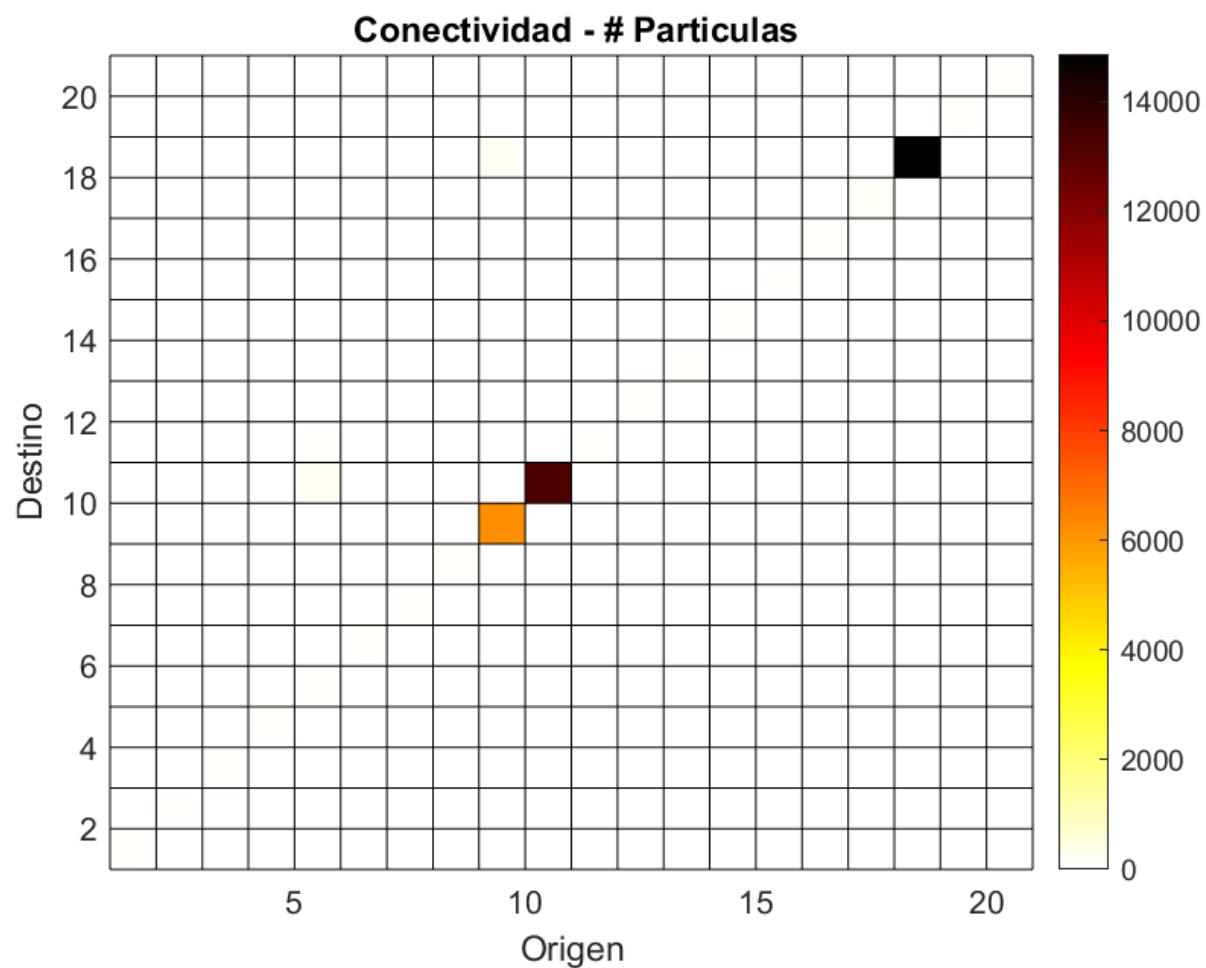


Figura 5: Matriz de conectividad potencial: Cantidad de Partículas

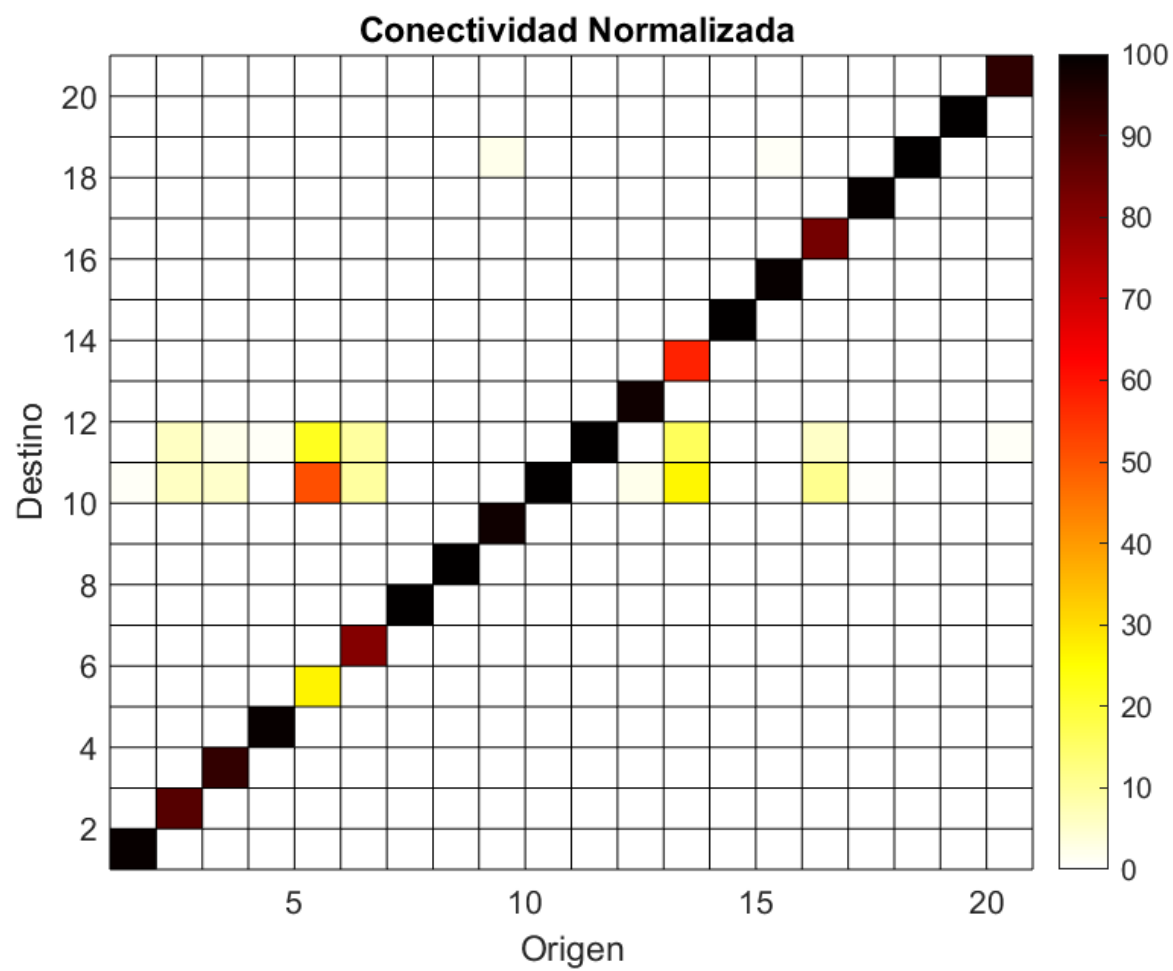


Figura 6: Matriz de conectividad potencial: Normalizado

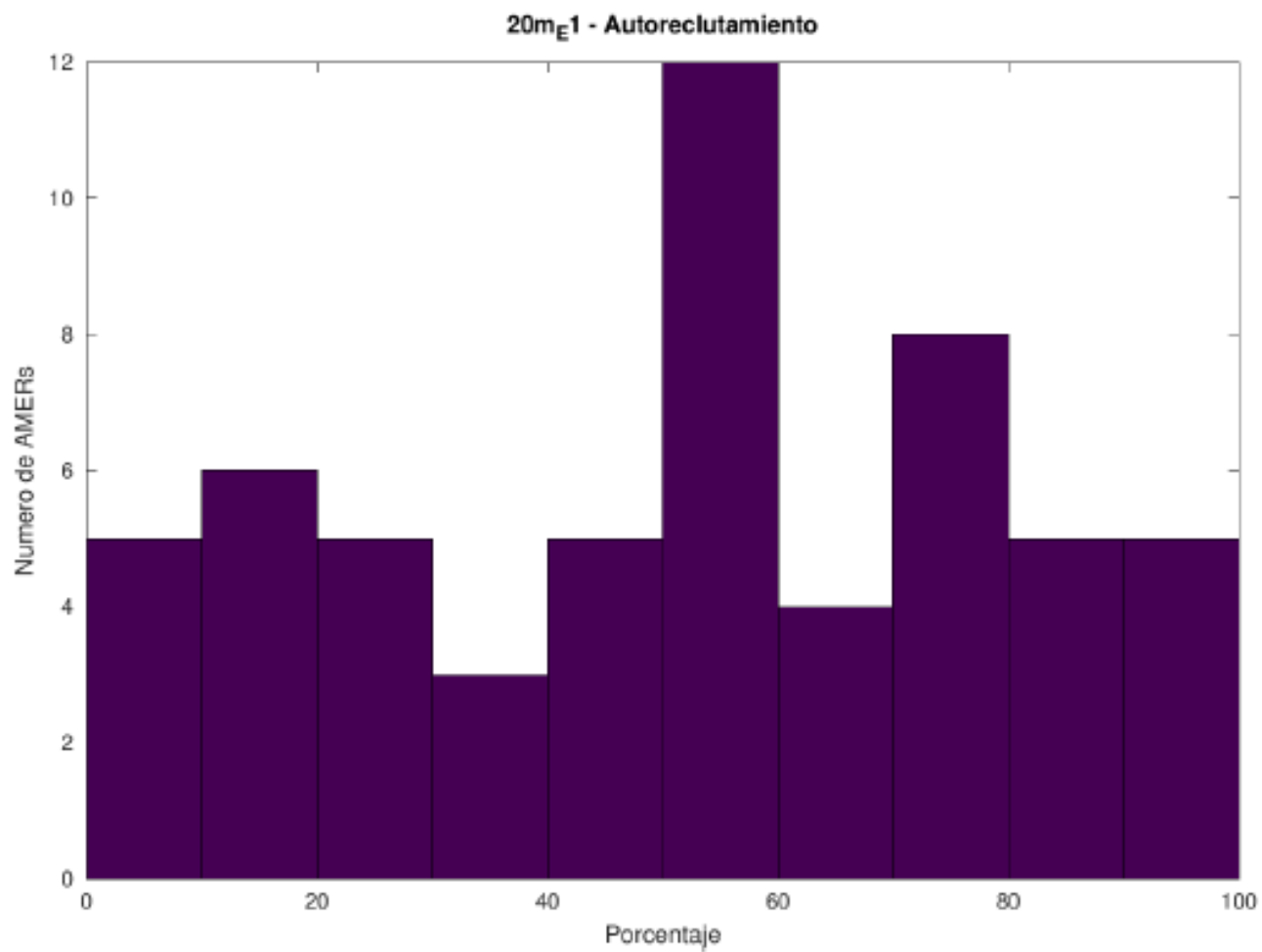


Figura 7: Histograma de Autoreclutamiento

<b>AMER</b>	<b>1ra</b>	<b>%</b>	<b>2da</b>	<b>%</b>	<b>3ra</b>	<b>%</b>
1	1	79.1	40	11.1	31	5.9
2	2	59.1	7	22.7	33	9.1
3	3	64.1	1	23.1	31	12.8
4	4	96.7	33	3.3	1	0.0
5	5	82.4	6	5.9	6	5.9
6	6	48.4	5	25.8	24	16.1
7	7	53.8	5	15.4	5	15.4
8	8	88.1	19	11.9	1	0.0
9	9	27.7	8	23.1	54	15.4
10	10	96.3	22	1.9	22	1.9
11	7	58.3	11	16.7	3	8.3
12	12	57.4	13	33.3	40	5.6
13	31	51.0	13	38.8	40	10.2
14	7	62.1	14	27.6	33	6.9
15	15	89.2	52	10.8	1	0.0
16	16	54.8	25	23.8	17	7.1
17	17	76.3	25	13.6	27	10.2
18	18	79.2	56	16.7	55	4.2
19	19	86.9	8	9.8	35	1.6
20	20	61.5	38	23.1	24	11.5
21	21	25.9	21	25.9	1	15.5
22	30	65.9	20	24.4	22	7.3

Figura 8: Tabla de primeras conectividades

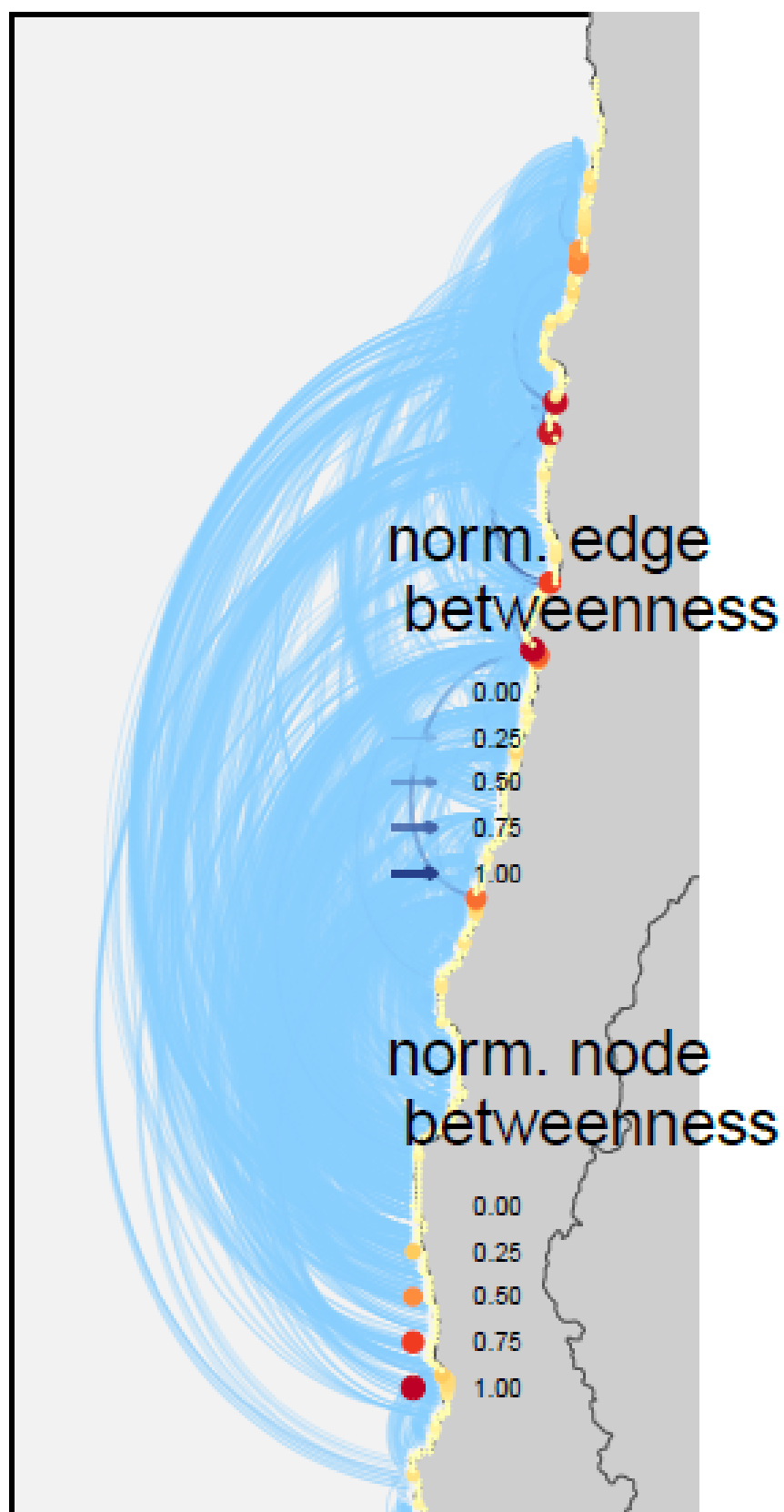


Figura 9: Teoría de Gráfos. Gráfico Betweenness

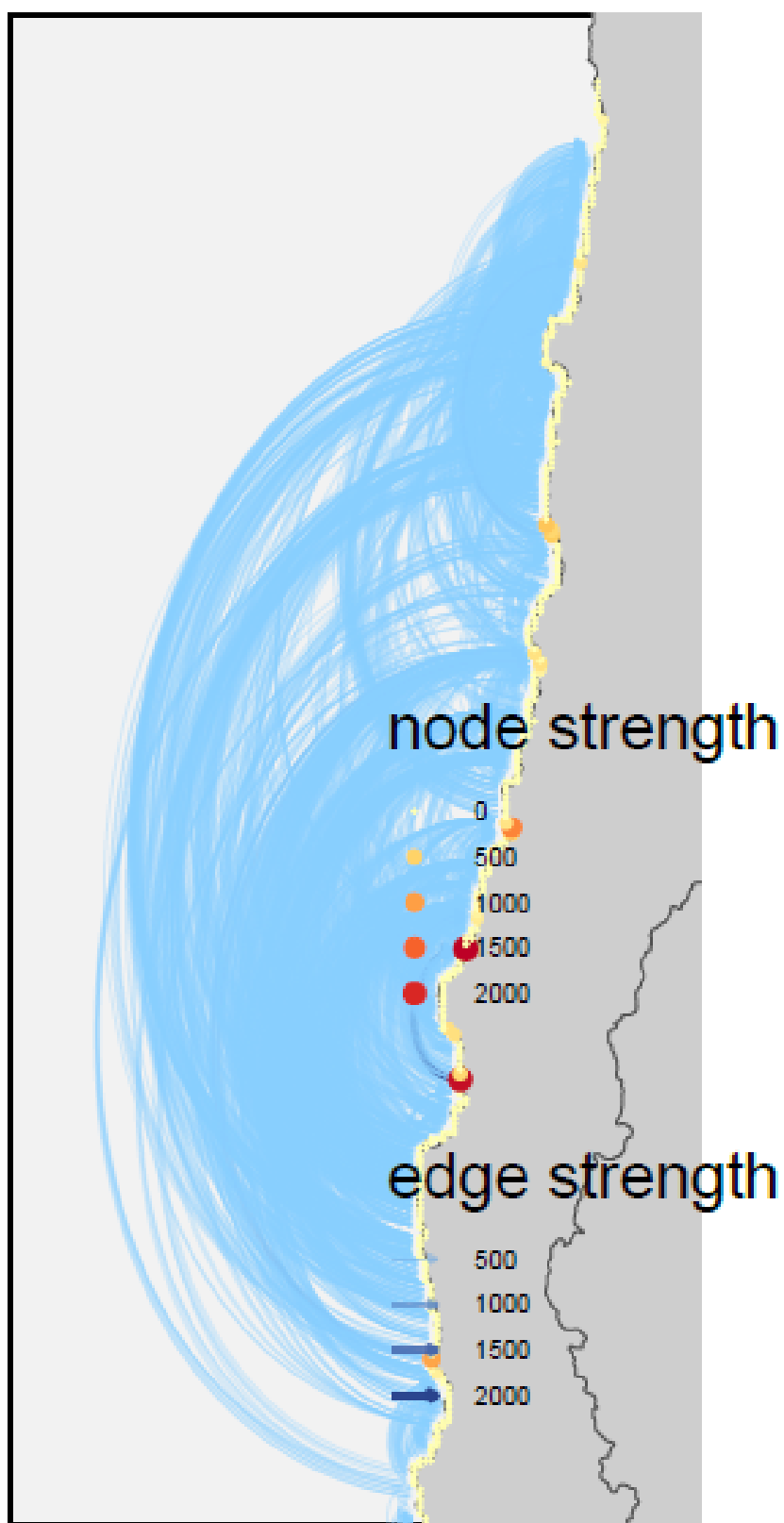


Figura 10: Teoría de Gráfos. Gráfico Strength

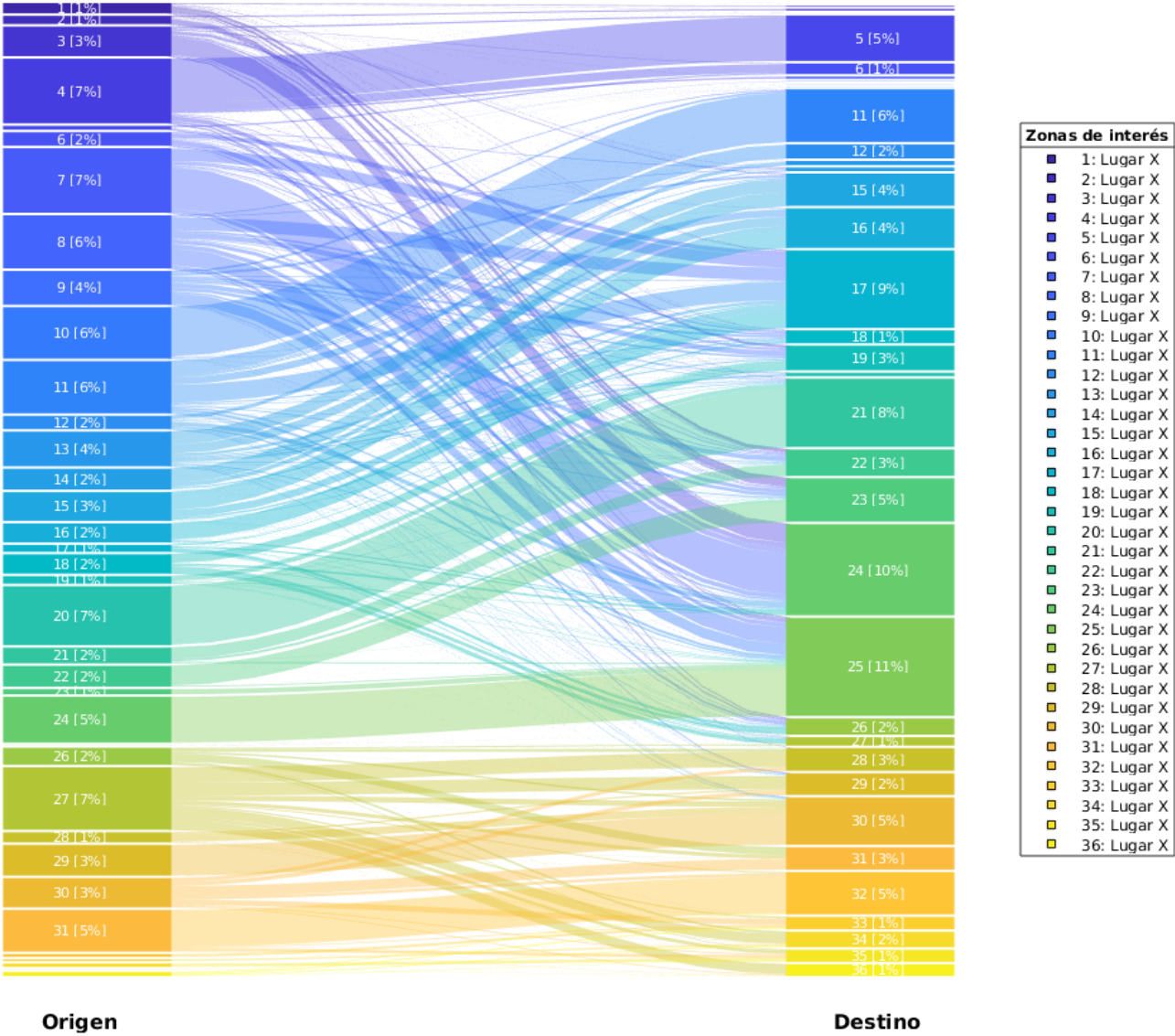


Figura 11: Diagrama Sankey