



Universidad Nacional Autónoma de México
Escuela Nacional de Estudios Superiores Unidad Juriquilla

Licenciatura en Neurociencias

Proyecto

Modelos computacionales

Alumna:
Mercedes Arantza Juárez Cerecedo

Juriquilla, Querétaro, 19 de noviembre de 2025

- Resumen

En este reporte se analizó datos de una dataset de un experimento donde medía en una tarea “three-armed bandit task” marcadores electrofisiológicos para personas bebedoras de alto riesgo. Tras el análisis se observó que existe menos conexiones en el grupo riesgo, y más en el grupo control, pero ambos mantienen un Hub CF1, lo que refiere a que hay integración funcional por ambos grupos

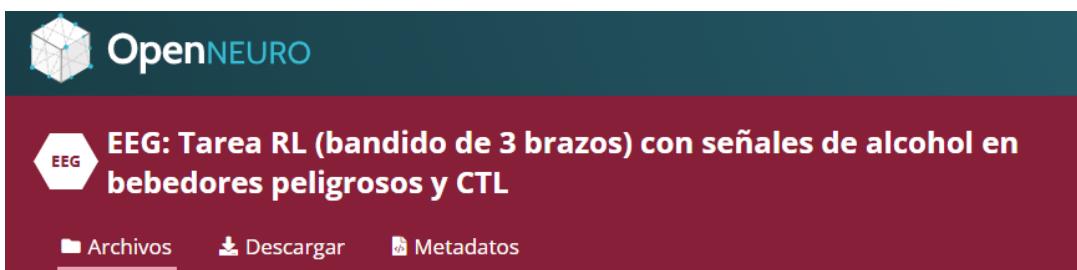
- Introducción.

El experimento trato de realizar una tarea computarizada llamada “three-armed bandit task” con estímulos de bebidas alcohólicas y no alcohólicas, esto con el objetivo de observar los marcadores electrofisiológicos de la exploración atípica de personas bebedoras de alto riesgo. Se realizaron 350 ensayos en 53 participantes divididos en, 27 participantes para bebedores de riesgo y 26 participantes en el grupo control. Se utilizó un electroencefalograma (EEG) con 64 canales para poder medir la actividad.

En la tarea se les pidió a los participantes que escogieran entre 3 imágenes que estaban asociadas a una recompensa (+1 punto o nada), y tenían 1500 ms para seleccionar las imágenes, presionando un botón dependiendo de donde apareciera (arriba, abajo, o en el centro). En ocasiones se metían imágenes de bebidas tanto alcohólicas como no alcohólicas.

El Dataset seleccionado fue encontrado en OpenNeuro con un número de acceso de ds005907. Los archivos del EEG se encontraban en .set, al igual que se encuentra un archivo .TSV con las coordenadas de los electrodos, que se utilizaron para cada sujeto. Ambos archivos fueron necesarios para la realización del análisis.

- Metodología y resultados
 1. La metodología que se utilizó fue encontrar el DataSet, y descargar los archivos necesarios.



2. Entramos a Python, donde importamos las bibliotecas necesarias para poder abrir y

realizar las actividades necesarias para el análisis; mne fue una de ellas, se utilizó para poder cargar el archivo. set de manera correcta. Convertí el archivo .TSV a .Xlsx

3. Definimos las rutas donde se encontraban los archivos .set para poder cargarlos, al igual que el archivo .TSV.

```
import os
import numpy as np
import pandas as pd
import mne
import networkx as nx
import matplotlib.pyplot as plt
from scipy import stats
from scipy.signal import butter, filtfilt
```

[2]:

```
ruta_datos=r"C:\Users\ecere\Documents\GitHub\Neurociencias-2026-1\S06_proyecto\projecto"
archivo_par = "{sub}_task-ThreeArmedBandit_eeg.set"
participantes_excel =r"C:\Users\ecere\Documents\GitHub\Neurociencias-2026-1\S06_proyecto"
eeg_coords=r"C:\Users\ecere\Documents\GitHub\Neurociencias-2026-1\S06_proyecto\projecto"
```

4. Se colocó 4 parámetros (BANDA, BANDPASS, TH, PORC) que se usaron para poder analizar correctamente la actividad cerebral y construir una red de conectividad. BANDASS se utilizó para medir Alpha (8.0 y 12.0), BANDPASS para limpiar señal que eran o muy bajas o muy altas, al igual que un umbral para cápsula, la cual permitió saber entre qué canales existe una conexión fuerte.
5. Cargamos la información de los participantes, para conocer en qué grupo se encontraban si en control o en el grupo de riesgo.

```
BANDA = (8.0, 12.0)      # alpha (fmin, fmax)
BANDPASS = (1.0, 40.0)    # filtro previo (low, high)
TH = 0.3                  # umbral para cápsula
PORC = 0.6
```

```
[6]: participants = pd.read_excel(participantes_excel)

# Limpiar nombres de columnas
participants.columns = participants.columns.str.strip()

# obtener listas de participantes por grupo
grupo_control = participants[participants["GROUP"]=="CTRL"]["participant_id"].tolist()
grupo_riesgo = participants[participants["GROUP"]=="AUD"]["participant_id"].tolist()

print("CTRL:", grupo_control)
print("AUD:", grupo_riesgo)
```

CTRL: ['sub-001', 'sub-002', 'sub-003', 'sub-004', 'sub-005', 'sub-006', 'sub-018', 'sub-019', 'sub-035', 'sub-036', 'sub-038', 'sub-039', 'sub-040', 'sub-041', 'sub-042', 'sub-043', 'sub-044', 'sub-045', 'sub-046', 'sub-047', 'sub-048', 'sub-049', 'sub-050', 'sub-051', 'sub-052', 'sub-053']
AUD: ['sub-007', 'sub-008', 'sub-009', 'sub-010', 'sub-011', 'sub-012', 'sub-013', 'sub-014', 'sub-015', 'sub-016', 'sub-017', 'sub-020', 'sub-021', 'sub-022', 'sub-023', 'sub-024', 'sub-025', 'sub-026', 'sub-027', 'sub-028', 'sub-029', 'sub-030', 'sub-031', 'sub-032', 'sub-033', 'sub-034', 'sub-037']

6. Se automatizó la carga de los archivos .set, utilizando mne, creando y ocupando la función "cargar_raw", permitiendo de esta manera cargar los datos crudos de cada sujeto de manera sencilla. Verifica que las señales y los canales hayan sido importados de manera eficiente
7. Con la función "Bandpass" se filtró la señal dejando solo la frecuencia para Alpha (8-12) y con otra función llamada "filtrar_banda (raw, low, high)" va a tomar una señal de EEG y le va a aplicar un filtro entre dos frecuencias (low y high), eliminando ruido de baja y alta frecuencia. Utilizada para analizar solo bandas Alfa.

8. Se calculó una matriz de conectividad funcional entre canales de EEG usando una correlación de Pearson con la función de "matriz_conectividad (datos, canales)". Nos

ayuda a saber cómo interactúan diferentes áreas cerebrales dentro de banda. Necesario para construir los grafos. Como resultado se obtendrá una lista de matrices de conectividad y con la función normalizar_matrices (lista_mats) se normalizó cada una para que los valores estuvieran entre 0 y 1. Evita que un sujeto domine estadísticas del grupo.

```
#Filtrado dependiendo de la banda
def bandpass(x, fs, low, high, order=4):
    b, a = butter(order, [low/(fs/2), high/(fs/2)], btype='band')
    return filtfilt(b, a, x)

def filtrar_banda(raw, low, high):
    data = raw.get_data()          # canales x muestras
    fs = raw.info['sfreq']

    filtrado = np.array([bandpass(ch, fs, low, high) for ch in data])
    return filtrado, raw.ch_names

def matriz_conectividad(datos, canales):
    N = len(canales)
    M = np.zeros((N, N))

    for i in range(N):
        for j in range(N):
            M[i, j] = np.corrcoef(datos[i], datos[j])[0, 1]

    return pd.DataFrame(M, index=canales, columns=canales)

def normalizar_matrices(lista_mats):
    return [(m - m.values.min()) / (m.values.max() - m.values.min()) for m in lista_mats]
```

9. Se realizó otra función llamada "cápsula", la cual nos permitió generar una matriz final con conexiones que aparecen consistentemente. Calcula el número mínimo de sujetos requeridos según el porcentaje.
 10. Con función "procesar_grupo(lista_sujetos)" se obtuvo matrices de conectividad normalizadas por sujeto, alineadas, y capsulada. Y también Data Frames, los cuales nos ayudaron para realizar los estadísticos.

```
def capsula(lista_dfs, TH, porcentaje_):
    stack = np.stack([df.values for df in lista_dfs])
    above = stack > TH

    N = len(lista_dfs)
    count_above = above.sum(axis=0)
    min_requerido = int(np.ceil(porcentaje_* N))

    result_bool = count_above >= min_requerido

    return pd.DataFrame(
        result_bool.astype(int),
        index=lista_dfs[0].index,
        columns=lista_dfs[0].columns
    )
```

```
def procesar_grupo(lista_sujetos):
    matrices_norm = []
    canales_ref = None

    for idx, sub in enumerate(lista_sujetos):
        print("Procesando:", sub)

        raw = cargar_raw(sub)

        df_raw = raw.to_dataframe()
        display(df_raw.head())

        datos_filt, canales = filtrar_banda(raw, BANDA[0], BANDA[1])
        df_filt = pd.DataFrame(datos_filt.T, columns=canales)
        display(df_filt.head())

        mat = matriz_conectividad(datos_filt, canales)
        mat_norm = normalizar_matrices([mat])[0]

        if idx == 0:
            canales_ref = set(mat_norm.index)
        else:
            # intersección: solo dejamos canales comunes
            canales_ref = canales_ref.intersection(mat_norm.index)

        matrices_norm.append(mat_norm)

    canales_ref = list(canales_ref)

    matrices_norm_iguales = [
        m.loc[canales_ref, canales_ref] for m in matrices_norm
    ]

    matriz_capsulada = capsula(matrices_norm_iguales, TH, PORC)

    return matrices_norm_iguales, matriz_capsulada
```

11. Posteriormente se calculó los estadísticos (th25, th50, th75, promedio, varianza) de ambos grupos.

GRUPO CONTROL

```

: th25_c = []
th50_c = []
th75_c = []
promedio_c = []
varianza_c = []
mu_sigma_c = []

for sub in mats_ctrl:
    data = sub.stack().values

    th25_c.append(np.percentile(data, 25))
    th50_c.append(np.percentile(data, 50))
    th75_c.append(np.percentile(data, 75))
    promedio_c.append(np.mean(data))
    varianza_c.append(np.var(data))
    mu_sigma_c.append(np.mean(data) + np.var(data))

: estadisticos_CTRL = calcular_estadisticos(
    th25_c, th50_c, th75_c, promedio_c, mu_sigma_c
)
print(estadisticos_CTRL)

      th25     th50     th75   promedio   mu_sigma
min  0.230243  0.490663  0.785525  0.507889  0.590315
max  0.609714  0.749537  0.89299   0.741859  0.770278
mediana  0.380179  0.586376  0.823351  0.590395  0.656191
cosa_rara  0.055834  0.053069  0.029211  0.036274  0.031099

```

GRUPO RIESGO

```

: th25_r = []
th50_r = []
th75_r = []
promedio_r = []
varianza_r = []
mu_sigma_r = []

for sub in mats_riesgo:
    data = sub.stack().values

    th25_r.append(np.percentile(data, 25))
    th50_r.append(np.percentile(data, 50))
    th75_r.append(np.percentile(data, 75))
    promedio_r.append(np.mean(data))
    varianza_r.append(np.var(data))
    mu_sigma_r.append(np.mean(data) + np.var(data))

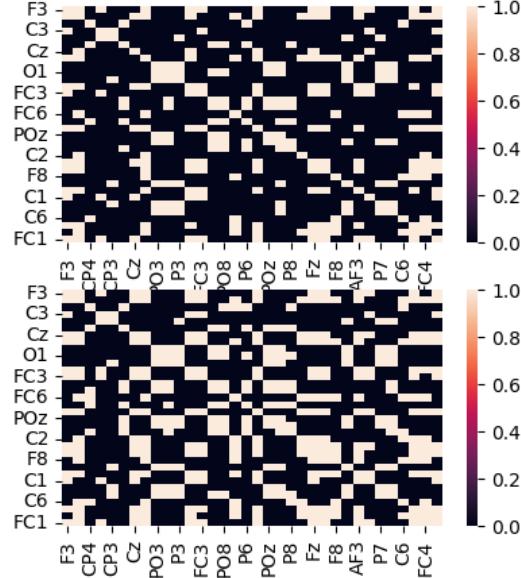
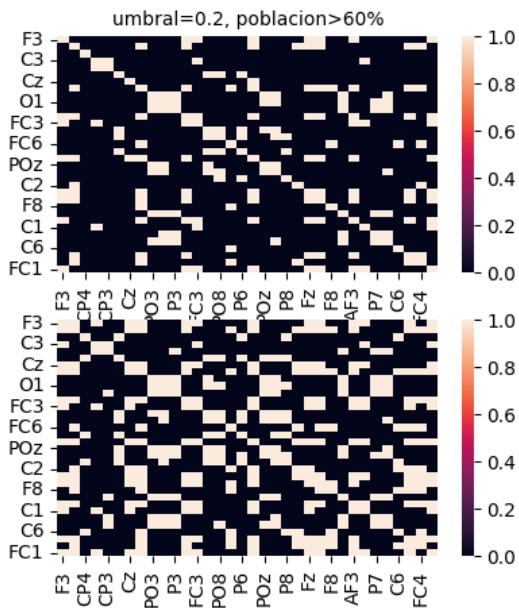
: estadisticos_RISK = calcular_estadisticos(
    th25_r, th50_r, th75_r, promedio_r, mu_sigma_r
)
print(estadisticos_RISK)

      th25     th50     th75   promedio   mu_sigma
min  0.325091  0.626811  0.828894  0.58418  0.660932
max  0.664197  0.851419  0.931241  0.77473  0.819259
mediana  0.494858  0.753019  0.89838  0.694273  0.749384
cosa_rara  0.048117  0.02571  0.013306  0.034848  0.025337

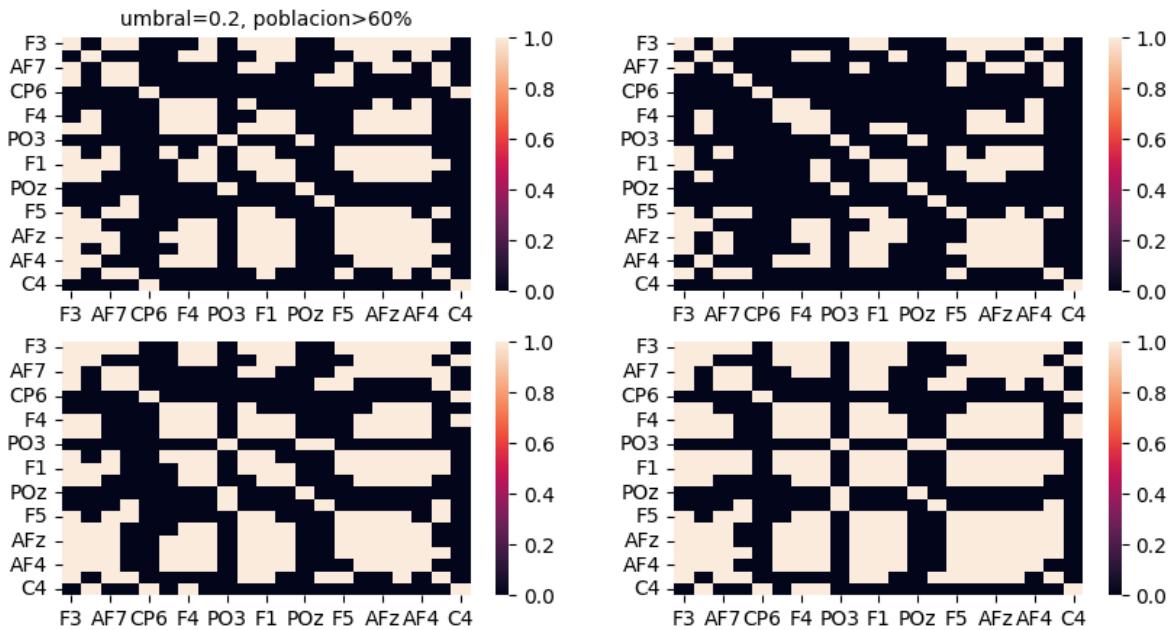
```

12. Se generaron cuatro matrices utilizando función "capsula", variando entre dos parámetros (TH y porcentaje) y se compararon con mapas de calor para observar patrones de conectividad. Se utilizaron esos umbrales porque supuse las conexiones iban a ser fuertes y consistentes.

GRUPO CONTROL



GRUPO RIESGO

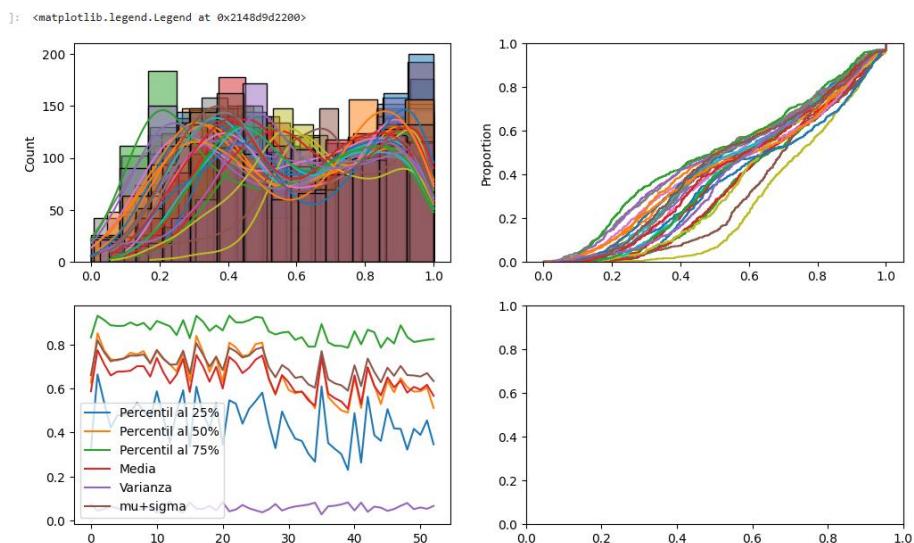


13. Se crearon grafos 2D, se convirtió de dataframe a grafo y se realizó grafos de comunidad, para ver las redes más conectadas y el punto que tiene más conexiones (Hub).

RESULTADOS

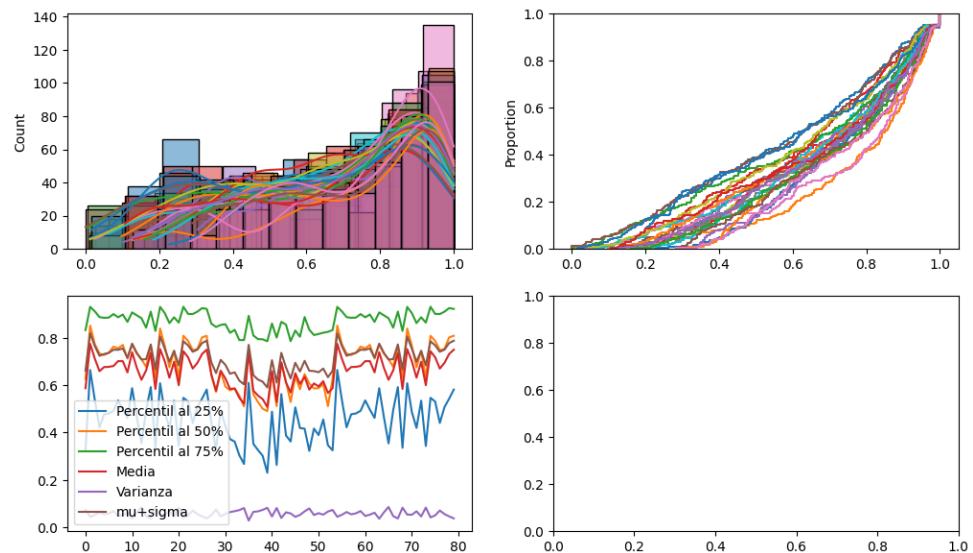
- Estadísticos de percentiles 25, 50 y 75, promedio, varianza y mu_sigma

GRUPO CONTROL



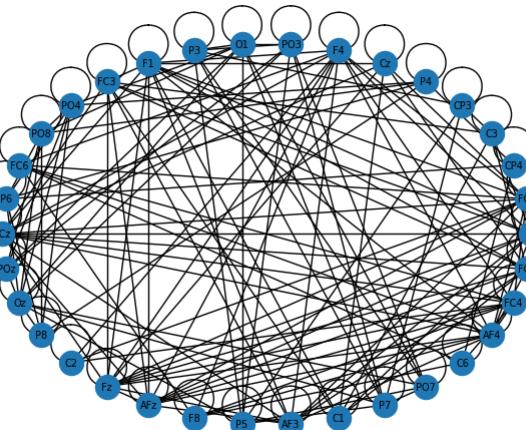
GRUPO DE RIESGO

<matplotlib.legend.Legend at 0x214b550a020>



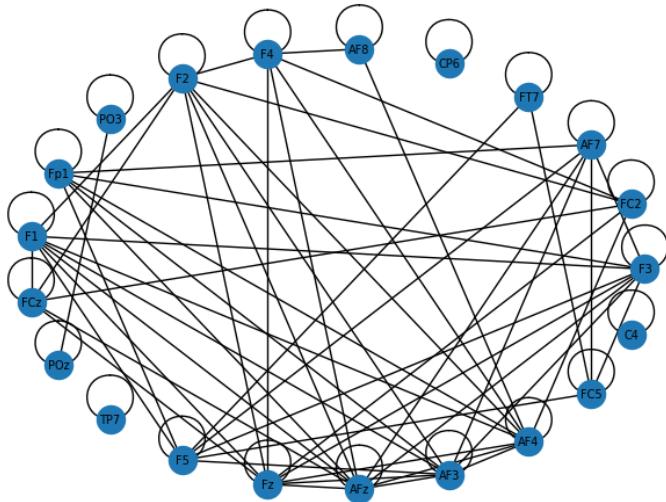
Grafos 2D

GRUPO CONTROL



Cada círculo en el gráfico representa una ubicación específica de un electrodo de EEG. Conectividad distribuida, sin embargo, no se ven muchas conexiones.

GRUPO RIESGO

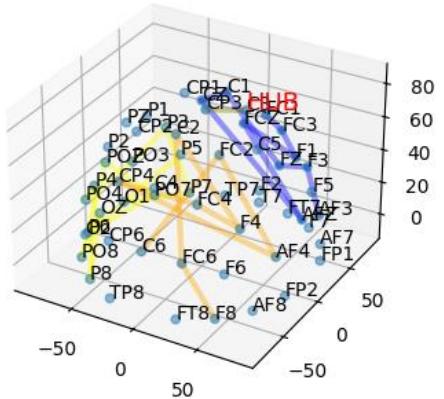


Poca conectividad, tanto que hay electrodos que no tienen conexiones.

Se observa más conexiones en el grupo control que en el de grupo de riesgo.

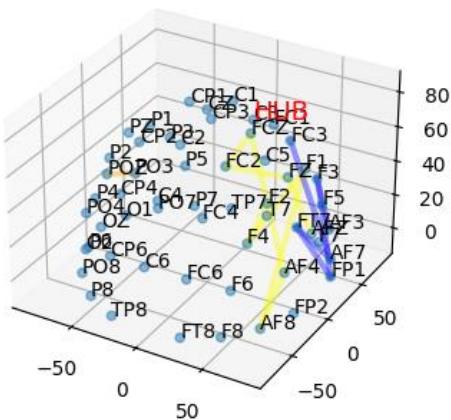
Grafo 3D y de comunidades

GRUPO CONTROL



En el grupo control se observan 3 comunidades con más conexiones y el hub se encuentra en FC1

GRUPO RIESGO



En el grupo riesgo se observan 2 comunidades con más conexiones y un hub en FC1

Discusión y conclusión

Los resultados obtenidos muestran un patrón de conectividad funcional muy diferente entre los dos grupos (control y riesgo) durante la tarea de toma de decisiones con estímulos relacionado con alcohol. En el grupo control, la red presenta una organización más distribuida y con más conexiones en general, entre hemisferios y participación equilibrada entre regiones frontales, parietales y occipitales, característico de un sistema funcional flexible.

En el grupo riesgo por el contrario se observa menor integración, viéndose solo 2 comunidades con pocas conexiones y más distribuidas. También se observan grandes números de conexiones entre zonas lejanas, lo que habla de procesamiento menos especializado y dificultad para integrar información eficientemente.

Sin embargo, se observa que comparten mismo hub en zona fronto-central- izquierda FC1, que sugiere que esta zona actúa como integradora en esta zona. FC1 se ha visto que está vinculada con funciones ejecutivas, control cognitivo y regulación de conductas.

CONCLUSIÓN

Los resultados muestran que grupo control presenta más conectividad, y que está más integrada entre diferentes áreas y también se encuentra más equilibrada, mientras que grupo riesgo muestran menos conectividad y se observan más distribuidas. Aun así, ambos grupos mantienen a FC1 como hub, lo que refiere que en ambos grupos hay integración funcional.

Campbell EM, Singh G, Claus ED, Witkiewitz K, Costa VD, Hogeveen J, Cavanagh JF. Electrophysiological Markers of Aberrant Cue-Specific Exploration in Hazardous Drinkers. Comput Psychiatr. 2023 Jul 28;7(1):47-59. doi: 10.5334/cpsy.96. PMID: 38774639; PMCID: PMC11104413.