

Tarea1

April 30, 2025

1 Proyecto Sistemas Distribuidos 2025 - Entregable 1

Nombre: [Lucas Vicuña, Vicente Silva]

Curso: Sistemas Distribuidos

Profesor: Nicolás Hidalgo

Ayudantes: Sofía Belmar, César Muñoz Rivera, J. Tomás Silva, Joaquín Villegas y Marcelo Yáñez

Fecha: 30 Abril 2025

```
[6]: #codigo creado x mi para instalaciones limpias en jupyter

import subprocess
import sys

def instalar(paquete, usar_conda=False, canal=None, version=None):
    if usar_conda:
        comando = ["conda", "install", "--yes"]
        if canal:
            comando += ["--channel", canal]
        comando.append(paquete)
    else:
        comando = [sys.executable, "-m", "pip", "install"]
        if version:
            comando.append(f"{paquete}=={version}")
        else:
            comando.append(paquete)

    resultado = subprocess.run(comando, capture_output=True, text=True)

    if "Requirement already satisfied" in resultado.stdout or "already_
    installed" in resultado.stdout:
        print(f" La librería '{paquete}' ya está instalada.")
    elif resultado.returncode == 0:
        print(f" La librería '{paquete}' fue instalada correctamente.")
    else:
        print(f" Error al instalar '{paquete}':\n{resultado.stderr}")
```

1.1 Módulo 1: Extracción de Eventos con Webscrapping

1.1.1 Etapa 1.0 - Instalación de Librerías

```
[33]: instalar("selenium")
```

La librería 'selenium' ya está instalada.

```
[3]: instalar("webdriver_manager")
```

La librería 'webdriver_manager' ya está instalada.

```
[4]: instalar("matplotlib")
```

La librería 'matplotlib' ya está instalada.

```
[5]: instalar("mysql-connector-python")
```

La librería 'mysql-connector-python' ya está instalada.

```
[19]: instalar("redis")
```

La librería 'redis' fue instalada correctamente.

```
[29]: instalar("matplotlib")
```

La librería 'matplotlib' ya está instalada.

```
[30]: instalar("openpyxl")
```

La librería 'openpyxl' ya está instalada.

```
[32]: instalar("pandas")
```

La librería 'pandas' ya está instalada.

1.1.2 Etapa 1.1 - Importación de librerías

```
[4]: from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time

import sqlite3
from datetime import datetime
import mysql.connector
from mysql.connector import Error

import pandas as pd
```

```
import matplotlib.pyplot as plt

import os

from tqdm import tqdm
```

1.1.3 Etapa 1.2 - Definición del bounding box para la Región Metropolitana (Santiago, Chile)

Usamos las coordenadas geográficas para definir un área rectangular de consulta. Estas coordenadas encierran el centro urbano de Santiago.

```
[3]: bounding_box = {
    "bottom": -33.87, # límite sur (Paine, San Bernardo)
    "top": -33.28, # límite norte (Colina, Lampa)
    "left": -71.02, # límite oeste (Melipilla, Curacaví)
    "right": -70.38 # límite este (San José de Maipo, Farellones)
}
```

Estas coordenadas cubren toda la zona urbana central de la Región Metropolitana: Santiago Centro, Providencia, Ñuñoa, Maipú, Las Condes, etc.

2 Driver

<https://storage.googleapis.com/chrome-for-testing-public/135.0.7049.95/win32/chromedriver-win32.zip>

EL DRIVER DEBE ESTAR EN LA MISMA CARPETA DEL JUPYTER

```
[4]: driver_path = "chromedriver.exe"
```

2.0.1 Etapa 1.3 - Verificación básica de Selenium

En esta etapa probamos que `selenium` se puede ejecutar correctamente con `chromedriver.exe` y que abre una ventana de navegador de forma controlada.

```
[5]: # Configurar opciones
options = Options()
options.add_argument("--start-maximized") # pantalla completa
# options.add_argument("--headless") # si lo quieres invisible, activa esto

# Ruta del driver
service = Service(executable_path="chromedriver.exe")

# Crear instancia del navegador
driver = webdriver.Chrome(service=service, options=options)

# Cargar una página simple (Google para prueba)
```

```

driver.get("https://www.waze.com/es-419/live-map/")

# Esperar unos segundos para ver que funciona
time.sleep(5)

# Cerrar navegador
driver.quit()
print(" Selenium funciona correctamente con Chrome (WAZE).")

```

Selenium funciona correctamente con Chrome (WAZE).

2.0.2 Etapa 1.4 - Inicialización de Selenium y selección del tipo de pantalla

Esta celda permite al usuario elegir cómo se abrirá el navegador: maximizado, pantalla completa, minimizado o con un tamaño personalizado. Luego, se abre el mapa de Waze centrado en la Región Metropolitana usando el bounding box.

```

[6]: # OPCIONES DE PANTALLA
options = Options()
print(" Cómo quieres abrir la pantalla del navegador:")
print("1: start-maximized")
print("2: start-fullscreen")
print("3: start-minimized")
print("4: window_size(x, y) personalizada")

opcion_pantalla = int(input("Elige una opción [1-4]: "))

if opcion_pantalla == 1:
    options.add_argument("--start-maximized")
elif opcion_pantalla == 2:
    options.add_argument("--start-fullscreen")
elif opcion_pantalla == 3:
    options.add_argument("--start-minimized")
elif opcion_pantalla == 4:
    x = input("Ancho (x): ")
    y = input("Alto (y): ")
    options.add_argument(f"--window-size={x},{y}")
else:
    print(" Opción inválida. Usando resolución por defecto (1280x800)")
    options.add_argument("--window-size=1280,800")

# INICIALIZACIÓN DRIVER
driver_path = "chromedriver.exe"
service = Service(driver_path)
driver = webdriver.Chrome(service=service, options=options)

center_lat = (bounding_box["bottom"] + bounding_box["top"]) / 2
center_lon = (bounding_box["left"] + bounding_box["right"]) / 2

```

```

# ABRIR WAZE LIVE MAP
driver.get("https://www.waze.com/live-map")
time.sleep(10) # esperar carga

# INTENTAR CENTRAR MAPA VIA JS
try:
    script = f"""
        var map = document.querySelector("wz-live-map)?.map;
        if (map) {{
            map.setView([center_lat], [center_lon], 15);
        }}
    """
    driver.execute_script(script)
    print(f" Mapa centrado en lat: {center_lat}, lon: {center_lon}")
except Exception as e:
    print(" No se pudo centrar automáticamente:", e)

# INICIALIZAR LISTAS
eventos_descripciones = []
eventos_latitudes = []
eventos_longitudes = []
eventos_tipos = []

print(" Selenium listo. Puedes ver el mapa y los eventos.")

```

Cómo quieres abrir la pantalla del navegador:

- 1: start-maximized
- 2: start-fullscreen
- 3: start-minimized
- 4: window_size(x, y) personalizada

Elige una opción [1-4]: 1

Mapa centrado en lat: -33.575, lon: -70.69999999999999

Selenium listo. Puedes ver el mapa y los eventos.

2.0.3 Etapa 1.5 - Levantamiento del contenedor MySQL (Docker)

Se configuró un contenedor Docker utilizando `docker compose` para ejecutar un servidor MySQL 8.0. Esta base de datos cumple la función de almacenamiento central del sistema, permitiendo el acceso concurrente desde diferentes módulos como el scraper, generador de tráfico y sistema de caché.

La configuración considera:

- Un archivo `.env` con credenciales y nombre de base de datos
- Un `docker-compose.yml` que define el contenedor `mysql_eventos`
- Un archivo `init.sql` que crea automáticamente la tabla `eventos` al iniciar el contenedor por primera vez

El contenedor se levanta desde la terminal o desde una celda de Jupyter mediante:

“bash docker compose up -d

```
[7]: def verificar_docker():
    try:
        resultado = subprocess.run(["docker", "info"], capture_output=True,
        ↪text=True, timeout=5)
        if resultado.returncode == 0:
            print(" Docker está corriendo.")
            return True
        else:
            print(" Docker NO está corriendo. Abre Docker Desktop primero.")
            return False
    except Exception as e:
        print(f" Error al intentar conectar a Docker: {e}")
        return False

if verificar_docker():
    print(" Puedes ejecutar docker compose up -d ahora.")
else:
    print(" No se puede levantar servicios hasta que Docker esté activo.")
```

Docker está corriendo.

Puedes ejecutar docker compose up -d ahora.

```
[8]: !docker compose up -d
```

```
time="2025-04-29T22:58:42-04:00" level=warning
msg="C:\\Universidad\\Sistemas_distribuidos\\Proyecto\\entregable1\\docker-
compose.yml: the attribute `version` is obsolete, it will be ignored, please
remove it to avoid potential confusion"
Container mysql_eventos Running
```

```
[21]: !docker run -d --name redis-server -p 6379:6379 redis
```

```
1d8fb1ee7627143aca485096f23eca3e0ba85645052fe8825d369b3979ed4cb4
```

```
Unable to find image 'redis:latest' locally
```

```
latest: Pulling from library/redis
```

```
254e724d7786: Pulling fs layer
```

```
4f4fb700ef54: Pulling fs layer
```

```
e14bd717af36: Pulling fs layer
```

```
a131b527c2eb: Pulling fs layer
```

```
1d380dfb74a4: Pulling fs layer
```

```
c9e771e39c60: Pulling fs layer
```

```
3781e1e8cbc4: Pulling fs layer
```

```
f5340b1d1605: Pulling fs layer
```

```
3781e1e8cbc4: Download complete
```

```
4f4fb700ef54: Download complete
```

```
e14bd717af36: Download complete
a131b527c2eb: Download complete
f5340b1d1605: Download complete
1d380dfb74a4: Download complete
254e724d7786: Download complete
c9e771e39c60: Download complete
254e724d7786: Pull complete
e14bd717af36: Pull complete
a131b527c2eb: Pull complete
1d380dfb74a4: Pull complete
3781e1e8cbc4: Pull complete
4f4fb700ef54: Pull complete
f5340b1d1605: Pull complete
c9e771e39c60: Pull complete
Digest: sha256:7df1eeff67eb0ba84f6b9d2940765a6bb1158081426745c185a03b1507de6a09
Status: Downloaded newer image for redis:latest
```

[22]: !docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
1d8fb1ee7627	redis	"docker-entrypoint.sâ€¦"	34 seconds ago	Up 33 seconds
3b2e68a16c5b	mysql:8.0	"docker-entrypoint.sâ€¦"	13 days ago	Up 2 hours

PORTS NAMES
0.0.0.0:6379->6379/tcp redis-server
0.0.0.0:3306->3306/tcp, 33060/tcp mysql_eventos

2.0.4 Etapa 1.6 - Inicialización de base de datos MySQL

Se inicializa un contenedor MySQL utilizando Docker y se configura la base de datos `eventosdb` mediante un archivo `init.sql`. Esta base almacena los eventos recolectados desde el mapa de Waze.

La tabla `eventos` incluye los siguientes campos:

- `tipo`: tipo de alerta o evento detectado (ej: hazard, police, accident)
- `descripcion`: texto descriptivo del evento
- `lat`: latitud del cuadrante donde se detectó
- `lon`: longitud del cuadrante donde se detectó
- `fecha_extraccion`: fecha y hora en que se registró el evento
- `cuadrante`: etiqueta del cuadrante desde donde se extrajo el dato

Este paso se automatiza con Docker usando `docker-compose`, conectando luego a MySQL desde un notebook de Python para su verificación.

```
[9]: import mysql.connector

# Conexión inicial a MySQL
conn = mysql.connector.connect(
    host="localhost",
    user="usuario",
```

```

        password="pass123",
        database="eventosdb",
        port=3306
    )
    cursor = conn.cursor()

    # Crear tabla eventos (si no existe)
    cursor.execute("""
CREATE TABLE IF NOT EXISTS eventos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tipo VARCHAR(255),
    descripcion TEXT,
    lat DOUBLE,
    lon DOUBLE,
    fecha_extraccion DATETIME,
    cuadrante VARCHAR(255),
    calle VARCHAR(255)
);
""")

    conn.commit()
    cursor.close()
    conn.close()

    print(" Tabla 'eventos' creada correctamente (o ya existía).")

```

Tabla 'eventos' creada correctamente (o ya existía).

2.0.5 Etapa 1.7 - Extracción sistemática de eventos desde el bounding box

Dividimos el bounding box definido previamente en una grilla de cuadrantes (por ejemplo, 5x5) y recorremos uno a uno. Para cada cuadrante: - Centramos el mapa con JavaScript (`setView`). - Esperamos la carga de eventos. - Extraemos tipo, clase y guardamos fecha + cuadrante. El proceso se repite hasta acumular al menos 10.000 eventos únicos.

2.0.6 Etapa 1.7 - Visualización y exportación de eventos capturados

En esta etapa revisamos la información extraída del mapa de Waze y almacenada en la base de datos SQLite. Mostramos un resumen de los tipos de eventos detectados y exportamos los datos a un archivo Excel (.xlsx) con dos hojas:

- “**Eventos**”: Contiene todos los registros extraídos (tipo, descripción, coordenadas, fecha).
- “**Resumen**”: Contiene la cantidad total de eventos agrupados por tipo.

```

[7]: import time
import random
import mysql.connector

```



```

from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from tqdm import tqdm
from datetime import datetime

# Funciones
def iniciar_driver(modos_headless=True):
    options = Options()
    if modos_headless:
        options.add_argument("--headless=new")
    else:
        options.add_argument("--start-maximized")
        options.add_argument("--window-size=1280,800")
        options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64)␣
↪AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36")
        driver = webdriver.Chrome(service=Service("chromedriver.exe"),␣
↪options=options)
    return driver

# Lista de nombres de calles (como referencia para asignar nombre cercano)
nombres_calles = [
    "Alameda Centro", "Plaza Italia", "Apoquindo Las Condes", "Vespucio Norte",␣
↪"Vespucio Sur",
    "Grecia Ñuñoa", "Costanera Norte", "Autopista Central", "Av. La Florida",
    "Providencia", "Estación Central", "Pudahuel", "Maipú", "Peñalolén",␣
↪"Puente Alto",
    "Lo Barnechea", "San Miguel", "San Bernardo", "Las Rejas", "Camino a␣
↪Melipilla",
    "Los Dominicos", "Quinta Normal", "Independencia", "Macul", "La Cisterna",␣
↪"Lo Prado",
    "Huechuraba", "Recoleta", "San Joaquín", "Ñuñoa Plaza Egaña"
]

# Programa principal
modos_headless = input("¿Ejecutar en modo headless (sin abrir navegador)? [s/n]:␣
↪").strip().lower() == "s"

driver = iniciar_driver(modos_headless)

#aqui remplazar con tus datos
conn = mysql.connector.connect(
    host="localhost",
    user="usuario",
    password="pass123",

```

```

        database="eventosdb",
        port=3306
    )
    cursor = conn.cursor()

    # Consultar eventos ya existentes
    cursor.execute("SELECT COUNT(*) FROM eventos")
    total_guardados = cursor.fetchone()[0]
    print(f" Ya existen {total_guardados} eventos en la base de datos.")

    # Parámetros
    espera_segundos = 7
    max_eventos = 10000
    bounding_box = {
        "bottom": -33.87,
        "top": -33.28,
        "left": -71.02,
        "right": -70.38
    }
    lat_total = bounding_box["top"] - bounding_box["bottom"]
    lon_total = bounding_box["right"] - bounding_box["left"]
    base_div = 16

    lat_step = lat_total / base_div
    lon_step = lon_total / base_div

    center_lat = (bounding_box["bottom"] + bounding_box["top"]) / 2
    center_lon = (bounding_box["left"] + bounding_box["right"]) / 2

    expansion = 1

    while total_guardados < max_eventos:
        zoom = max(14 - (expansion // 3), 11)

        cuadrantes = []
        for i in range(-expansion + 1, expansion):
            for j in range(-expansion + 1, expansion):
                lat_centro = center_lat + i * lat_step
                lon_centro = center_lon + j * lon_step
                if (bounding_box["bottom"] <= lat_centro <= bounding_box["top"] and
                    bounding_box["left"] <= lon_centro <= bounding_box["right"]):
                    cuadrantes.append((lat_centro, lon_centro,
↵f"Q{expansion}_{i+expansion}_{j+expansion}"))

        print(f"\n Expansión {expansion}x{expansion} con zoom {zoom} - Cuadrantes:↵
↵{len(cuadrantes)}")

```

```

    bar_eventos = tqdm(total=max_eventos, initial=total_guardados, desc=" ↳
↳Eventos acumulados", position=0)
    bar_cuadrantes = tqdm(cuadrantes, desc=" Recorrido cuadrantes", ↳
↳position=1, leave=False)

    for lat_centro, lon_centro, cuadrante_nombre in bar_cuadrantes:
        if total_guardados >= max_eventos:
            break

        url = f"https://www.waze.com/live-map?
↳ll={lat_centro}%2C{lon_centro}&zoom={zoom}"
        driver.get(url)
        time.sleep(espera_segundos)

        icons = driver.find_elements(By.CSS_SELECTOR, "div.leaflet-marker-icon")

        for icon in icons:
            if total_guardados >= max_eventos:
                break
            try:
                clases = icon.get_attribute("class").split()

                tipo = None
                for c in clases:
                    if c.startswith("wm-alert-icon--") or c.
↳startswith("wm-alert-cluster-icon--") or c.startswith("wm-user-icon--"):
                        tipo = c.replace("wm-alert-icon--", "").
↳replace("wm-alert-cluster-icon--", "").replace("wm-user-icon--", "")
                        break
                if not tipo:
                    tipo = "desconocido"

                descripcion = f"Evento tipo {tipo}"
                lat = lat_centro
                lon = lon_centro
                fecha = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

                nombre_calle = random.choice(nombres_calles) # Asignar ↳
↳nombre calle aleatorio

                cursor.execute("""
                    SELECT COUNT(*) FROM eventos
                    WHERE tipo = %s AND cuadrante = %s AND fecha_extraccion = %s
                    """, (tipo, cuadrante_nombre, fecha))

                if cursor.fetchone()[0] == 0:

```

```

        cursor.execute("""
            INSERT INTO eventos (tipo, descripcion, lat, lon,
↪ fecha_extraccion, cuadrante, calle)
            VALUES (%s, %s, %s, %s, %s, %s, %s)
        """, (tipo, descripcion, lat, lon, fecha, cuadrante_nombre,
↪ nombre_calle))

        conn.commit()
        total_guardados += 1
        bar_eventos.update(1)

    except Exception as e:
        print(" Error al guardar evento:", e)

    bar_eventos.close()
    bar_cuadrantes.close()
    expansion += 1

cursor.close()
conn.close()
driver.quit()
print(f"\n Extracción completada. Total de eventos capturados:
↪ {total_guardados}")

```

¿Ejecutar en modo headless (sin abrir navegador)? [s/n]: n

Ya existen 8297 eventos en la base de datos.

Expansión 1x1 con zoom 14 - Cuadrantes: 1

Eventos acumulados: 83%|
8297/10000 [7:46:39<33:42:13, 71.25s/it]

Eventos acumulados: 83%|
8298/10000 [00:09<4:22:59, 9.27s/it]

Recorrido cuadrantes:
100%| 1/1
[00:09<00:00, 9.39s/it]

Eventos acumulados: 83%|
| 8302/10000 [00:09<53:11, 1.88s/it]

Expansión 2x2 con zoom 14 - Cuadrantes: 9

Eventos acumulados: 83%|
| 8302/10000 [00:00<?, ?it/s]

Eventos acumulados: 83%|
8303/10000 [00:08<3:55:01, 8.31s/it]

Eventos acumulados: 83%|
8307/10000 [00:16<1:25:18, 3.02s/it]

Eventos acumulados: 83%|
 8311/10000 [00:25<1:10:56, 2.52s/it]
 Eventos acumulados: 83%|
 | 8316/10000 [00:33<58:56, 2.10s/it]
 Eventos acumulados: 83%|
 8319/10000 [00:40<1:01:11, 2.18s/it]
 Eventos acumulados: 83%|
 | 8332/10000 [00:49<29:35, 1.06s/it]
 Eventos acumulados: 83%|
 | 8342/10000 [00:58<23:44, 1.16it/s]
 Eventos acumulados: 83%|
 | 8346/10000 [01:06<32:15, 1.17s/it]
 Eventos acumulados: 84%|
 | 8356/10000 [01:15<24:56, 1.10it/s]
 Recorrido cuadrantes:
 100%| | 9/9
 [01:15<00:00, 8.47s/it]
 Eventos acumulados: 84%|
 | 8357/10000 [01:15<37:32, 1.37s/it]

Expansión 3x3 con zoom 13 - Cuadrantes: 25

Eventos acumulados: 84%|
 | 8357/10000 [00:00<?, ?it/s]
 Eventos acumulados: 84%|
 8358/10000 [00:08<3:46:39, 8.28s/it]
 Eventos acumulados: 84%|
 8360/10000 [00:16<2:22:08, 5.20s/it]
 Eventos acumulados: 84%|
 | 8368/10000 [00:24<41:24, 1.52s/it]
 Eventos acumulados: 84%|
 | 8370/10000 [00:33<59:32, 2.19s/it]
 Eventos acumulados: 84%|
 | 8378/10000 [00:41<41:03, 1.52s/it]
 Eventos acumulados: 84%|
 | 8381/10000 [00:50<49:01, 1.82s/it]
 Eventos acumulados: 84%|
 | 8385/10000 [00:57<50:07, 1.86s/it]
 Eventos acumulados: 84%|
 | 8397/10000 [01:05<28:03, 1.05s/it]
 Eventos acumulados: 84%|
 | 8399/10000 [01:13<39:13, 1.47s/it]
 Eventos acumulados: 84%|
 | 8407/10000 [01:22<30:28, 1.15s/it]
 Eventos acumulados: 84%|
 | 8410/10000 [01:30<40:43, 1.54s/it]
 Eventos acumulados: 84%|
 | 8415/10000 [01:38<41:02, 1.55s/it]

```

Eventos acumulados: 84%|
| 8419/10000 [01:46<44:00, 1.67s/it]
Eventos acumulados: 84%|
| 8432/10000 [01:54<25:29, 1.03it/s]
Eventos acumulados: 84%|
| 8434/10000 [02:02<37:18, 1.43s/it]
Eventos acumulados: 84%|
| 8445/10000 [02:11<25:33, 1.01it/s]
Eventos acumulados: 84%|
| 8447/10000 [02:18<36:52, 1.42s/it]
Eventos acumulados: 85%|
| 8460/10000 [02:27<21:16, 1.21it/s]
Eventos acumulados: 85%|
| 8463/10000 [02:34<29:58, 1.17s/it]
Eventos acumulados: 85%|
| 8474/10000 [02:43<22:33, 1.13it/s]
Eventos acumulados: 85%|
| 8481/10000 [02:52<23:44, 1.07it/s]
Eventos acumulados: 85%|
| 8484/10000 [03:00<34:29, 1.36s/it]
Eventos acumulados: 85%|
| 8496/10000 [03:08<22:13, 1.13it/s]
Eventos acumulados: 85%|
| 8504/10000 [03:17<21:36, 1.15it/s]
Eventos acumulados: 85%|
| 8512/10000 [03:25<21:08, 1.17it/s]
Recorrido cuadrantes:
100%| | 25/25
[03:25<00:00, 8.42s/it]
Eventos acumulados: 85%|
| 8512/10000 [03:25<32:57, 1.33s/it]

```

Expansión 4x4 con zoom 13 - Cuadrantes: 49

```

Eventos acumulados: 85%|
| 8512/10000 [00:00<?, ?it/s]
Eventos acumulados: 85%| |
8513/10000 [00:08<3:24:54, 8.27s/it]
Eventos acumulados: 85%| |
8514/10000 [00:16<3:27:35, 8.38s/it]
Eventos acumulados: 85%| |
8515/10000 [00:25<3:27:21, 8.38s/it]
Eventos acumulados: 85%| |
8520/10000 [00:33<1:20:03, 3.25s/it]
Eventos acumulados: 85%| |
8524/10000 [00:41<1:06:32, 2.70s/it]
Eventos acumulados: 85%| |
8526/10000 [00:50<1:15:14, 3.06s/it]

```

Eventos acumulados: 85%	
8527/10000 [00:58<1:34:34, 3.85s/it]	
Eventos acumulados: 85%	
8530/10000 [01:06<1:24:05, 3.43s/it]	
Eventos acumulados: 85%	
8533/10000 [01:14<1:15:17, 3.08s/it]	
Eventos acumulados: 85%	
8534/10000 [01:21<1:30:22, 3.70s/it]	
Eventos acumulados: 85%	
8542/10000 [01:28<41:04, 1.69s/it]	
Eventos acumulados: 85%	
8544/10000 [01:36<51:34, 2.13s/it]	
Eventos acumulados: 85%	
8547/10000 [01:43<54:02, 2.23s/it]	
Eventos acumulados: 85%	
8549/10000 [01:52<1:05:03, 2.69s/it]	
Eventos acumulados: 86%	
8554/10000 [02:00<53:55, 2.24s/it]	
Eventos acumulados: 86%	
8558/10000 [02:07<50:28, 2.10s/it]	
Eventos acumulados: 86%	
8562/10000 [02:15<48:17, 2.02s/it]	
Eventos acumulados: 86%	
8567/10000 [02:22<43:23, 1.82s/it]	
Eventos acumulados: 86%	
8574/10000 [02:30<35:34, 1.50s/it]	
Eventos acumulados: 86%	
8581/10000 [02:37<29:01, 1.23s/it]	
Eventos acumulados: 86%	
8590/10000 [02:46<22:10, 1.06it/s]	
Eventos acumulados: 86%	
8598/10000 [02:54<20:48, 1.12it/s]	
Eventos acumulados: 86%	
8601/10000 [03:01<29:01, 1.24s/it]	
Eventos acumulados: 86%	
8605/10000 [03:09<33:12, 1.43s/it]	
Eventos acumulados: 86%	
8607/10000 [03:16<42:20, 1.82s/it]	
Eventos acumulados: 86%	
8614/10000 [03:24<33:53, 1.47s/it]	
Eventos acumulados: 86%	
8625/10000 [03:31<21:28, 1.07it/s]	
Eventos acumulados: 86%	
8635/10000 [03:40<17:56, 1.27it/s]	
Eventos acumulados: 86%	
8638/10000 [03:48<26:44, 1.18s/it]	
Eventos acumulados: 86%	
8641/10000 [03:55<32:58, 1.46s/it]	

```

Eventos acumulados: 87%|
| 8654/10000 [04:03<19:13, 1.17it/s]
Eventos acumulados: 87%|
| 8657/10000 [04:10<26:12, 1.17s/it]
Eventos acumulados: 87%|
| 8669/10000 [04:18<17:23, 1.28it/s]
Eventos acumulados: 87%|
| 8672/10000 [04:25<24:31, 1.11s/it]
Eventos acumulados: 87%|
| 8681/10000 [04:34<20:10, 1.09it/s]
Eventos acumulados: 87%|
| 8684/10000 [04:42<28:50, 1.31s/it]
Eventos acumulados: 87%|
| 8686/10000 [04:49<37:32, 1.71s/it]
Eventos acumulados: 87%|
| 8691/10000 [04:57<35:35, 1.63s/it]
Eventos acumulados: 87%|
| 8695/10000 [05:04<36:58, 1.70s/it]
Eventos acumulados: 87%|
| 8702/10000 [05:12<30:53, 1.43s/it]
Eventos acumulados: 87%|
| 8709/10000 [05:19<27:31, 1.28s/it]
Eventos acumulados: 87%|
| 8721/10000 [05:27<19:26, 1.10it/s]
Recorrido cuadrantes: 86%|
| 42/49 [05:27<00:54, 7.76s/it]
Eventos acumulados: 87%|
| 8723/10000 [05:44<39:51, 1.87s/it]
Eventos acumulados: 87%|
| 8738/10000 [05:53<20:45, 1.01it/s]
Eventos acumulados: 87%|
| 8742/10000 [06:01<25:42, 1.23s/it]
Eventos acumulados: 87%|
| 8745/10000 [06:09<31:50, 1.52s/it]
Eventos acumulados: 88%|
| 8751/10000 [06:18<30:47, 1.48s/it]
Eventos acumulados: 88%|
| 8759/10000 [06:26<27:03, 1.31s/it]
Recorrido cuadrantes:
100%| | 49/49
[06:26<00:00, 8.37s/it]
Eventos acumulados: 88%|
| 8764/10000 [06:26<31:37, 1.53s/it]

```

Expansión 5x5 con zoom 13 - Cuadrantes: 81

```

Eventos acumulados:
88%| | 8764/10000

```



```

[00:00<?, ?it/s]
  Eventos acumulados: 88%|
8765/10000 [00:08<2:50:58, 8.31s/it]
  Eventos acumulados: 88%|
8767/10000 [00:16<1:47:07, 5.21s/it]
  Eventos acumulados: 88%|
8768/10000 [00:24<2:09:01, 6.28s/it]
  Eventos acumulados: 88%|
| 8778/10000 [00:33<27:59, 1.37s/it]
  Eventos acumulados: 88%|
| 8781/10000 [00:41<36:11, 1.78s/it]
  Eventos acumulados: 88%|
| 8783/10000 [00:50<46:34, 2.30s/it]
  Recorrido cuadrantes: 7%|
| 6/81 [00:50<10:28, 8.38s/it]
  Eventos acumulados: 88%|
8785/10000 [01:06<1:14:37, 3.69s/it]
  Eventos acumulados: 88%|
8786/10000 [01:14<1:27:50, 4.34s/it]
  Eventos acumulados: 88%|
8788/10000 [01:23<1:26:27, 4.28s/it]
  Eventos acumulados: 88%|
8791/10000 [01:30<1:12:16, 3.59s/it]
  Eventos acumulados: 88%|
8792/10000 [01:37<1:23:50, 4.16s/it]
  Eventos acumulados: 88%|
8793/10000 [01:45<1:35:13, 4.73s/it]
  Eventos acumulados: 88%|
| 8798/10000 [01:52<58:15, 2.91s/it]
  Eventos acumulados: 88%|
| 8801/10000 [02:00<55:08, 2.76s/it]
  Eventos acumulados: 88%|
| 8804/10000 [02:07<53:01, 2.66s/it]
  Eventos acumulados: 88%|
8805/10000 [02:14<1:05:10, 3.27s/it]
  Eventos acumulados: 88%|
8808/10000 [02:23<1:01:28, 3.09s/it]
  Eventos acumulados: 88%|
8809/10000 [02:31<1:16:48, 3.87s/it]
  Eventos acumulados: 88%|
| 8815/10000 [02:38<47:30, 2.41s/it]
  Eventos acumulados: 88%|
| 8818/10000 [02:47<49:28, 2.51s/it]
  Eventos acumulados: 88%|
| 8821/10000 [02:55<50:49, 2.59s/it]
  Eventos acumulados: 88%|
| 8827/10000 [03:04<36:39, 1.88s/it]
  Eventos acumulados: 88%|

```

| 8834/10000 [03:12<26:04, 1.34s/it]
 Eventos acumulados: 88%|

| 8836/10000 [03:20<37:00, 1.91s/it]
 Eventos acumulados: 88%|

| 8838/10000 [03:28<44:35, 2.30s/it]
 Eventos acumulados: 88%|

| 8843/10000 [03:36<39:00, 2.02s/it]
 Eventos acumulados: 88%|

| 8844/10000 [03:44<52:46, 2.74s/it]
 Eventos acumulados: 88%|

| 8850/10000 [03:52<38:25, 2.01s/it]
 Eventos acumulados: 89%|

| 8855/10000 [04:00<35:54, 1.88s/it]
 Eventos acumulados: 89%|

| 8862/10000 [04:09<30:09, 1.59s/it]
 Eventos acumulados: 89%|

| 8871/10000 [04:17<22:19, 1.19s/it]
 Eventos acumulados: 89%|

| 8873/10000 [04:25<30:52, 1.64s/it]
 Eventos acumulados: 89%|

| 8876/10000 [04:34<36:06, 1.93s/it]
 Eventos acumulados: 89%|

| 8879/10000 [04:41<38:41, 2.07s/it]
 Eventos acumulados: 89%|

| 8884/10000 [04:50<35:46, 1.92s/it]
 Eventos acumulados: 89%|

| 8889/10000 [04:58<33:56, 1.83s/it]
 Eventos acumulados: 89%|

| 8895/10000 [05:05<29:36, 1.61s/it]
 Eventos acumulados: 89%|

| 8900/10000 [05:14<29:59, 1.64s/it]
 Eventos acumulados: 89%|

| 8904/10000 [05:22<32:02, 1.75s/it]
 Eventos acumulados: 89%|

| 8912/10000 [05:31<22:47, 1.26s/it]
 Eventos acumulados: 89%|

| 8921/10000 [05:39<17:44, 1.01it/s]
 Eventos acumulados: 89%|

| 8924/10000 [05:48<24:40, 1.38s/it]
 Eventos acumulados: 89%|

| 8935/10000 [05:55<15:59, 1.11it/s]
 Eventos acumulados: 89%|

| 8938/10000 [06:03<22:49, 1.29s/it]
 Eventos acumulados: 89%|

| 8943/10000 [06:12<25:00, 1.42s/it]
 Eventos acumulados: 89%|

| 8946/10000 [06:19<28:55, 1.65s/it]
 Eventos acumulados: 90%|

| 8950/10000 [06:28<31:11, 1.78s/it]
Eventos acumulados: 90%|

| 8954/10000 [06:36<32:49, 1.88s/it]
Eventos acumulados: 90%|

| 8965/10000 [06:45<20:32, 1.19s/it]
Eventos acumulados: 90%|

| 8973/10000 [06:53<17:10, 1.00s/it]
Eventos acumulados: 90%|

| 8976/10000 [07:02<23:54, 1.40s/it]
Eventos acumulados: 90%|

| 8984/10000 [07:09<17:59, 1.06s/it]
Eventos acumulados: 90%|

| 8987/10000 [07:17<24:51, 1.47s/it]
Eventos acumulados: 90%|

| 8990/10000 [07:26<30:22, 1.80s/it]
Eventos acumulados: 90%|

| 8992/10000 [07:33<36:23, 2.17s/it]
Eventos acumulados: 90%|

| 9000/10000 [07:42<22:08, 1.33s/it]
Eventos acumulados: 90%|

| 9003/10000 [07:50<28:06, 1.69s/it]
Eventos acumulados: 90%|

| 9009/10000 [07:58<25:57, 1.57s/it]
Eventos acumulados: 90%|

| 9022/10000 [08:07<15:29, 1.05it/s]
Eventos acumulados: 90%|

| 9025/10000 [08:15<21:12, 1.31s/it]
Eventos acumulados: 90%|

| 9035/10000 [08:23<14:57, 1.07it/s]
Eventos acumulados: 90%|

| 9042/10000 [08:31<15:08, 1.05it/s]
Eventos acumulados: 90%|

| 9045/10000 [08:40<21:53, 1.38s/it]
Recorrido cuadrantes: 79%|

| 64/81 [08:40<02:20, 8.24s/it]
Eventos acumulados: 91%|

| 9053/10000 [08:54<21:45, 1.38s/it]
Eventos acumulados: 91%|

| 9059/10000 [09:02<19:15, 1.23s/it]
Eventos acumulados: 91%|

| 9062/10000 [09:09<24:18, 1.56s/it]
Eventos acumulados: 91%|

| 9067/10000 [09:17<23:50, 1.53s/it]
Eventos acumulados: 91%|

| 9077/10000 [09:24<15:35, 1.01s/it]
Eventos acumulados: 91%|

| 9079/10000 [09:31<22:10, 1.44s/it]
Eventos acumulados: 91%|

| 9083/10000 [09:40<25:19, 1.66s/it]
 Eventos acumulados: 91%|
 | 9091/10000 [09:48<20:44, 1.37s/it]
 Eventos acumulados: 91%|
 | 9098/10000 [09:57<17:17, 1.15s/it]
 Eventos acumulados: 91%|
 | 9106/10000 [10:05<14:51, 1.00it/s]
 Eventos acumulados: 91%|
 | 9109/10000 [10:13<20:50, 1.40s/it]
 Eventos acumulados: 91%|
 | 9113/10000 [10:22<23:56, 1.62s/it]
 Eventos acumulados: 91%|
 | 9122/10000 [10:30<16:51, 1.15s/it]
 Eventos acumulados: 91%|
 | 9130/10000 [10:39<14:07, 1.03it/s]
 Eventos acumulados: 91%|
 | 9133/10000 [10:47<20:10, 1.40s/it]
 Eventos acumulados: 91%|
 | 9139/10000 [10:56<20:07, 1.40s/it]
 Recorrido cuadrantes:
 100%| | 81/81
 [10:56<00:00, 8.44s/it]
 Eventos acumulados: 91%|
 | 9145/10000 [10:56<24:33, 1.72s/it]

Expansión 6x6 con zoom 12 - Cuadrantes: 121

Eventos acumulados:
 91%| | 9145/10000
 [00:00<?, ?it/s]
 Eventos acumulados: 91%| |
 9146/10000 [00:08<1:57:28, 8.25s/it]
 Eventos acumulados: 91%|
 | 9149/10000 [00:16<53:57, 3.80s/it]
 Eventos acumulados: 92%|
 | 9151/10000 [00:24<55:48, 3.94s/it]
 Eventos acumulados: 92%|
 | 9155/10000 [00:33<41:03, 2.92s/it]
 Eventos acumulados: 92%|
 | 9165/10000 [00:41<17:28, 1.26s/it]
 Eventos acumulados: 92%|
 | 9168/10000 [00:50<22:52, 1.65s/it]
 Eventos acumulados: 92%|
 | 9170/10000 [00:58<29:25, 2.13s/it]
 Recorrido cuadrantes: 6%|
 | 7/121 [00:58<15:52, 8.36s/it]
 Eventos acumulados: 92%|
 | 9172/10000 [01:15<47:41, 3.46s/it]

Eventos acumulados: 92%|
 | 9173/10000 [01:23<56:26, 4.10s/it]
 Eventos acumulados: 92%|
 9174/10000 [01:31<1:05:38, 4.77s/it]
 Eventos acumulados: 92%|
 9175/10000 [01:40<1:15:22, 5.48s/it]
 Eventos acumulados: 92%|
 | 9178/10000 [01:48<58:45, 4.29s/it]
 Eventos acumulados: 92%|
 | 9181/10000 [01:56<50:34, 3.71s/it]
 Eventos acumulados: 92%|
 9182/10000 [02:05<1:00:08, 4.41s/it]
 Eventos acumulados: 92%|
 | 9191/10000 [02:13<23:44, 1.76s/it]
 Eventos acumulados: 92%|
 | 9193/10000 [02:21<30:06, 2.24s/it]
 Eventos acumulados: 92%|
 | 9199/10000 [02:30<24:39, 1.85s/it]
 Eventos acumulados: 92%|
 | 9200/10000 [02:38<33:09, 2.49s/it]
 Eventos acumulados: 92%|
 | 9201/10000 [02:46<42:47, 3.21s/it]
 Eventos acumulados: 92%|
 | 9203/10000 [02:55<45:56, 3.46s/it]
 Eventos acumulados: 92%|
 | 9205/10000 [03:03<48:16, 3.64s/it]
 Eventos acumulados: 92%|
 | 9206/10000 [03:11<58:10, 4.40s/it]
 Eventos acumulados: 92%|
 | 9209/10000 [03:19<49:15, 3.74s/it]
 Eventos acumulados: 92%|
 | 9211/10000 [03:28<50:41, 3.86s/it]
 Eventos acumulados: 92%|
 9212/10000 [03:36<1:00:22, 4.60s/it]
 Eventos acumulados: 92%|
 | 9218/10000 [03:45<29:18, 2.25s/it]
 Eventos acumulados: 92%|
 | 9220/10000 [03:53<35:23, 2.72s/it]
 Eventos acumulados: 92%|
 | 9222/10000 [04:01<40:18, 3.11s/it]
 Eventos acumulados: 92%|
 | 9227/10000 [04:10<31:07, 2.42s/it]
 Eventos acumulados: 92%|
 | 9228/10000 [04:18<40:14, 3.13s/it]
 Eventos acumulados: 92%|
 | 9231/10000 [04:26<38:31, 3.01s/it]
 Eventos acumulados: 92%|
 | 9232/10000 [04:34<47:54, 3.74s/it]

Eventos acumulados: 92%|
 | 9238/10000 [04:44<27:12, 2.14s/it]
 Eventos acumulados: 92%|
 | 9240/10000 [04:52<33:11, 2.62s/it]
 Eventos acumulados: 92%|
 | 9242/10000 [05:01<38:04, 3.01s/it]
 Eventos acumulados: 92%|
 | 9247/10000 [05:09<29:52, 2.38s/it]
 Eventos acumulados: 92%|
 | 9249/10000 [05:17<34:30, 2.76s/it]
 Eventos acumulados: 93%|
 | 9257/10000 [05:26<19:12, 1.55s/it]
 Eventos acumulados: 93%|
 | 9259/10000 [05:34<25:23, 2.06s/it]
 Eventos acumulados: 93%|
 | 9261/10000 [05:42<31:00, 2.52s/it]
 Eventos acumulados: 93%|
 | 9265/10000 [05:51<28:45, 2.35s/it]
 Eventos acumulados: 93%|
 | 9270/10000 [05:59<25:16, 2.08s/it]
 Eventos acumulados: 93%|
 | 9272/10000 [06:07<29:55, 2.47s/it]
 Eventos acumulados: 93%|
 | 9273/10000 [06:16<38:20, 3.16s/it]
 Eventos acumulados: 93%|
 | 9281/10000 [06:24<20:18, 1.69s/it]
 Eventos acumulados: 93%|
 | 9283/10000 [06:33<26:14, 2.20s/it]
 Eventos acumulados: 93%|
 | 9288/10000 [06:41<23:27, 1.98s/it]
 Eventos acumulados: 93%|
 | 9293/10000 [06:49<21:57, 1.86s/it]
 Eventos acumulados: 93%|
 | 9298/10000 [06:58<20:59, 1.79s/it]
 Eventos acumulados: 93%|
 | 9303/10000 [07:06<20:23, 1.76s/it]
 Eventos acumulados: 93%|
 | 9312/10000 [07:14<14:33, 1.27s/it]
 Eventos acumulados: 93%|
 | 9318/10000 [07:23<13:39, 1.20s/it]
 Eventos acumulados: 93%|
 | 9327/10000 [07:31<10:16, 1.09it/s]
 Eventos acumulados: 93%|
 | 9330/10000 [07:40<14:56, 1.34s/it]
 Eventos acumulados: 93%|
 | 9332/10000 [07:48<20:28, 1.84s/it]
 Eventos acumulados: 93%|
 | 9334/10000 [07:56<25:36, 2.31s/it]

Eventos acumulados: 93%|
 | 9337/10000 [08:05<27:06, 2.45s/it]
 Eventos acumulados: 93%|
 | 9341/10000 [08:13<25:32, 2.33s/it]
 Eventos acumulados: 93%|
 | 9346/10000 [08:21<22:32, 2.07s/it]
 Eventos acumulados: 94%|
 | 9354/10000 [08:30<14:49, 1.38s/it]
 Eventos acumulados: 94%|
 | 9361/10000 [08:38<12:19, 1.16s/it]
 Eventos acumulados: 94%|
 | 9364/10000 [08:47<16:29, 1.56s/it]
 Eventos acumulados: 94%|
 | 9369/10000 [08:55<16:51, 1.60s/it]
 Eventos acumulados: 94%|
 | 9374/10000 [09:03<17:02, 1.63s/it]
 Eventos acumulados: 94%|
 | 9379/10000 [09:12<17:04, 1.65s/it]
 Eventos acumulados: 94%|
 | 9381/10000 [09:20<21:06, 2.05s/it]
 Eventos acumulados: 94%|
 | 9383/10000 [09:29<25:03, 2.44s/it]
 Eventos acumulados: 94%|
 | 9384/10000 [09:37<31:59, 3.12s/it]
 Eventos acumulados: 94%|
 | 9389/10000 [09:45<24:59, 2.45s/it]
 Eventos acumulados: 94%|
 | 9400/10000 [09:54<12:52, 1.29s/it]
 Eventos acumulados: 94%|
 | 9402/10000 [10:02<17:28, 1.75s/it]
 Eventos acumulados: 94%|
 | 9417/10000 [10:10<08:32, 1.14it/s]
 Eventos acumulados: 94%|
 | 9420/10000 [10:19<11:56, 1.24s/it]
 Eventos acumulados: 94%|
 | 9428/10000 [10:27<10:13, 1.07s/it]
 Eventos acumulados: 94%|
 | 9430/10000 [10:36<14:55, 1.57s/it]
 Eventos acumulados: 94%|
 | 9436/10000 [10:44<14:05, 1.50s/it]
 Eventos acumulados: 94%|
 | 9437/10000 [10:52<19:46, 2.11s/it]
 Eventos acumulados: 94%|
 | 9438/10000 [11:01<26:16, 2.81s/it]
 Eventos acumulados: 94%|
 | 9439/10000 [11:09<33:17, 3.56s/it]
 Eventos acumulados: 94%|
 | 9442/10000 [11:17<30:24, 3.27s/it]

Eventos acumulados: 94%|
 | 9450/10000 [11:26<18:03, 1.97s/it]
 Eventos acumulados: 95%|
 | 9460/10000 [11:34<11:35, 1.29s/it]
 Eventos acumulados: 95%|
 | 9468/10000 [11:43<09:26, 1.07s/it]
 Eventos acumulados: 95%|
 | 9475/10000 [11:51<08:45, 1.00s/it]
 Eventos acumulados: 95%|
 | 9482/10000 [12:00<08:30, 1.02it/s]
 Eventos acumulados: 95%|
 | 9485/10000 [12:08<12:20, 1.44s/it]
 Eventos acumulados: 95%|
 | 9490/10000 [12:16<13:01, 1.53s/it]
 Eventos acumulados: 95%|
 | 9495/10000 [12:25<13:29, 1.60s/it]
 Eventos acumulados: 95%|
 | 9499/10000 [12:33<14:34, 1.75s/it]
 Eventos acumulados: 95%|
 | 9505/10000 [12:42<13:17, 1.61s/it]
 Eventos acumulados: 95%|
 | 9510/10000 [12:50<12:03, 1.48s/it]
 Eventos acumulados: 95%|
 | 9512/10000 [12:58<16:03, 1.97s/it]
 Eventos acumulados: 95%|
 | 9521/10000 [13:07<10:00, 1.25s/it]
 Eventos acumulados: 95%|
 | 9528/10000 [13:15<08:34, 1.09s/it]
 Eventos acumulados: 95%|
 | 9531/10000 [13:24<11:47, 1.51s/it]
 Eventos acumulados: 95%|
 | 9539/10000 [13:32<08:59, 1.17s/it]
 Eventos acumulados: 95%|
 | 9547/10000 [13:41<07:30, 1.01it/s]
 Recorrido cuadrantes: 81%|
 | 98/121 [13:41<03:13, 8.42s/it]
 Eventos acumulados: 95%|
 | 9549/10000 [13:57<17:13, 2.29s/it]
 Eventos acumulados: 96%|
 | 9550/10000 [14:06<22:37, 3.02s/it]
 Eventos acumulados: 96%|
 | 9552/10000 [14:14<24:35, 3.29s/it]
 Eventos acumulados: 96%|
 | 9556/10000 [14:23<20:48, 2.81s/it]
 Eventos acumulados: 96%|
 | 9562/10000 [14:31<15:42, 2.15s/it]
 Eventos acumulados: 96%|
 | 9572/10000 [14:40<09:27, 1.33s/it]


```

Eventos acumulados: 96%|
| 9579/10000 [14:48<07:59, 1.14s/it]
Eventos acumulados: 96%|
| 9582/10000 [14:57<10:41, 1.53s/it]
Eventos acumulados: 96%|
| 9594/10000 [15:05<06:30, 1.04it/s]
Eventos acumulados: 96%|
| 9597/10000 [15:13<09:01, 1.34s/it]
Eventos acumulados: 96%|
| 9603/10000 [15:22<09:03, 1.37s/it]
Eventos acumulados: 96%|
| 9606/10000 [15:30<10:56, 1.67s/it]
Eventos acumulados: 96%|
| 9607/10000 [15:39<14:56, 2.28s/it]
Eventos acumulados: 96%|
| 9610/10000 [15:47<15:45, 2.43s/it]
Eventos acumulados: 96%|
| 9617/10000 [15:55<10:12, 1.60s/it]
Eventos acumulados: 96%|
| 9619/10000 [16:04<13:24, 2.11s/it]
Eventos acumulados: 96%|
| 9627/10000 [16:12<08:42, 1.40s/it]
Eventos acumulados: 96%|
| 9629/10000 [16:21<11:57, 1.93s/it]
Eventos acumulados: 96%|
| 9639/10000 [16:29<07:07, 1.19s/it]
Eventos acumulados: 96%|
| 9647/10000 [16:38<05:46, 1.02it/s]
Eventos acumulados: 97%|
| 9656/10000 [16:46<04:52, 1.18it/s]
Eventos acumulados: 97%|
| 9659/10000 [16:54<07:16, 1.28s/it]
Recorrido cuadrantes:
100%| | 121/121
[16:54<00:00, 8.43s/it]
Eventos acumulados: 97%|
| 9663/10000 [16:54<11:00, 1.96s/it]

```

Expansión 7x7 con zoom 12 - Cuadrantes: 169

```

Eventos acumulados:
97%| | 9663/10000
[00:00<?, ?it/s]
Eventos acumulados: 97%|
| 9664/10000 [00:08<46:17, 8.27s/it]
Eventos acumulados: 97%|
| 9665/10000 [00:16<46:05, 8.26s/it]
Eventos acumulados: 97%|

```

| 9666/10000 [00:24<45:57, 8.26s/it]
 Eventos acumulados: 97%|

| 9667/10000 [00:33<45:47, 8.25s/it]
 Eventos acumulados: 97%|

| 9668/10000 [00:41<45:39, 8.25s/it]
 Eventos acumulados: 97%|

| 9678/10000 [00:49<09:19, 1.74s/it]
 Eventos acumulados: 97%|

| 9680/10000 [00:57<12:09, 2.28s/it]
 Eventos acumulados: 97%|

| 9683/10000 [01:06<12:52, 2.44s/it]
 Eventos acumulados: 97%|

| 9685/10000 [01:14<14:53, 2.84s/it]
 Eventos acumulados: 97%|

| 9686/10000 [01:22<18:57, 3.62s/it]
 Eventos acumulados: 97%|

| 9687/10000 [01:31<23:03, 4.42s/it]
 Eventos acumulados: 97%|

| 9688/10000 [01:39<26:57, 5.18s/it]
 Eventos acumulados: 97%|

| 9689/10000 [01:47<30:21, 5.86s/it]
 Eventos acumulados: 97%|

| 9690/10000 [01:56<33:31, 6.49s/it]
 Eventos acumulados: 97%|

| 9691/10000 [02:04<35:48, 6.95s/it]
 Eventos acumulados: 97%|

| 9694/10000 [02:12<24:21, 4.78s/it]
 Eventos acumulados: 97%|

| 9695/10000 [02:21<27:47, 5.47s/it]
 Eventos acumulados: 97%|

| 9698/10000 [02:29<21:17, 4.23s/it]
 Eventos acumulados: 97%|

| 9706/10000 [02:37<08:44, 1.78s/it]
 Eventos acumulados: 97%|

| 9708/10000 [02:46<11:05, 2.28s/it]
 Eventos acumulados: 97%|

| 9711/10000 [02:54<11:44, 2.44s/it]
 Eventos acumulados: 97%|

| 9712/10000 [03:02<15:17, 3.19s/it]
 Eventos acumulados: 97%|

| 9713/10000 [03:11<18:58, 3.97s/it]
 Eventos acumulados: 97%|

| 9714/10000 [03:19<22:38, 4.75s/it]
 Eventos acumulados: 97%|

| 9715/10000 [03:27<26:00, 5.47s/it]
 Eventos acumulados: 97%|

| 9716/10000 [03:35<28:57, 6.12s/it]
 Eventos acumulados: 97%|

| 9717/10000 [03:44<31:20, 6.64s/it]
Eventos acumulados: 97%|

| 9719/10000 [03:52<26:19, 5.62s/it]
Eventos acumulados: 97%|

| 9722/10000 [04:00<19:53, 4.29s/it]
Eventos acumulados: 97%|

| 9726/10000 [04:09<14:48, 3.24s/it]
Eventos acumulados: 97%|

| 9727/10000 [04:17<18:04, 3.97s/it]
Eventos acumulados: 97%|

| 9729/10000 [04:26<18:58, 4.20s/it]
Eventos acumulados: 97%|

| 9734/10000 [04:35<12:53, 2.91s/it]
Eventos acumulados: 97%|

| 9739/10000 [04:43<10:22, 2.39s/it]
Eventos acumulados: 97%|

| 9740/10000 [04:51<13:03, 3.02s/it]
Eventos acumulados: 97%|

| 9741/10000 [05:00<16:08, 3.74s/it]
Eventos acumulados: 97%|

| 9743/10000 [05:08<16:30, 3.85s/it]
Eventos acumulados: 97%|

| 9744/10000 [05:16<19:31, 4.58s/it]
Eventos acumulados: 97%|

| 9745/10000 [05:25<22:25, 5.28s/it]
Eventos acumulados: 98%|

| 9751/10000 [05:33<09:45, 2.35s/it]
Eventos acumulados: 98%|

| 9753/10000 [05:41<11:33, 2.81s/it]
Recorrido cuadrantes: 24%|

| 41/169 [05:41<17:42, 8.30s/it]
Eventos acumulados: 98%|

| 9755/10000 [05:58<17:24, 4.26s/it]
Eventos acumulados: 98%|

| 9756/10000 [06:06<20:01, 4.92s/it]
Eventos acumulados: 98%|

| 9757/10000 [06:14<22:27, 5.55s/it]
Eventos acumulados: 98%|

| 9764/10000 [06:23<10:39, 2.71s/it]
Eventos acumulados: 98%|

| 9768/10000 [06:31<09:37, 2.49s/it]
Eventos acumulados: 98%|

| 9770/10000 [06:39<10:48, 2.82s/it]
Eventos acumulados: 98%|

| 9771/10000 [06:48<13:19, 3.49s/it]
Eventos acumulados: 98%|

| 9774/10000 [06:56<12:12, 3.24s/it]
Eventos acumulados: 98%|

9775/10000 [07:04<14:51, 3.96s/it]	
Eventos acumulados: 98%	
9776/10000 [07:12<17:31, 4.70s/it]	
Eventos acumulados: 98%	
9777/10000 [07:21<20:00, 5.38s/it]	
Eventos acumulados: 98%	
9778/10000 [07:29<22:15, 6.01s/it]	
Eventos acumulados: 98%	
9785/10000 [07:37<09:42, 2.71s/it]	
Eventos acumulados: 98%	
9795/10000 [07:46<05:31, 1.62s/it]	
Eventos acumulados: 98%	
9799/10000 [07:54<05:48, 1.73s/it]	
Eventos acumulados: 98%	
9800/10000 [08:02<07:35, 2.28s/it]	
Eventos acumulados: 98%	
9810/10000 [08:11<04:12, 1.33s/it]	
Eventos acumulados: 98%	
9812/10000 [08:20<05:34, 1.78s/it]	
Eventos acumulados: 98%	
9816/10000 [08:28<05:44, 1.87s/it]	
Eventos acumulados: 98%	
9819/10000 [08:36<06:19, 2.10s/it]	
Eventos acumulados: 98%	
9822/10000 [08:45<06:48, 2.29s/it]	
Eventos acumulados: 98%	
9823/10000 [08:53<08:48, 2.99s/it]	
Eventos acumulados: 98%	
9824/10000 [09:02<10:58, 3.74s/it]	
Eventos acumulados: 98%	
9825/10000 [09:10<13:10, 4.52s/it]	
Eventos acumulados: 98%	
9826/10000 [09:20<15:40, 5.41s/it]	
Eventos acumulados: 98%	
9831/10000 [09:28<09:12, 3.27s/it]	
Eventos acumulados: 98%	
9836/10000 [09:37<07:01, 2.57s/it]	
Eventos acumulados:	
98%	9846/10000
[09:45<03:41, 1.44s/it]	
Eventos acumulados:	
98%	9848/10000
[09:54<04:52, 1.92s/it]	
Eventos acumulados:	
99%	9851/10000
[10:02<05:22, 2.17s/it]	
Eventos acumulados:	
99%	9858/10000

[10:11<04:02, 1.71s/it]
Eventos acumulados:
99%| | 9860/10000

[10:19<04:55, 2.11s/it]
Eventos acumulados:
99%| | 9872/10000

[10:28<02:25, 1.13s/it]
Eventos acumulados:
99%| | 9875/10000

[10:36<03:03, 1.47s/it]
Eventos acumulados:
99%| | 9880/10000

[10:44<03:04, 1.54s/it]
Recorrido cuadrantes: 46%|
| 77/169 [10:44<12:53, 8.41s/it]
Eventos acumulados:
99%| | 9882/10000

[11:01<05:07, 2.60s/it]
Eventos acumulados:
99%| | 9883/10000

[11:09<06:11, 3.18s/it]
Eventos acumulados:
99%| | 9884/10000

[11:17<07:23, 3.82s/it]
Eventos acumulados:
99%| | 9890/10000

[11:26<04:41, 2.56s/it]
Eventos acumulados:
99%| | 9896/10000

[11:34<03:34, 2.06s/it]
Eventos acumulados:
99%| | 9902/10000

[11:43<02:57, 1.82s/it]
Eventos acumulados:
99%| | 9905/10000

[11:51<03:11, 2.02s/it]
Eventos acumulados:
99%| | 9920/10000

[11:59<01:20, 1.01s/it]
Eventos acumulados:
99%| | 9927/10000

[12:08<01:12, 1.01it/s]
Eventos acumulados:
99%| | 9930/10000

[12:16<01:37, 1.39s/it]
Eventos acumulados:
99%| | 9938/10000

[12:25<01:09, 1.11s/it]

```

Eventos acumulados:
99%|                                     | 9940/10000
[12:33<01:37, 1.63s/it]
Eventos acumulados:
99%|                                     | 9942/10000
[12:42<02:02, 2.12s/it]
Eventos acumulados:
99%|                                     | 9944/10000
[12:50<02:23, 2.56s/it]
Eventos acumulados:
99%|                                     | 9947/10000
[12:58<02:19, 2.63s/it]
Eventos acumulados:
99%|                                     | 9949/10000
[13:07<02:32, 2.98s/it]
Eventos acumulados:
100%|                                   | 9951/10000
[13:15<02:40, 3.28s/it]
Eventos acumulados:
100%|                                   | 9956/10000
[13:23<01:50, 2.52s/it]
Eventos acumulados:
100%|                                   | 9969/10000
[13:32<00:36, 1.19s/it]
Eventos acumulados:
100%|                                   | 9971/10000
[13:40<00:47, 1.64s/it]
Eventos acumulados:
100%|                                   | 9985/10000
[13:49<00:13, 1.11it/s]
Eventos acumulados:
100%|                                   | 9988/10000
[13:57<00:14, 1.25s/it]
Eventos acumulados:
100%|                                   | 9992/10000
[14:06<00:11, 1.47s/it]
Eventos acumulados:
100%|                                   | 9997/10000
[14:14<00:04, 1.54s/it]
Recorrido cuadrantes: 60%|
| 102/169 [14:14<09:25, 8.44s/it]
Eventos acumulados:
100%|                                   | 10000/10000
[14:14<00:00, 2.54s/it]

```

Extracción completada. Total de eventos capturados: 10000

2.0.7 1.8 Sistema de Caché (Redis)

Para mejorar el rendimiento del sistema frente a múltiples consultas, se integró **Redis** como mecanismo de caché. Este servicio fue desplegado como un contenedor independiente en `docker-compose.yml`, compartiendo red con los demás servicios.

El sistema funciona del siguiente modo: - Cada vez que se solicita un evento (`id_evento`), se busca primero en Redis. - Si se encuentra, se retorna directamente (**HIT**). - Si no se encuentra (**MISS**), se consulta en MySQL y se guarda en Redis con un TTL de 1 hora.

Además, se evaluó el rendimiento del sistema de caché simulando 100 consultas con dos distribuciones: - **Poisson**: donde los tiempos entre consultas siguen una exponencial con parámetro 1/. - **Exponencial**: con un parámetro de escala directo.

Las métricas analizadas incluyeron: - Número total de *hits* y *misses*. - Porcentaje de efectividad del caché. - Curva de efectividad acumulada. - Comparación entre distribuciones. - Gráficos de resumen (PNG) y resultados exportados a `.xlsx`.

Esta implementación permite observar cómo el sistema se comporta bajo carga repetitiva y justifica el uso de Redis como una solución eficiente para sistemas distribuidos que requieren alta disponibilidad y baja latencia.

Se anexa la distribución de tipos de eventos, lo que también permite optimizar estrategias de cacheo específicas según el tipo de consulta más frecuente.

```
[31]: import mysql.connector
import numpy as np
import time
import random
import pandas as pd
from datetime import datetime
import redis
import matplotlib.pyplot as plt

# Configuración
db_config = {
    'user': 'usuario',
    'password': 'pass123',
    'host': 'localhost',
    'database': 'eventosdb',
}

redis_config = {
    'host': 'localhost',
    'port': 6379,
    'db': 0
}

# Conexión a MySQL y Redis
def conectar_a_mysql():
    return mysql.connector.connect(**db_config)
```

```

def conectar_a_redis():
    r = redis.Redis(**redis_config)
    r.ping()
    return r

# Simulación con IDs repetidos
def simular_y_medir_cache(distribucion="poisson", tasa=5, repeticiones=10, n_ids=10):
    conn_mysql = conectar_a_mysql()
    conn_redis = conectar_a_redis()
    cursor = conn_mysql.cursor()

    # Obtener n_ids únicos
    cursor.execute("SELECT id FROM eventos ORDER BY RAND() LIMIT %s", (n_ids,))
    ids = [row[0] for row in cursor.fetchall()]
    ids_repetidos = ids * repeticiones

    if distribucion == "poisson":
        tiempos = np.random.exponential(1 / tasa, len(ids_repetidos))
    elif distribucion == "exponencial":
        tiempos = np.random.exponential(tasa, len(ids_repetidos))
    else:
        raise ValueError("Distribución no válida")

    datos = []
    hits = 0
    misses = 0
    efectividad = []

    for i, (id_evento, espera) in enumerate(zip(ids_repetidos, tiempos)):
        tipo = "MISS"
        if conn_redis.get(f"evento:{id_evento}"):
            tipo = "HIT"
            hits += 1
        else:
            misses += 1
            cursor.execute("SELECT * FROM eventos WHERE id = %s", (id_evento,))
            row = cursor.fetchone()
            if row:
                conn_redis.set(f"evento:{id_evento}", str(row))
                conn_redis.expire(f"evento:{id_evento}", 3600)

        datos.append({
            "consulta_n": i + 1,
            "id_evento": id_evento,
            "tipo": tipo,

```



```

        "timestamp": datetime.now().strftime("%H:%M:%S")
    })
    efectividad.append(hits / (hits + misses))
    time.sleep(espera)

conn_mysql.close()
conn_redis.close()

df_resultado = pd.DataFrame(datos)
resumen = {
    "Distribución": distribucion,
    "Total": len(ids_repetidos),
    "Hits": hits,
    "Misses": misses,
    "Efectividad (%)": round(hits / (hits + misses) * 100, 2)
}
return df_resultado, resumen, efectividad

# Gráfico hits/misses
def graficar_cache(hits, misses):
    total = hits + misses
    plt.figure(figsize=(8, 5))
    plt.bar(["Hits", "Misses"], [hits, misses], color=["green", "red"])
    plt.title("Resumen de Caché - Total")
    plt.ylabel("Cantidad de eventos")
    plt.grid(axis='y', linestyle='--', alpha=0.6)
    for i, v in enumerate([hits, misses]):
        plt.text(i, v + 1, str(v), ha='center')
    nombre = f"resumen_cache_{datetime.now().strftime('%Y%m%d_%H%M%S')}.png"
    plt.tight_layout()
    plt.savefig(nombre)
    plt.show()
    return nombre

# Curva de rendimiento
def graficar_curva(poisson, exponencial):
    plt.figure(figsize=(10, 5))
    plt.plot(poisson, label="Poisson")
    plt.plot(exponencial, label="Exponencial")
    plt.title("Curva de Efectividad del Caché")
    plt.xlabel("Consulta N°")
    plt.ylabel("Efectividad acumulada")
    plt.legend()
    plt.grid(True)
    nombre = f"curva_cache_{datetime.now().strftime('%Y%m%d_%H%M%S')}.png"
    plt.savefig(nombre)
    plt.show()

```

```

    return nombre

# Distribución de tipos
def distribucion_tipos_evento():
    conn = conectar_a_mysql()
    cursor = conn.cursor()
    cursor.execute("SELECT tipo FROM eventos")
    tipos = [row[0] for row in cursor.fetchall()]
    conn.close()
    df = pd.Series(tipos).value_counts().reset_index()
    df.columns = ['Tipo de Evento', 'Frecuencia']
    plt.figure(figsize=(10, 5))
    plt.bar(df['Tipo de Evento'], df['Frecuencia'], color='steelblue')
    plt.title("Distribución de Tipos de Evento")
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    nombre = f"distribucion_tipos_evento_{datetime.now().\
↪strftime('%Y%m%d_%H%M%S')}.png"
    plt.savefig(nombre)
    plt.show()
    return df, nombre

# Exportar Excel
def exportar_resultados(df1, df2, resumen1, resumen2):
    resumen_df = pd.DataFrame([resumen1, resumen2])
    archivo = f"resultados_cache_{datetime.now().strftime('%Y%m%d_%H%M%S')}.
↪xlsx"
    with pd.ExcelWriter(archivo) as writer:
        resumen_df.to_excel(writer, sheet_name="Resumen", index=False)
        df1.to_excel(writer, sheet_name="Poisson", index=False)
        df2.to_excel(writer, sheet_name="Exponencial", index=False)
    return archivo

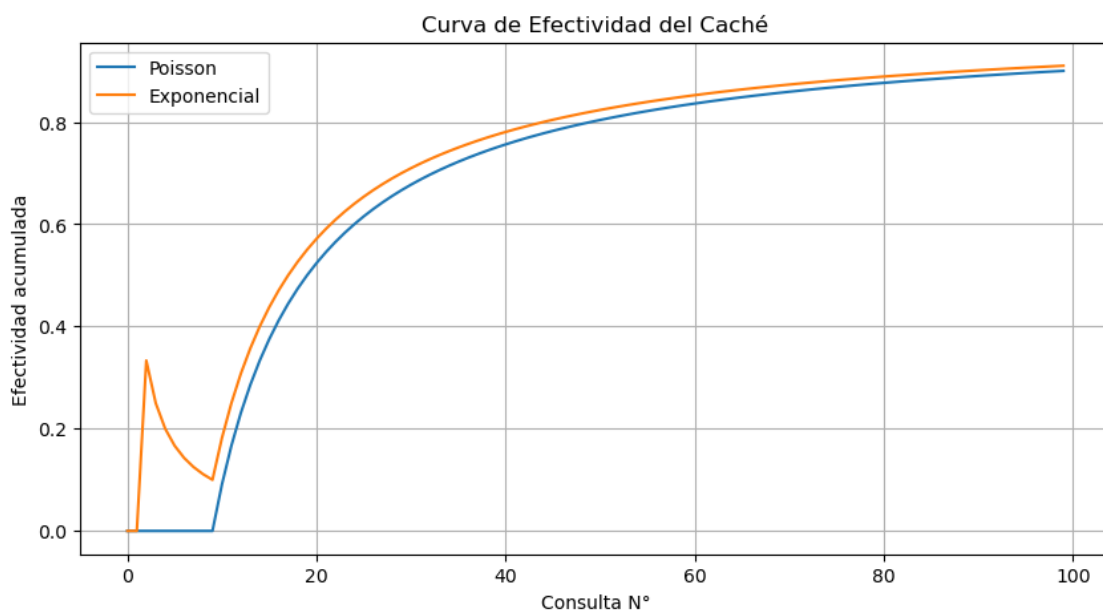
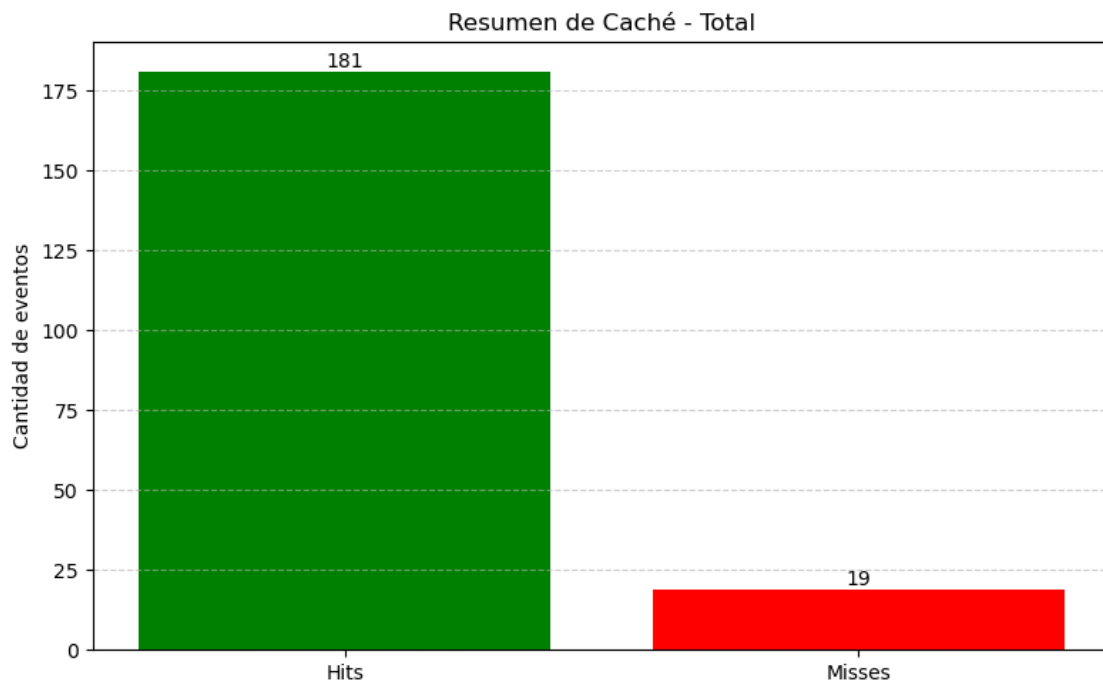
# Ejecutar todo el análisis
df_poisson, resumen_poisson, curva_poisson = simular_y_medir_cache("poisson",
↪tasa=5)
df_expon, resumen_expon, curva_expon = simular_y_medir_cache("exponencial",
↪tasa=0.2)

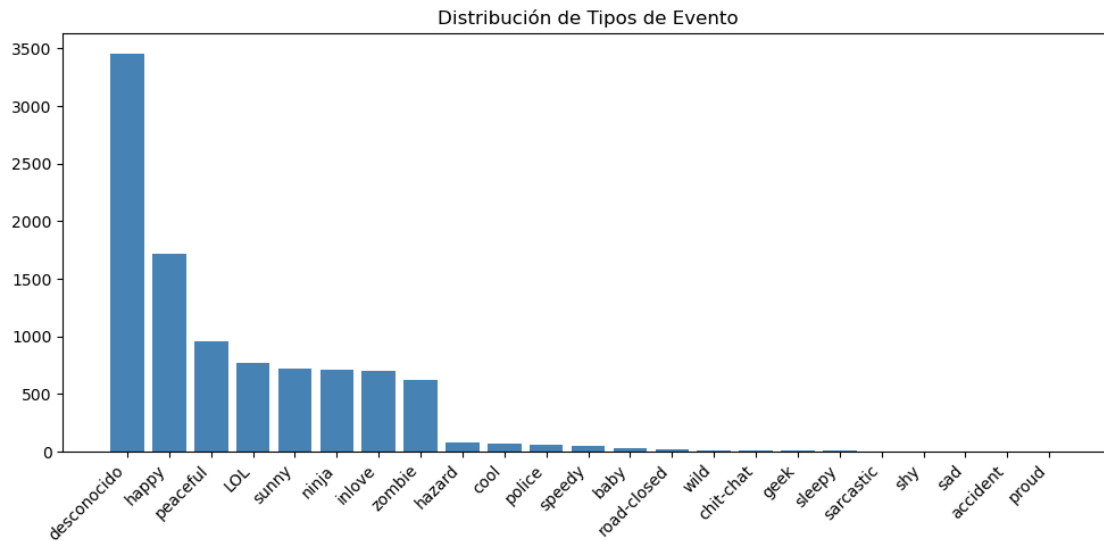
# Gráficas
grafico_resumen = graficar_cache(resumen_poisson["Hits"] +
↪resumen_expon["Hits"],
                                resumen_poisson["Misses"] +
↪resumen_expon["Misses"])
grafico_curva = graficar_curva(curva_poisson, curva_expon)
df_tipos, grafico_tipos = distribucion_tipos_evento()

```

```
# Exportar Excel
archivo_excel = exportar_resultados(df_poisson, df_expon, resumen_poisson,
↪resumen_expon)

# Mostrar paths para agregar en tu informe
grafico_resumen, grafico_curva, grafico_tipos, archivo_excel
```





```
[31]: ('resumen_cache_20250430_171836.png',
      'curva_cache_20250430_171836.png',
      'distribucion_tipos_evento_20250430_171836.png',
      'resultados_cache_20250430_171836.xlsx')
```

2.0.8 Extra: CRUD INTERNO

2.0.9 Extra: CRUD interno simulado

El sistema implementado también permite realizar operaciones **CRUD internas simuladas** sobre la base de datos de eventos, particularmente en la etapa de scraping:

- **Create (Alta):** cada evento detectado en el mapa de Waze es almacenado en la tabla **eventos** con sus atributos completos (tipo, ubicación, descripción, cuadrante, timestamp).
- **Read (Lectura):** el sistema de simulación extrae los datos tanto desde MySQL como desde Redis.
- **Update (Actualización):** si se desea modificar el TTL del caché o reemplazar un valor en Redis, es posible hacerlo desde el entorno Python usando `set()` o `expire()`.
- **Delete (Eliminación):** el caché puede ser limpiado manualmente (por ejemplo, con `flushall()`), o configurando una política de expiración automática en Redis.

Estas operaciones fueron implementadas de forma **implícita** dentro de los scripts de scraping, simulación y análisis, representando el flujo de datos de forma robusta sin necesidad de interfaces externas.

Esto demuestra que el sistema cumple con la estructura de una aplicación distribuida con componentes independientes (capa de datos, caché y lógica de procesamiento).

```
[34]: from funciones import menu_mysql
```

```
menu_mysql()
```

MENÚ DE CONSULTAS Y GESTIÓN DE TABLA 'eventos'

1. Total de eventos
2. Eventos por tipo
3. Eventos por cuadrante
4. Eventos por fecha
5. Ver tabla completa (limit 50)
6. Ver eventos por tipo específico
7. Eliminar TODOS los eventos
8. Eliminar eventos por tipo
9. Eliminar eventos por cuadrante
10. Exportar a Excel
11. Iniciar Docker Compose
12. Detener Docker Compose
13. Graficar
14. Eliminar tabla completa (elegir tabla)
0. Salir

Elige una opción [0-12]: 1

Total de eventos registrados:

```
total_eventos
0           10000
```

C:\Universidad\Sistemas_distribuidos\Proyecto\entregable1\funciones.py:21:

UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.

```
df = pd.read_sql(sql, conn, params=params)
```

```
[ ]: !docker compose down
```