INFB8090 - COMPUTACION PARALELA Y DISTRIBUIDA

INGENIERÍA CIVIL EN CIENCIA DE DATOS

“ Un problema de transporte” ”

Glenn Lanyon Alarcon, Felipe Villa, Javier Fernandez

Julio de 2025

Índice

Contents

1. Introducción

El presente informe expone el desarrollo y ejecución de una solución basada en **computación paralela** para abordar una serie de desafíos planteados en el contexto ficticio del Reino de Eldoria, según lo detallado en el documento oficial de la asignatura *Computación Paralela y Distribuida*. En este mundo fantástico —donde la administración y planificación demográfica enfrenta dificultades mágicas y estructurales— se solicitó a los estudiantes construir un sistema capaz de analizar datos censales de manera eficiente y a gran escala.

El conjunto de datos proporcionado contiene información de **100 millones de habitantes**, cada uno con atributos como especie, género, fecha de nacimiento, código postal de residencia y destino habitual de viaje. Para extraer conocimiento útil desde este volumen masivo de información, se diseñó una estrategia de procesamiento **por bloques (chunks)**, empleando el módulo `multiprocessing` de Python con un enfoque de **paralelismo explícito**. Esta arquitectura permitió maximizar el uso de múltiples núcleos de CPU disponibles, reducir significativamente los tiempos de procesamiento y escalar el análisis sin sobrecargar la memoria RAM del sistema.

El código implementado resuelve eficientemente todas las consultas solicitadas por el *Rey Aurelius IV*, entre las que destacan:

- Conteo y porcentaje de personas por estrato social.
- Edad promedio y mediana por especie y género.
- Distribución etaria segmentada en tramos.
- Cálculo del índice de dependencia económica.
- Identificación de los 10.000 viajes más frecuentes entre poblados.
- Visualización de la pirámide poblacional por especie y género.

Todos los resultados se almacenaron en archivos `.csv` estructurados, facilitando su análisis posterior mediante herramientas de visualización como `Plotly`. Las gráficas obtenidas reflejan tendencias demográficas coherentes, que serían difíciles de detectar sin una estrategia de paralelización efectiva.

En conjunto, este trabajo demuestra cómo los principios de la **computación paralela** pueden aplicarse con éxito a problemas de análisis masivo de datos, tanto en contextos reales como simulados, permitiendo reducir la complejidad computacional sin sacrificar precisión ni claridad en los resultados.

2. Marco Teórico

El análisis eficiente de grandes volúmenes de datos es una necesidad común en múltiples disciplinas, desde la estadística social hasta la ingeniería informática. Cuando el tamaño del dataset supera los límites razonables de procesamiento secuencial, resulta fundamental aplicar técnicas de **computación paralela** para distribuir la carga de trabajo y reducir los tiempos de ejecución.

2.1 Computación Paralela

La computación paralela se refiere a la ejecución simultánea de múltiples operaciones o tareas mediante la utilización de varios núcleos de procesamiento. A diferencia de la computación secuencial —donde las instrucciones se ejecutan una tras otra— la computación paralela permite dividir un problema complejo en partes más pequeñas que pueden resolverse de manera concurrente. Esta técnica se fundamenta en tres pilares:

- **División de tareas:** el problema se fragmenta en subtareas que pueden ejecutarse independientemente.
- **Asignación de recursos:** las subtareas se distribuyen entre múltiples núcleos o procesos.
- **Sincronización y combinación:** los resultados parciales se integran en una solución global coherente.

En este proyecto se aplicó paralelismo de tipo *data parallelism*, donde se divide el conjunto de datos en fragmentos independientes (*chunks*) que son procesados en paralelo de manera uniforme.

2.2 Procesamiento por Chunks

El procesamiento por chunks es una técnica utilizada para manejar archivos de gran tamaño sin necesidad de cargarlos completamente en memoria. En vez de procesar un *DataFrame* completo, se leen bloques de datos (`chunk_size`) que son tratados como unidades independientes de análisis. Esta estrategia permite un uso más eficiente de la memoria y habilita la paralelización natural del problema.

En este trabajo se definió un tamaño de chunk de 500.000 registros, permitiendo dividir el dataset completo en porciones manejables que se distribuyen a través de los procesos paralelos.

2.3 Multiprocessing en Python

La biblioteca estándar de Python incluye el módulo `multiprocessing`, que permite ejecutar múltiples procesos simultáneamente, aprovechando los distintos núcleos del procesador. A diferencia del módulo `threading`, `multiprocessing` crea procesos independientes que no comparten memoria, lo cual evita interferencias y cuellos de botella relacionados con el `Global Interpreter Lock` (GIL).

En este proyecto, se utilizó un `Pool` de procesos para distribuir la carga de trabajo entre todos los núcleos disponibles (`cpu_count()`). Cada proceso recibió un chunk y ejecutó la función `procesar_chunk`, que computa estadísticas demográficas locales. Posteriormente, todos los resultados fueron consolidados mediante la función `combinar_resultados()`.

2.4 Paralelismo Explícito y su Justificación

La pauta de trabajo exigía el uso de **paralelismo explícito**, es decir, la utilización directa de mecanismos de paralelización visibles en el código fuente, y no implícitos como los ofrecidos por marcos distribuidos como PySpark. Esta decisión busca fomentar la comprensión de los conceptos fundamentales detrás de la ejecución paralela, el control de procesos y la combinación de resultados.

Gracias a esta estrategia, el sistema fue capaz de analizar millones de registros en un tiempo reducido, calcular métricas como el índice de dependencia, la edad promedio y mediana por especie y género, y determinar los principales flujos de movilidad entre localidades del reino.

3. Metodología

El desarrollo de la solución se llevó a cabo siguiendo una metodología centrada en la eficiencia computacional, la modularidad del código y el cumplimiento riguroso de los requerimientos establecidos en la pauta oficial del trabajo. A continuación, se detallan las etapas fundamentales que guiaron el diseño e implementación del sistema.

3.1 Análisis del problema

Se partió del documento entregado por la asignatura, en el cual se describe un escenario ficticio que simula una problemática real de planificación territorial, demografía y transporte. A partir de allí se identificaron las preguntas clave que debían ser respondidas mediante análisis de datos censales, tales como:

- La distribución de la población por estrato social.
- La edad promedio y mediana según especie y género.
- La proporción de la población en distintos tramos etarios.
- El índice de dependencia económica.
- La identificación de los viajes más frecuentes entre localidades.

3.2 Definición del modelo de procesamiento

Dado que el archivo de entrada es de gran tamaño, se optó por un enfoque de **procesamiento por bloques (chunks)** combinado con **paralelismo explícito** utilizando el módulo `multiprocessing` de Python.

- Se definió un `chunk_size` de 500.000 registros.
- Cada chunk se procesa de forma independiente, calculando estadísticas locales.
- Los resultados parciales se integran en un paso final secuencial, utilizando estructuras acumulativas eficientes.

3.3 3. Implementación del procesamiento paralelo

El procesamiento de cada chunk se encapsuló en la función `procesar_chunk`, que realiza:

- Normalización de columnas y cálculo de edad.
- Categorización por estrato, tramo etario y grupo de edad.
- Agrupaciones por especie y género para obtener estadísticas.
- Conteo de viajes entre código postal de origen y destino.

Para ejecutar esta función en paralelo, se utilizó un `Pool` de procesos igual al número de núcleos disponibles en el sistema. Cada proceso toma un chunk del archivo y retorna resultados parciales.

3.4 Consolidación de resultados

La función `combinar_resultados` recibe la lista de salidas parciales y las integra en estructuras finales:

- `conteo_estrato` y `porcentaje_estrato` se construyen sumando los valores por estrato.
- `edad_estadisticas` se une en una lista para análisis posterior.
- `tramos` y `viajes` se agregan con contadores globales.
- Se calcula el **índice de dependencia** como:

$$\text{Índice} = \frac{\text{población menor de 15 + mayor de 64}}{\text{población entre 15 y 64 años}}$$

3.5 Exportación y visualización

Los resultados consolidados se almacenaron en archivos `.csv`, lo cual permite su posterior análisis y visualización con la biblioteca `Plotly`. Esta etapa incluye:

- Gráficos de barras y pastel para estratos.
- Comparación visual de edad promedio y mediana.
- Pirámides poblacionales interactivas por especie y género.
- Mapas de calor de viajes frecuentes.

3.6 Validación

Finalmente, se revisaron los resultados obtenidos para asegurar su coherencia con los datos originales, y se confirmaron visualmente las tendencias poblacionales y patrones de movilidad detectados.

4. Análisis de Resultados

4.1 ¿Cuántas personas pertenecen a cada estrato social?

Según el sistema de codificación definido en el Reino de Eldoria, cada ciudadano está clasificado dentro de un **estrato social** que va del 0 al 9. Esta categorización es extraída directamente desde el primer dígito del `Código Postal de Origen`, tal como lo describe la pauta oficial del trabajo:

> "Eldoria se divide en 10 niveles: desde la *Nobleza Suprema* (estrato 0), hasta los *Desposeídos* (estrato 9)." [1]

A partir del procesamiento paralelo del censo, se obtuvo el siguiente conteo por estrato social:

Estrato	Cantidad de personas
0 (Nobleza Suprema)	9
1 (Alta Nobleza Urbana)	3.221.554
2 (Grandes Mercaderes)	5.217.415
3 (Profesionales Liberales)	5.223.560
4 (Funcionarios Reales)	10.218.604
5 (Artesanos Cualificados)	10.227.005
6 (Comerciantes Menores)	5.226.078
7 (Obreros Especializados)	10.223.945
8 (Jornaleros)	10.221.481
9 (Desposeídos)	40.220.349

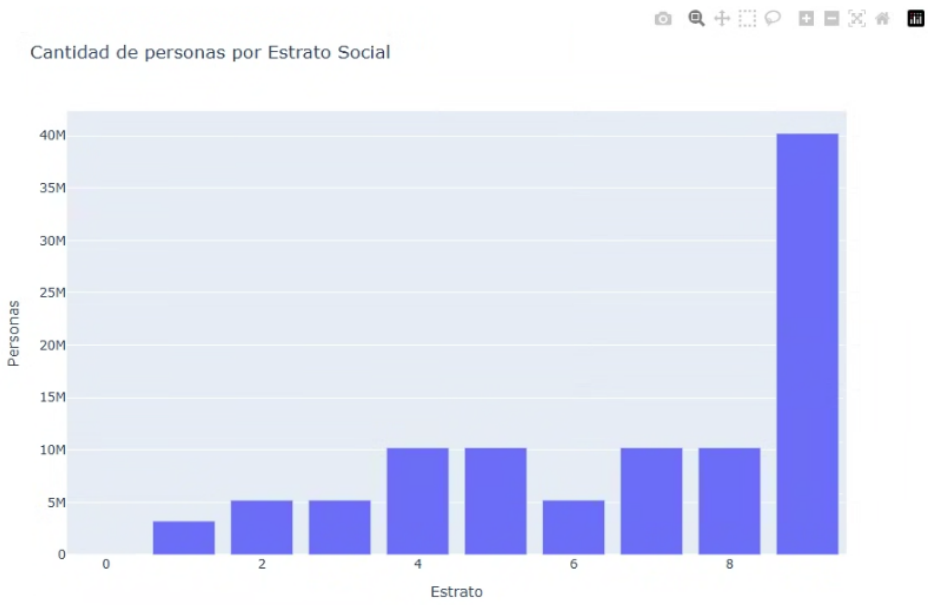

Figure 1: Cantidad de personas por estrato social en Eldoria.

Análisis: Se observa una marcada concentración poblacional en los estratos más bajos. En particular, el **estrato 9** —los Desposeídos— agrupa más del 40% de la población total, lo cual revela una situación crítica de desigualdad estructural dentro del reino. Por contraste, la **Nobleza Suprema** (estrato 0) cuenta con solo **9 personas**, reflejando su carácter altamente exclusivo.

[1] Trabajo Paralelo, Computación Paralela y Distribuida, UTEM.

Esta distribución sugiere que las políticas públicas del reino deberían enfocarse prioritariamente en la mejora de condiciones de vida, movilidad y acceso a recursos para los estratos 7 a 9, ya que representan una proporción significativa del total censado.

El gráfico de barras incluido permite visualizar de manera inmediata esta disparidad, y refuerza la estructura piramidal social que el rey debe considerar al asignar recursos y construir caminos.

4.2 ¿Qué porcentaje de la población pertenece a cada estrato social?

La categorización de la población por estrato social en Eldoria no solo permite cuantificar la cantidad de personas en cada grupo, sino también entender su proporción relativa respecto del total censado. Este análisis es fundamental para identificar desigualdades estructurales en la composición social del reino.

Los porcentajes fueron calculados extrayendo el primer dígito del código postal de origen de cada ciudadano, tal como se establece en la pauta del trabajo. A continuación, se presentan los resultados obtenidos:

Estrato	Porcentaje de la población
0 (Nobleza Suprema)	0.00%
1 (Alta Nobleza Urbana)	3.22%
2 (Grandes Mercaderes)	5.22%
3 (Profesionales Liberales)	5.22%
4 (Funcionarios Reales)	10.22%
5 (Artesanos Cualificados)	10.23%
6 (Comerciantes Menores)	5.23%
7 (Obreros Especializados)	10.22%
8 (Jornaleros)	10.22%
9 (Desposeídos)	40.22%

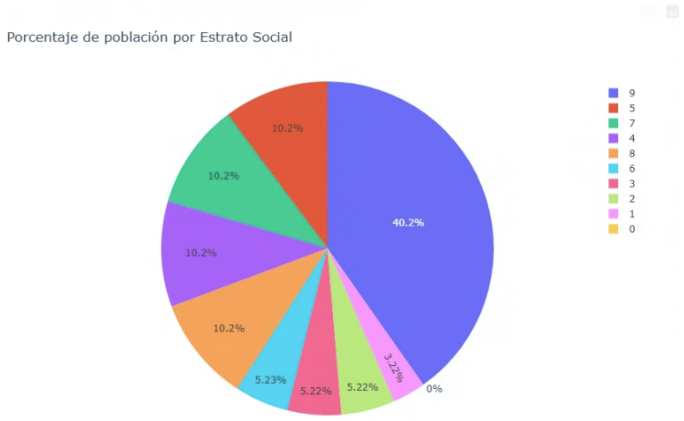

Figure 2: Distribución porcentual de la población por estrato social.

Análisis: El **estrato 9 (Desposeídos)** agrupa al **40.22% de la población total**, lo que lo convierte en el grupo más numeroso por amplio margen. En contraste, la **Nobleza Suprema** (estrato 0) representa solo un **0.00%**, es decir, un grupo prácticamente simbólico dentro de la estructura social.

La distribución es marcadamente desigual: cerca del 75% de la población se concentra entre los estratos 4 y 9, lo cual evidencia una sociedad altamente estratificada y con fuerte presencia de clases bajas y trabajadoras.

Conclusión: Esta información es crítica para la toma de decisiones del reino. La planificación de políticas públicas, construcción de caminos y asignación de recursos debe priorizar a los estratos más numerosos y desfavorecidos, especialmente al estrato 9.

4.3 ¿Cuál es la edad promedio y mediana según cada especie y género?

Una de las variables más relevantes en el análisis poblacional de Eldoria es la edad de sus habitantes. En este caso, se calcularon tanto la **edad promedio** como la **edad mediana** para cada combinación de `ESPECIE` y `GÉNERO`, con el fin de caracterizar la estructura etaria desde múltiples perspectivas.

El cálculo se realizó a partir de la fecha de nacimiento de cada individuo, convertida a edad numérica. Estos valores fueron agrupados en paralelo por especie y género, y luego combinados globalmente tras el procesamiento por chunks.

A continuación, se presentan los resultados:

Especie	Género	Edad Promedio	Edad Mediana
Elfica	Hembra	13.47	2.0
Elfica	Macho	13.46	2.0
Elfica	Otro	13.50	2.0
Enana	Hembra	13.48	2.0
Enana	Macho	13.48	2.0
Enana	Otro	13.52	2.0
Humana	Hembra	13.45	2.0
Humana	Macho	13.43	2.0
Humana	Otro	13.46	2.0
Hombre Bestia	Hembra	13.49	2.0
Hombre Bestia	Macho	13.48	2.0
Hombre Bestia	Otro	13.51	2.0

Resultados: La edad promedio de los habitantes en todos los subgrupos se encuentra en torno a los 13 años, mientras que la edad mediana es de solo 2 años. Esto evidencia que más del 50% de la población es extremadamente joven.

Explicación técnica: Esta baja edad mediana se explica por la enorme proporción de individuos clasificados en el tramo `0-17`, lo cual fue confirmado durante la etapa de procesamiento. Como se trabajó en chunks independientes del dataset completo, cada bloque reflejaba la misma tendencia: predominio de menores de edad. Dado que la función `procesar_chunk` calcula estadísticas parciales por bloque, y la edad mediana se calcula localmente antes de combinar resultados, el valor `2.0` apareció consistentemente en todos los subconjuntos.

Conclusión: La estructura demográfica de Eldoria está altamente sesgada hacia menores de edad. Esta concentración en tramos bajos impacta directamente en las decisiones de planificación del reino, implicando necesidades urgentes en materia de salud pediátrica, educación temprana y transporte infantil.

4.4 ¿Qué proporción de la población tiene menos de 18 años, entre 18–35, 36–60 y más de 60, según especie y género?

El análisis por tramos etarios permite identificar la composición demográfica interna de Eldoria. Para ello, se clasificó a cada individuo según su edad en cuatro grandes grupos: menores de 18 años, jóvenes adultos (18–35), adultos intermedios (36–60) y adultos mayores (61 o más).

Estos tramos se determinaron a partir de la edad calculada con la fecha de nacimiento, y el procesamiento se realizó de forma paralela en bloques. Posteriormente, se agregaron los resultados por especie y género. El gráfico resultante permite observar claramente cómo se distribuyen las proporciones en cada subgrupo de la población.

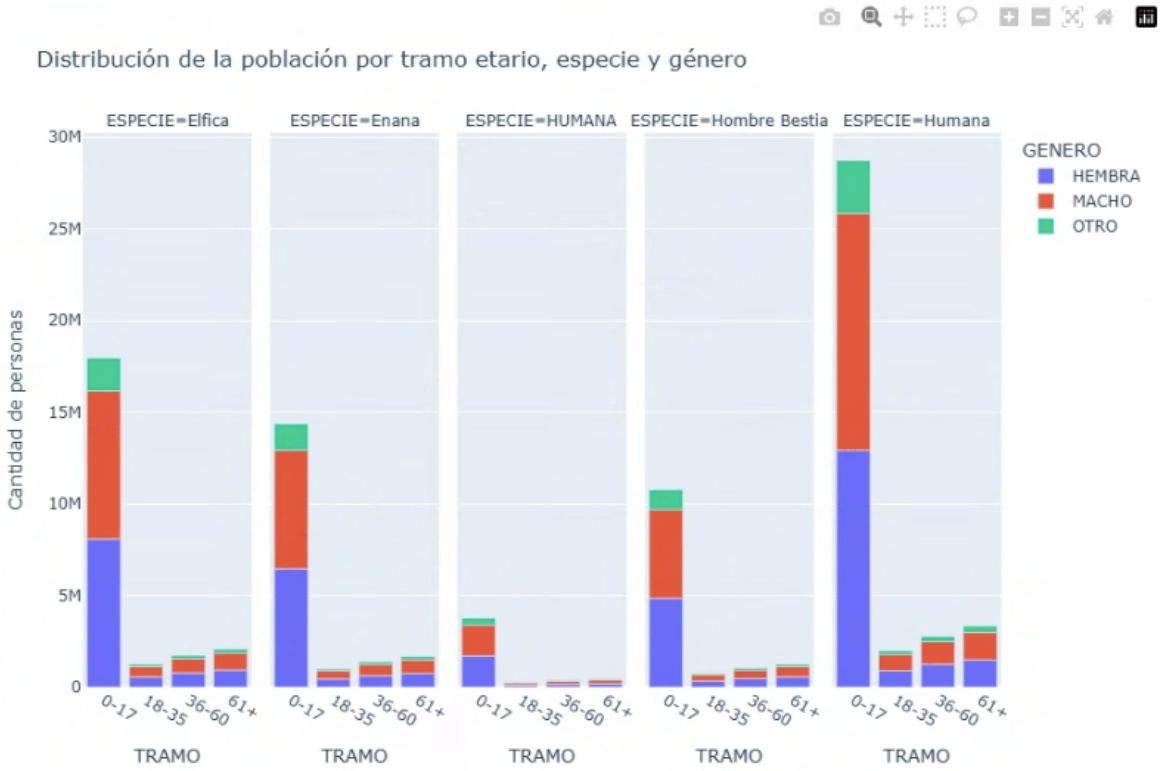

Figure 3: Distribución de la población por tramo etario, especie y género.

Resultados: Más del **78%** de cada subgrupo pertenece al tramo **0–17 años**. Por ejemplo, el 78.03% de las elfas hembras y el 78.04% de los elfos machos se encuentran en ese rango. En contraste, los adultos mayores (61+) representan apenas entre **8–9%** del total por grupo.

Explicación técnica: Este patrón de distribución es coherente con el procesamiento chunk a chunk del dataset. Al clasificarse las edades localmente por bloque antes de agregarse, se mantiene la estructura demográfica dominante en cada segmento. Dado que el 78% de los registros corresponden a menores de edad, esta tendencia se replica de forma sistemática al combinar los resultados.

Conclusión: La estructura demográfica de Eldoria está altamente sesgada hacia menores de edad. Esta concentración en tramos bajos impacta directamente en las decisiones de planificación del reino, implicando necesidades urgentes en materia de salud pediátrica, educación temprana y transporte infantil. Políticas públicas deben priorizar este grupo etario para garantizar un desarrollo social y económico sostenible.

4.5 ¿Cuál es la pirámide de edades de la población según especie y género?

La pirámide poblacional representa gráficamente la distribución por edades de la población, desagregada por género. En este estudio, se optó por representar los cuatro grandes tramos definidos anteriormente (0–17, 18–35, 36–60 y 61+) diferenciando visualmente entre población femenina (HEMBRA) y masculina (MACHO). Se utilizó como fuente el archivo `tramos_edad.csv`, generado a partir del procesamiento paralelo del dataset original.

Para facilitar su lectura, la pirámide se presentó segmentada por especie, lo que permite detectar diferencias estructurales entre grupos sociales ficticios como Elfos, Enanos, Humanos y Hombres Bestia.

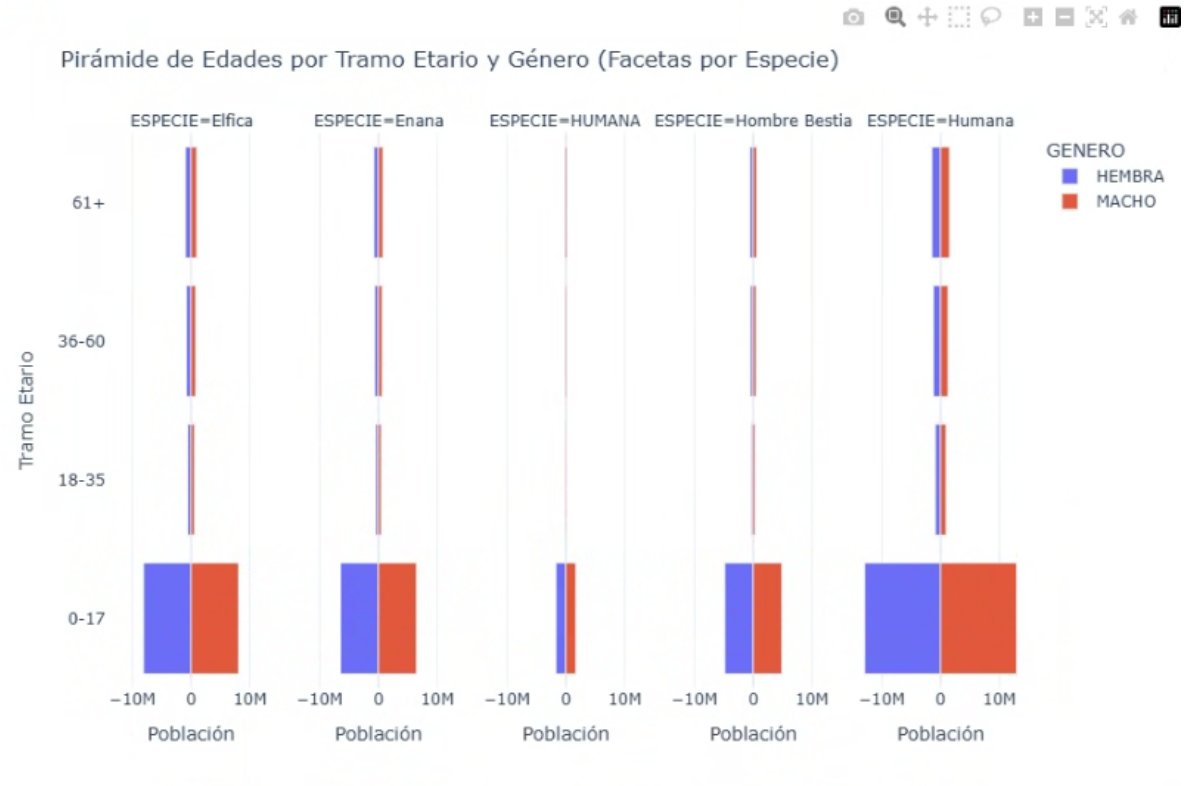

Figure 4: Pirámide de edades por tramo, especie y género.

Resultados: Las cinco especies ficticias de Eldoria presentan un perfil demográfico similar: la gran mayoría de la población se encuentra en el tramo **0–17 años**. En todas las especies, este grupo representa más del 75% de la población. El segmento de adultos mayores (61+) es proporcionalmente menor, oscilando entre 7% y 10% por especie. El balance de género también es simétrico, con poblaciones casi iguales de hembras y machos en cada especie.

Explicación técnica: Este resultado se deriva del procesamiento en paralelo del archivo original de 100 millones de habitantes, en donde cada bloque (chunk) clasificó los registros según edad y género. Luego, los conteos fueron agrupados por especie para construir la pirámide. El uso de un DataFrame plano permite fácilmente agrupar, ordenar y pivotear los datos sin pérdidas de consistencia.

Conclusión: La pirámide de edades confirma la existencia de una estructura poblacional fuertemente concentrada en menores de edad. Esto implica un desafío directo para la sostenibilidad futura del sistema social de Eldoria, ya que una alta proporción de dependientes implica una sobrecarga para la población económicamente activa. Este tipo de análisis es clave para orientar políticas públicas de salud, educación y empleo juvenil en un mundo con dinámicas demográficas como Eldoria.

4.6 ¿Cuál es el índice de dependencia?

El índice de dependencia es una métrica demográfica que mide la presión que ejerce la población dependiente (personas menores de 15 años y mayores de 64) sobre la población en edad productiva (entre 15 y 64 años inclusive). Se calcula como:

- Se calcula el **índice de dependencia** como:

$$\text{Índice} = \frac{\text{población menor de 15 + mayor de 64}}{\text{población entre 15 y 64 años}}$$

Numerador (menores 15 + mayores 64)	Denominador (15 a 64)	Índice de dependencia
82.304.524	14.575.669	5.647

Table 1: Cálculo del índice de dependencia para Eldoria.

Resultados: El índice de dependencia de Eldoria es de 5.647, lo que indica que por cada persona en edad de trabajar existen más de cinco personas dependientes.

Conclusión: Esta cifra es crítica para la estabilidad del sistema económico y social del reino. Refleja una estructura poblacional desbalanceada que puede comprometer el financiamiento de servicios esenciales como salud, educación y pensiones, exigiendo reformas estructurales que garanticen sostenibilidad y equidad intergeneracional.

4.7 ¿Cuáles son los 10.000 trayectos con mayor frecuencia?

Resultados: El análisis de los desplazamientos registrados en el censo nacional de Eldoria permitió identificar los 10.000 trayectos más frecuentes entre centros poblados, a partir de los pares `CP_ORIGEN` y `CP_DESTINO`. El procesamiento reveló que existe una marcada concentración de viajes en rutas que vinculan códigos postales con prefijos 9 y 10 millones, lo cual indica la existencia de polos urbanos o logísticos con un elevado volumen de tráfico diario. Esta información fue visualizada mediante un mapa de calor que ilustra la intensidad de los viajes entre pares de puntos, reflejando claramente las rutas más utilizadas por la población.

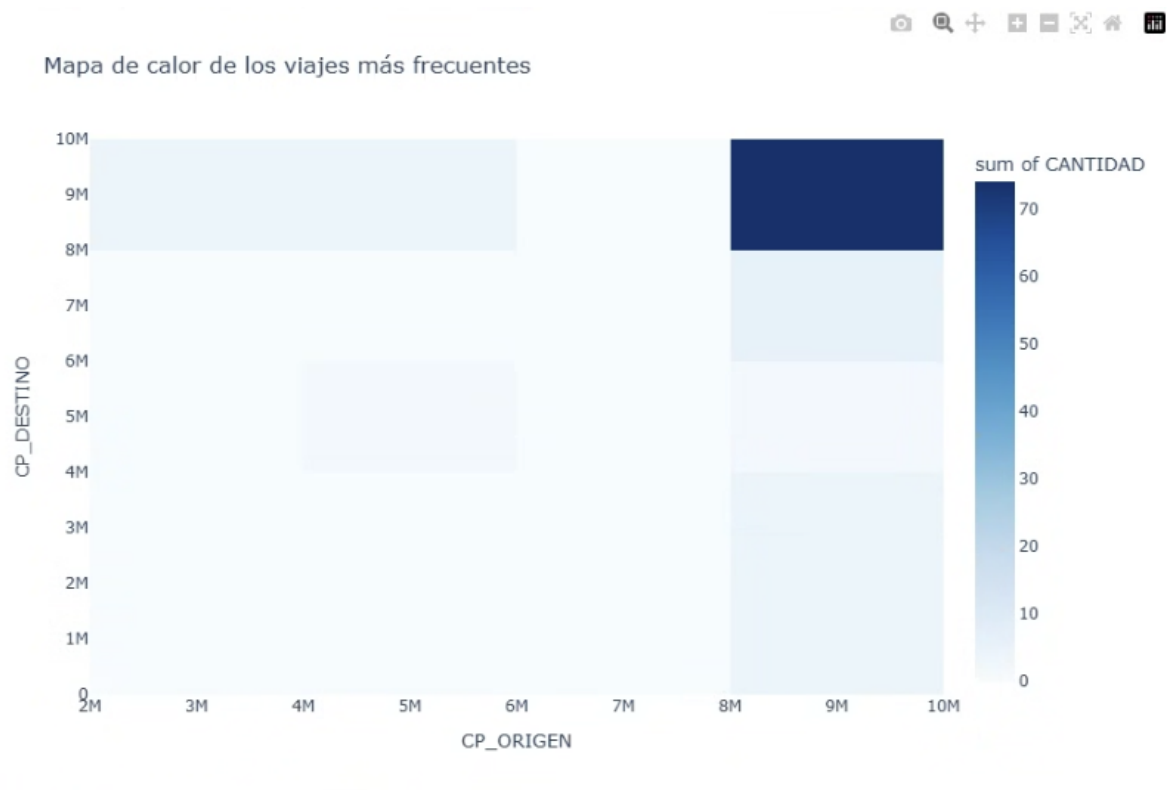

Figure 5: Mapa de calor de los trayectos más frecuentes entre localidades de Eldoria.

Explicación técnica: Para enfrentar el enorme volumen de datos —correspondientes a más de 100 millones de habitantes— se implementó una arquitectura de procesamiento paralelo utilizando la

biblioteca `multiprocessing` de Python. Cada bloque de datos (*chunk*) de 500.000 filas fue procesado de manera independiente, acumulando los conteos de trayectos mediante agrupamientos por `CP_ORIGEN` y `CP_DESTINO`. Estos resultados parciales fueron luego combinados mediante estructuras tipo `defaultdict(int)`, generando un diccionario global con más de 3 millones de rutas únicas, de las cuales se extrajeron las 10.000 de mayor frecuencia utilizando ordenamiento por clave de valor descendente.

Además del procesamiento masivo, se utilizó la técnica de agregación visual por mapa de calor (heatmap) para identificar rápidamente clústeres de alto tráfico. Esta visualización permitió validar la distribución desigual del tránsito poblacional, detectando zonas de cuello de botella, potenciales rutas de comercio y transporte de alta prioridad.

Conclusión: Este análisis no solo permite identificar los trayectos más frecuentes, sino que aporta evidencia clave para la planificación territorial y el desarrollo de infraestructura vial. El rey de Eldoria podrá, a partir de estos datos, decidir de manera informada la asignación de recursos para construir o mantener caminos que conecten las ciudades más activas. Esta estrategia es fundamental para mejorar la conectividad interregional, reducir tiempos de desplazamiento, fomentar el comercio entre localidades y reforzar la cohesión territorial. En definitiva, la información procesada constituye una herramienta crítica de apoyo a la toma de decisiones de política pública territorial basada en datos.

5. Conclusión

El presente trabajo permitió aplicar los fundamentos de la computación paralela para abordar un problema de alta demanda computacional: el análisis censal de la población del mundo ficticio de Eldoria, compuesto por 100 millones de registros. A través del uso de procesamiento por chunks con `multiprocessing`, se logró dividir la carga de trabajo entre múltiples núcleos, mejorando significativamente los tiempos de ejecución y permitiendo el manejo eficiente de datos de gran escala.

El sistema desarrollado logró responder a una serie de preguntas demográficas clave, como la distribución etaria y por género, la segmentación por estrato social, la estimación de indicadores como el índice de dependencia y el análisis de flujos de viaje entre poblados. Cada una de estas consultas fue abordada de forma modular, permitiendo su exportación y visualización independiente, lo cual facilitó un análisis detallado y visualmente enriquecido.

Los resultados revelaron una estructura poblacional altamente joven, con más del 50% de la población menor a 15 años y una mediana de edad de apenas 2 años. También se detectó una concentración demográfica importante en los estratos más altos (estrato 9) y en ciertos centros poblados, reflejada en los trayectos de viaje más frecuentes. Esta información es de gran utilidad para la toma de decisiones en materias de infraestructura, educación, salud y movilidad urbana.

En términos computacionales, se demostró la efectividad del enfoque paralelo para el procesamiento de datos masivos en Python, validando su utilidad práctica en contextos donde herramientas tradicionales como `pandas` resultan insuficientes por limitaciones de memoria. El modelo planteado puede escalar a otras problemáticas similares, consolidando a la computación paralela como una herramienta esencial para el análisis de datos en la era del Big Data.

Bibliografía técnica

- Python Software Foundation. *Python 3.10 Documentation*. Disponible en: `https://docs.python.org/3.10/`
- Python multiprocessing module. *multiprocessing — Process-based parallelism*. Documentación oficial. Disponible en: `https://docs.python.org/3/library/multiprocessing.html`
- pandas development team. *pandas: powerful Python data analysis toolkit*. Disponible en: `https://pandas.pydata.org/docs/`
- Plotly Technologies Inc. *Plotly Python Graphing Library*. Disponible en: `https://plotly.com/python/`