

Segunda Parte del Proyecto Final

LEA BIEN las indicaciones :

EVALUACIÓN:

- La evaluación se realizará el día miércoles 25 en horas de laboratorio en PUNTO:
- El protocolo de evaluación es el siguiente:
 - Los grupos se revisarán por orden en laboratorio, se empezará por la derecha o izquierda. (si es posible podrían empezar a presentar antes de la hora)
 - Se elegirá un alumno al azar de entre los 2 integrantes
 - El alumno seleccionado anteriormente deberá responder una pregunta directa sobre la implementación(MANUAL) del código.
 - Finalmente se mostrará los resultados alcanzados del grupo (el proyecto debería estar precargado, si en ese momento se ponen a cargar el proyecto se pasará a otro grupo)

OBJETIVOS:

5. Propiedades y Métricas de la Red

- Detección de **Comunidades**: Implementar los algoritmos como Girvan-Newman o Louvain para detectar comunidades dentro de la red.(sin uso de librerías)

6. Análisis Avanzado

- Análisis de **Camino Más Corto**: Calcular la longitud promedio del camino más corto. (sin uso de librerías)
- Árboles de expansión Mínima (sin uso de librerías)

7. Visualización

- Visualizaciones **Interactivas**: Crear visualizaciones interactivas usando Plotly o Gephi para explorar la red.
- Visualización de **Comunidades**: Visualizar las comunidades detectadas con diferentes colores para resaltar la estructura de la comunidad.

RÚBRICA:

Puntaje total: 20 puntos (5 criterios, 4 puntos cada uno)

1. Carga Masiva Eficiente (Python)(mejoraron la carga ?)

- **Excelente (4 pts)**: Carga rápida, robusta y eficiente de 10M registros. Uso de procesamiento en lote, stream o paralelismo, formatos más ligeros.
- **Bueno (3 pts)**: Funcional con eficiencia razonable. Puede mejorar en tiempos o escalabilidad.
- **Aceptable (2 pts)**: Carga muy lenta o no optimizada.
- **Insuficiente (1 pt)**: No completa la carga o presenta fallos graves.

2. Visualización

- **Excelente (4 pts):** Excelente manejo de memoria, fluidez en la visualización de una gran cantidad de nodos.(uso de tecnología adicional por ejemplo GPU)
- **Bueno (3 pts):** Manejo adecuado de memoria pero puede fallar con una cantidad masiva de nodos.
- **Aceptable (2 pts):** Uso ineficiente de visualización.
- **Insuficiente (1 pt):** Sin manejo de memoria ni control de recursos.

3. Propiedades y Métricas de la Red(Análisis avanzado)

- **Excelente (4 pts):** Completo: incluye aplicación de algoritmos de camino más corto y comunidades.
- **Bueno (3 pts):** Básico, camino más corto.
- **Aceptable (2 pts):** Incompleto, superficial o con errores conceptuales.
- **Insuficiente (1 pt):** Solo muestra datos.

4. Legibilidad y Calidad del Código

- **Excelente (4 pts):** Código comprensible, presentación clara de código(por alumno).
- **Bueno (3 pts):** Código medianamente comprensible, con algunas áreas de mejora en estilo o estructura.
- **Aceptable (2 pts):** Código funcional pero poco legible o mal organizado.
- **Insuficiente (1 pt):** Código desordenado, sin comentarios ni estructura clara.

5. Documentación y Presentación

- **Excelente (4 pts):** README claro, instrucciones de uso, descripción del flujo
 - **Bueno (3 pts):** Documentación suficiente para reproducir el proyecto.
 - **Aceptable (2 pts):** Documentación parcial o poco clara.
 - **Insuficiente (1 pt):** No hay documentación útil o comprensible.
-