

Aquí tienes un **README.md** breve y claro que explica cómo ejecutar el script **run.sh**, qué hace y cómo usar la opción **--pasos**:

Agrupamiento Jerárquico Aglomerativo

Este proyecto implementa un algoritmo de **clustering jerárquico aglomerativo** en C++, con visualización de resultados paso a paso usando Python.

🛠️ Requisitos

- `g++` (C++17 o superior)
- Python 3 con las siguientes bibliotecas:
 - `matplotlib`
 - `numpy`
 - `scipy`

Puedes instalar las dependencias de Python con:

```
```bash
pip install matplotlib numpy scipy
```

## 🚀 Ejecución del script

Para compilar, ejecutar y graficar los resultados, usa el script:

```
./run.sh archivo_entrada.txt [--pasos]
```

## 🔧 Parámetros

- **archivo\_entrada.txt**: archivo de entrada con la matriz de distancias.
- **--pasos** (*opcional*): muestra las gráficas paso a paso del clustering.

## 🧠 Ejemplo de uso

```
./run.sh datos_ejemplo.txt --pasos
```

Este comando:

1. Limpia las carpetas de imágenes.
2. Compila el código C++ (**main.cpp**).
3. Ejecuta el programa con el archivo de entrada.

4. Genera gráficas por cada método (**single**, **complete**, **average**), incluyendo pasos si se usa **--pasos**.

## Salida

- Los resultados visuales se guardan en la carpeta **img/**, organizados por método y paso.

## Notas

- Si el archivo de entrada no existe, el script se detiene con un mensaje de error.
- Si no se encuentra un archivo **.json** esperado (resultado del programa en C++), el método correspondiente será omitido con advertencia.

---

¿Quieres que también incluya un ejemplo del formato del archivo de entrada (``archivo_entrada.txt``)?