1. Estructura General del Programa

El programa se basa en una clase llamada ConsolaInteractiva, que administra todos los comandos disponibles y su funcionamiento. Al iniciar el programa, el usuario ve un símbolo de $, indicando que puede ingresar un comando. El sistema espera que el usuario escriba un comando y, según lo que se ingrese, se ejecuta la acción correspondiente.

Para que esto funcione, el programa utiliza un mapa de comandos (unordered_map) que asocia cada palabra clave (como cargar_imagen, info_imagen, etc.) con una función específica. Esto permite que el programa sea fácilmente extensible: para añadir un nuevo comando, simplemente se agrega una entrada al mapa con la palabra clave y la función correspondiente.

2. Comandos Implementados

El sistema cuenta con una serie de comandos para interactuar con las imágenes y el volumen de imágenes. Estos comandos son:

ayuda: Muestra una lista de todos los comandos disponibles o información detallada de un comando específico. Esto facilita al usuario aprender a usar el sistema sin tener que revisar el código fuente.

salir: Finaliza la ejecución del programa de manera ordenada. Simplemente imprime un mensaje de despedida y utiliza exit(0) para cerrar el programa.

clr: Limpia la pantalla de la consola, mejorando la experiencia del usuario al permitir un área de trabajo más organizada.

cargar_imagen: Este comando carga una imagen en formato PGM en la memoria del programa. Si ya hay una imagen cargada, esta se reemplaza por la nueva. Se verifica el formato del archivo para asegurarse de que sea compatible antes de cargarla.

info_imagen: Muestra el nombre, ancho y alto de la imagen actualmente cargada. Si no hay una imagen cargada, el sistema informa al usuario del problema.

proyeccion2D: Aunque el funcionamiento interno no está completamente desarrollado, la intención de este comando es generar una proyección 2D de una imagen o volumen cargado y guardarla como un archivo PGM.

decodificar_archivo: Este comando está diseñado para decodificar un archivo comprimido (.huf) y convertirlo de nuevo en una imagen PGM. Se verifica la existencia del archivo antes de proceder, y si ocurre algún problema, se informa al usuario.

cargar_volumen y info_volumen: Se incluyen comandos para cargar y obtener información de un volumen de imágenes. Aunque la funcionalidad detallada no está implementada, el esqueleto del comando permite expandir el proyecto en el futuro.

codificar_imagen: Este comando se planificó para codificar una imagen utilizando el algoritmo de Huffman y guardarla como un archivo .huf.

segmentar: Esta funcionalidad tiene como objetivo segmentar una imagen utilizando semillas y grafos, lo cual es útil para separar objetos dentro de una imagen.

3. Procesamiento de Comandos

Cuando el usuario ingresa un comando, el programa:

Divide la entrada en palabras clave: Se utiliza istringstream para separar la línea completa en palabras (tokens). La primera palabra se considera el comando, mientras que las siguientes son sus argumentos.

Busca el comando en el mapa: Utiliza la palabra clave para buscar en el mapa de comandos. Si el comando existe, se ejecuta la función correspondiente; de lo contrario, se muestra un mensaje de error indicando que el comando no es reconocido.

Ejecuta la función asociada: Cada comando está asociado a una función que recibe los argumentos del usuario como un vector de strings (vector<string>). Esto permite al programa manejar comandos con distintos números de parámetros de manera flexible.