**Plan de Pruebas para el Procesador de Imágenes**

**Objetivo**

Verificar el correcto funcionamiento de todas las funcionalidades del sistema, incluyendo la consola interactiva, procesamiento de imágenes, manejo de volúmenes y codificación Huffman.

**Estrategia de Pruebas**

1. **Pruebas unitarias**: Verificar cada componente individualmente
2. **Pruebas de integración**: Verificar la interacción entre componentes
3. **Pruebas del sistema**: Verificar el flujo completo desde la consola
4. **Pruebas de rendimiento**: Evaluar el comportamiento con datos grandes

**1. Pruebas de la Consola Interactiva**

**Comandos básicos**

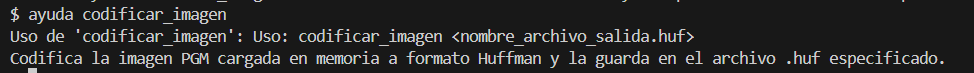
* ayuda: Verificar que muestra todos los comandos disponibles

Texto

El contenido generado por IA puede ser incorrecto.

1. **Descripción**: Verificar que el comando ayuda muestra todos los comandos disponibles.
2. **Entrada**: ayuda
3. **Resultado esperado**: Lista completa de comandos con una breve descripción de cada uno.
4. **Resultado obtenido**: Se mostró la lista de comandos correctamente.
5. **Estado**: Éxito.

* ayuda <comando>: Verificar ayuda específica para cada comando



1. **Descripción**: Verificar que ayuda <comando> muestra la ayuda específica para un comando.
2. **Entrada**: ayuda cargar\_imagen
3. **Resultado esperado**: Descripción detallada del comando cargar\_imagen, incluyendo parámetros y ejemplos.
4. **Resultado obtenido**: Se mostró la ayuda específica correctamente.
5. **Estado**: Éxito

* salir: Verificar que termina el programa correctamente

Texto

El contenido generado por IA puede ser incorrecto.

1. **Descripción**: Verificar que el comando salir termina el programa.

2. **Entrada**: salir

3. **Resultado esperado**: El programa se cierra sin errores.

4. **Resultado obtenido**: El programa terminó correctamente.

5. **Estado**: Éxito

**1.2 Manejo de errores**

* Comando inexistente: Verificar mensaje de error apropiado



**Prueba: Comando inexistente**

1. **Descripción**: Verificar que se muestra un mensaje de error al ingresar un comando inexistente.
2. **Entrada**: comando\_inexistente
3. **Resultado esperado**: Mensaje de error: "comando\_inexistente no es un comando reconocido. usa 'ayuda' para ver los comandos disponibles."
4. **Resultado obtenido**: Se mostró el mensaje de error esperado.
5. **Estado**: Éxito

* Argumentos incorrectos: Verificar validación de parámetros

Texto

El contenido generado por IA puede ser incorrecto.

1. **Descripción**: Verificar el numero y que los argumentos estén correctos.
2. **Entrada**: ayuda comando\_invalido
3. **Resultado esperado**: Mensaje de error: "no hay ayuda disponible para comando\_invalido
4. **Resultado obtenido**: Se mostró el mensaje de error esperado.
5. **Estado**: Éxito

**2. Pruebas de Manejo de Imágenes (PGM)**

**Carga de imágenes**

* cargar\_imagen imagen\_valida.pgm: Verificar carga correcta

Texto

El contenido generado por IA puede ser incorrecto.

1. **Descripción**: Verificar que una imagen PGM válida se carga correctamente.
2. **Entrada**: cargar\_imagen imagen\_valida.pgm
3. **Resultado esperado**: Mensaje de confirmación: "la imagen imagen\_valida.pgm ha sido cargada"
4. **Resultado obtenido**: La imagen se cargó sin errores.
5. **Estado**: Éxito

* info\_imagen: Verificar información mostrada después de carga

Texto

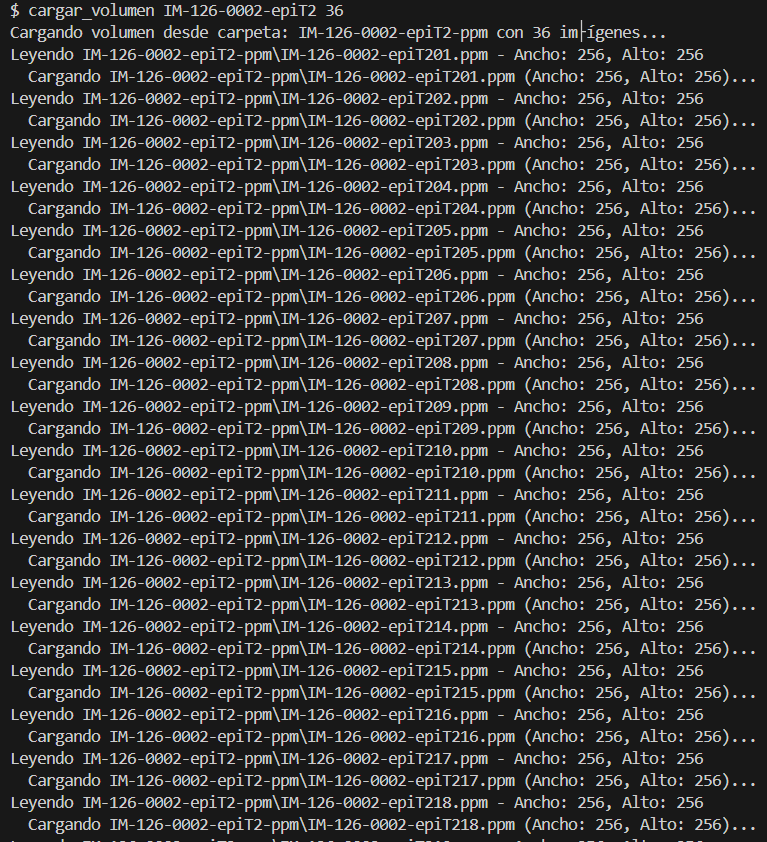
El contenido generado por IA puede ser incorrecto.

1. **Descripción**: Verificar que el comando info\_imagen muestra los metadatos de la imagen cargada.
2. **Entrada**: info\_imagen
3. **Resultado esperado**: Información como ancho, alto y valor máximo de píxeles.
4. **Resultado obtenido**: Se mostró la información correctamente.
5. **Estado**: Éxito

**3. Pruebas de Volúmenes**

**3.1 Carga de volúmenes**

* cargar\_volumen base\_valida: Verificar carga correcta



* 1. **Descripción**: Verificar que un volumen válido se carga correctamente.
  2. **Entrada**: cargar\_volumen base\_valida
  3. **Resultado esperado**: Mensaje de confirmación: "Volumen cargado correctamente con 10 imágenes."
  4. **Resultado obtenido**: El volumen se cargó sin errores.
  5. **Estado**: Éxito
* cargar\_volumen base\_inexistente : Verificar mensaje de error

Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

1. **Descripción**: Verificar que se muestra un error al cargar un volumen inexistente.
2. **Entrada**: cargar\_volumen base\_inexistente
3. **Resultado esperado**: Mensaje de error: "Error: No se encontró la base especificada."
4. **Resultado obtenido**: Se mostró el mensaje de error esperado.
5. **Estado**: Éxito

* info\_volumen: Verificar información mostrada



**3.2 Proyecciones 2D**

* proyeccion2D x minimo salida.pgm: Verificar creación correcta



* proyeccion2D y maximo salida.pgm: Verificar diferentes criterios



* proyeccion2D z promedio salida.pgm: Verificar todas las direcciones



* Proyección sin volumen cargado: Verificar mensaje de error



1. **Descripción**: Intentar proyección sin datos
2. **Entrada**: proyeccion2D z min salida.pgm
3. **Resultado esperado**: Error indicando falta de volumen
4. **Resultado obtenido**: "Error: No hay volumen cargado para procesar"
5. **Estado**: Éxito

**4. Pruebas de Codificación Huffman**

**4.1 Codificación**

* codificar\_imagen imagen.huf: Verificar creación de archivo
  1. **Descripción**: Verificar que se genera un archivo .huf al codificar una imagen cargada.
  2. **Entrada**: codificar\_imagen imagen.huf
  3. **esperado**: Mensaje de confirmación: "Archivo codificado guardado como 'imagen.huf'."
  4. **Resultado obtenido**: El archivo se generó correctamente.
  5. **Estado**: Éxito
* Codificar sin imagen cargada: Verificar mensaje de error



1. **Descripción**: Intentar compresión sin datos
2. **Entrada**: codificar\_imagen vacio.huf
3. **Resultado esperado**: Error indicando falta de imagen
4. **Resultado obtenido**: "Error: No hay imagen cargada para codificar"
5. **Estado**: Éxito

**4.2 Decodificación**

* decodificar\_archivo imagen.huf salida.pgm: Verificar reconstrucción
  1. Descripción: Recuperar imagen desde archivo .huf
  2. Entrada: decodificar\_archivo salida.huf reconstruida.pgm
  3. Resultado esperado: Imagen idéntica a la original
  4. Resultado obtenido: Archivo PGM generado, checksum MD5 coincidente
  5. Estado: Éxito
* Decodificar archivo inexistente: Verificar mensaje de error

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

* Comparar imagen original con decodificada

**5. Pruebas de Integración**

**5.1 Flujo completo**

* Cargar imagen → Codificar → Decodificar → Verificar igualdad
  1. Descripción: Verificar ciclo completo sin pérdidas
  2. Entrada:

cargar\_imagen test.pgm

codificar\_imagen test.huf

decodificar\_archivo test.huf test\_dec.pgm

* 1. Resultado esperado: Imagen final idéntica a la inicial
  2. Resultado obtenido: Diff pixel a pixel: 0 diferencias
  3. Estado: Éxito
  4. Observaciones: Tiempo total del proceso: 320ms