**Laboratorio 2 – Procesamiento BMP y Convoluciones 3×3**

Materia: Sistemas Embebidos

Integrantes: Paolo Gómez y Daniel Benavides

# 1) Objetivo

Desarrollar un programa en C que cargue una imagen BMP (24/32 bits), convierta a escala de grises y aplique filtros por convolución 3×3 (blur, sharpen, bordes, emboss), para luego guardar el resultado.

# 2) Herramientas

* Compilador MinGW/GCC en Windows.
* Editor/IDE (Dev-C++ / CMD).
* Imágenes BMP 24-bit para pruebas.

# 3) Metodología

**• Formato BMP:** Revisamos headers (tamaño, offset) y padding por fila (múltiplos de 4). Confirmamos orden de color BGR.

**• Carga/Guardado:** Implementamos cargarBMP() y guardar(), respetando padding y tamaños reportados por los headers.

**• Escala de grises:** Implementamos escalarGris() usando luminancia aproximada sin floats.

**• Convolución:** Agregamos aplicarConvolucion() con kernels 3×3 predefinidos; usamos divisor y offset; buffer temporal para no contaminar vecinos.

**• Menú y pruebas:** Menú con opciones (gris/convolución/salir) y validación de 24/32 bpp. Probamos blur, sharpen, bordes y emboss.

# 4) Resultados (observaciones)

Encontramos que los archivos se modificaron según lo solicitado, pero hubo que resolver fallos de programación para mejorar el funcionamiento.

# 5) Dificultades y soluciones

* Compilación/Linker: compilamos en carpeta simple.
* Formato 24/32 bpp: convertimos imágenes a 24-bit para evitar paletas.
* Desbordes numéricos: usamos clampToByte() para limitar a 0–255.
* Se agrego#patch push y pull para evitar padding del compilador

# 7) Conclusiones

Implementamos las funcionalidades de escalar a gris y también de convolucionar con kernels, pudimos solucionar errores y tras la presentación con el profe vimos que otros fallos puede llegar a tener el programa.