

# Framebuffers y Bitmaps

---

Gráficas por Computadora (CC2018) - 2026

## Framebuffers y Bitmaps

---

Semestre 02, 2026

### Utilidad

---

- Es la memoria intermedia entre el proceso de renderizado y la pantalla.
- Permite construir la imagen completa en memoria antes de mostrarla.
- Es fundamental en el software rendering (renderizado por CPU, sin GPU).

### Definición

---

Un framebuffer es una región de memoria reservada donde se almacena la imagen que está en proceso de construcción antes de enviarse a la pantalla.

Guarda el valor de color de cada píxel de la imagen.

### Importancia

---

- Permite representar gráficamente escenas complejas sin dibujar directamente sobre la pantalla.
- Es la base para técnicas modernas como el doble buffer y el triple buffer.

- Optimiza el rendimiento al desacoplar el renderizado de la actualización de la pantalla.

## Funcionamiento

---

El framebuffer actúa como un lienzo digital:

- Cada píxel se representa con uno o varios bytes según la profundidad de color (color depth).
- La CPU o la GPU escribe directamente los valores de color de cada píxel.
- Durante la fase de refresco de la pantalla, el contenido del framebuffer se envía al monitor.

## Flujo de Datos

- Los datos generados por los shaders y el pipeline gráfico se almacenan en el framebuffer.
- En las pantallas analógicas clásicas (VGA), un RAMDAC lee el framebuffer: multiplexores seleccionan la señal de cada canal.
- Conversores digital-analógico (DAC) transforman los valores en señales para la pantalla.
- La imagen renderizada se muestra en la pantalla.

*Nota: las pantallas digitales modernas (HDMI, DisplayPort) reciben la señal digital directamente, sin DAC.*

### Framebuffer

- El framebuffer es una cuadrícula de memoria que guarda un valor de color por cada píxel.

- En el modo indexado (palettized), esos valores son índices que apuntan a una tabla de colores.
- La tabla (Color Map o paleta) contiene los valores RGB completos de cada índice.

### Color map

- El Color Map alimenta cada canal de color (R, G, B) con valores digitales.
- En pantallas analógicas, cada canal pasa por su propio conversor D/A (Digital a Analógico) para generar la señal.
- Cada canal tiene su propio D/A.

## Aplicaciones

---

- **Doble buffering:** se alterna entre dos framebuffers (uno se muestra mientras el otro se dibuja) para evitar el parpadeo. Combinado con la sincronización vertical (VSync) evita el desgarro de la imagen (tearing).
- **Post-procesado:** efectos como bloom, blur o HDR se aplican sobre el framebuffer ya renderizado.
- **Captura de pantalla:** se lee el contenido del framebuffer para guardar la imagen.

## Bitmap

---

El formato más simple para representar imágenes digitales: una cuadrícula de píxeles almacenada en memoria.

### Definición

- Una imagen rasterizada formada por una matriz de píxeles.

- Cada píxel tiene un valor que representa su color.
- Se almacena como datos binarios en la memoria o en archivos.

Bitmap

## Profundidad de color

La cantidad de bits por píxel determina el número de colores posibles:

- 1 bit: 2 colores (blanco y negro).
- 4/8 bits: paletas de colores (16 o 256 colores).
- 16 bits: High Color (~65 mil colores).
- 24 bits: True Color (~16.7 millones de colores).

*Mayor profundidad de color → mayor tamaño de archivo.*

## Formato BMP

Bitmap (BMP) es un formato estándar para guardar imágenes rasterizadas:

- Introducido por Microsoft junto con Windows 2.0 en 1987, compatible con OS/2 (desarrollado con IBM).
- Amplio soporte en sistemas operativos y programas.

## Estructura del archivo

- **File Header:** identifica el archivo como BMP e indica su tamaño.
- **Info Header:** contiene las dimensiones, la profundidad de color y el tipo de compresión.
- **Color Table (opcional):** la paleta de colores, presente en imágenes indexadas.
- **Pixel Data:** la matriz de colores de los píxeles.

## Ventajas y desventajas

- □ **Simplicidad:** fácil de leer y escribir.
- □ **Sin pérdida de calidad:** normalmente no usa compresión.
- □ **Ideal para pruebas y enseñanza.**
- □ **Archivos grandes:** cada píxel ocupa espacio completo.
- □ **No soporta transparencias ni capas.**