

# Imágenes soportadas en LIBGDX

## 1. Formatos de Imagen Soportados por LibGDX

En general, LibGDX soporta cualquier formato estándar que pueda leer Java mediante [ImageIO](#) o sus propios decodificadores internos. Los más utilizados son:

### 1.1 PNG (Portable Network Graphics)

- **Compatibilidad:** Total.
- **Transparencia:** Sí (canal alfa 8 bits).
- **Compresión:** Sin pérdida.
- **Ventajas:**
  - Excelente calidad para sprites.
  - Soporta transparencias precisas.
  - Ideal para elementos UI y assets con detalles.
- **Desventajas:**
  - Tamaño de archivo relativamente grande.

### 1.2 JPG / JPEG (Joint Photographic Experts Group)

- **Compatibilidad:** Total.
- **Transparencia:** No.
- **Compresión:** Con pérdida.
- **Ventajas:**
  - Archivos muy ligeros.
  - Buen rendimiento en texturas grandes (ej.: fondos).
- **Desventajas:**
  - No apto para sprites.
  - Artefactos visibles debido a la compresión.

### 1.3 BMP (Bitmap)

- **Compatibilidad:** Sí.
- **Transparencia:** No.
- **Compresión:** Ninguna.
- **Ventajas:**
  - Rápido de cargar.
- **Desventajas:**
  - Archivos enormes.
  - No recomendado en la práctica.

## 1.4 GIF (Graphics Interchange Format)

- **Compatibilidad:** Parcial (no recomendado para producción).
- **Transparencia:** 1 bit (sí/no).
- **Compresión:** Sin pérdida.
- **Ventajas:**
  - Soporta imágenes simples.
- **Desventajas:**
  - Soporte limitado.
  - Calidad baja (256 colores).
  - No apto para animaciones complejas.

## 1.5 TGA (Targa)

- **Compatibilidad:** Sí, especialmente para pipelines profesionales.
- **Transparencia:** Sí.
- **Compresión:** Opcional (RLE).
- **Ventajas:**
  - Usado en muchos pipelines 3D.
  - Compatible con canal alfa.
- **Desventajas:**
  - No ofrece ventajas relevantes sobre PNG para 2D.

## 1.6 KTX / KTX2 (Khronos Texture Format)

- **Compatibilidad:** A través de extensiones.
  - **Transparencia:** Sí.
  - **Compresión:** Altamente optimizada (GPU-ready).
  - **Ventajas:**
    - Se cargan muy rápido en GPU.
    - Permite compresión específica por plataforma (ETC2, ASTC, S3TC...).
    - Ideal para proyectos profesionales o móviles de alto rendimiento.
  - **Desventajas:**
    - Requiere herramientas externas para convertir.
    - No recomendado para principiantes.
- 

# 2. Recomendaciones de Uso

La elección del formato depende del tipo de recurso gráfico.

## 2.1 Sprites, personajes y elementos con transparencia

**Formato recomendado: PNG**

- Excelente calidad y transparencia real.

- Perfecto para animaciones (spritesheets).

## 2.2 Fondos, ilustraciones grandes sin transparencia

**Formato recomendado: JPG**

- Archivos ligeros.
- Buen desempeño en dispositivos móviles.

## 2.3 Elementos 3D o texturas avanzadas

**Formato recomendado: KTX / KTX2**

- Carga optimizada para GPU.
- Muy útil para reducir el uso de memoria.

## 2.4 Prototipos o debugging

**Formato recomendado: BMP** (solo para desarrollo por simplicidad y velocidad de exportación). No usar en producción.

## 2.5 Imágenes simples o con paleta reducida

**Formato recomendado: PNG** (GIF queda descartado por sus limitaciones severas).

---

# 3. Consideraciones de Rendimiento en LibGDX

- Mantén texturas en potencias de 2 si necesitas compatibilidad amplia en Android.
  - Minimiza texturas enormes (> 4096px) para garantizar soporte móvil.
- 

# 4. Conclusión

Los formatos más recomendables para LibGDX son:

- **PNG** → principal para sprites, UI y recursos con transparencia.
- **JPG** → fondos grandes donde la pérdida de calidad no afecta.
- **KTX/KTX2** → rendimiento alto en proyectos avanzados.

Elegir el formato correcto optimiza:

- Carga de recursos.
- Consumo de memoria.
- Rendimiento gráfico.
- Calidad visual final.

Este manual sirve como guía para seleccionar el formato adecuado según el caso dentro de proyectos LibGDX.