

Documentación de la Función Dispose()

¿Qué es dispose()?

En LibGDX, `dispose()` es un método utilizado para **liberar recursos** que consumen memoria en la GPU o en la memoria nativa del sistema.

LibGDX no usa un *garbage collector* para muchos de sus recursos gráficos (texturas, fuentes, música, etc.), por lo que **el programador debe liberarlos manualmente**.

¿Por qué es importante?

Si no llamas a `dispose()` cuando ya no necesitas ciertos recursos, tu juego o aplicación puede sufrir:

- **Fugas de memoria (memory leaks)**
 - **Caídas de rendimiento**
 - **Crash en dispositivos con poca memoria**, especialmente móviles
-

¿Dónde se usa normalmente?

1. En la clase principal (`ApplicationListener` o `Game`):

Cuando creas un juego en LibGDX, tu clase principal tiene un método especial llamado `dispose()`. LibGDX lo llama automáticamente **cuando la aplicación está cerrando**.

Dentro de ese método, tú le dices a LibGDX:

“Oye, ya estoy cerrando el juego, puedes borrar estos recursos de la memoria.”

```
@Override  
public void dispose() {  
    spriteBatch.dispose(); // Borrar spriteBatch de memoria  
    texture.dispose(); // Borrar la textura de memoria  
    font.dispose(); // Borrar la fuente de memoria  
}
```

2. En pantallas (Screen):

Cuando cambias de pantalla, tú decides cuándo llamar a `dispose()`.

```
@Override  
public void dispose() {  
    texture.dispose();  
    font.dispose();  
}
```

¿Cómo se invoca?

Sintaxis general

```
objeto.dispose();
```

Ejemplo para un texture

```
Texture texture = new Texture("fondo.png");  
// ... usar textura ...  
texture.dispose(); // Liberación
```

¿Qué recursos deben ser liberados?

Debes llamar `dispose()` a objetos como:

- `Texture`
- `SpriteBatch`
- `BitmapFont`
- `ShaderProgram`
- `Music`
- `Sound`
- `FrameBuffer`
- `Pixmap`
- `ShapeRenderer`
- Cualquier clase que implemente `Disposable`

Como regla general:

Si ves que un objeto tiene un método `.dispose()`, DEBES llamarlo cuando ya no lo necesites.

Ejemplo completo

```
public class MyGame extends ApplicationAdapter {  
    private Texture texture;  
    private SpriteBatch batch;  
  
    @Override  
    public void create() {  
        batch = new SpriteBatch();  
        texture = new Texture("image.png");  
    }  
  
    @Override  
    public void render() {  
        batch.begin();  
        batch.draw(texture, 0, 0);  
        batch.end();  
    }  
  
    @Override  
    public void dispose() {  
        batch.dispose();  
        texture.dispose();  
    }  
}
```

Resumen

- `dispose()` libera recursos manualmente.
- Es esencial para evitar fugas de memoria.
- Debes usarlo para texturas, sonidos, fuentes y cualquier objeto que implemente `Disposable`.
- LibGDX no lo hace automáticamente por ti.