

Ciclo de vida de una aplicación en LibGDX

Métodos principales del ciclo de vida (`ApplicationListener`)

1. `create()`

Momento de invocación:

Se ejecuta una única vez, al iniciar la aplicación.

Responsabilidades:

- Inicialización de recursos: texturas, sonidos, fuentes, cámaras, viewports, batches.
- Configuración del estado inicial del juego.
- Registro de listeners o sistemas internos.

Notas técnicas:

No debe contener lógica dependiente del tiempo.

Los recursos creados aquí deben liberarse en `dispose()`.

2. `resize(int width, int height)`

Momento de invocación:

- Cuando la aplicación cambia de tamaño (desktop).
- En rotaciones de pantalla o cambios de resolución (Android/iOS).
- Se invoca al menos una vez al iniciar la aplicación.

Responsabilidades:

- Actualizar viewports y cámaras.

- Ajustar escalados y proporciones de UI.
- Recalcular matrices de proyección.

3. `render()`

Momento de invocación:

Es llamado de forma continua por el game loop de LibGDX, normalmente 60 veces por segundo (dependiendo del framerate real).

Responsabilidades:

- Actualización lógica del juego.
- Procesamiento de entradas (`Gdx.input`).
- Redibujado completo de la escena.
- Sincronización temporal mediante `delta` (usando `Gdx.graphics.getDeltaTime()` o el parámetro si se usa `Screen`).

4. `pause()`

Momento de invocación:

Principalmente en plataformas móviles:

- Aplicación pasa a segundo plano.
- Interrupciones del sistema (llamadas, notificaciones).

Responsabilidades:

- Guardar estado necesario.
- Pausar música, animaciones o tareas en segundo plano.
- Suspende lógicas dependientes del tiempo.

5. `resume()`

Momento de invocación:

Cuando la aplicación vuelve a primer plano después de un `pause()`.

Responsabilidades:

- Restaurar estado pausado.
- Reanudar música o procesos detenidos.
- Reconstruir recursos perdidos si la plataforma los ha destruido (especialmente en Android con OpenGL ES).

6. `dispose()`

Momento de invocación:

Se ejecuta una única vez al cerrar la aplicación.

Responsabilidades:

- Liberar todos los recursos cargados manualmente:
`Texture`, `SpriteBatch`, `Sound`, `Music`, `BitmapFont`, `ShapeRenderer`, etc.
- Cerrar conexiones, streams, sockets o archivos abiertos.

LibGDX nunca libera recursos gráficos automáticamente; depende completamente del desarrollador.