

NGN_LOAD.H

NGN_TextureData* Texture(std::string filepath);

Carga una textura en formato PNG.

```
NGN_TextureData* bg_grid = ngn->load->Texture("data/png/grid.png");
```

NGN_TiledBgData* TiledBg(std::string filepath);

Carga un fondo de tiles en formato .tbg (Ver utilidades adjuntas).

```
NGN_TiledBgData* tiles_bg_front = ngn->load->TiledBg("data/bg/bg_front.tbg");
```

NGN_SpriteData* Sprite(std::string filepath);

Carga un Sprite en formato .spr (Ver utilidades adjuntas).

```
NGN_SpriteData* wizard_sprite = ngn->load->Sprite("data/spr/wizard.spr");
```

NGN_CollisionMapData* CollisionMap(std::string filepath);

Carga un mapa de colisiones en formato .map (Ver utilidades adjuntas)

```
NGN_CollisionMapData* collision_map = ngn->load->CollisionMap("data/collision/mainmap.map");
```

NGN_AudioClipData* AudioClip(std::string filepath);

Carga un archivo de audio en formato WAV, FLAC o OGG para usarlo como efecto de sonido. Devuelve FALSE en caso de error en la carga del archivo.

```
NGN_AudioClipData* coin_sfx = ngn->load->AudioClip("data/wav/coin.wav");
```

```
NGN_TextFont* TrueTypeFont(  
    std::string filepath,    // Archivo a cargar  
    uint32_t height,        // Altura de la fuente (en pixeles)  
    bool antialias = true,   // Antialias?  
    uint32_t font_color = 0xFFFFFFFF,    // Color base  
    uint32_t outline = 0,     // Borde? (en pixeles)  
    uint32_t outline_color = 0x000000    // Color del borde  
);
```

Carga y convierte a BITMAP un archivo de fuente tipográfica TRUE TYPE con la altura y propiedades especificadas. Devuelve NULL en caso de no poder cargar o convertir la fuente.

```
NGN_TextFont* font = ngn->load->TrueTypeFont("data/consolas.ttf", 24);
```