

Desarrollo N64 nativo y exploración de Memoria

Esta guía cubre desde la instalación en macOS hasta la creación de una herramienta para volcar el contenido de un Controller Pak.

[Enlace al controller pak.](#)

1. Instalación y Configuración del Entorno (macOS)

Si usas una Mac con chip Apple Silicon (M1/M2/M3), Docker necesita ser forzado a la plataforma amd64.

Bash

```
# 1. Crear y entrar a la carpeta del proyecto
mkdir -p ~/write && cd ~/write
```

```
# 2. Configurar plataforma para Mac modernos
export DOCKER_DEFAULT_PLATFORM=linux/amd64
```

```
# 3. Inicializar el contenedor de Libdragon
libdragon init
```

2. El Fichero de Compilación (Makefile)

Crea un archivo llamado Makefile en la raíz de la carpeta write. Este archivo coordina al compilador MIPS dentro de Docker.

Makefile

```
PROG_NAME = explorador_pak
SRCDIR = src
BUILDDIR = build
OBJS = $(BUILDDIR)/main.o

# Configuración interna de Libdragon
N64_INST = /n64_toolchain
include $(N64_INST)/include/n64.mk

# Flags de estabilidad para hardware real
CFLAGS += -Iinclude -falign-functions=32

all: $(PROG_NAME).z64

$(PROG_NAME).elf: $(OBJS)

$(PROG_NAME).z64: N64_ROM_TITLE = "Pak Explorer N64"

$(BUILDDIR)/%.o: $(SRCDIR)/%.c
    @mkdir -p $(dir $@)
    @echo "      [CC] $<"
    $(CC) $(CFLAGS) -c -o $@ $<

clean:
    rm -rf $(BUILDDIR) *.z64 *.elf
.PHONY: clean
```

3. El Código Fuente (src/main.c)

Crea la carpeta src y dentro el archivo main.c. Este programa recorre los **1024 bloques** (32KB) del Controller Pak y los muestra en pantalla ignorando errores de detección.

C

```
#include <stdio.h>
#include <string.h>
#include <libdragon.h>
```

```

#define TOTAL_BLOCKS 1024
#define BLOCKS_PER_PAGE 10

int main(void) {
    timer_init();
    display_init(RESOLUTION_320x240, DEPTH_16_BPP, 2,
GAMMA_NONE, ANTIALIAS_RESAMPLE);
    joypad_init();

    int current_page = 0;
    uint8_t buffer[32] __attribute__((aligned(16)));
    char info[64];

    while (1) {
        joypad_poll();

        joypad_buttons_t btn =
joypad_get_buttons_pressed(JOYPAD_PORT_1);

        if (btn.d_down && current_page < (TOTAL_BLOCKS /
BLOCKS_PER_PAGE) - 1) current_page++;
        if (btn.d_up && current_page > 0) current_page--;

        surface_t *disp = display_get();
        graphics_fill_screen(disp, 0);

        graphics_draw_text(disp, 20, 10, "== EXPLORADOR DE
CONTROLLER PAK ==");
        sprintf(info, "Bloques: %d a %d (de 1024)",
            current_page * BLOCKS_PER_PAGE,
            (current_page * BLOCKS_PER_PAGE) +
BLOCKS_PER_PAGE - 1);
        graphics_draw_text(disp, 20, 25, info);

        for (int i = 0; i < BLOCKS_PER_PAGE; i++) {
            int block_to_read = (current_page *
BLOCKS_PER_PAGE) + i;
            int y_pos = 50 + (i * 15);

            // Lectura física directa del bloque de 32 bytes
            int res = joybus_accessory_read(JOYPAD_PORT_1,
block_to_read * 32, buffer);

```

```

        if (res == 0) {
            char hex_preview[64];
            sprintf(hex_preview, "%04X: %02X %02X %02X
%02X | %c%c%c%c",
                    block_to_read * 32, buffer[0],
buffer[1], buffer[2], buffer[3],
                    (buffer[0]>31?buffer[0]:'. '),
                    (buffer[1]>31?buffer[1]:'. '),
                    (buffer[2]>31?buffer[2]:'. '),
                    (buffer[3]>31?buffer[3]:'. '));
            graphics_draw_text(disg, 20, y_pos,
hex_preview);
        } else {
            graphics_draw_text(disg, 20, y_pos, "ERROR: No
detectado o sucio");
        }
    }

    graphics_draw_text(disg, 20, 210, "D-PAD ARRIBA/ABAJO:
Navegar");
    display_show(disg);
}
}

```

4. Comandos para Compilar y Ejecutar

Ejecuta esto en tu terminal cada vez que quieras probar cambios:

Bash

```
# Limpiar binarios antiguos
libdragon make clean
```

```
# Compilar la nueva ROM
libdragon make
```

Esto generará el archivo **explorador_pak.z64**.

5. Resumen de Lecciones para el Post

1. **Limpieza de Docker:** Usamos `docker system prune` y `docker rm` para evitar conflictos de imágenes antiguas.
2. **Alineación de Memoria:** El uso de `__attribute__((aligned(16)))` es obligatorio. Sin esto, el hardware real de la N64 suele dar errores de lectura aleatorios.
3. **Detección de Hardware:** Si el Pak se detecta como «Rumble», es un problema de resistencia eléctrica (suciedad). Limpiar con alcohol o goma de borrar es el 50% del éxito.
4. **Estructura del Pak:** La memoria son 32KB divididos en bloques de 32 bytes. La dirección `0x0000` es la cabecera; a partir de `0x0100` es zona segura para pruebas.