



DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE CONCEPCIÓN
CONCEPCIÓN, CHILE.



Código - Nombre del Curso

Título del Trabajo

Profesor: Vincenzo Caro Fuentes

Integrantes: Alumno1

Alumno2

1 de octubre de 2025

Concepción, 1 de octubre de 2025

Resumen

 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

 Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Índice

1. Introducción	1
2. Actividad n°1: Ahorcado	2
2.1. Ítem 1	2
2.2. Ítem 2	3
2.3. Solución	4
2.3.1. Ítem 1	4
2.3.2. Ítem 2	6
3. Actividad n°2: La boveda de Mr.Phantom	7
4. Actividad n°3: Transformando Raspberry Pi en una consola de juegos	8
4.1. Materiales a utilizar	8
4.2. Montaje de la carcasa NESPI	11
4.3. Instalación del software	14
4.4. Configuración retropie en la raspberry	17
4.5. Instalación de juegos	18
4.5.1. Configuración WiFi	18
4.5.2. Instalación de juegos en la consola	20
4.6. Configuración de la Interfaz	22
5. Conclusión	24
AnexoA: Información Adicional	26
A.5: Tipos de LEDs	26
A.5.0: RGB LEDs	26
A.5.0: LEDs con circuitos integrados	26
A.5.0: RGB LEDs	26

1. Introducción

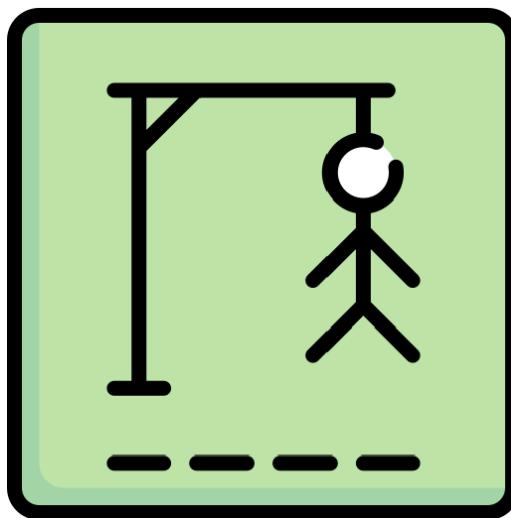
 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

 Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

 Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2. Actividad n°1: Ahorcado

El tradicional juego del ahorcado consiste en descubrir una palabra secreta adivinando sus letras una por una, dentro de un número limitado de intentos. Esta primera actividad consiste en implementar una versión digital del juego en Raspberry Pi, en la que se comenzará con un modo básico por consola, para luego evolucionar a una versión interactiva que incorpore LEDs, botones y un buzzer. La dinámica general del juego es la siguiente: al inicio, el jugador seleccionará un nivel de dificultad que determinará la extensión de las palabras a descubrir y la cantidad de intentos permitidos. Durante cada partida, el jugador ingresará o seleccionará letras para intentar completar la palabra. Cada acierto acercará al jugador a la victoria, mientras que cada error reducirá su número de intentos disponibles. El juego terminará cuando el jugador complete correctamente la palabra o bien pierda todos sus intentos.



2.1. Ítem 1

En este Ítem deberán preparar la base de su juego del ahorcado utilizando únicamente un teclado y la consola en Raspberry Pi. Para programar la lógica de su juego, desarrollem un programa en Python que opere de acuerdo con los siguientes pasos:

1. Configuración inicial:

- El juego deberá comenzar solicitando al usuario ingresar el número de letras de las palabras a utilizar, y el número de intentos permitidos antes de perder.
- A partir de estos parámetros, el programa seleccionará aleatoriamente una palabra de una lista predefinida, ocultando sus letras con guiones bajos en pantalla.

2. Desarrollo de la partida:

- El jugador ingresará letras por teclado utilizando la función `input()`.
- Si la letra pertenece a la palabra, se revelará en todas las posiciones correspondientes.
- Si la letra no pertenece, se restará un intento y se mostrará un mensaje indicando cuántos intentos quedan.
- En cada turno, la consola deberá mostrar el estado actual de la palabra, las letras descubiertas y las que faltan.

3. Condiciones de fin de la partida:

- El jugador ganará cuando logre descubrir toda la palabra antes de agotar los intentos.
- El jugador perderá si se queda sin intentos disponibles.
- En ambos casos, deberá mostrarse un mensaje de victoria o derrota.
- Tras terminar, el programa deberá consultar al usuario si desea iniciar una nueva partida (reiniciando la configuración) o finalizar.

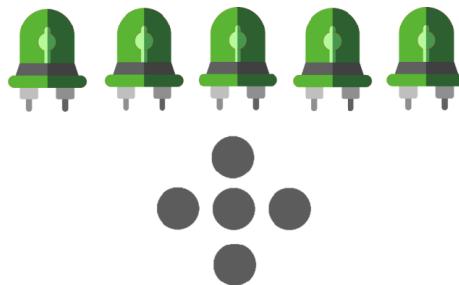
4. Consideraciones adicionales:

- Utilizar lo mostrado en el Anexo “Reescritura en consola” para imprimir un abecedario o la palabra oculta del juego de forma más interactiva por consola.
- Obtener la palabra a utilizar durante una partida del juego de manera dinámica desde internet.

2.2. Ítem 2

Utilizando como base el código del Ítem anterior, modifiquen su juego para incorporar elementos del kit de 45 sensores, utilizando al menos los siguientes módulos:

- 5 LEDs (KY-009, KY-011, KY-016 o KY-029)
- 5 botones (KY-004, KY-023, KY-040 o los entregados por el profesor).
- 1 buzzer activo (KY-012).



Con estos elementos, preparen un circuito en su protoboard que siga la estructura de la imagen anterior y desarrollen un programa en Python utilizando la biblioteca gpiozero (u otra equivalente) para controlar los LEDs, los botones y el buzzer, teniendo en cuenta las nuevas funcionalidades que se describen a continuación:

1. Indicador visual de vidas con LEDs:

- Los LEDs representarán el número de intentos disponibles (máximo 5).
- Se deberán definir criterios de color: por ejemplo, verde cuando se tengan todas las vidas, amarillo a la mitad, y rojo cuando quede solo 1 vida.
- A medida que se pierdan vidas, los LEDs deberán apagarse de acuerdo con el número restante (ejemplo: apagar un LED por cada intento perdido).

2. Control de juego con botones:

- Los 5 botones reemplazarán parcialmente la entrada por teclado, permitiendo navegar por las letras del alfabeto:

- Botón 1: avanzar una letra.
- Botón 2: retroceder una letra.
- Botón 3: avanzar 5 letras.
- Botón 4: retroceder 5 letras.
- Botón 5: seleccionar la letra actual.
- Al seleccionar, el programa validará si la letra pertenece o no a la palabra y actuará en consecuencia.

3. Señales sonoras con buzzer:

- Emitir un tono personalizado cada vez que se pierda una vida.
- Emitir un tono personalizado de victoria al completar la palabra.
- Emitir un tono personalizado de derrota cuando se acaben los intentos.

4. Consideraciones adicionales

- Implementar un contador de racha de victorias, que registre cuántas palabras se han descubierto consecutivamente sin perder. La racha deberá guardarse en un archivo y mostrarse tanto al inicio de cada juego como al finalizar una partida.

2.3. Solución

Primero, se creó la base del juego del ahorcado para que funcionara en la consola. Una vez hecho esto, se comenzaron a agregar los elementos propuestos en el ítem 2 y se procedió a adaptar el código para que funcionara con los nuevos elementos. El código funciona de la siguiente manera:

2.3.1. Ítem 1

1. Funciones utilizadas:

- (a) Función para sacar las palabras de un repositorio online:

```
def get_spanish_pokemon_name(pokemon_id):
    url = f"https://pokeapi.co/api/v2/pokemon-species/{pokemon_id}/"
    with urllib.request.urlopen(url, timeout=5) as response:
        raw_data = response.read()
        data = json.loads(raw_data)

    for item in data["names"]:
        if item["language"]["name"] == "es":
            return item["name"].lower()

    return data["name"].lower()

def get_random_spanish_pokemon(max_id=1010):
    pid = randint(1, max_id)
    return get_spanish_pokemon_name(pid)
```

(a) Nombres de pokemon

```
# Github palabras
SPANISH_WORDS_URL = "https://raw.githubusercontent.com/words/an-array-of-spanish-words/master/index.json"

spanish_words = []

def load_spanish_words():
    global spanish_words
    print("Fetching Spanish word list from the internet...")
    with urllib.request.urlopen(SPANISH_WORDS_URL, timeout=5) as response:
        data = response.read()
        words = json.loads(data)
        spanish_words = [w.strip().lower() for w in words if isinstance(w, str) and w.strip()]

    print(f"Loaded {len(spanish_words)} words from online source!")

def get_random_spanish_word_by_length(length):
    global spanish_words

    if not spanish_words:
        load_spanish_words()

    filtered = [w for w in spanish_words if len(w) == length]

    if not filtered:
        raise ValueError(f"No Spanish words found with length {length}")

    return choice(filtered)
```

(b) Palabras en español

(b) Función para elegir la dificultad:

```

def dificultad(palabra_objetivo):
    intentos = validador(1,5,"Número de intentos (máximo 5): ")
    print("\n")
    progreso = ["_"]*len(palabra_objetivo)
    contador = 0
    while intentos > 0:
        if intentos == 5:
            red.off()
            green.on()
            blue.on()
        elif intentos == 3 or intentos == 4:
            red.off()
            green.on()
            blue.off()
        else:
            red.on()
            green.off()
            blue.off()
        abecedario(progreso,intentos)
        letra_candidata = alfabeto[pos]
        if letra_candidata in palabra_objetivo and " ".join(progreso) != palabra_objetivo:
            print("Fantástico! Sigue así!")
            melodía_buzzer("letra_correcta")
            for i in range(len(palabra_objetivo)):
                if palabra_objetivo[i] == letra_candidata:
                    progreso[i] = letra_candidata
    print("".join(progreso))
    if "" .join(progreso) == palabra_objetivo:
        print("Has ganado! La palabra era: " + palabra_objetivo)
        melodía_buzzer("victoria")
        break
    else:
        print("No te rindas! Por favor, continua")
        melodía_buzzer("letra_incorrecta")
        if "" .join(progreso) == palabra_objetivo:
            contador += 1
            with open("racha.txt", "w", encoding="utf-8") as f:
                f.write(str(contador))
            print("Has ganado! La palabra era: " + palabra_objetivo)
            melodía_buzzer("victoria")
            break
        if intentos == 0:
            contador = 0
            with open("racha.txt", "w", encoding="utf-8") as f:
                f.write(str(contador))
            print("Has perdido! La palabra era: " + palabra_objetivo)
            print("\n")
            melodía_buzzer("derrota")

```

(c)

(d)

Fig. 1: Código para elegir la dificultad del juego

(c) Función para contar la racha de victorias:

```

while True:
    if btn == 0:
        sleep(1)
    else:
        if btn == "1":
            pos = (pos + 1) % 26
        elif btn == "2":
            pos = (pos - 1) % 26
        elif btn == "3":
            pos = (pos + 5) % 26
        elif btn == "4":
            pos = (pos - 5) % 26
        elif btn == "5":
            historial_letras.append(alfabeto[pos])
            btn = 0
            break
        print("Racha histórica de victorias: " + str(racha))
        print("Palabra actual: " + " ".join(progreso))
        print("Intentos restantes: " + str(intentos))
        print("Letra actual: " + alfabeto[pos].upper())
        if len(historial_letras) > 0:
            print("Letras usadas hasta ahora: " + " ".join(historial_letras).upper())
        print("Presione botón: ")
        print("1: Siguiente | 2:Anterior | 3: +5 | 4: -5 | 5: Seleccionar")
        print("\n")

```

(a)

(b)

Fig. 2: Código para almacenar la racha de victorias

(d) Función para validar las entradas:

```

def validador(linf,lsup,mensaje):
    x = int(input(mensaje))
    while x < linf or x > lsup:
        print("Error. Ingrese un valor dentro del rango")
        x = int(input(mensaje))
    return x

```

Fig. 3: Función "validador"

2. Parte principal del código:

```
#Interfaz del usuario
aux = False
while aux == False:
    modo = str(input("SELECCIONE MODO DE JUEGO (POKEMON o NORMAL): ")).lower()
    if modo == "pokemon":
        palabra_objetivo = get_random_spanish_pokemon()
        dificultad(palabra_objetivo)
        jugar_de_nuevo = str(input("Desea empezar una nueva partida: ")).lower()
        if jugar_de_nuevo == "si":
            historial_letras = []
            pos = 0
            aux = False
        else:
            aux = True

    elif modo == "normal":
        try:
            longitud_palabra = int(input("Cantidad de letras deseada en la palabra: "))
            palabra_objetivo = get_random_spanish_word_by_length(longitud_palabra)
            dificultad(palabra_objetivo)
            jugar_de_nuevo = str(input("Desea empezar una nueva partida: ")).lower()
            if jugar_de_nuevo == "si":
                historial_letras = []
                pos = 0
                aux = False
            else:
                aux = True
        except ValueError as ve:
            print(ve)
        except Exception as e:
            print("Error fetching word:", e)

    else:
        print("MODO DE JUEGO INVALIDO. POR FAVOR, SELECCIONA 'POKEMON' o 'NORMAL'")
```

Fig. 4: Parte principal del código

2.3.2. Ítem 2

3. Actividad n°2: La bóveda de Mr.Phantom

4. Actividad n°3: Transformando Raspberry Pi en una consola de juegos

4.1. Materiales a utilizar

1. RaspBerry PI 4B 2G

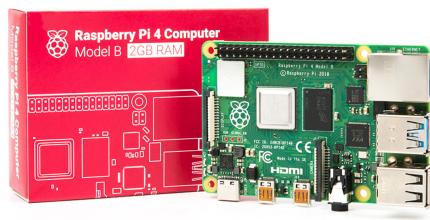


Fig. 5: RaspBerry PI 4B 2G. Fuente: [1]

2. Carcasa NESPI



Fig. 6: Carcasa NESPI. Fuente: [2]

3. Fuente Alimentación con cortador de corriente



Fig. 7: Fuente de alimentación. Fuente: [3]

4. Mandos a elección (nes,nintendo,etc)



Fig. 8: Mandos snes. Fuente: [3]

5. SD de 64GB



Fig. 9: Tarejeta micro SD. Fuente: [3]

6. Lector de tarjetas SD



Fig. 10: Lector SD. Fuente: [3]

7. Cable HDMI/Micro HDMI

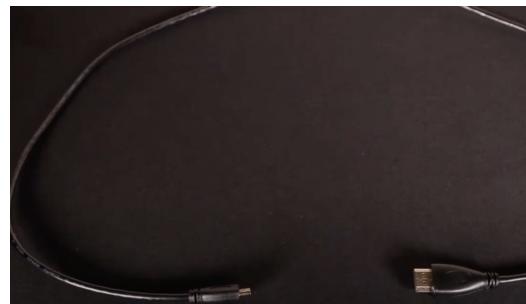


Fig. 11: Cable HDMI. Fuente: [3]

8. Ventilador + disipador

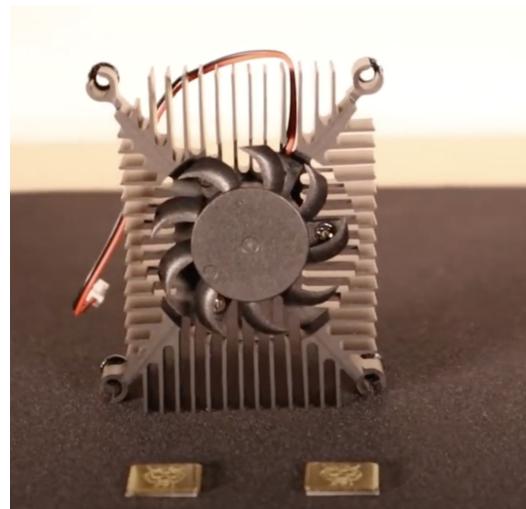


Fig. 12: Disipador y pasta térmica. Fuente: [3]

Para esta guía se utilizará una carcasa NESPI como la que se muestra en 6, la cual no es indispensable. Sin embargo, se recomienda ya que mejora la circulación del aire y proteje la raspberry utilizada.

4.2. Montaje de la carcasa NESPI

Este procedimiento se realizará teniendo la carcasa NESPI 4 CASE como referencia. Los pasos a seguir son los siguientes:

1. Al abrir la carcasa NESPI, se aprecia un espacio destinado a la Raspberry Pi.



Fig. 13: Interior de la carcasa NESPI. Fuente: [4]

2. Se deben colocar las dos pastas térmicas incluidas con el disipador sobre el procesador de la Raspberry Pi.



Fig. 14: Interior de la carcasa NESPI. Fuente: [4]

3. La Raspberry Pi debe introducirse en el espacio de la carcasa NESPI mencionado anteriormente, conectando posteriormente los dos cables USB y el cable Ethernet a la placa.

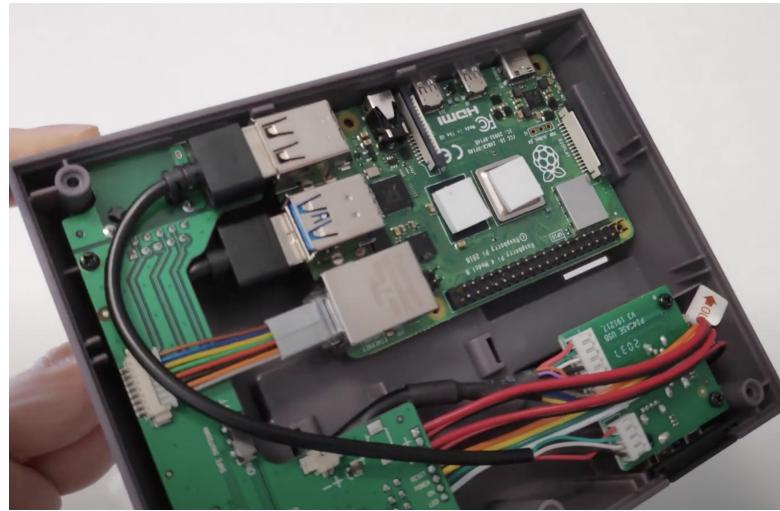
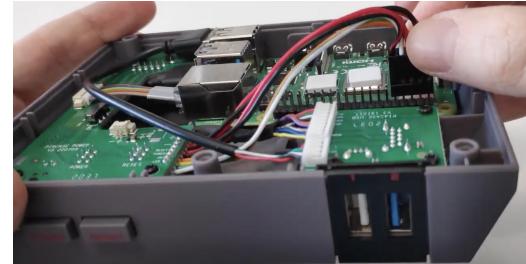


Fig. 15: Ensamblaje de la raspberry. Fuente: [4]

4. En el interior de la carcasa se encuentra un cable rojo, el cual debe conectarse a la Raspberry Pi según lo indicado en la imagen 16.



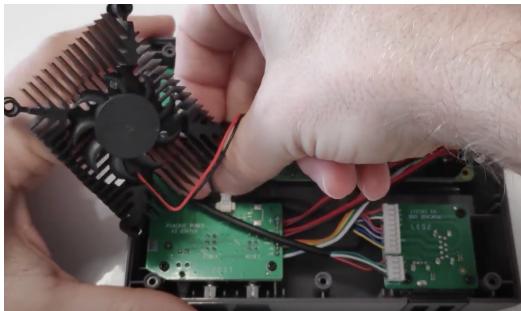
(a) Cable



(b) Conexión a la raspberry

Fig. 16: Conexión del cable. Fuente: [4]

5. Se debe conectar el cable del disipador a la carcasa, tal como se indica en la imagen 17, y posteriormente fijar el disipador a la Raspberry Pi usando los tornillos correspondientes. Es importante retirar el adhesivo de la pasta térmica utilizada anteriormente, en caso de que este estuviera presente.



(a) Conexión del disipador



(b) Ensamblaje del disipador

Fig. 17: Conexión del disipador. Fuente: [4]

6. Finalmente, se debe conectar el cable USB ubicado en la parte superior de la carcasa a la Raspberry Pi. A continuación, se procede a ensamblar completamente la carcasa NESPI.



(a) Conexión del cable USB



(b) Ensamblaje de la carcasa NESPI

Fig. 18: Ensamblaje de la carcasa NESPI. Fuente: [4]

4.3. Instalación del software

Primero, se debe ingresar a la página oficial de Raspberry Pi a través del siguiente enlace: [Página oficial de descargas de Raspberry Pi](#). Una vez dentro, se deberán seguir los pasos que se indican a continuación:

1. Descargar Raspberry PI Imager.

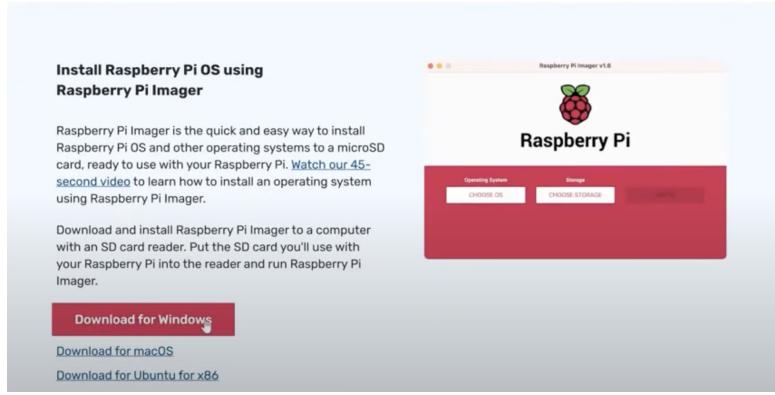


Fig. 19: RaspBerry PI Imager. Fuente: [3]

2. Ejecutar el archivo instalado e instalar el software



Fig. 20: Instalación del software. Fuente: [3]

3. Elegir el sistema operativo llamado RetroPie

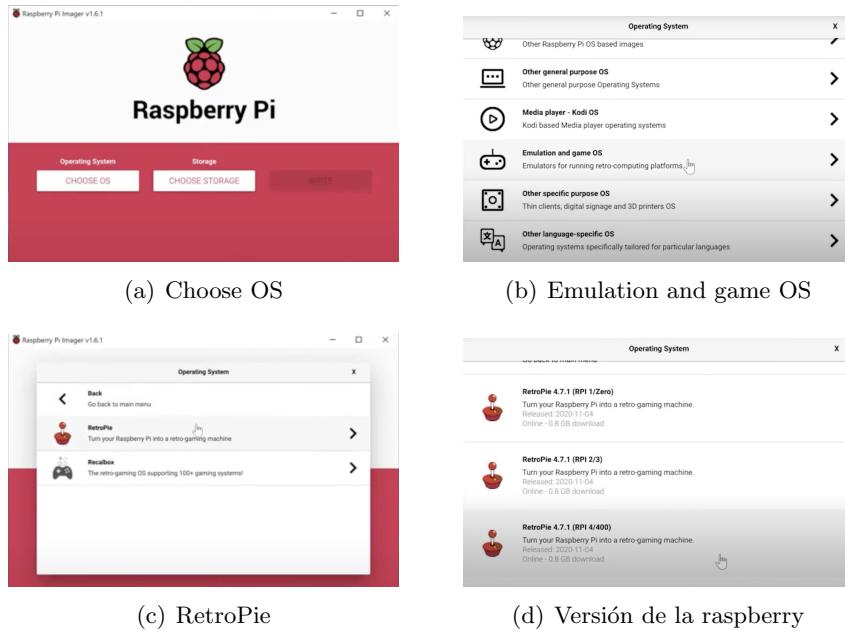


Fig. 21: Elección del sistema operativo. Fuente: [3]

en la imagen 21(d) se debe elegir la versión dependiendo de cual Raspberry se este usando (en este caso se usa RPI4).

4. Se debe seleccionar el dispositivo de almacenamiento: para ello, la tarjeta SD debe insertarse en el lector SD y conectarse al computador. A continuación, se selecciona la opción “Choose Storage” y se elige la tarjeta SD previamente conectada.

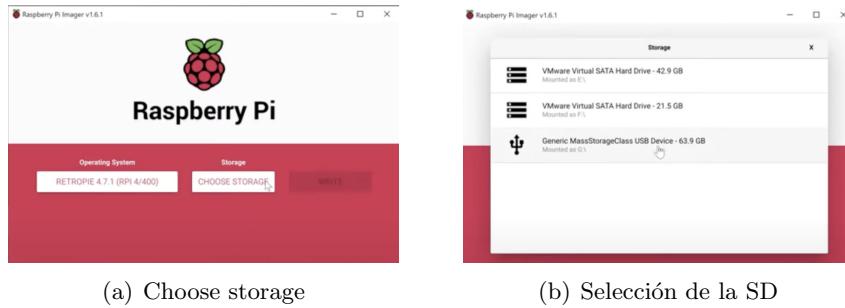


Fig. 22: Elección del almacenamiento. Fuente: [3]

5. Finalmente, se selecciona la opción "Write".



Fig. 23: Instalación del software. Fuente: [3]

Una vez finalizado este proceso, se debe retirar la tarjeta micro SD del lector y colocarla en la carcasa NESPI o directamente en la Raspberry Pi.

4.4. Configuración retropie en la raspberry

Para llevar a cabo esta etapa, es necesario conectar un control a la carcasa NESPI o a la Raspberry Pi. Los pasos a seguir son los siguientes:

1. Conectar la carcasa NESPI a una pantalla usando el cable HDMI.
2. Con la tarjeta SD insertada en la carcasa NESPI, se debe conectar la carcasa a la fuente de alimentación.
3. Al encender la consola, la raspberry instalará y creará automáticamente el sistema y particiones necesarias para el correcto funcionamiento de retropie. Cuando finalice este proceso, se mostrará una pantalla para configurar el control.



Fig. 24: Configuración del control. Fuente: [3]

4. Dejar pulsado cualquier botón del control y empezar a configurarlo



Fig. 25: Configuración del control. Fuente: [3]

Es importante asignar un botón a la opción "HOTKEY ENABLE" como se muestra en la imagen 25(b), ya que esta es necesaria para acceder a funciones importantes de la consola (por ejemplo guardar, cargar y salir de partidas).

4.5. Instalación de juegos

Existen diversas formas de instalar juegos en la Raspberry Pi, en este ítem se explicará el procedimiento para realizar la instalación a través de una conexión WiFi. Los pasos a seguir son los siguientes:

4.5.1. Configuración WIFI

1. Al encender la consola, se visualizará la opción para acceder a la configuración de RetroPie. Para ello, se debe presionar el botón asignado a la función Select.



Fig. 26: Configuración RetroPie. Fuente: [3]

2. Seleccionar la opción WiFi al final del menú.

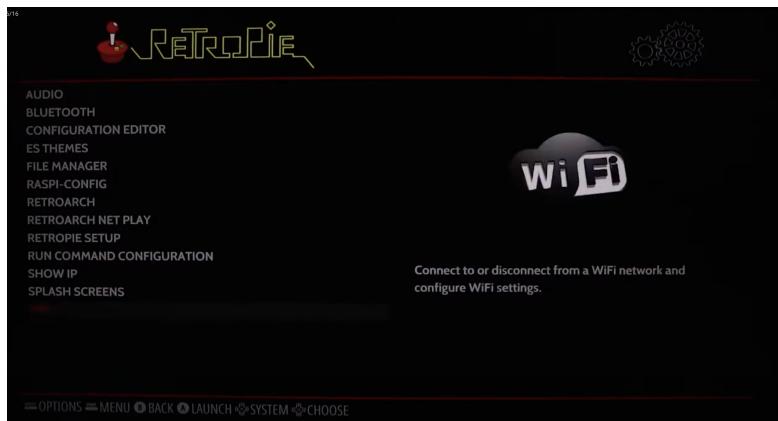


Fig. 27: Opción WiFi. Fuente: [3]

3. Seguir la siguiente secuencia:

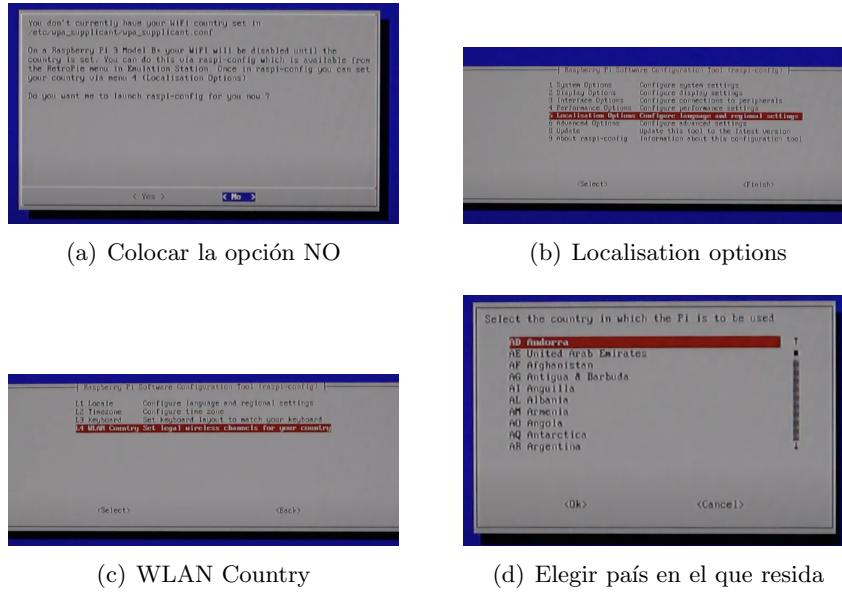


Fig. 28: Selección del País. Fuente: [3]

Despues de haber hecho esto, dar a la opción Finish y reiniciar la consola.



Fig. 29: Configuración del control. Fuente: [3]

4. Finalmente, en la esquina superior izquierda de la pantalla se mostrará la dirección IP identificada como Wireless IP. Esta dirección deberá ser registrada para su posterior uso.

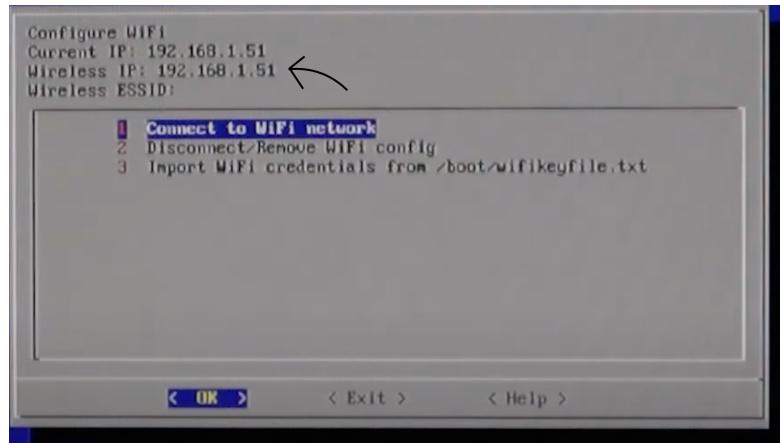


Fig. 30: Dirección IP. Fuente: [3]

4.5.2. Instalación de juegos en la consola

Para llevar a cabo este proceso será necesario disponer de un computador con sistema operativo Windows. A continuación, se detallan los pasos a seguir:

1. Encender la consola. Luego, desde el escritorio de Windows, se debe presionar la combinación de teclas Ctrl + R e introducir la dirección IP anotada previamente en el siguiente formato: \\Dirección IP. Finalmente, se debe seleccionar la opción Aceptar.

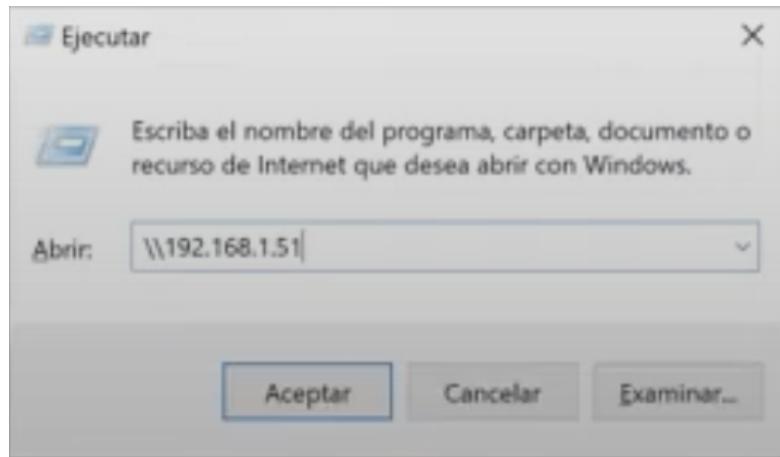


Fig. 31: Ejecutar la IP. Fuente: [5]

2. Seleccionar la carpeta roms.

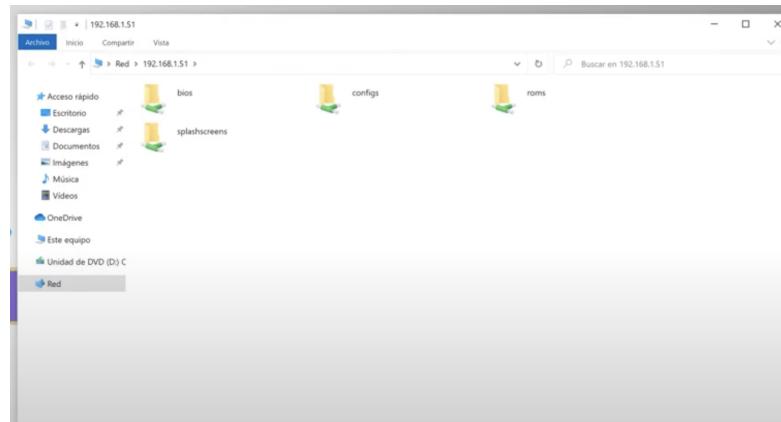


Fig. 32: Carpeta roms. Fuente: [5]

3. Se debe seleccionar la carpeta correspondiente al emulador de la consola que se desea utilizar y copiar en ella los juegos. Es importante verificar que los juegos sean compatibles con el emulador correspondiente. Una vez hecho esto, reiniciar la consola.

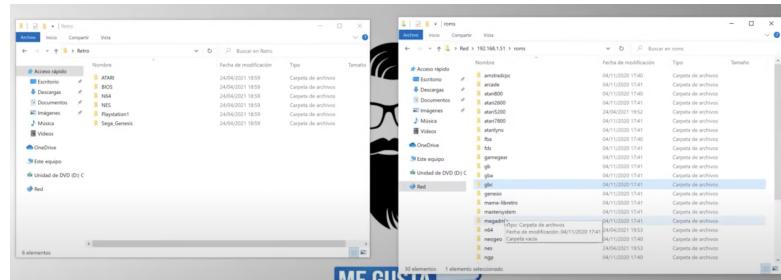


Fig. 33: Carpeta roms. Fuente: [5]

4. Al encender la consola, se podrán visualizar los diferentes emuladores en los que se han colocado juegos. Finalmente, se debe seleccionar el emulador y el juego deseado.



(a) Selección del emulador



(b) Selección del juego

Fig. 34: Interfaz de los juegos. Fuente: [5]

4.6. Configuración de la Interfaz

Es posible modificar la interfaz de la consola para hacerla más atractiva. A continuación, se detallan los pasos a seguir:

1. En el menú principal (este se puede apreciar en la imagen 34(a)) pulsar el botón asignado a la función Start. Esto abrirá el menú de RetroPie. Aquí, se deberá colocar la opción SCRAPER.

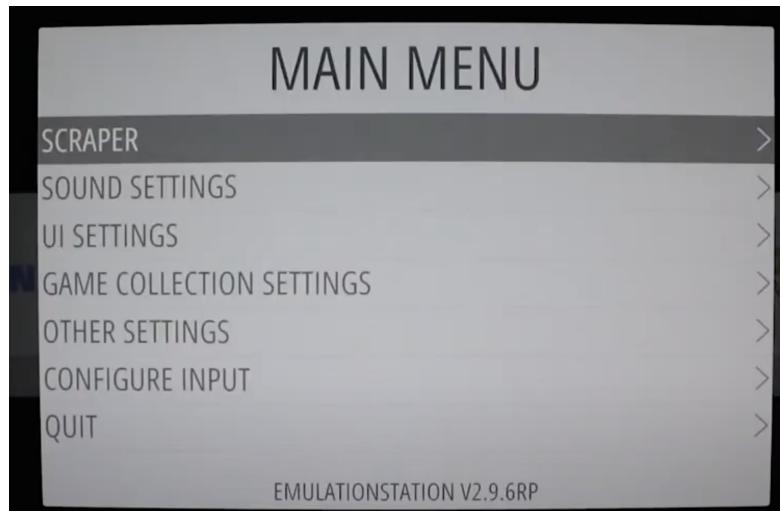


Fig. 35: Menú de RetroPie. Fuente: [5]

2. Seguir la siguiente secuencia:

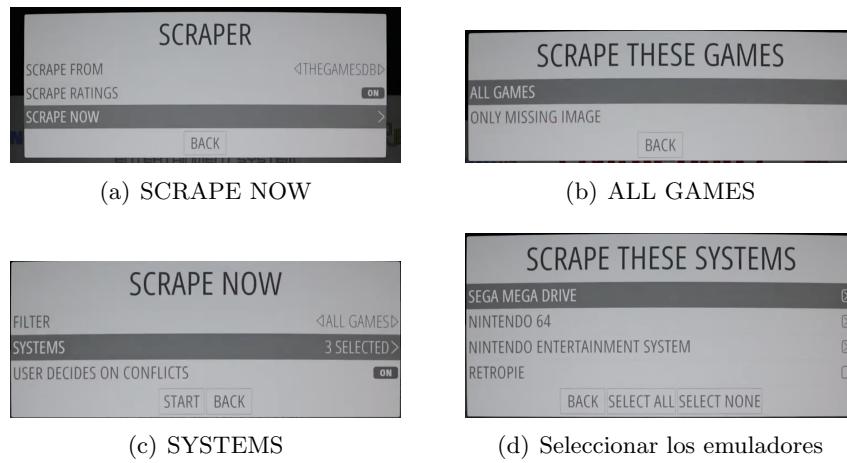


Fig. 36: Configuración de la interfaz. Fuente: [5]

Despues de haber hecho esto, se deberá colocar la opción "USED DECIDES ON CONFLICTS" en off (ver imagen 36(c)).

3. Finalmente, seleccionar la opción start. Una vez hecho todo este proceso, los juegos tendrán una imagen y descripción asociadas, como se puede observar en la figura 37.

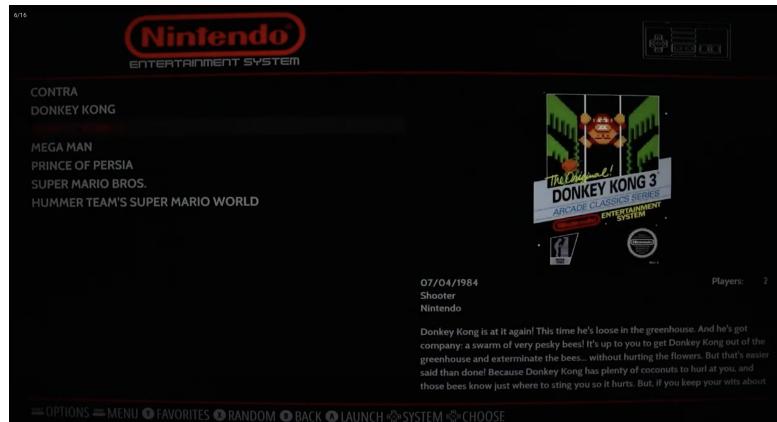


Fig. 37: Imagen de referencia. Fuente: [5]

5. Conclusión

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Referencias

- [1] P. Electric, “Imagen raspberry pi 4b,” <https://cdn11.bigcommerce.com/.../RaspberryPi4B.jpg>, 2025, accedido el 22 de septiembre de 2025.
- [2] Starware, “Imagen nespi,” <https://tienda.starware.com.ar/wp-content/uploads/2021/09/gabinete-carcasa-retroflag-nespi-4-case-raspberry-pi-4-2375-4579.jpg>, 2025, accedido el 22 de septiembre de 2025.
- [3] SysBeards, “Como instalar retropie en raspberry pi 4,” <https://www.youtube.com/watch?v=X-UD7Xmrl5w&t=300s>, 2025, accedido el 22 de septiembre de 2025.
- [4] G. Gamer, “Como instalar retropie en raspberry pi 4,” https://www.youtube.com/watch?v=S0_oJ6S8sSc, 2025, accedido el 22 de septiembre de 2025.
- [5] SysBeards, “Como pasar juegos a retropie — como configurar retropie,” https://www.youtube.com/watch?v=qY7GvgmeYI0&list=PL18-KiZG_nIMiqJ2Ex8ijsEUNBM2A0WBI&index=6, 2025, accedido el 22 de septiembre de 2025.

Anexo A: Información Adicional

A.5: Tipos de LEDs

A.5.0: RGB LEDs

¡Los LED RGB (Rojo-Verde-Azul) son en realidad tres LED en uno! Pero eso no significa que solo pueda hacer tres colores. Debido a que el rojo, el verde y el azul son los colores primarios aditivos, puede controlar la intensidad de cada uno para crear todos los colores del arco iris. La mayoría de los LED RGB tienen cuatro pines: uno para cada color y un pin común. En algunos, el pin común es el ánodo, y en otros, es el cátodo.

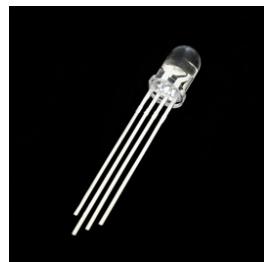


Fig. 38: LED de cátodo transparente común RGB. Fuente:

A.5.0: LEDs con circuitos integrados

Algunos LED son más inteligentes que otros. Tomemos el LED de ciclismo, por ejemplo. Dentro de estos LED, en realidad hay un circuito integrado que permite que el LED parpadee sin ningún controlador externo. Aquí hay un primer plano del IC (el gran chip cuadrado negro en la punta del yunque) que controla los colores.

¡Simplemente enciéndalo y míralo ir! Estos son excelentes para proyectos en los que desea un poco más de acción, pero no tiene espacio para circuitos de control. ¡Incluso hay LED parpadeantes RGB que recorren miles de colores!



Fig. 39: Primer plano del LED de ciclo lento de 5 mm. Fuente:

A.5.0: RGB LEDs

Los LED SMD no son tanto un tipo específico de LED sino un tipo de paquete. A medida que la electrónica se hace cada vez más pequeña, los fabricantes han descubierto cómo meter más componentes en un espacio más pequeño. Las piezas SMD (dispositivo de montaje en superficie) son versiones pequeñas de sus contrapartes estándar. Aquí hay un primer plano de un LED direccionable WS2812B empaquetado en un pequeño paquete 5050.

Los LED SMD vienen en varios tamaños, ¡desde bastante grandes hasta más pequeños que un grano de arroz! Debido a que son tan pequeños y tienen almohadillas en lugar de piernas, no son tan fáciles de trabajar, pero si tiene poco espacio, podrían ser justo lo que recetó el médico.

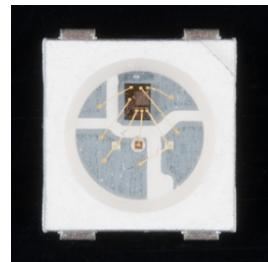


Fig. 40: Primer plano direccional WS2812B. Fuente: