**UNA IMPLEMENTACIÓN HARDWARE-SOFTWARE DEL JUEGO DE MESA MASTERMIND**

**Manuel Guerrero Moñús**

**Máster en Ingeniería Informática; 2020 – 2021**

Índice

[​ 1. Descripción del proyecto 3](#__RefHeading___Toc571_975996222)

[​ 2. Palabras clave 4](#__RefHeading___Toc689_4221368095)

[​ 3. Estado del arte 4](#__RefHeading___Toc940_3167806940)

[​ 4. Sistemas empotrados distribuidos 5](#__RefHeading___Toc715_2108017989)

[​ 4.1 Raspberry Pi 5](#__RefHeading___Toc573_975996222)

[​ 4.2 Elegoo 6](#__RefHeading___Toc609_2108017989)

[​ 5. Periféricos, componentes electrónicos y otros 7](#__RefHeading___Toc1029_2108017989)

[​ 5.1 Diodos LED RGB 7](#__RefHeading___Toc1031_2108017989)

[​ 5.2 Buzzer 7](#__RefHeading___Toc1033_2108017989)

[​ 5.3 Resistencias de 220 Ω 7](#__RefHeading___Toc1035_2108017989)

[​ 5.4 Potenciómetro de 10 KΩ 7](#__RefHeading___Toc1037_2108017989)

[​ 5.5 Protoboard 7](#__RefHeading___Toc1039_2108017989)

[​ 5.6 Teclado matricial 8](#__RefHeading___Toc1146_2108017989)

[​ 5.7 LCD Display 8](#__RefHeading___Toc1148_2108017989)

[​ 6. Comunicación 9](#__RefHeading___Toc1150_2108017989)

[​ 7. Arquitectura del sistema 10](#__RefHeading___Toc1152_2108017989)

[​ 7.1 Diagrama de casos de uso 10](#__RefHeading___Toc1154_2108017989)

[​ 7.2 Diagrama de clases 11](#__RefHeading___Toc1156_2108017989)

[​ 7.3 Diagrama de actividades 13](#__RefHeading___Toc1158_2108017989)

[​ 7.4 Diagrama de despliegue 14](#__RefHeading___Toc642_969233285)

[​ 8. Costes de los componentes del producto 14](#__RefHeading___Toc669_3821925571)

[​ 9. Funcionamiento del sistema 15](#__RefHeading___Toc1279_969233285)

[​ 9.1 Raspberry Pi 16](#__RefHeading___Toc1288_969233285)

[​ 9.2 Elegoo 21](#__RefHeading___Toc1283_969233285)

[​ 10. Conclusiones 22](#__RefHeading___Toc1298_2108017989)

[​ 11. Trabajo futuro 23](#__RefHeading___Toc1300_2108017989)

[​ 12. Referencias 24](#__RefHeading___Toc1302_2108017989)

# 1. Descripción del proyecto

Este proyecto ha consistido en la elaboración de una versión hardware-software de un clásico de los juegos de mesa, Mastermind. Un juego de dos jugadores en el que un jugador “codificador” le propone adivinar a otro jugador “decodificador” una clave de color, cuya longitud habitual suele estar entre los 4 y 6 colores.

Por norma general, el jugador “decodificador” dispone de hasta 10 rondas para averiguar esta clave secreta, durante las cuales, este irá recibiendo *feedback* del jugador “codificador” para saber cómo de cerca está de adivinar la clave, indicándole mediante un código de color negro-blanco-nada si cada color que ha escogido está correctamente ubicado en la clave secreta, o en otro lugar en la clave, o si no aparece en esta. Finalmente, el juego acaba cuando el jugador “decodificador” agota todas sus rondas de juego sin adivinar la clave, en cuyo caso ganaría el jugador “codificador”, o bien cuando el jugador “decodificador” adivina la clave secreta al completo, en cuyo caso ganaría éste.

La implementación de este juego descrito anteriormente en el ámbito de los sistemas empotrados distribuidos, se ha realizado de la siguiente manera:

Por un lado, contamos con un ordenador Raspberry Pi, sistema que está pensado para ser utilizado por el jugador “codificador”, pues este cuenta con una aplicación software hecha en Python que incorpora diversas funcionalidades, como el acceso a una ayuda para conocer qué es el Mastermind y como se utiliza la aplicación en sí, un menú de configuración para crear un perfil propio de partida y así poder variar la dificultad o la forma en la que se juega, un menú para lanzar la partida y, finalmente, también nos otorga la posibilidad de detectar alguna partida que haya quedado incompleta ante la pérdida de la comunicación con el otro extremo, de esta forma, podremos restaurarla y acabarla correctamente.

Por otro lado, contamos con un microcontrolador Elegoo, sistema que está pensado para ser utilizado por el jugador “decodificador”, que ha sido programado en C++ y contiene tanto la lógica del juego como la necesaria para comunicarse con los periféricos de entrada/salida requeridos para recrear esta versión electrónica del juego. En concreto, estos periféricos han sido los siguientes: un teclado matricial 4x4, a través del cual el jugador puede realizar sus elecciones para tratar de adivinar el código secreto, 4 leds RGB, que informan de cuanto se aproxima la clave escogida a la clave secreta, un *buzzer* para la emisión de sonidos de acierto o fallo, pero también para la emisión de breves melodías de victoria o derrota y, por último, una pantalla LCD de 2x16 a través de la cual informar al usuario de la ronda del juego y también de sus pulsaciones en el teclado matricial.

Los dos sistemas se comunican de forma bidireccional a través del puerto serie, de esta manera, ambos forman un todo.

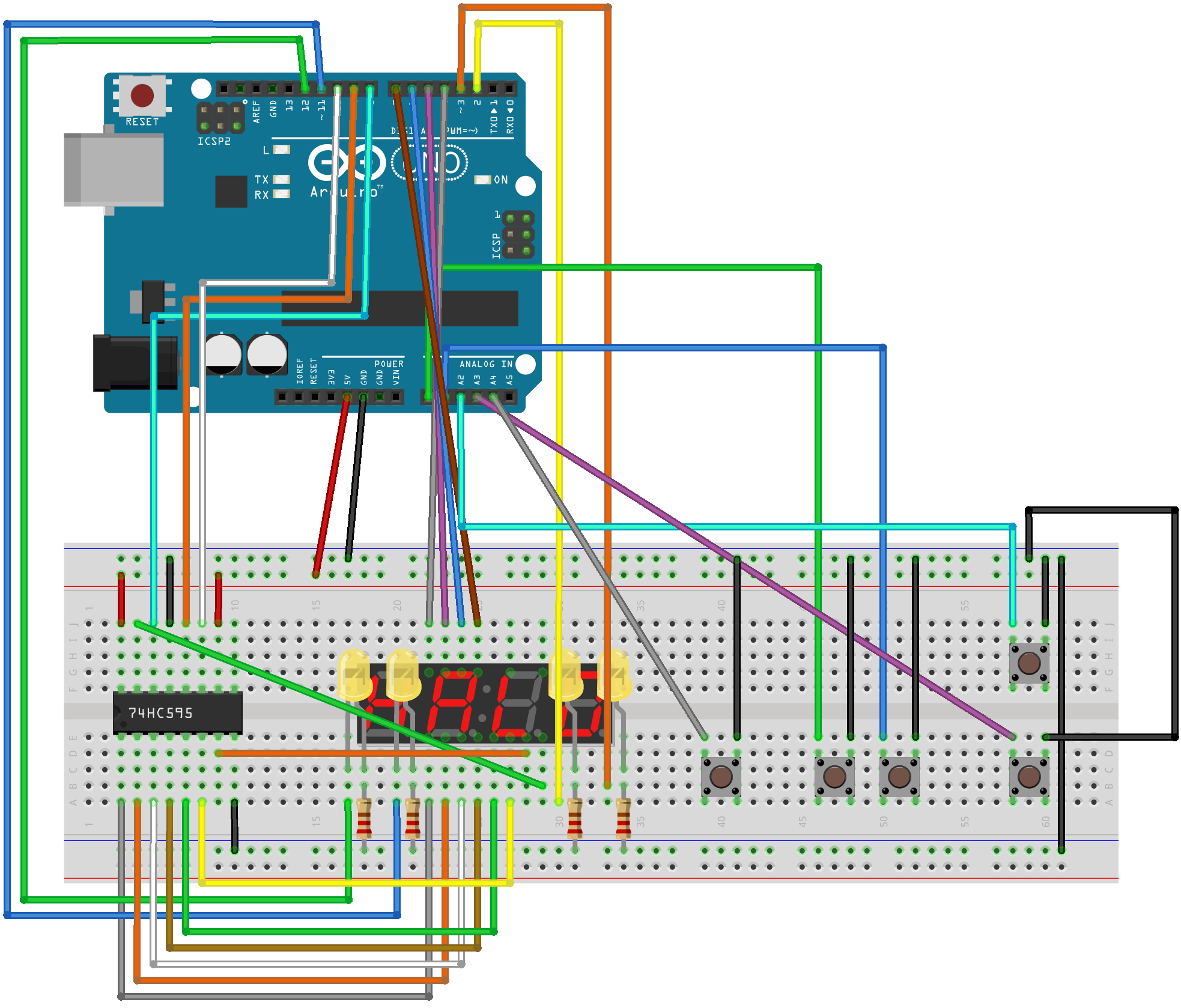
# **2. Palabras clave**

Mastermind, Raspberry, Elegoo, Python, C++, UML, sistema.

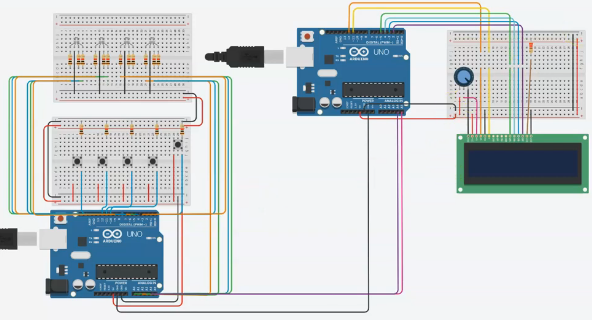
# 3. Estado del arte

En esta sección se ha realizado una investigación sobre otros proyectos Mastermind que sirvan como inspiración para la elaboración de este trabajo. Los proyectos que me resultaron más ilustrativos son mostrados a continuación.

Una de las implementaciones que más me llamó la atención fue la siguiente, ya que en esta versión se jugaba con códigos numéricos, leds y un 7 segment 4 digit display en la misma protoboard, una idea similar a la mía en sus orígenes:

En este diagrama de cableado se puede observar cómo hacer una implementación viable mediante el uso de un 7 segment 4 digit display para mostrar los dígitos numéricos, 5 pulsadores, 2 de los cuales son para desplazarse por el código numérico, otros 2 para seleccionar un dígito de forma ascendente o descendente y, otro para confirmar la selección del código numérico actual. Un registro de desplazamiento para poder controlar con menos conexiones el 7 segment 4 digit display, y 4 leds de color amarillo, que indican mediante un brillo fijo si el dígito está correctamente ubicado en el código secreto, un estado parpadeante para informarnos que el dígito está colocado en otra posición, y la ausencia de brillo para señalar que el dígito no está en el código.

En esta segunda implementación ya se hace uso de un par de microcontroladores Arduino Uno para crear un Mastermind como un único sistema formado por otros subsistemas distribuidos:

Como puede verse, este prototipo hace uso de varias placas de pruebas para distribuir de forma más cómoda los componentes que forman este sistema de entretenimiento.

En este caso, el sistema usa un LCD display como pantalla informativa para dar a conocer los aciertos y los fallos, un potenciómetro marcado en azul para regular la intensidad del LCD display y, por último, leds RGB cuyo color se cambia con los botones que están frente a ellos.

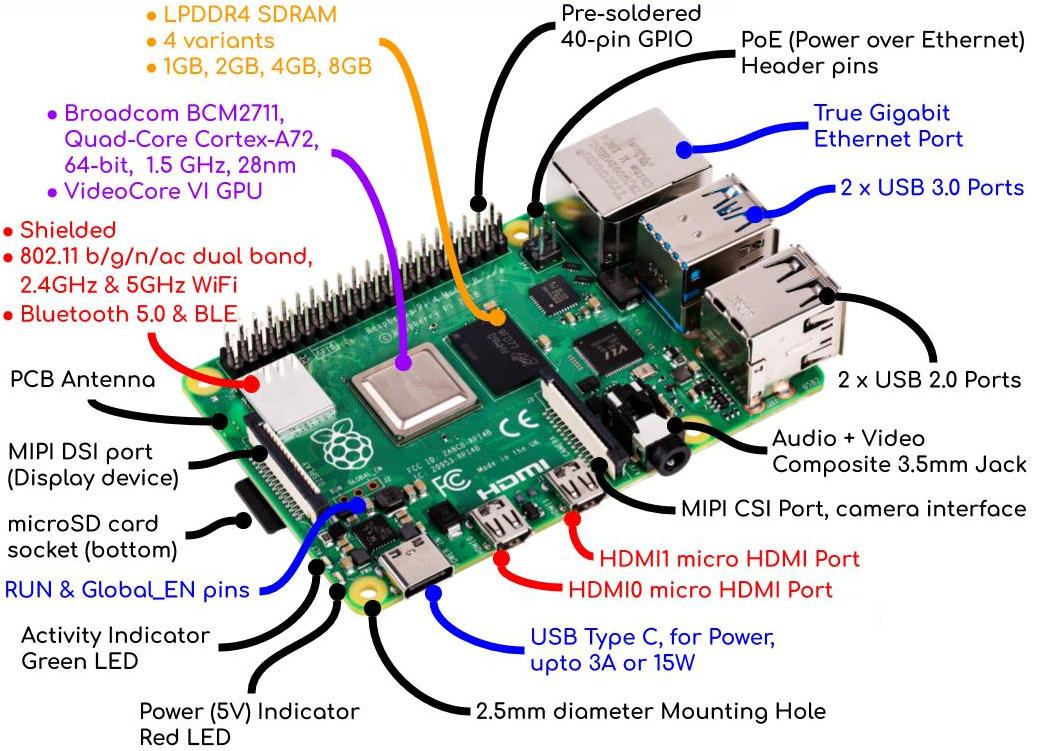
# 4. Sistemas empotrados distribuidos

En este capítulo se expone la historia y las principales características de los sistemas empotrados distribuidos utilizados para construir el sistema de entretenimiento.

### 4.1 Raspberry Pi

Raspberry Pi es un ordenador del tamaño de una tarjeta de crédito que apareció por primera vez en el mercado en el año 2012. Se suele utilizar para aprender a programar aplicaciones o videojuegos, pero también es muy utilizado para construir proyectos de electrónica. Su desarrollo es realizado por la Fundación Raspberry Pi, fundada en 2009 como una organización benéfica con sede en Caldecote (Reino Unido) que trabaja por el afán de potenciar la informática y la creatividad.

Para la realización de este proyecto se ha utilizado concretamente el modelo 4 B de la Raspberry Pi, cuyos principales elementos y prestaciones pueden apreciarse en la siguiente imagen.



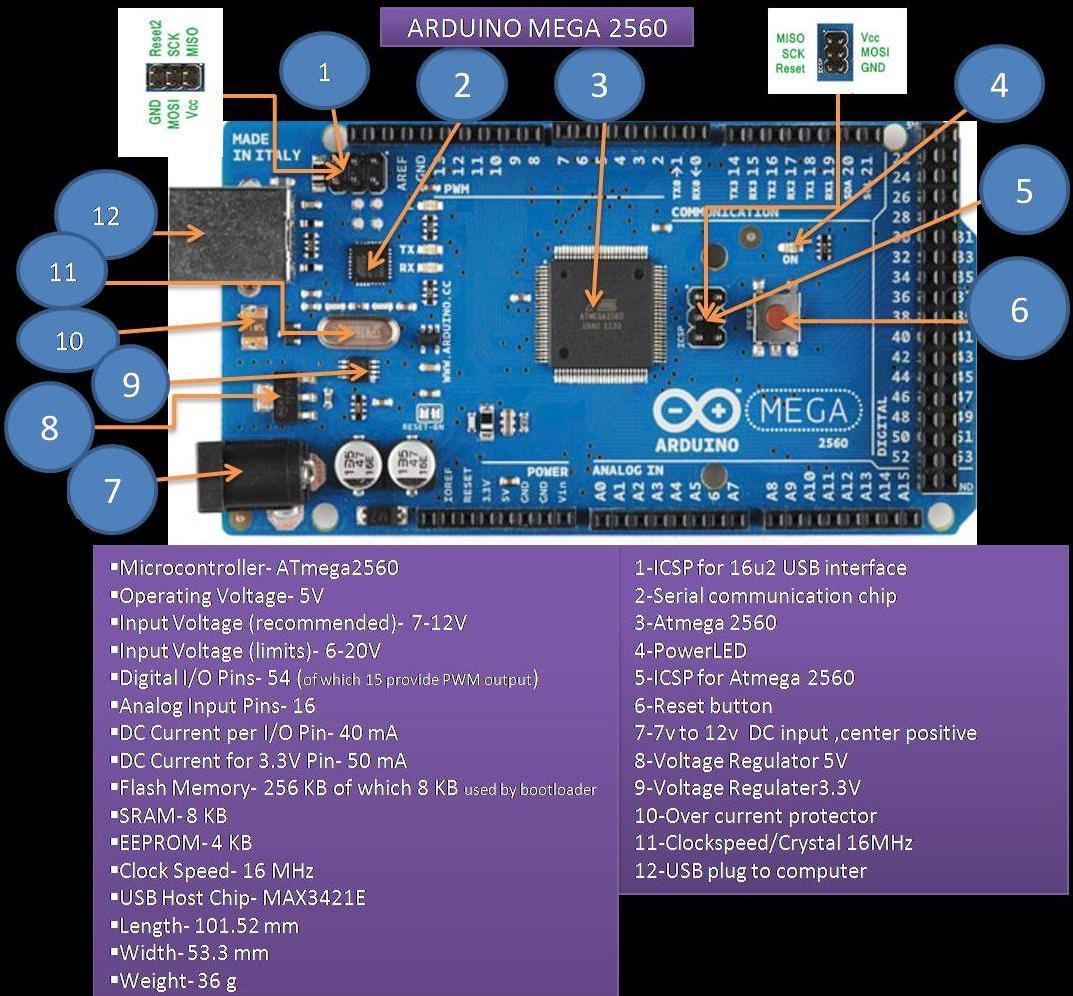
La elección de este sistema para la realización del proyecto se debió a lo siguiente:

* Gran comunidad de usuarios y mucha documentación al respecto.
* Ideal para introducirse poco a poco en proyectos de electricidad y electrónica.
* Dispositivo estandarizado y comunidad libre.

### 4.2 Elegoo

Elegoo es una compañía, fundada en 2015 por Chris Hong en Shenzhen (China), que se dedica a la venta de kits para el desarrollo de proyectos de electrónica para que pudieran ser utilizados por estudiantes y otros entusiastas de esta área. De entre sus productos más cotizados, podemos destacar los kits de las placas Elegoo, que son unos microcontroladores compatibles con Arduino. También venden algunos kits de iniciación a la robótica e impresoras 3D.

Para la realización de este proyecto se ha utilizado concretamente el modelo Mega 2560 R3 de Elegoo, cuyos principales elementos y prestaciones pueden apreciarse en la siguiente imagen.



La elección de este sistema para la realización del proyecto se debió a lo siguiente:

* Precio bastante económico, tanto del dispositivo, como de los kits Elegoo.
* Clon del microcontrolador Arduino, ideal para aprender electrónica.
* Permite valerse de la comunidad de usuarios y la documentación de Arduino.

# 5. Periféricos, componentes electrónicos y otros

Aquí se detallan los componentes utilizados para la implementación hardware.

### 5.1 Diodos LED RGB



Utilizar varios diodos led de color rojo, verde y azul, complicaría el diseño del juego, por ello se ha optado por utilizar diodos que pueden actuar como cualquiera de ellos e incluso como una combinación de estos colores básicos. Estos diodos son realmente la unión de tres diodos led, cada uno de los cuales aporta uno de los colores básicos, y todos comparten una toma a tierra (GND / Cathode) común.

### 5.2 Buzzer



Un *buzzer* es un dispositivo que emite una onda de sonido que varía en función de la frecuencia y la duración de una señal eléctrica a la que es sometido. Para utilizar este componente, Elegoo dispone de algunos pines digitales que pueden ser utilizados para emitir señales de salida analógicas (están marcados con el símbolo “~”, Pulse Width Modulation).

### 5.3 Resistencias de 220 Ω

Una resistencia es aquel componente cuya misión es oponerse en mayor o menor medida al paso de la corriente eléctrica.

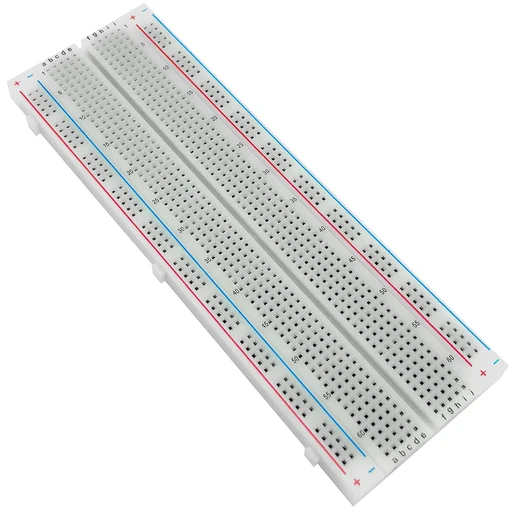


### 5.4 Potenciómetro de 10 KΩ



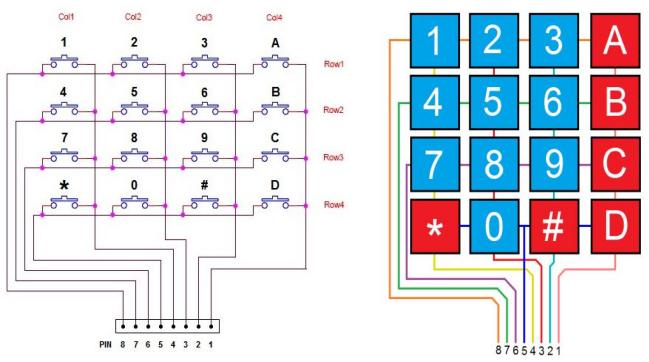
Un potenciómetro es un componente que actúa como una resistencia variable, la cual puede ser ajustada manualmente para regular su valor entre los 0 Ω y los 10 KΩ.

### 5.5 Protoboard

****

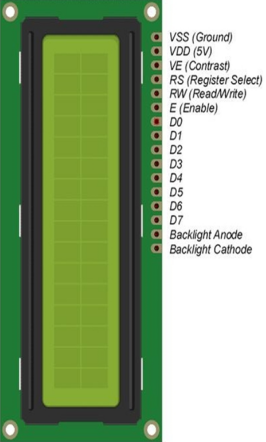
Una placa de pruebas es un tablero con varios orificios que se encuentran conectados eléctricamente entre sí de forma interna siguiendo una línea, la cual permite insertar componentes para construir prototipos de circuitos.

### 5.6 Teclado matricial



El teclado matricial funciona de la siguiente manera: los pines de las filas (8 - 5) son sometidos a tensión de forma secuencial de esta manera, cuando coincide que una fila es sometida a tensión y un botón de la misma es presionado, entonces se podrá leer en los pines de las columnas (1 - 4), la cual es la que posee un voltaje de salida. Llegados hasta aquí, gracias a la intersección del voltaje en la fila y la columna se puede saber que tecla es la que se está pulsando.

### 5.7 LCD Display



La pantalla de cristal líquido está formada por 2 filas de 16 celdas de píxeles monocromo, esta se configura a través de los pines:

VSS: Conexión a tierra (GND).  
VCC: Conexión a la alimentación (5V).  
V0: Conectado a un potenciómetro para regular el contraste (brillo del color blanco).  
RS: Indica si se están enviando caracteres o comandos de control.  
R/W (1/0): Modo escritura (GND).  
E: Habilita la recepción de datos.  
D0 – D4: Para la transmisión de datos.  
BA “o” LED(+): Ánodo del LED que ilumina desde el fondo (Retroiluminación; 5V).  
BC “o” LED(-): Cátodo del LED que ilumina desde fondo (Retroiluminación; GND).

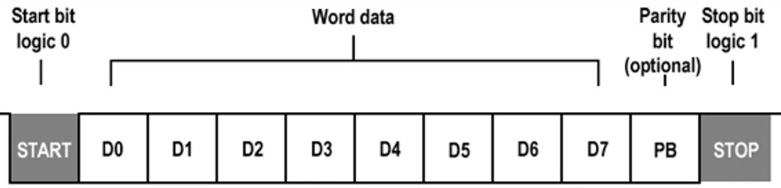
# 6. Comunicación

La comunicación del proyecto se realiza a través del puerto serie, también conocido como UART (Universally Asynchronous Receiver/Transmitter), a través de este se comunicarán el microcontrolador Elegoo y el computador Raspberry Pi.

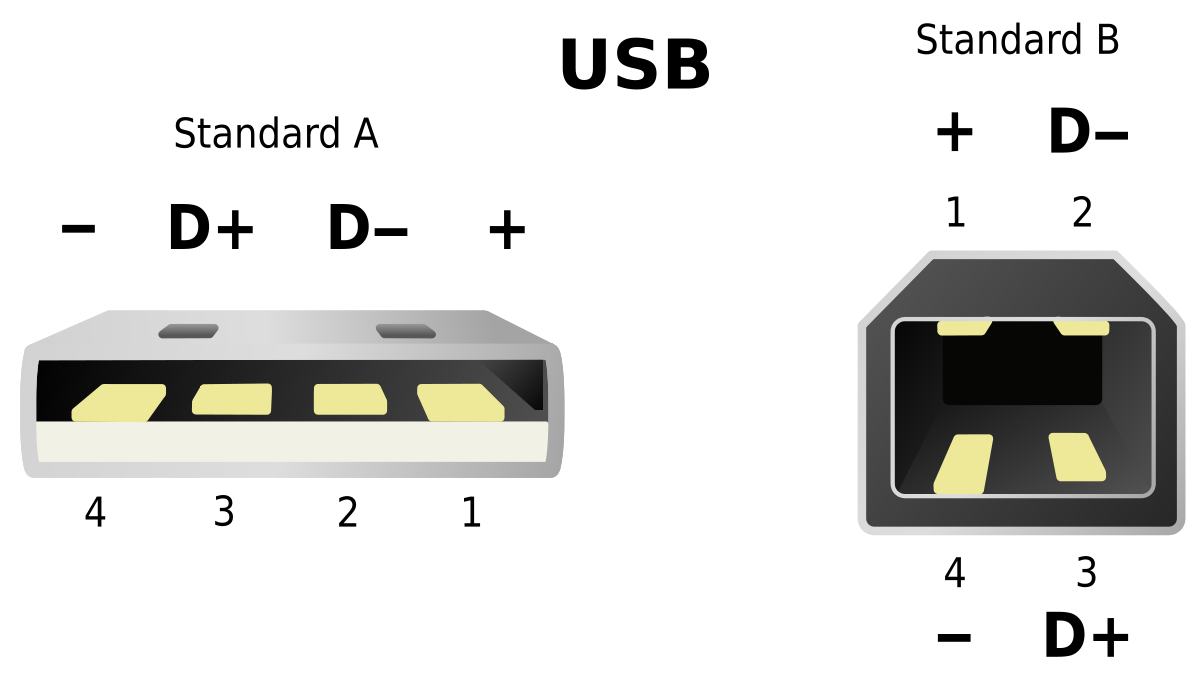
Para hacer uso del puerto serie, en Elegoo hay que utilizar los pines 0 (Rx) y 1 (Tx) para poder comunicarse, mientras que en la Raspberry Pi se utilizan los pines 14 (Tx) y 15 (Rx), sin embargo, en este proyecto se ha optado por utilizar un interfaz USB del computador Raspbery Pi, ya que este nos permite disponer de un puerto serie virtual que sirve tanto para comunicar Raspberri Py con Elegoo, como para poder proporcionarle a este último su alimentación.

Cuando uno de los dispositivos hace uso del puerto serie para transmitir, lo que hace es enviar a través de él bytes de datos individuales de forma secuencial, ideal para transmitir desde caracteres individuales hasta frases enteras, pero para que ambos dispositivos puedan comunicarse a través del puerto serie, ambos tienen que estar configurados de igual forma para poder tanto codificar como decodificar los datos que son transmitidos.

En este proyecto, la configuración para la comunicación a través del puerto serie es la siguiente para ambos dispositivos:

* Velocidad de la comunicación: 9600 (baudios).
* N.º Bits de datos: 8 (1 byte).
* N.º Bits de paridad: no.
* N.º Bits de parada: 1.
* N.º Bits de inicio: 1.

El cable a través del cual tiene lugar la comunicación es un cable USB de tipo A-B, estando el extremo del standard A conectado al puerto USB de la Raspberry Pi, y el standard B conectado al puerto de Elegoo. A continuación se indica el propósito de cada uno de sus pines.



Pin:

1 (+): alimentación (5V).

2 (-): tierra (GND).

3 (D+): datos (host → device).

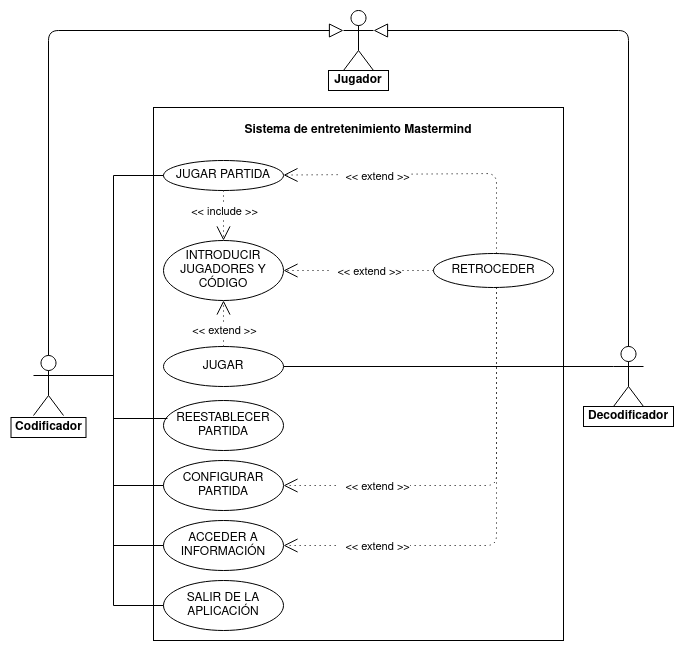
4 (D-): datos (device → host).

# 7. Arquitectura del sistema

Este capítulo recoge diferentes aspectos relacionados con el software implementado en los sistemas Raspberry Pi y Elegoo, para elaborarlos se ha recurrido al estándar UML (*Unified Modeling Language*).

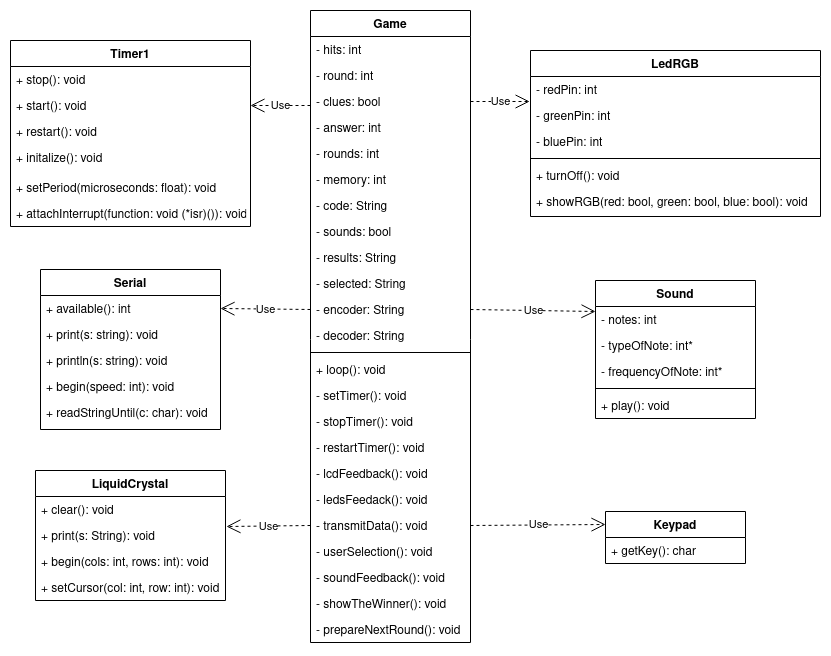
### 7.1 Diagrama de casos de uso

Este diagrama sirve para representar como interactuarán los usuarios con el sistema.

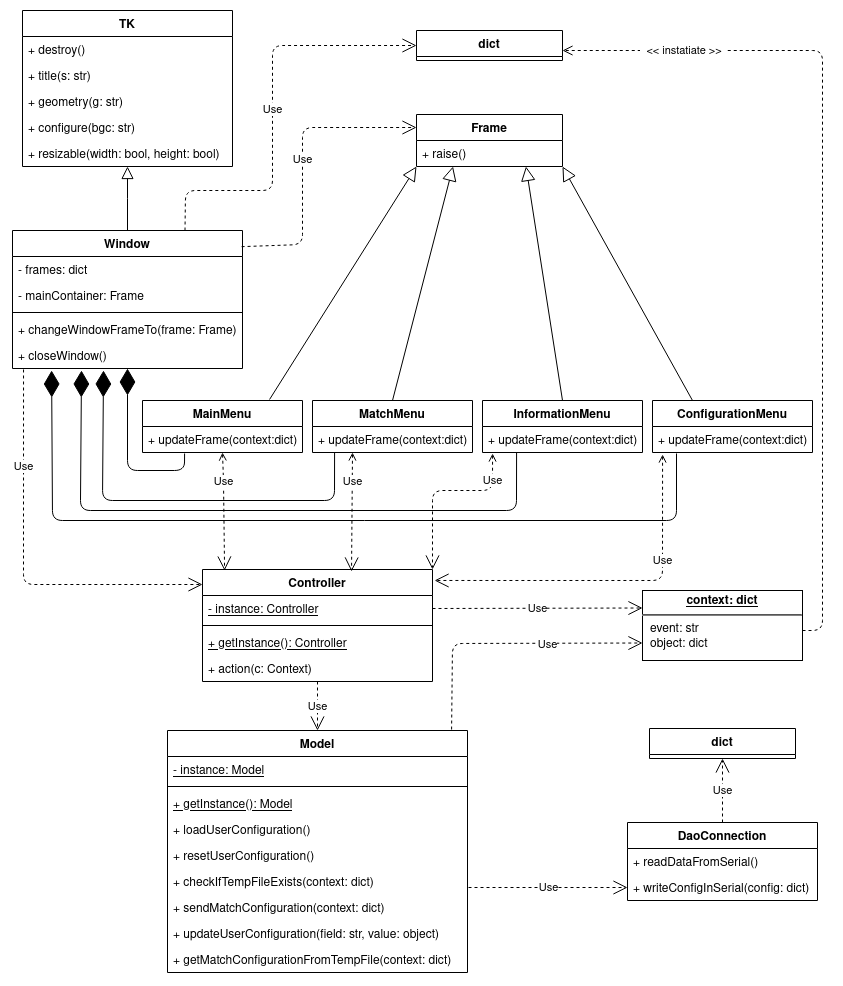
****

### 7.2 Diagrama de clases

El siguiente diagrama es relativo a la implementación hecha en el microcontrolador Elegoo, su objetivo es ilustrar, con un alto nivel de abstracción, los recursos que son necesarios para que el juego pueda funcionar.

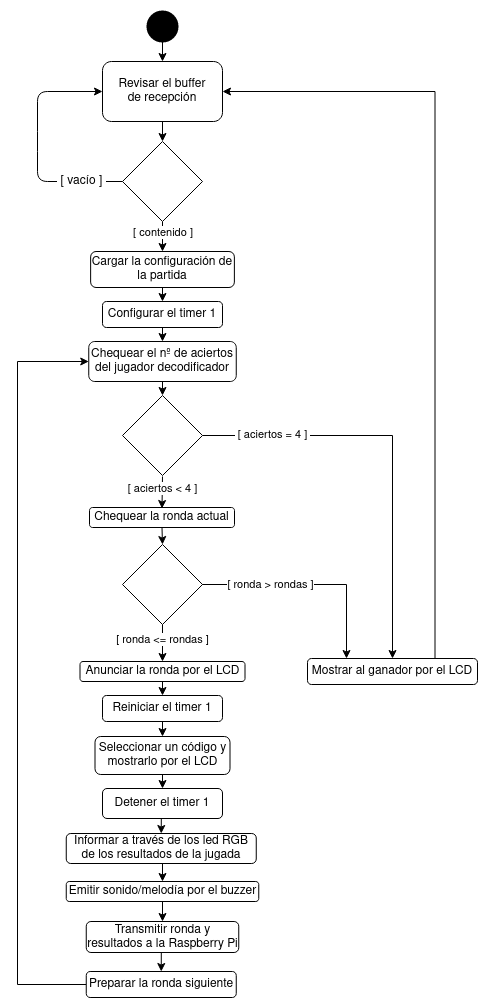
****

El siguiente diagrama es relativo a la implementación realizada en el computador Raspberry Pi, cuyo objetivo es ilustrar, con un alto nivel de abstracción, la arquitectura software de la aplicación que sirve para configurar y lanzar, o restablecer el juego. Esta arquitectura es: Model View Controller.

****

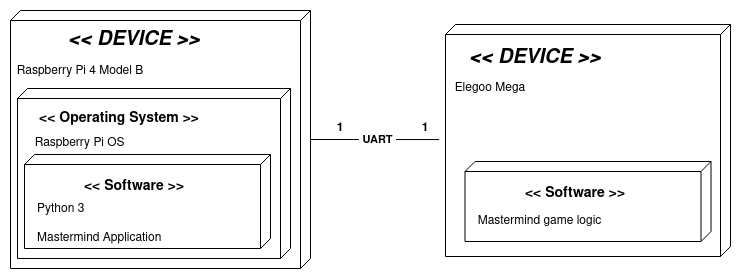
### 7.3 Diagrama de actividades

El siguiente diagrama de actividades muestra, en líneas generales, el comportamiento completo de una partida de Masterind. La secuencia de acciones se ejecuta en el microcontrolador Elegoo y transmite el interactuar con un sistema de entretenimiento.



### 7.4 Diagrama de despliegue

El siguiente diagrama muestra cómo se distribuyen y comunican los nodos físicos, así como el software que emplea cada uno.



# 8. Costes de los componentes del producto

En este capítulo se detallan los costes de los componentes del producto:

* Raspberry Pi 4 Desktop Kit: 130€
  + Incluye:
    - Teclado y ratón.
    - Tarjeta micro-SD.
    - Cable de alimentación.
    - 2 Cables micro-HDMI.
    - Raspberry Pi 4 model B (4 GB RAM)
* Elegoo Mega: 14€
* Resto de los componentes: 35€ \*

TOTAL: 179€

\* Se toma como precio de referencia el coste del kit de electrónica de nivel medio de la marca Elegoo, ya que el precio de los componentes de forma individual es muy variable, pues depende mucho de la plataforma de comercio electrónico y la cantidad de unidades que se quieran meter en la oferta.

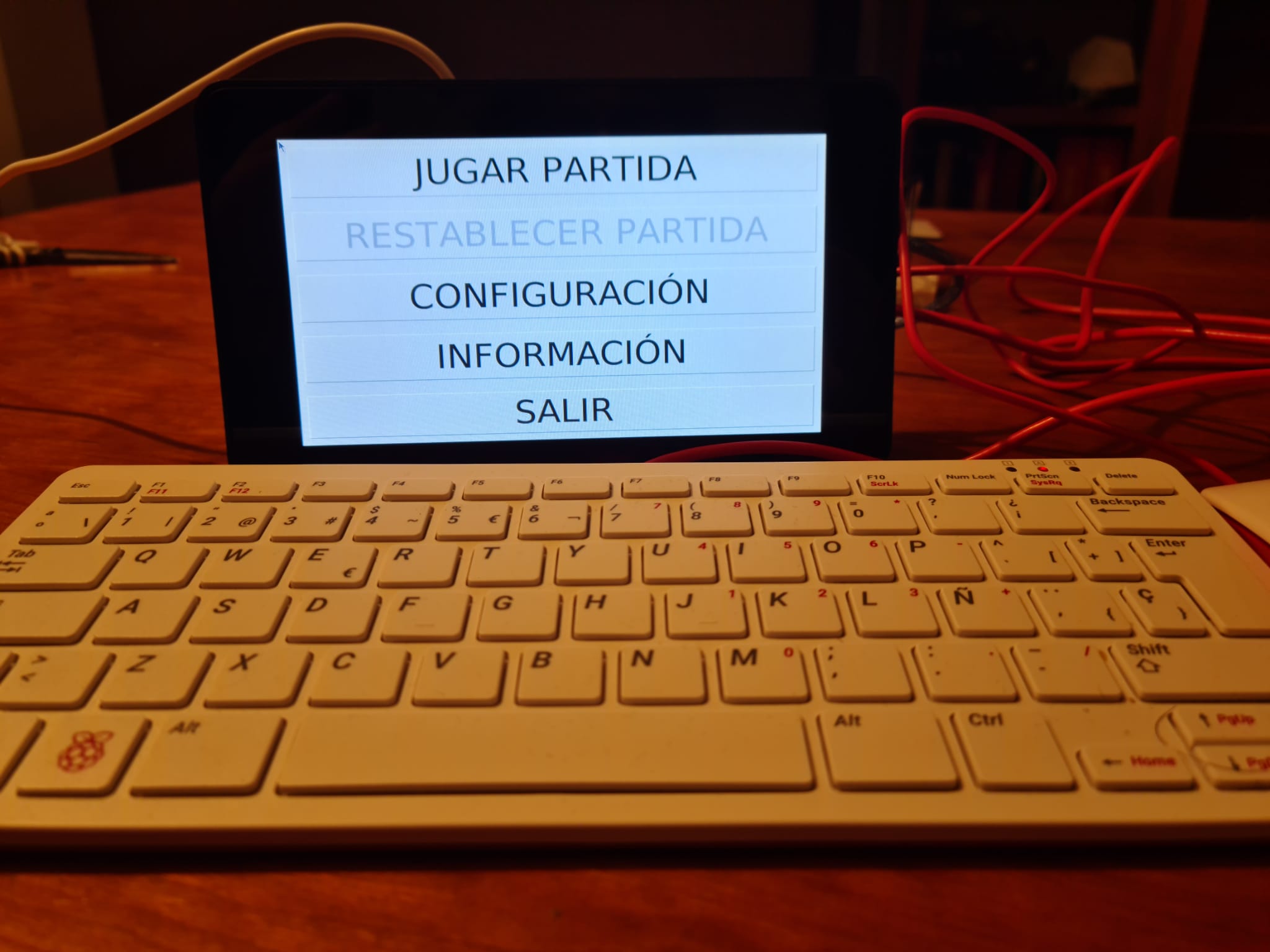
# 9. Funcionamiento del sistema

En este capítulo se expone brevemente con ilustraciones, cómo un usuario podría interactuar con los sistemas empotrados distribuidos Elegoo y Raspberry Pi, los cuales conforman este sistema de entretenimiento Mastermind, que se muestra en la siguiente imagen.



### **9.1 Raspberry Pi**

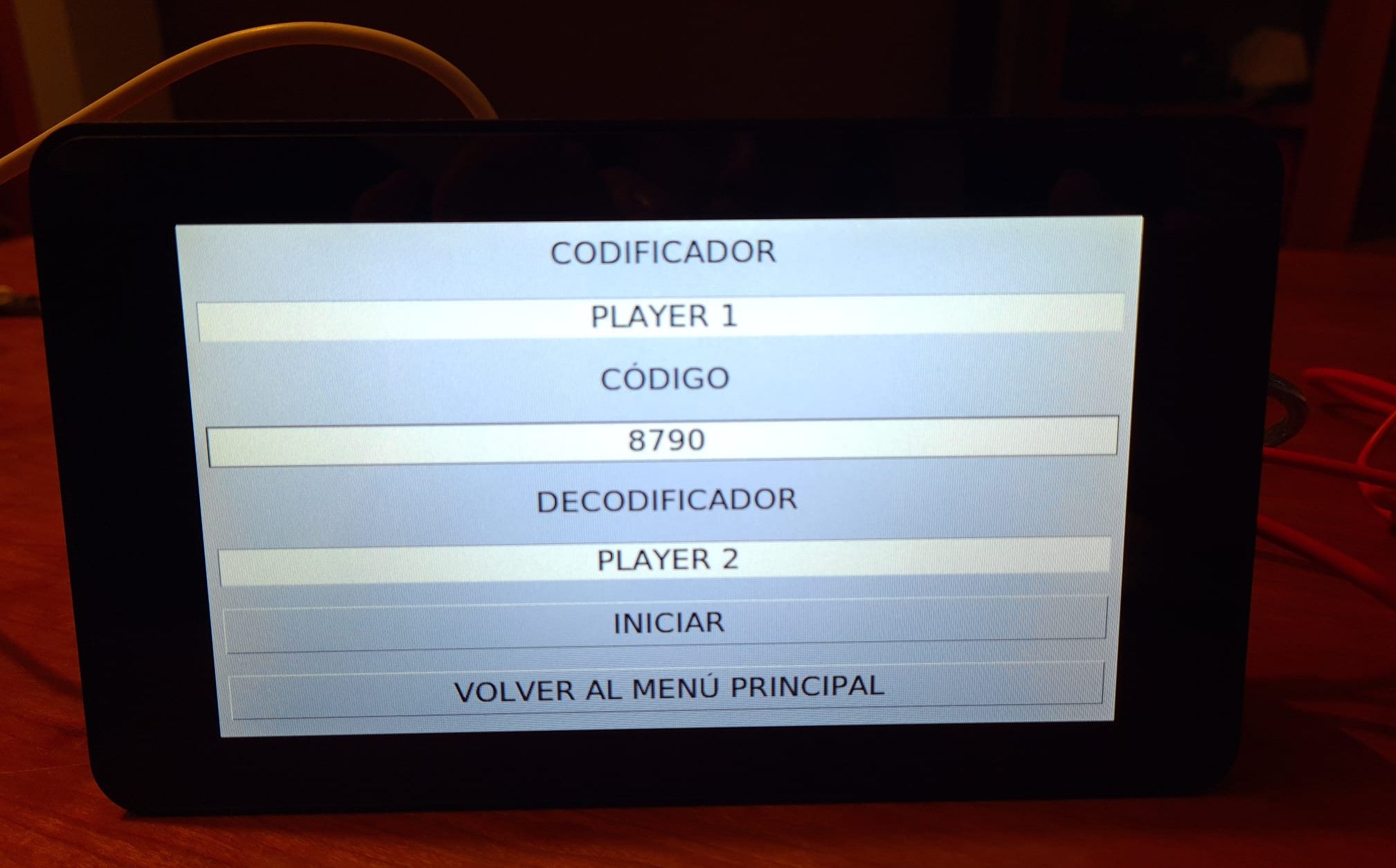
En esta primera imagen se muestra el menú principal de la aplicación desarrollada para la interacción con el sistema Raspberry Pi, una aplicación con un aspecto muy básico pero intuitiva que le permitirá al usuario elegir entre diversas opciones.



Cuando el jugador “codificador” seleccione la opción “JUGAR PARTIDA”, entonces acabará en el menú de la siguiente imagen, en el cual deberá introducir ciertos datos:

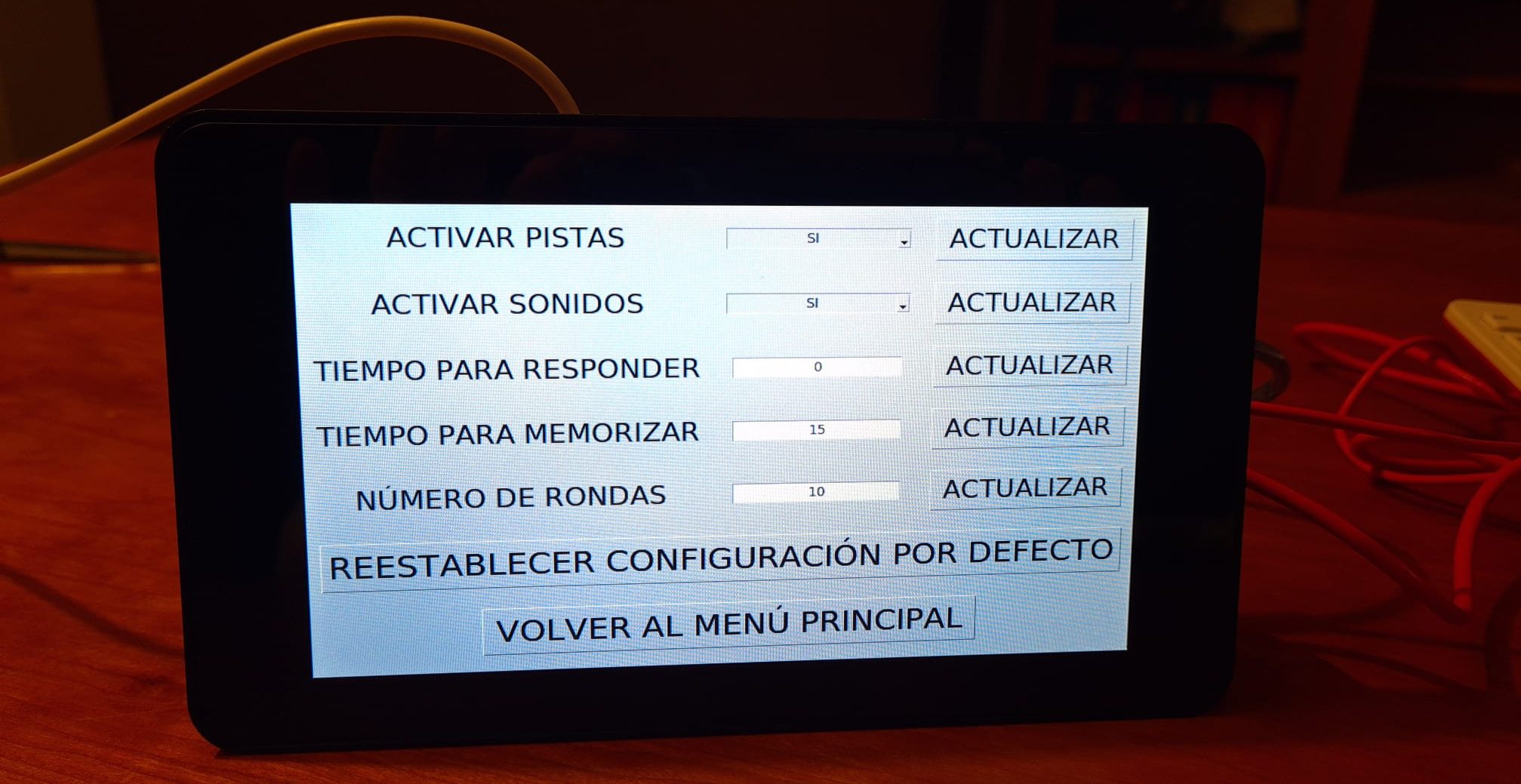
* Nombre del jugador “codificador” (máximo 16 caracteres).
* Código que el jugador “decodificador” deberá de averiguar (4 dígitos).
* Nombre del jugador “decodificador” (máximo 16 caracteres).

Aviso: hay controles implementados que garantizan que las condiciones se cumplan.



Una vez se han rellenado los campos de texto correctamente puede iniciarse la partida pulsando el botón “INICIAR”, a partir de ese momento la aplicación dejará de responder al jugador “codificador” hasta que esta le avise gráficamente a través de un mensaje de texto que la partida ha terminado en el lado del microcontrolador Elegoo. Mientras la partida está en marcha, el sistema Raspberry Pi recoge ronda a ronda la siguiente información: código elegido por el jugador “decodificador” y resultado de elegir dicho código. Esta recolección de información se realiza para garantizar la supervivencia de la partida aun cuando la conexión entre los dispositivos se pierde, es decir, una cuestión de resiliencia.

Si el jugador “codificador” hubiera seleccionado en el menú principal el botón de “CONFIGURACIÓN”, entonces llegaría a la siguiente pantalla.



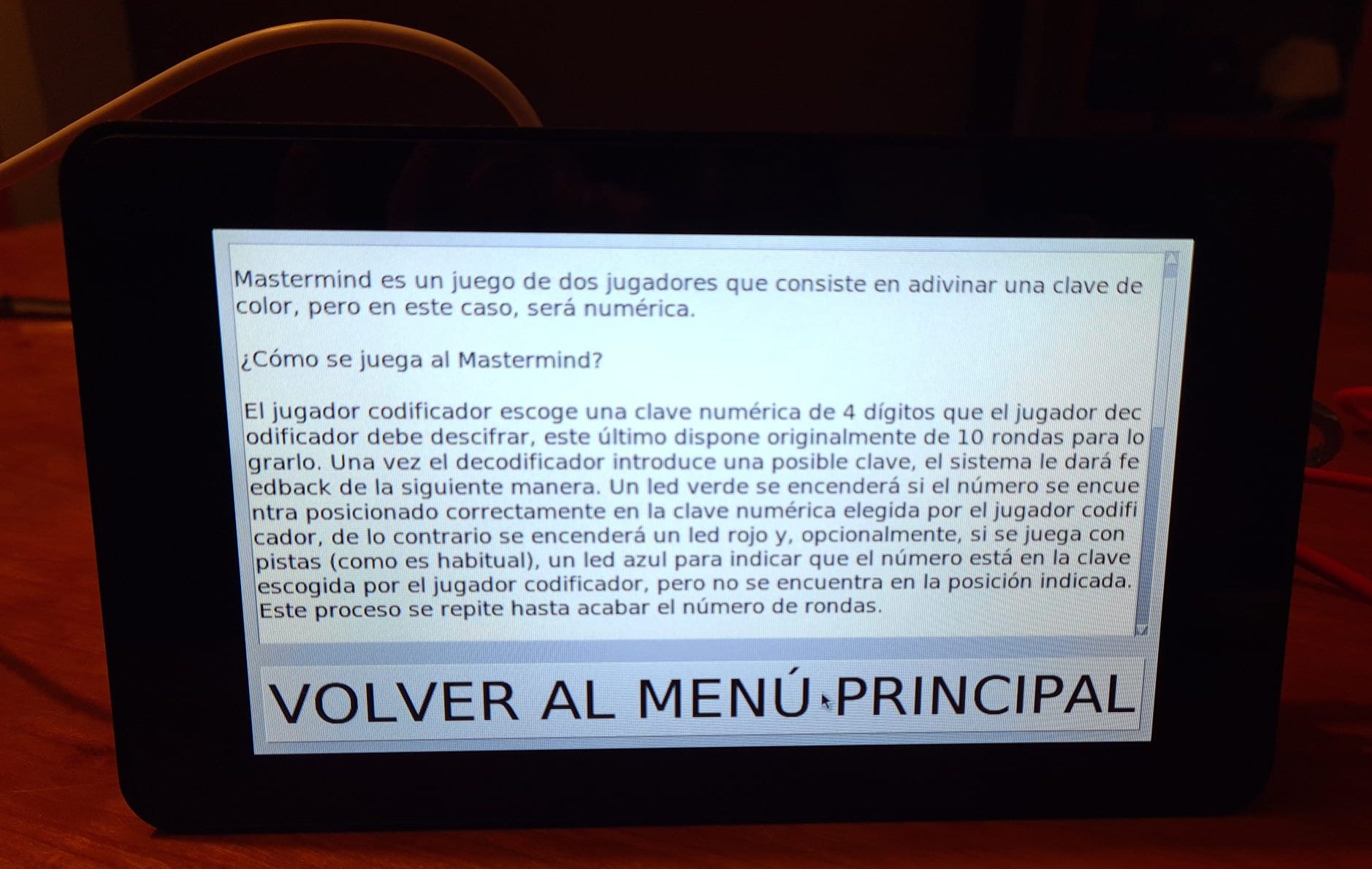
En ella, el jugador “codificador” podrá ajustar las siguientes opciones de juego, pero debe de acordarlas de antemano con el jugador “decodificador”:

* Pistas: por defecto se jugará con ellas, pues es la forma habitual. En este modo los leds RGB brillarán de 3 formas distintas: en verde, si el dígito seleccionado por el jugador “decodificador” está en la posición correcta en la clave escogida por el jugador “codificador”, azul, si el dígito está en la clave secreta pero no en la posición correcta y, finalmente, rojo, si el número no está en la clave. Por otra parte, se ha implementado otra forma de juego sin pistas, en este modo de juego no se indica cuando un dígito está fuera de lugar, es decir, solo se avisa de si un dígito está o no en su sitio en la clave secreta.
* Sonidos: por defecto se juega con sonidos. Después de que el “decodificador” realice una elección de un código de 4 dígitos, se escuchará un sonido a modo de acierto si este ha conseguido al menos acertar un dígito de la clave secreta, de lo contrario se emitirá un sonido a modo de fallo. Por otra parte, se emitirá una melodía de victoria si el jugador “decodificador” logra adivinar la clave secreta antes de que se le agoten todas las rondas de juego, de lo contrario, esta melodía será una melodía de derrota.
* Tiempo para responder: su valor por defecto es 0, lo cual indica que no se va a presionar al jugador “decodificador” para que responda en un tiempo limitado. Cuando este valor se varía, entonces el “decodificador” deberá responder rápido antes de que el juego pase automáticamente de ronda. Este tiempo se mide en segundos y su valor puede ajustarse en el intervalo [0 – 60].
* Tiempo para memorizar: por defecto son 4 segundos, pero el valor puede ser ajustado en el rango [4 – 60].
* Número de rondas: por defecto se juega a 10 rondas como en el juego original, pero el número puede ajustarse en el intervalo [4 – 20].

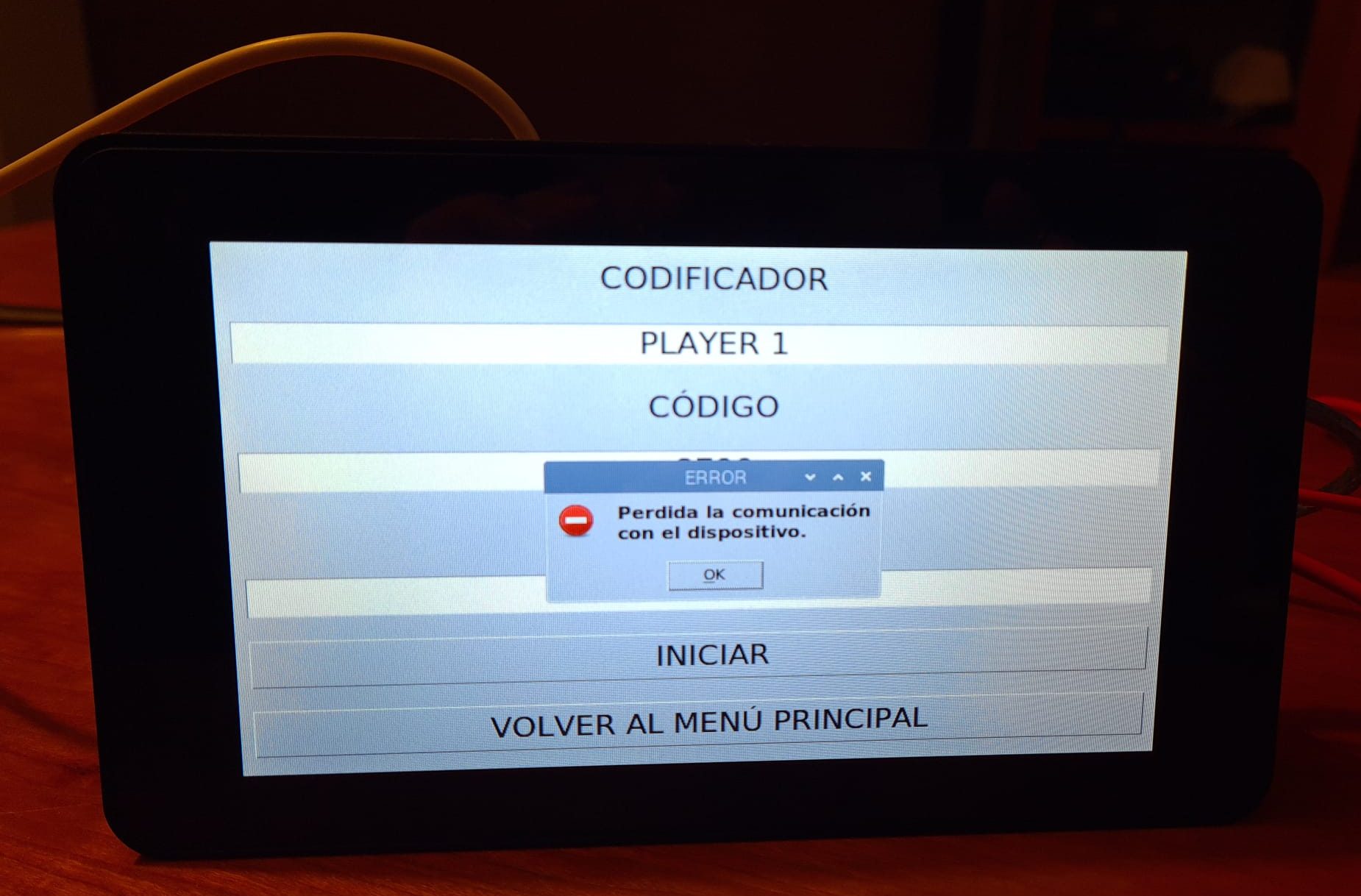
Aviso: hay comprobaciones que garantizan que los ajustes no se salgan de los rangos.

Para realizar un cambio en alguna de las opciones de configuración, primero hay que cambiar el valor del campo de texto o desplegable y luego pulsar el botón actualizar de la fila para que dicho cambio entre en vigor. En última instancia, destacar que siempre se podrá volver a dejar el juego en su configuración por defecto si se pulsa el botón que expresa este mismo deseo.

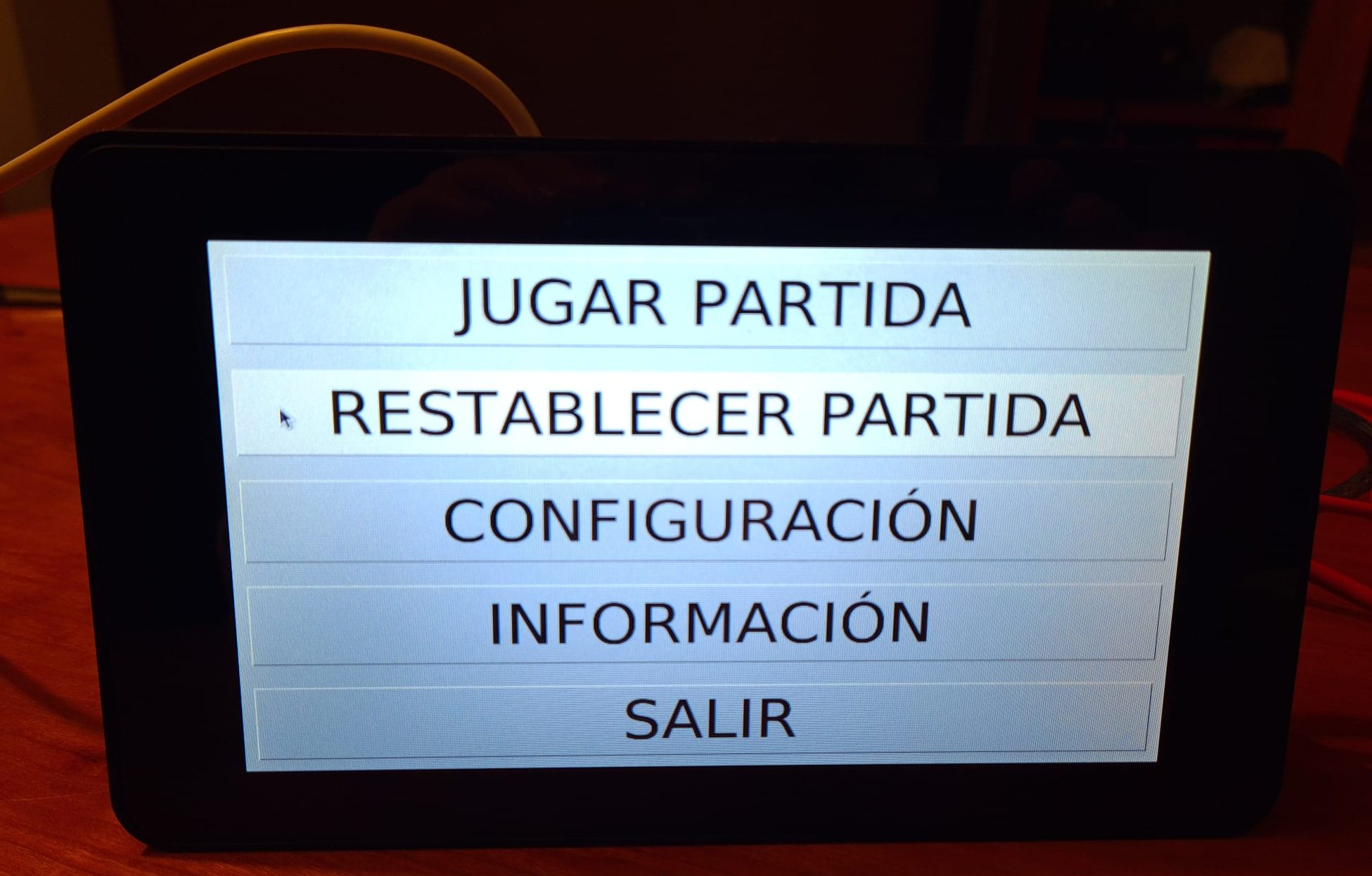
El último menú de la aplicación que el jugador “codificador” puede visitar es el menú de “INFORMACIÓN”. Este menú, cuya imagen se muestra a continuación, tiene las siguientes utilidades: explicar al usuario qué es el Mastermind, cómo se juega a esta versión electrónica del juego y cómo se utiliza esta aplicación.



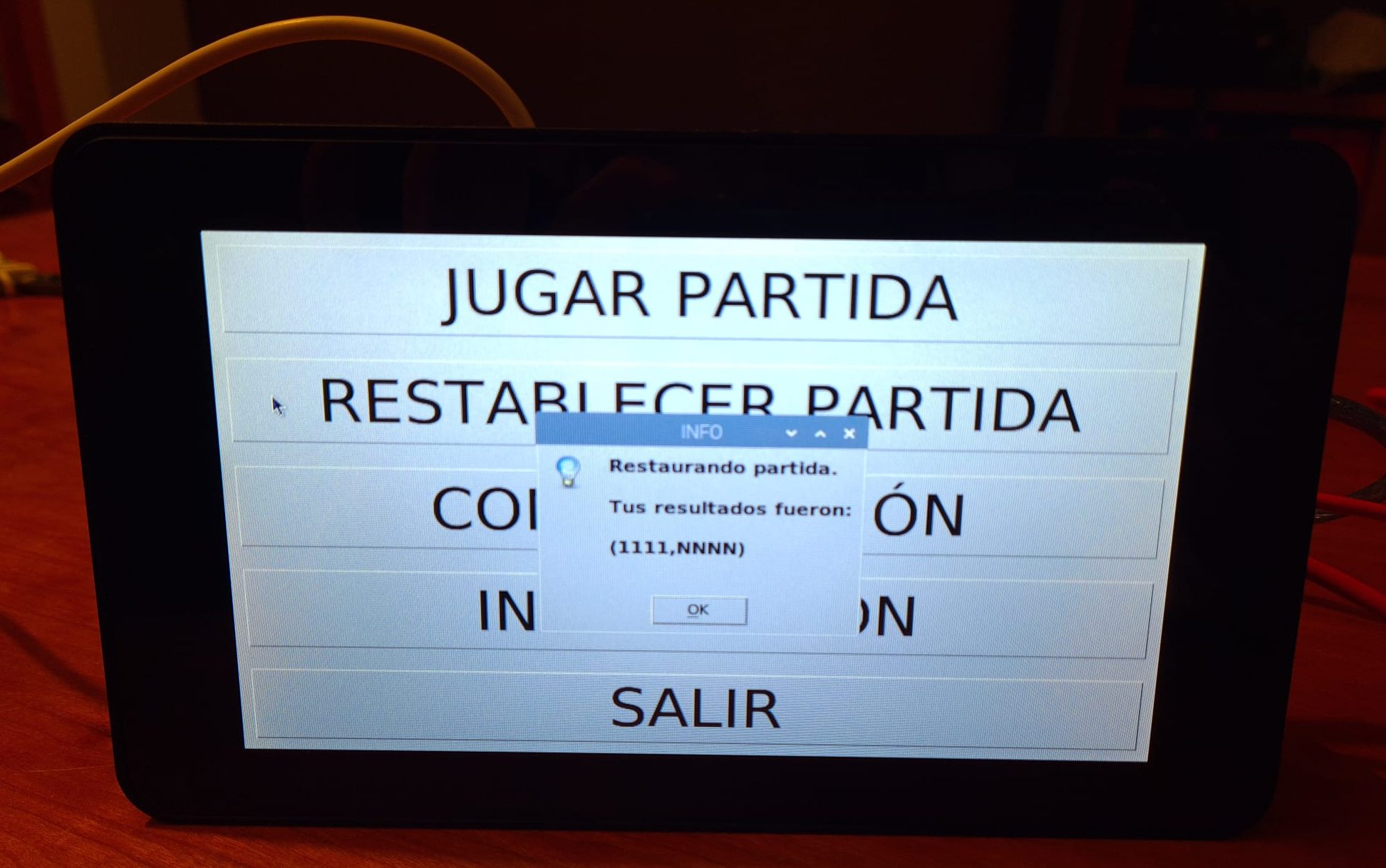
Aunque llegados a este punto el usuario “codificador” ya no tiene más menús por los que moverse, aún quedaría por explicar la última opción del menú principal, que es la del botón “RESTABLECER PARTIDA”, una opción que solamente está disponible en una partida en la que el “decodificador” ha pasado de la primera ronda y se produce la pérdida de la conexión entre los dos dispositivos, por ejemplo, soltando un extremo del cable usb A–B, en cuyo caso veríamos lo siguiente.



Cuando dicho error se manifiesta, se produce un volcado de los datos transmitidos cada ronda desde la Elegoo a la Raspberry Pi, a un archivo de tipo JSON en el cual se almacena toda la información necesaria para volver a restablecer la partida cuando se recupere la comunicación. Después de este aviso, la aplicación nos llevaría al menú principal, donde ya estaría habilitada la opción para la restauración de la partida.

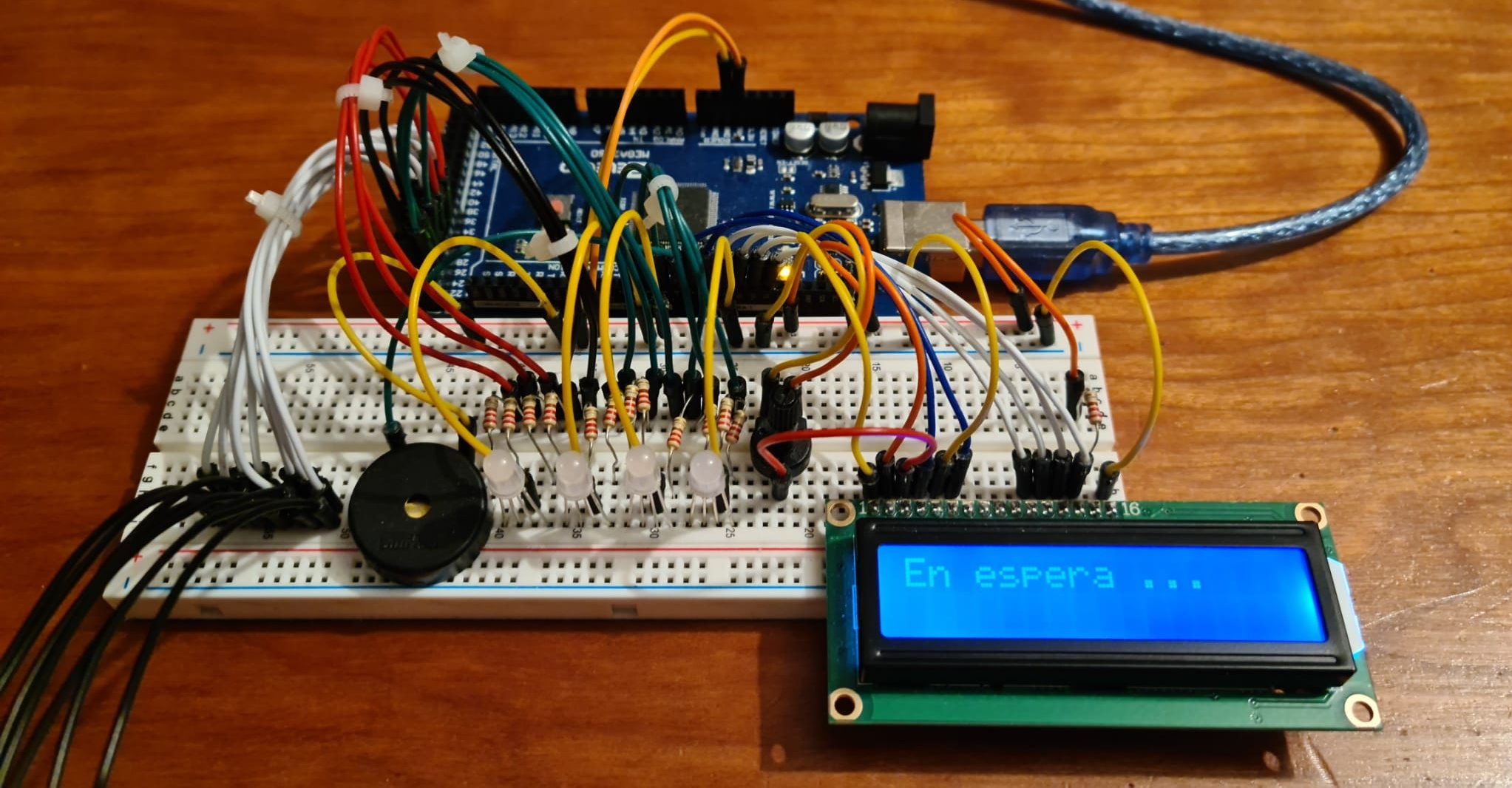


Una vez se pulsa en el botón de esta opción, se muestra un cuadro de diálogo que el jugador “codificador” deberá de leer al “decodificador”, pues contiene un resumen de las selecciones que realizó cada ronda y como de bien le fue con cada selección, esto último se indica con las letras: C, O, N y M. Las cuales nos indican si un dígito está Correctamente ubicado, en Otro lugar, o bien si No está en el código o si solo está Mal ubicado (esto último solo se indica en el modo de juego sin pistas).

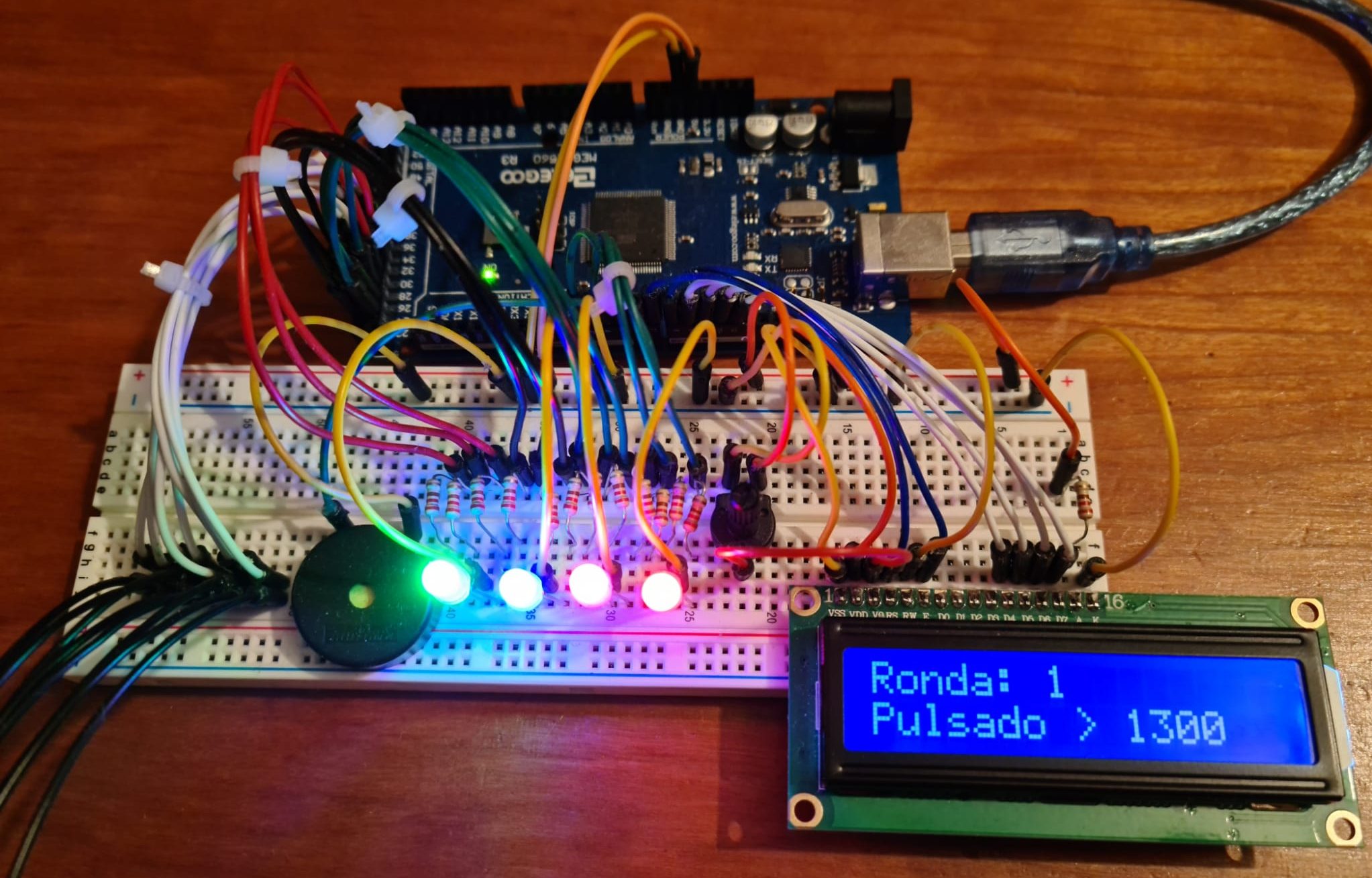


### **9.2 Elegoo**

En la siguiente imagen se muestra el estado inicial en el que se encontraría siempre el microcontrolador Elegoo antes de comenzar una partida.

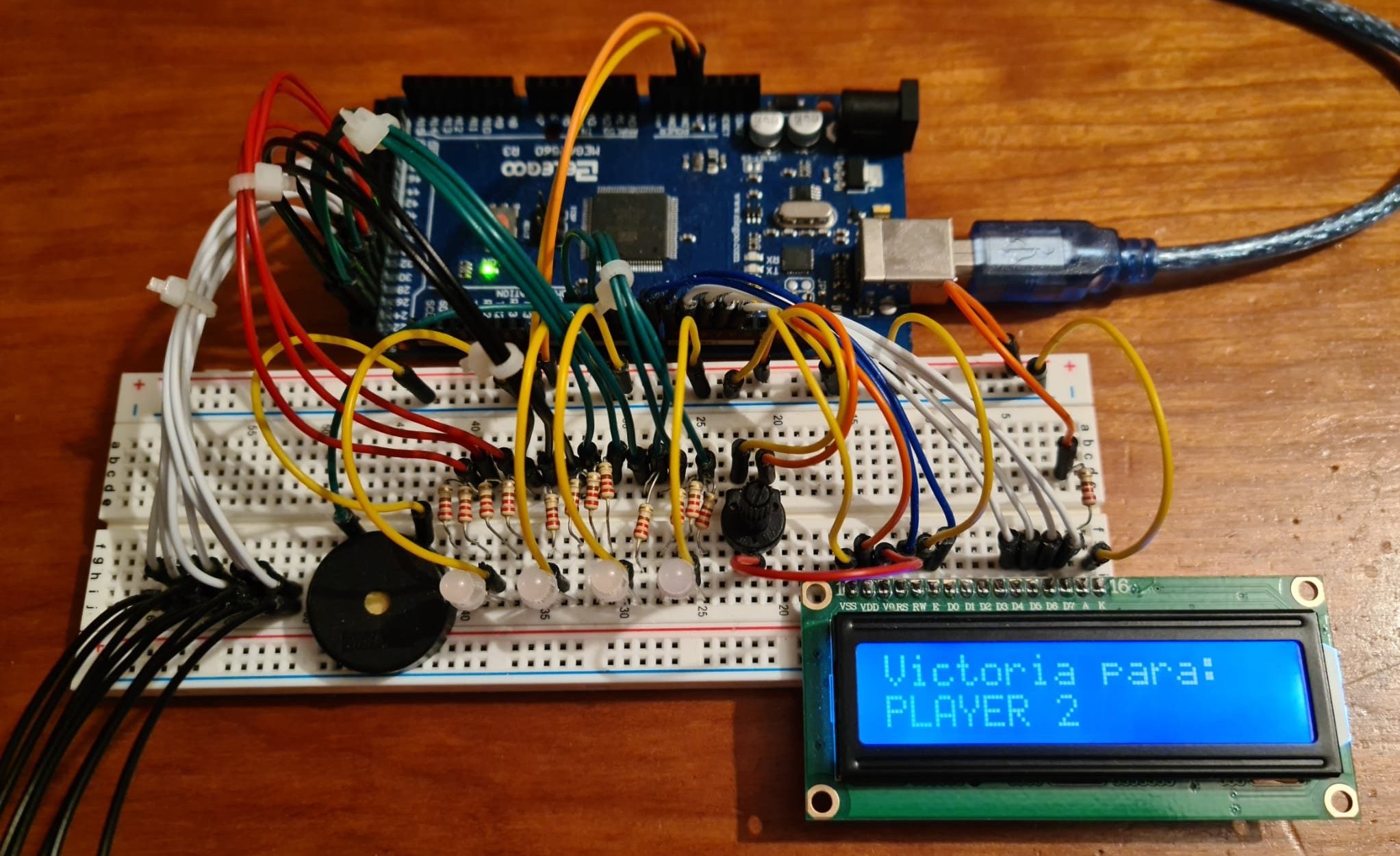


En este momento Elegoo estaría realizando continuamente una lectura activa del buffer de recepción del puerto serie hasta que detecta en él la presencia de datos. Cuando esto ocurre, la configuración de la partida, el código numérico y los nombres de los jugadores han comenzado a transmitirse hacia este sistema, el cual recoge toda esa información y prepara la partida, es entonces cuando el sistema pasa de estar en estado de espera a estar en un estado interactivo. En la imagen mostrada a continuación podemos ver como informa el sistema al interactuar con él al seleccionarse con el teclado matricial los dígitos 1300 (Correcto, Otro, No y No).



Una vez El juego acaba porque el “decodificador” averiguó el código o bien porque agotó sus rondas, el microcontrolador emite la correspondiente melodía de victoria o derrota y, posteriormente, muestra por el LCD display el nombre del jugador ganador.

Después de eso, el juego volvería al estado de espera.



# 10. Conclusiones

Se ha elaborado un sistema empotrado distribuido capaz de comportarse como una implementación electrónica del juego de mesa Mastermind, el cual permite a los usuarios configurar ciertos aspectos del juego (sonidos, dificultad, n.º rondas, tiempo para memorizar y tiempo para responder) así como persistir una partida aun cuando la comunicación entre los dispositivos se pierde (resiliencia).

Fue necesario tomar algunas decisiones que implicaron cambios con respecto a lo que se indicó en la propuesta de este proyecto. Ninguno se debió a la necesidad de más memoria, potencia de cálculo u otros factores similares. Estos motivos fueron los siguientes:

* Ha sido necesario utilizar un microcontrolador Elegoo Mega en lugar de un Arduino o Elegoo simple, esto se debió a la necesidad de disponer de más puertos de entrada/salida digitales para el envío de señales a los componentes que utilizo en la protoboard.
* La vida útil de la memoria EEPROM de Elegoo decrece a medida que se realizan sobre ella operaciones de lectura o escritura, por lo que la comunidad desaconseja su uso para realizar estas operaciones de forma frecuente, es por eso por lo que decidí que la persistencia de la partida la realizara el computador Raspberry Pi, de forma que cada ronda del juego Elegoo proporciona a este ciertos datos (código elegido y resultado obtenido) para restablecer la partida en el futuro en caso de pérdida de la comunicación. Por otra parte, con este enfoque también se ha conseguido que la comunicación entre los sistemas no se reduzca a una comunicación de tipo dispositivo principal a dispositivo secundario, logrando así la bidireccional en lo que respecta a la comunicación.
* Se han descartado los siguientes componentes y periféricos: 4 digit 8-Segment LED dispay y pulsadores. El motivo por el que se descartó el LED display de 4 dígitos de 8 segmentos fue debido a la falta de espacio en la protoboard, por lo que consideré que su utilidad fuera recogida por el LCD display. En cuanto a la renuncia al uso de los pulsadores, me pareció más adecuado que el usuario solo tuviera que manejar el teclado matricial en lugar de tener que utilizar algún botón complementario ubicado en la protoboard.

Finalmente, se ha obtenido un sistema robusto, bien estructurado y amigable de cara a los usuarios, pues este cumple con su finalidad, la de funcionar como un sistema que está destinado al entretenimiento.

# 11. Trabajo futuro

Quedan pendientes de realizar las siguientes tareas:

* Hacer nuevas simulaciones para futuros montajes de los circuitos en la parte de Elegoo y la protoboard. Esto se realizará con algún software específico con el que poder considerar otros esquemas de cableado y distribución de los componentes, lo cual permitirá saber si es posible ampliar el juego en la misma protoboard con otros componentes adicionales o si bien se necesitará una nueva para la ampliación del juego. Dicho software también será utilizado para obtener de forma automática el diagrama de cableado.
* Concebir una forma justa de evaluar las partidas y así poder crear un ranking de jugadores, este se almacenaría en la Raspberry Pi. Habría que pensar pues en un modelo lógico que permita representar la información a almacenar y, en base a la complejidad del esquema resultante, decidir la mejor tecnología para poder almacenar toda la información, ya que esta podría variar desde un simple almacenamiento en un fichero utilizando algún estándar para el intercambio de información, como XML o JSON, hasta una BDD relacional o NoSQL.
* Hacer un uso más eficiente del tiempo del procesador añadiendo nuevos tratamientos de eventos mediante el uso de interrupciones, en concreto, para la gestión del teclado matricial.

# 12. Referencias

* [0] Estado del arte:

https://create.arduino.cc/projecthub/liquidje/mastermind-codebreaker-game-on-uno-r3-93f6cc

<https://devpost.com/software/mastermind-uc2b6e>

* [1] ¿Qué es Raspberry Pi?:

[https://www.raspberrypi.org/documentation/faqs/#introduction](https://www.raspberrypi.org/documentation/faqs/" \l "introduction)

https://www.raspberrypi.org/about/

* [2] Especificaciones de la Raspberry Pi 4 model B:

<https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications>  
https://datasheets.raspberrypi.org/rpi4/raspberry-pi-4-product-brief.pdf

* [3] ¿Qué es Elegoo?:

https://www.elegoo.com/pages/about

* [4] Especificaciones de Elegoo Mega 2560 R3 / Arduino Mega 2560 R3:

<https://store.arduino.cc/arduino-mega-2560-rev3>

* [5] Comuniación Elegoo – Raspberry Pi:

https://www.arduino.cc/reference/en/language/functions/communication/serial/

https://pythonhosted.org/pyserial/

* [6] Componentes electrónicos, periféricos y otros:

Fitzgerald, S., Shiloh, M. (2013). Arduino Projects Book, Arduino , Torino.

<https://playground.arduino.cc/Code/Keypad/>

* [7] Modelado con UML:

Arlow, J., Neustadt, I. (2006). UML 2, Anaya Multimedia, Madrid.

* [8] Coste de los componentes del producto:

<https://www.kubii.es/215-raspberry-pi>

https://www.elegoo.com/collections/arduino-learning-sets