

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE GRADO

SISTEMA INTERACTIVO DE ESTIMULACIÓN COGNITIVA

Autora: Sara García Fernández

Titulación: Grado en Ingeniería en Tecnologías Industriales

Profesor Tutor: Ricardo Vergaz Benito (Departamento de Tecnología Electrónica)

Fecha: Septiembre 2016



Universidad
Carlos III de Madrid
www.uc3m.es

Título: Sistema Interactivo de Estimulación Cognitiva.

Autora: Sara García Fernández

Profesor Tutor: Ricardo Vergaz Benito

EL TRIBUNAL

Presidente: Ramiro Pareja Pareja

Vocal: Marta Ruiz Llata

Secretario: Francisco José Marcellán Español

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 4 de Octubre de 2016 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

VOCAL

SECRETARIO

PRESIDENTE

Índice

Índice de figuras	6
Índice de tablas	8
Agradecimientos	9
Resumen.....	10
Abstract	11
Capítulo 1. Introducción y objetivos del proyecto	12
1.1 <i>Introducción</i>	12
1.1.1 <i>Desarrollo de la idea</i>	12
1.1.2 <i>Soluciones electrónicas en este ámbito para personas dependientes</i>	14
1.2 <i>Objetivos</i>	17
1.2.1 <i>Colaboración con los centros</i>	17
1.2.2 <i>Centro de Día Peñagrande</i>	17
1.2.3 <i>Centro de Día de Alzheimer de Inicio Temprano Doctor Salgado Alba</i>	18
1.3 <i>Especificaciones del sistema</i>	18
1.3.1 <i>Requerimientos de los Centros colaboradores</i>	18
1.3.2 <i>Requerimientos finales</i>	19
1.4 <i>Etapas de desarrollo del proyecto</i>	19
1.5 <i>Medios utilizados</i>	21
1.6 <i>Descripción de la presente memoria</i>	22
Capítulo 2. Diseño del sistema	24
2.1 <i>Visión general</i>	24
2.1.1 <i>Explicación del juego</i>	24
2.2 <i>Diseño del sistema</i>	25
2.3 <i>Interfaz de uso</i>	28
2.3.1 <i>Pulsadores</i>	28
2.3.3 <i>Pantalla LCD</i>	34
2.3.3 <i>Botón Start/Reset</i>	37
2.3.4 <i>Batería externa</i>	39
Capítulo 3. Implementación y pruebas iniciales	42
3.1 <i>Software del microcontrolador</i>	42
3.1.1 <i>Elección del microcontrolador</i>	42
3.1.2 <i>Funciones</i>	43
3.1.3 <i>Bucle principal</i>	44

3.2 Montaje mecánico del conjunto	47
3.2.1 Impresión de piezas en 3D Witbox	47
3.2.2 Fabricación de la caja	52
3.3 Implementación electrónica	58
3.3.1 Placa de pruebas	58
3.3.2 Placa de puntos	60
Capítulo 4. Resultados experimentales e implementación final.....	62
4.1 Pruebas desarrolladas	62
4.2 Pruebas en los Centros	63
4.3 Presupuesto	66
4.3.1 Parte Mecánica	66
4.3.2 Parte Electrónica	68
4.3.3 Total	69
Capítulo 5. Conclusiones y líneas futuras.....	70
5.1 Conclusiones	70
5.2 Líneas futuras	70
Bibliografía	72
Créditos Fotográficos	73
Índice de acrónimos	73
Anexos	74
A.1 Manual de usuario	74
A.2 Encuesta de satisfacción	84
A.3 Esquemático del circuito	85
A.4 Programa del microcontrolador.....	86
A.5 Hojas de características (Datasheets).....	98

Índice de figuras

Figura 1. Diseño para Todos.....	13
Figura 2. Comparativa Diseño Medio/Diseño Universal	14
Figura 3. Áreas del CEAPAT	15
Figura 4. Similitud teclado convencional y teclado RehaCom	16
Figura 5. Usuario de RehaCom.....	16
Figura 6. Diagrama de Gantt del proyecto	21
Figura 7. Elementos del sistema.....	24
Figura 8. Juego con pulsador iluminado.....	25
Figura 9. Diagrama de bloques.....	26
Figura 10. Juego final.....	26
Figura 11. Vista lateral: display y botón	27
Figura 12. Pulsador descartado forma de seta	28
Figura 13. Pulsador en reposo.....	29
Figura 14. Pulsador presionado.....	29
Figura 15. Conexión pulsador eléctrico.....	30
Figura 16. Esquemático del 74HC595.....	31
Figura 17. Descripción de pines del 74HC595	32
Figura 18. Esquema de conexión LED.....	32
Figura 19. Conexión registro de desplazamiento y LEDs	33
Figura 20. Extracto datasheet LCD	34
Figura 21. Cableado LCD con Arduino	37
Figura 22. Conexión botón Start/Reset	39
Figura 23. Características batería externa	39
Figura 24. Detalle batería externa.....	40
Figura 25. Esquema de conexión final.....	41
Figura 26. Descripción pines del ATmega328P	42
Figura 27. Diagrama de flujo función principal loop	46
Figura 28. Impresora 3D Witbox	47
Figura 29. Diseño pulsador SketchUp	47
Figura 30. Diseño tapa SketchUp	48
Figura 31. Detalle impresión 3D.....	49
Figura 32. Piezas descartadas	49
Figura 33. Pulsador final con plástico sobrante	50

Figura 34. Pulsador final limado.....	50
Figura 35. Pulsadores finales.....	51
Figura 36. Comparativa tapas impresas.....	51
Figura 37. Diseño caja inicial SketchUp.....	52
Figura 38. Detalle del mecanismo de los pulsadores.....	52
Figura 39. Detalle de los conductos en los pulsadores.....	53
Figura 40. Diseño caja final SketchUp.....	54
Figura 41. Vista de la tabla de apoyo.....	54
Figura 42. Vista frontal caja contrachapado.....	55
Figura 43. Vista posterior caja contrachapado.....	55
Figura 44. Vista interior rejilla de ventilación.....	56
Figura 45. Detalle pulsador con clavos de sujeción.....	56
Figura 46. Detalle de patillas regulables.....	57
Figura 47. Pintura Acualux e imprimación Gesso.....	57
Figura 48. Vista superior de la caja terminada.....	58
Figura 49. Placa de pruebas inicial.....	58
Figura 50. Prueba 74HC595 con LEDs apagados.....	59
Figura 51. Prueba 74HC595 con LEDs iluminados.....	59
Figura 52. Placa de puntos inicial.....	60
Figura 53. Soldadura placa de puntos.....	60
Figura 54. Placa de puntos final.....	61
Figura 55. Descripción elementos placa de puntos.....	62
Figura 56. Interior de la caja con tabla fijada.....	62
Figura 57. Pruebas Centro Dr. Salgado Alba.....	63
Figura 58. Pruebas con usuarios Centro Dr. Salgado Alba.....	64
Figura 59. Prueba usuaria Centro de Día Peñagrande.....	65
Figura 60. Programación sonido.....	71

Índice de tablas

Tabla 1. Requerimientos del sistema final	19
Tabla 2. Pines de Arduino para los pulsadores	30
Tabla 3. Pines de Arduino para registro de desplazamiento de los LEDs	34
Tabla 4. Conexión inicial LCD con Arduino	35
Tabla 5. Conexión final LCD con Arduino	36
Tabla 6. Pines de Arduino para registro de desplazamiento del display	37
Tabla 7. Descripción funciones del programa	44
Tabla 8. Resultados encuestas	66
Tabla 9. Materiales parte mecánica	67
Tabla 10. Materiales parte electrónica	68
Tabla 11. Desglose coste del proyecto	69

Agradecimientos

En primer lugar, me gustaría agradecer a mi tutor Ricardo Vergaz. Puedo asegurar que sin él, este proyecto no se habría llevado a cabo. Gracias por confiar en mí, por creer en mi trabajo, por motivarme y por enseñarme que ninguna idea es pequeña. Ha sido un placer trabajar contigo.

Gracias al Ayuntamiento de Madrid por permitirme trabajar con los dos centros asignados. Ojalá este sea el inicio de muchas más colaboraciones con futuros alumnos.

A Ana Cruz, de Asispa, por su amabilidad y su cercanía. Por todas las ideas que sacamos en aquella reunión y que me sirvieron para comenzar esta aventura.

A Mónica Sánchez, por tu paciencia y tu ilusión. Por dedicar tu tiempo a contestar mis preguntas y atender mis peticiones.

A Ainhoa Hernández, por hacerme sentir como en casa cada vez que fui a tu centro. Gracias por esa charla final donde me aportaste tanto.

Gracias a ambas por valorar mi trabajo y por haberos involucrado tanto en este proyecto. Por permitirme presenciar las pruebas en vuestros centros y ver que este proyecto realmente es de utilidad para todas esas personas. Esos momentos hicieron que todo el trabajo de estos meses valiera la pena, y los guardaré conmigo siempre.

Gracias a Patricia Vega y a Gema Serrano, terapeutas de los centros, por vuestras opiniones y sobre todo por el tiempo que me dedicasteis en las pruebas y reuniones previas.

Con este proyecto he podido conocer a todas estas grandes profesionales y comprobar el valor de la labor que hacen. Por eso, desde aquí, todo mi reconocimiento para vosotras.

A Cipriano Cobo, por dejarme ocupar tu garaje durante un mes entero, por haberte involucrado tanto en la construcción de la caja y por ayudarme a hacer que quedase mejor de lo que hubiera imaginado.

Por último, por supuesto, gracias a mi familia. Por haber celebrado mis logros como si fueran vuestros y por haberme respetado mientras aprendía cuando las cosas no salían bien. Por haber estado conmigo a lo largo de todo el camino. ¡Al final hemos llegado!

A todos los que habéis puesto un poquito de vosotros en este proyecto del que me siento tan orgullosa,

¡Muchas gracias!

Resumen

El presente Trabajo de Fin de Grado que a continuación se expone consiste en un Sistema Interactivo de Estimulación Cognitiva, tanto para personas con déficit cognitivo como para usuarios de edad avanzada.

El principal objetivo del proyecto es proporcionar un sistema lúdico donde también se trabajen las habilidades cognitivas de los usuarios, siempre bajo las directrices de los Centros colaboradores adscritos al Ayuntamiento de Madrid.

El sistema está formado por una caja con 8 pulsadores donde se iluminarán 8 luces LED, un display LCD y un botón de Start/Reset. A lo largo de 5 niveles de dificultad, el usuario deberá recordar las diferentes secuencias de encendido de LEDs correctamente y repetirlas presionando los pulsadores.

El sistema iluminará primero un led aleatoriamente y, a continuación, el jugador debe presionar el pulsador correspondiente al led iluminado. Conforme el jugador avanza de nivel se irá aumentando progresivamente el número de leds que se encienden en la secuencia. Cada nivel constará de 3 rondas con secuencias diferentes. Si se produce un fallo, se volverá a la primera ronda, siempre dentro del nivel en el que se encuentre el jugador.

La duración de los tiempos de cada nivel, así como el tiempo total del juego, se mostrarán en un display LCD situado en el lateral derecho. Además, el sistema consta de un botón de Juego nuevo o Reset.

El sistema está gobernado por el microcontrolador ATmega328 de Arduino, la plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles. El microcontrolador contiene el código del juego previamente programado en lenguaje de Arduino y posteriormente cargado desde un PC.

El montaje del conjunto se ha realizado en una caja elaborada con tablas de contrachapado de 10mm de grosor. Los pulsadores se han diseñado por ordenador y fabricado en una impresora 3D. La parte electrónica se ha sintetizado en una placa de puntos con todos sus componentes soldados y conectados al microcontrolador y se encuentra en la parte inferior de la caja.

El sistema fue supervisado por los centros colaboradores durante su fabricación y posteriormente puesto a prueba en un entorno real con excelentes resultados. Se han seguido estrictamente los principios del Diseño para Todos y el sistema se ha diseñado y fabricado cuidando especialmente sus características de cara al perfil de usuario final al que se destina.

Abstract

The current Project here exposed consists on an Interactive System for Cognitive Stimulation. It is designed for elderly people with no cognitive impairment at all or in early stages, as well as for people with some types of dementia or with a higher degree of cognitive impairment.

The main purpose of this Project is to provide the final users with an interactive system which they can work with and, eventually, improve their cognitive skills, always under supervision of the professionals from both Day Centres of Madrid's City Council.

The system consists on a plywood box with 8 press buttons on it, 8 white LEDs that will light individually, an LCD display and a Start/Reset button. Throughout 5 levels of different degrees of difficulty, the final user will have to remember the sequences of the LEDs lighting and will be asked to repeat them by pressing the buttons.

In the first level – first round, the system will lighten a LEDs first, and right after the player shall press the corresponding button from where the light was emitted. As the player reaches for higher levels, the number of LEDs on each sequence will be increased. Each level will have 3 rounds with 3 different sequences in each one. If there is an error, the player will go back to the first round, always remaining on the same current level.

The game time on each level, as well as the total game time, will be showed on the LCD display located at the right side of the box. Moreover, the system will have a Start/Reset button right under the display.

The system will be ruled by the ATmega328 microcontroller from the Arduino board, the open-source electronics platform based on easy-to-use hardware and software. The microcontroller will contain the game's code previously programmed on Arduino language and later uploaded from a PC.

The assembly of the whole system has been set on a box made of 10 mm thickness plywood panels. The pressing buttons have been designed using a computer software and printed on a Witbox 3D printer. The electronic part has been put together on a stripboard with all their components soldered, connected to the microcontroller and it is located on the lower part of the box.

The system was supervised during the whole process by both Centres and, later on, it was successfully tested on final users. The principles of the Universal Design have been strictly followed and the system has been carefully designed and produced always keeping in mind the personal features and skills of the final users.

Capítulo 1. Introducción y objetivos del proyecto

1.1 Introducción

En este capítulo se analizará la idea en la que se basa ese proyecto, el tipo de soluciones que se emplean en la actualidad y se explicará en qué consiste la colaboración con los Centros asignados por el Ayuntamiento de Madrid.

1.1.1 Desarrollo de la idea

Este Trabajo de Fin de Grado tiene como objetivo proporcionar a los Centros colaboradores un sistema electrónico que permita a sus usuarios actividades lúdicas de entretenimiento a la vez que se trabaja la estimulación cognitiva de los mismos.

El deterioro cognitivo es la pérdida de funciones cognitivas, específicamente en memoria, atención y velocidad de procesamiento de la información.

Como consecuencia al desarrollo social y al incremento de la calidad de vida en las sociedades occidentales, la longevidad en las últimas décadas es mayor que en años anteriores. Este incremento ha dado lugar a la aparición de una gran masa de población envejecida, con los problemas de deterioro físico, psíquico y/o sensorial que ello conlleva.

En este contexto surge con fuerza el término “persona dependiente”, utilizado con frecuencia en cuanto a la problemática social y económica que supone en la actualidad, pero sobre todo en un futuro próximo. Es por ello que tan importante es asistir a la persona que ha adquirido el estatus de dependiente como prevenir, o retrasar, la aparición del mismo.

Científicos aseguran que tareas como razonamiento lógico y actividades donde se trabajen la memoria objetiva, la atención, la velocidad de procesamiento de la información y la función ejecutiva, pueden tener considerables beneficios en el día a día de las personas.

El trabajo de investigación, del que se hablará con más detalle en el apartado [1.1.2 Soluciones electrónicas](#), presentado en la Conferencia Internacional de la Asociación del Alzheimer celebrada en Julio de 2016 en Toronto, es el más concluyente acerca de que estas técnicas funcionan.

El Dr. Doug Brown, director de investigación y Desarrollo de la Asociación del Alzheimer, afirmó: “Existe un creciente interés sobre el potencial que tienen las actividades de entrenamiento del cerebro contra el deterioro cognitivo [...]. Esta investigación proporciona la primera evidencia que el entrenamiento cerebral basado en sistemas informáticos mejora la velocidad del cerebro para procesar la información, lo cual puede reducir la tendencia a desarrollar demencia en una década.” (1)

El desarrollo tecnológico en la actualidad que, por un lado, nos facilita y simplifica la realización de muchas tareas, por otro nos obliga a adquirir nuevas habilidades que nos permitan utilizarlo correctamente y poder aprovechar lo que nos ofrece. Por ello es de gran importancia tanto que el diseño de la tecnología cumpla los principios de “*Diseño para todos*”, como la disposición al entrenamiento y al aprendizaje de su uso para que ésta pueda ser de utilidad.

Diseño para Todos

Los 7 Principios del Diseño Universal o Diseño para Todos (Fig. 1), son principios generales de diseño, aplicables en ámbitos científicos como arquitectura, ingeniería o páginas y aplicaciones Web, entre otros campos de aplicación. A continuación, se describen brevemente los principios del Centro para el Diseño Universal (2):

1^{er} Principio: *Uso equiparable.* El diseño es útil a personas con diversas capacidades.

2^o Principio: *Uso flexible.* El diseño se acomoda a un amplio rango de preferencias y habilidades individuales.

3^{er} Principio: *Simple e intuitivo.* El uso del diseño es fácil de entender, atendiendo a la experiencia, conocimientos, habilidades lingüísticas o grado de concentración actual del usuario, eliminando complejidades innecesarias y siendo consistente con las expectativas y con la intuición del usuario.

4^o Principio: *Información perceptible.* El diseño comunica eficazmente la información necesaria para el usuario, atendiendo a las condiciones ambientales o a las capacidades sensoriales del usuario.

5^o Principio: *Con tolerancia al error.* El diseño minimiza los riesgos y las consecuencias adversas de acciones involuntarias o accidentales.

6^o Principio: *Que exija poco esfuerzo físico.* El diseño puede ser usado eficaz y cómodamente con un mínimo de fatiga.

7^o Principio: *Tamaño y espacio para el acceso y uso.* Que proporcione un tamaño y espacio apropiados para el acceso, alcance, manipulación y uso, atendiendo al tamaño del cuerpo, la postura o la movilidad del usuario.



Figura 1. Diseño para Todos

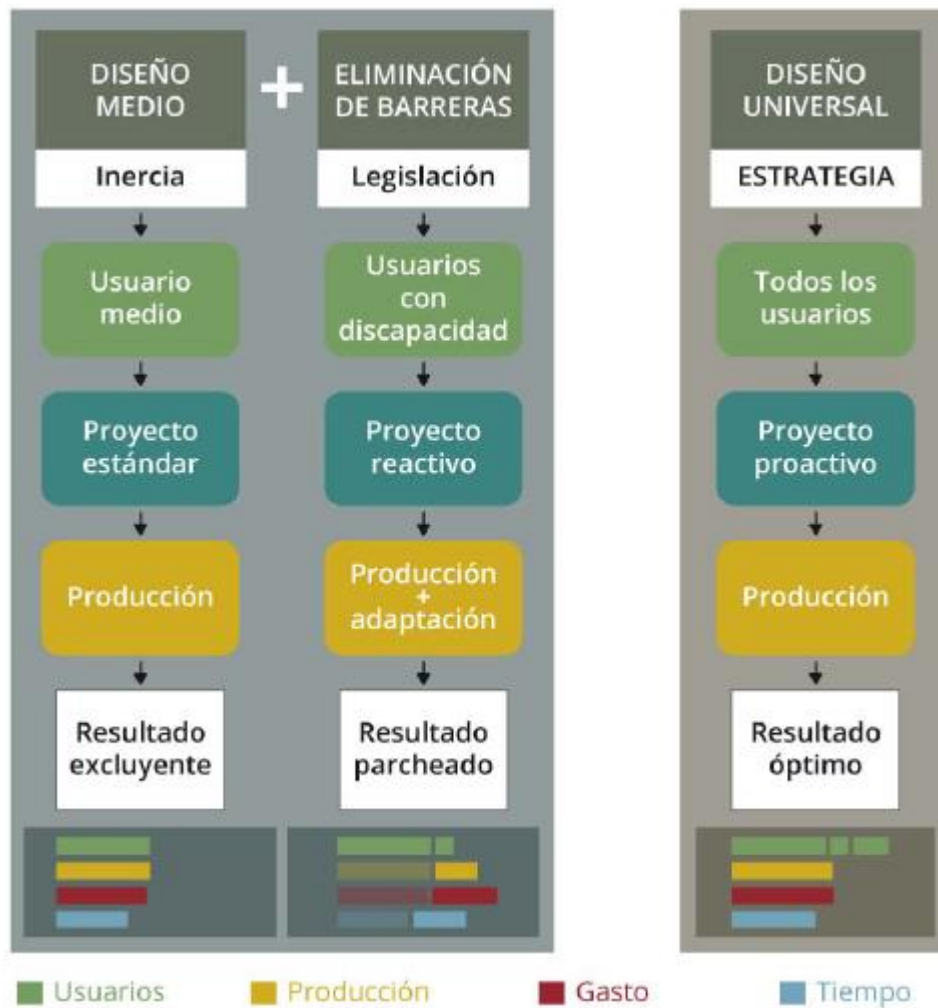


Figura 2. Comparativa Diseño Medio/Diseño Universal

Así, durante el desarrollo de este trabajo se ha tenido en cuenta desarrollar una ayuda técnica de carácter tecnológico para compensar funciones aminoradas por el proceso de envejecimiento o deficiencia, además del cumplimiento del principio de “Diseño para todos” al no introducir barreras para su uso, lo que permitirá alargar la vida activa de los usuarios.

En la Figura 2 se ilustran las ventajas de realizar proyectos basados en el Diseño Universal frente a un proyecto diseñado sin tener en cuenta la eliminación de barreras y que más tarde debe ser modificado, con los inconvenientes que ello supone en tiempo, gastos, etc.

1.1.2 Soluciones electrónicas en este ámbito para personas dependientes

El concepto de accesibilidad digital no existía hace 25 años. La revolución de las tecnologías y su rápida penetración en todos los ámbitos de la vida están suponiendo cambios en la manera de vivir, de relacionarse, de formarse, de trabajar y de participar en la sociedad.

Uno de los ejemplos nacionales de preocupación en este nuevo ámbito es el CEAPAT (Centro Estatal de Autonomía Personal y Ayudas Técnicas), creado en 1989, tiene como misión contribuir a hacer efectivos los derechos de las personas con discapacidad y personas mayores, a través de la accesibilidad integral, los productos y tecnologías de apoyo y el diseño pensado

para todas las personas, sirviendo de referencia y excelencia, con la colaboración e implicación de todos los agentes. En la Figura 3 se muestran sus áreas de actuación.

El centro ha desarrollado herramientas de referencia como el Test de Accesibilidad a la Web (TAW) y participado en la elaboración de normas legales y técnicas sobre accesibilidad web, formación a distancia accesible, subtítulo, audiodescripción, domótica, teleasistencia móvil, ciudades inteligentes accesibles y las plataformas en la nube (3).



Figura 3. Áreas del CEAPAT

Los trabajos de Feuerstein² (1970) sobre experiencias de aprendizaje mediado han demostrado que personas con déficit intelectual, independientemente de que sean consecuencia de una situación sobrevenida o no, son capaces de adquirir destrezas en tareas que se podrían considerar complejas. Así, Regel, H & Fritsch, A. , trabajando con el sistema RehaCom³ (Fig. 4 y Fig. 5) encontraron mejoras significativas o de inhibición del proceso degenerativo en habilidades cognitivas como memoria, atención, secuenciación, asociación, etc, que se traducen en mejoras en la autonomía en las actividades de la vida diaria.

² Clínico rumano cuyo trabajo teórico se basó mayoritariamente en el desarrollo psicológico cognitivo.

³ RehaCom consiste en más de 20 terapias diferentes llevadas a cabo a través de un sistema electrónico que ayudan a mejorar funciones cognitivas en los usuarios.

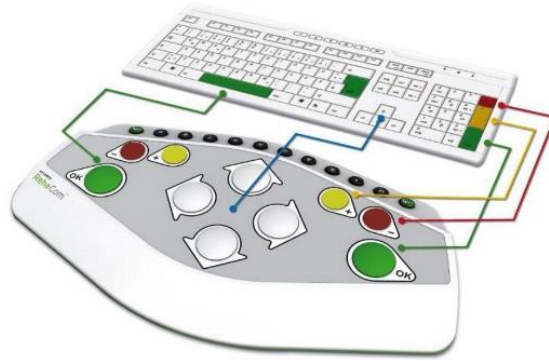


Figura 4. Similitud teclado convencional y teclado RehaCom



Figura 5. Usuario de RehaCom

Recientemente, en la Conferencia Internacional de la Asociación del Alzheimer (Toronto, Julio 2016), se presentó un estudio basado en un juego llamado “tarea de velocidad de procesamiento”. Fue uno de los 3 tipos de entrenamiento cognitivo que 2800 personas realizaron durante el estudio ACTIVE (Advanced Cognitive Training for Independent and Vital Elderly).

Los 2800 participantes, de una media de edad de 74 años cuando comenzaron el ensayo, fueron analizados durante 10 años por los científicos encargados del estudio, con la esperanza de conocer cómo el entrenamiento cognitivo podría afectar al funcionamiento de la salud de las personas de avanzada edad.

Dichos participantes fueron separados en 4 grupos. Un grupo realizó juegos de velocidad de procesamiento y otros dos grupos tomaron clases de memoria o razonamiento. El último grupo no realizó ninguna actividad y sirvió como referencia.

El estudio concluyó con que la utilización de este tipo de juegos electrónicos reduciría a la mitad la probabilidad de desarrollar demencia en usuarios de los 3 primeros grupos, comparado con un usuario del cuarto grupo sin entrenamiento de este ámbito.

1.2 Objetivos

El objetivo principal del Proyecto es **diseñar, implementar y probar un juego específicamente adaptado a personas mayores y usable tanto con deterioro cognitivo como sin él.**

Actualmente, el porcentaje de personas mayores de 65 años en España es del 18.4% (8.573.985) sobre el total de la población (46.624.382)⁴.

Inicialmente, la idea de este proyecto estaba centrada a personas pertenecientes a este sector sin deterioro cognitivo o con deterioro cognitivo leve, pero se amplió a personas con algún tipo de demencia leve o deterioro cognitivo leve-moderado debido a la asignación para la realización de este Proyecto del Centro de Día Dr. Salgado Alba, como se explicará posteriormente.

El mayor fin de este proyecto era probar su utilidad en usuarios y comprobar que cumplía los objetivos de entretenimiento y estimulación cognitiva. Para ello, se solicitó mediante un documento en la Junta Municipal del Distrito de Fuencarral-El Pardo, la colaboración con Centros de Día o Residencias de la Comunidad de Madrid. En él se propone por parte de la autora de esta memoria el diseño y fabricación de un juego electrónico, basado en el tradicional Simón, que estimule las funciones cognitivas mediante una serie de secuencias aleatorias de iluminación de botones, y que guarde datos relevantes sobre los tiempos de respuesta de los usuarios. El documento solicitaba además que dichos centros colaboraran en la realización del Diseño y las pruebas con usuarios, en base a los principios de Diseño para Todos.

1.2.1 Colaboración con los centros

Tras recibir respuesta, se concertó una reunión con una persona responsable del Centro de Servicios Sociales, el 19 de abril de 2016. En dicha reunión se proporcionaron los contactos con los dos centros asignados para realizar la colaboración, el Centro de Día Municipal Peñagrande, y el Centro de Día de Alzheimer de Inicio Temprano Doctor Salgado Alba.

A partir de entonces, el 25 de abril tiene lugar una reunión a la que asiste Ana Cruz, responsable de los Centros de Día de Asispa⁵; Ainhoa Hernández, directora del Centro de Día Peñagrande; y Mónica Sánchez, directora del Centro de Día Doctor Salgado Alba. Se acuerdan varias reuniones para ir informando de los avances realizados y también una reunión de prueba con usuarios.

1.2.2 Centro de Día Peñagrande

En el Centro de Día Municipal Peñagrande se atienden a los mayores del distrito de Fuencarral – El Pardo. El perfil mínimo son personas mayores de 65 años que no presentan deterioro cognitivo, o con deterioro cognitivo leve, al alta.

⁴ Según datos del Padrón Continuo (INE), a 1 de Enero de 2015.

⁵ Asispa es una asociación privada sin ánimo de lucro que tiene por misión la realización de actividades conducentes a promover tanto la autonomía personal como la mejor atención a las personas (mayores, adultos, juventud, infancia, etc.), siendo una Entidad de Economía Social.

Para conocer la existencia de deterioro se realiza una valoración cognitiva, donde entre otros test, destaca el MEC⁶, siendo necesario tener una puntuación mayor a 20, sobre un máximo de 35 puntos.

En el centro se ofrece una atención individual integral, dando mayor atención a aquellos aspectos donde el usuario/a presenta una menor autonomía. Entre las atenciones destacan la terapia ocupacional, basada en mantener o mejorar la capacidad cognitiva, y es aquí donde se encuentra el objetivo del proyecto.

1.2.3 Centro de Día de Alzheimer de Inicio Temprano Doctor Salgado Alba

El Centro de Día de Alzheimer de inicio temprano Dr. Salgado Alba, es un recurso de atención diurna, que presta servicio a personas menores de 60 años que sufren algún tipo de demencia o deterioro cognitivo degenerativo en fase leve o leve-moderado.

La demencia es un síndrome orgánico cerebral que se caracteriza por la pérdida progresiva de capacidades mentales superiores (cognitivas y funcionales): lenguaje, memoria, coordinación, atención, concentración, praxias, etc.

Las terapias no farmacológicas empleadas en los tratamientos pasan por la estimulación y entrenamiento de todas estas capacidades. Las actividades que se llevan a cabo deben ser atractivas y motivadoras para consolidar una buena adhesión al tratamiento. Es por ello que se precisan de herramientas novedosas que faciliten la estimulación y entrenamiento de dichas capacidades (4).

1.3 Especificaciones del sistema

A continuación se detallan las especificaciones del sistema acordadas con los Centros y siguiendo las bases del Diseño para Todos.

1.3.1 Requerimientos de los Centros colaboradores

En la reunión celebrada con las directoras de ambos centros, se resolvieron los siguientes puntos en referencia al sistema:

- 1)** Un máximo de 8 pulsadores cuadrados, organizados en 2 filas.
- 2)** Una pantalla o display que mostrase los tiempos por nivel y el tiempo total del juego.
- 3)** Botón o pulsador de inicio de nuevo juego.
- 4)** Una inclinación (regulable o no) para mayor comodidad de los usuarios al utilizar el juego en una mesa.
- 5)** Luces LED de color blanco que iluminasen con suficiente intensidad para poder ser percibida sin problemas por los usuarios.

⁶ Mini Examen Cognitivo. Valores normales 30-35 puntos. Sugiere deterioro cognitivo: >65 años: <24 puntos.

- 6) Pulsadores numerados del 1 al 8, comenzando por la fila de arriba a la izquierda.
- 7) Mínimo cableado posible, evitando cables sueltos con los que se pudiese tropezar.
- 8) Velocidad de juego estándar ajustada a las capacidades de los usuarios.

Además, en una posterior reunión con Patricia Vega, terapeuta del Centro de Día Peñagrande, se acordó un tamaño adecuado para los pulsadores de 55mm de lado y una separación entre ellos de 30mm, suficiente para evitar que se presionasen varios a la vez en condiciones de movilidad reducida, decisiones que fueron corroboradas posteriormente por Gema Serrano, terapeuta del Centro de Día Dr. Salgado Alba.

1.3.2 Requerimientos finales

Una vez acordados los detalles de lo que será el sistema, se adjunta la Tabla 1, en la cual se basó el diseño final:

Dimensiones	<i>39x13x20 cm</i>
Inclinación	<i>Sí, regulable a ser posible</i>
Iluminación	<i>Luz blanca</i>
Niveles de juego	<i>5</i>
Rondas de juego	<i>3 (excepto nivel 5)</i>
Numeración	<i>Del 1 al 8</i>
Número de filas	<i>2</i>
Número de columnas	<i>4</i>
Display	<i>LCD 2X16</i>
Botón Start/Reset	<i>Sí</i>
Alimentación	<i>Batería externa recargable</i>

Tabla 1. Requerimientos del sistema final

1.4 Etapas de desarrollo del proyecto

A continuación, se detallan las etapas en las que tuvo lugar el desarrollo del proyecto y las tareas principales que se dieron en cada una.

Etapas 0: Planificación. Se define el objetivo del proyecto y se realiza una evaluación tecnológica de las opciones disponibles.

- Presentación de la idea y asignación de TFG.
- Solicitud de colaboración y asignación de centros.

En esta etapa se realizó la solicitud de colaboración al Ayuntamiento de Madrid, asignando después los dos centros colaboradores y tuvieron lugar las primeras reuniones con ellos.

Etapas 1: Desarrollo del concepto. Se describe la forma, función y características del sistema.

- Reunión con las responsables de los centros.

- Determinación de las especificaciones del sistema.

Se reciben los requisitos que debe cumplir el sistema, tanto electrónicos como mecánicos. Se especifican los 5 niveles de juego y se realiza una estimación del tamaño global del sistema, así como de detalles tales como el tamaño de los pulsadores, el nivel de inclinación y las dimensiones finales.

Etapa 2: Diseño del sistema. Definición de su arquitectura y división en subsistemas de los componentes.

- Elección y compra de componentes.

En esta etapa se divide el sistema en partes, cada cual con un componente principal. Se eligen dichos componentes para conseguir los más adecuados y con menor coste posible.

Etapa 3: Diseño detallado. Se especifica completamente la geometría, los materiales, etc.

- Programación del juego.
- Impresión de piezas.
- Fabricación de la caja.
- Pruebas sobre protoboard.
- Soldadura de componentes.

Se realiza el diseño mecánico final, se concreta la programación del sistema y se realiza el montaje electrónico sobre la placa de pruebas para posteriormente soldarlo e integrarlo en el conjunto.

Etapa 4: Pruebas y afinación. Pruebas del sistema y eliminación de errores del mismo.

- Pruebas finales sobre el sistema montado.
- Ajustes de programación.

En esta etapa tienen lugar las pruebas sobre el sistema ya montado, se comprueba la iluminación y se ajusta la velocidad de las secuencias.

Etapa 5: Producción de transición. Se valida el proceso de fabricación con los usuarios finales.

- Pruebas en los centros.

Se realizan las pruebas en los centros y se recogen sus impresiones sobre el sistema creado.

A continuación, se muestra en la Figura 6 el Diagrama de Gantt del proyecto donde el eje de abscisas es el eje temporal y el eje de ordenadas representa las tareas. La fecha de inicio es 22/02/2016, y la fecha de fin 19/09/2016.

Como medios propios que se han ido adquiriendo o de los que ya se disponía se encuentran:

- Software: Google SketchUp para diseño de piezas en 3D.
- Kit de soldadura: soldador eléctrico, bomba de succión de soldadura, pasta fundente para soldar, soporte de soldador y estaño en espiral.
- Sierra de corte y segueta.
- Lima, taladradora y brocas de diferentes tamaños.
- Pelacables y alicates.
- Pinzas de sujeción.
- Martillo y destornilladores.
- Clavos y tornillos.
- Regla y calibre.
- Pistola de pegamento termofusible.
- Imprimación Gesso para madera y pintura Acualux.

1.6 Descripción de la presente memoria

La presente memoria describe el proceso de creación, desarrollo e implementación de un Sistema Interactivo de Estimulación Cognitiva.

Así, el **Capítulo 1. Introducción**, desarrolla la idea inicial y analiza la situación actual en lo referente a soluciones tecnológicas para las personas a las que este proyecto está dedicado. Se pone a su vez en conocimiento la tarea que realizan los Centros colaboradores, así como informa del proceso que se llevó a cabo para conseguir esta colaboración. También describe las reuniones informativas con las directoras de los centros y los puntos que se resolvieron con respecto a las especificaciones del sistema.

En el **Capítulo 2. Diseño del Sistema**, se muestra una visión general del proyecto. Incluye una explicación detallada del juego en el apartado [2.1.1 Explicación del juego](#). Además, se analizan individualmente los elementos principales que componen este proyecto y las soluciones por las que se optaron para optimizar su diseño frente a las alternativas barajadas.

El **Capítulo 3. Implementación y pruebas iniciales**, se centra en los aspectos físicos de la construcción de la caja que alberga la circuitería interna. Se analiza la elección del microcontrolador y la programación del juego con la descripción de sus funciones principales. En la segunda parte del capítulo se muestran las pruebas realizadas inicialmente sobre una placa protoboard y se prosigue con su desarrollo hasta la implementación en la placa de puntos.

En el **Capítulo 4. Resultados experimentales e implementación final**, se muestran los resultados de las pruebas realizados en los centros colaboradores y de las encuestas de satisfacción posteriores tanto a los usuarios directos como a las directoras y demás personal del centro. Asimismo, se calcula un presupuesto general del proyecto, analizando por separado las partes mecánica y electrónica hasta llegar al coste total.

Finalmente, en el **Capítulo 5. Conclusiones y líneas futuras**, se analizan los resultados finales del sistema y se valora si se han cumplido con los objetivos del proyecto, así como las pautas del Diseño Para Todos. Además, se proporcionan una serie de ideas para proyectos que puedan seguir esta misma línea.

En los **Anexos** se encuentra adjunta la programación del juego, el esquemático de conexiones y un manual de usuario que se entregará a los centros una vez se les ceda el sistema al fin de este proyecto. Además se añade el modelo de encuesta de satisfacción y los links de acceso a las hojas de características de los componentes.

Capítulo 2. Diseño del sistema

2.1 Visión general

Atendiendo siempre las peticiones y sugerencias de los Centros, se adjunta en la siguiente imagen simplificada (Fig. 7) los elementos principales del conjunto, que se irán analizando en profundidad más adelante.

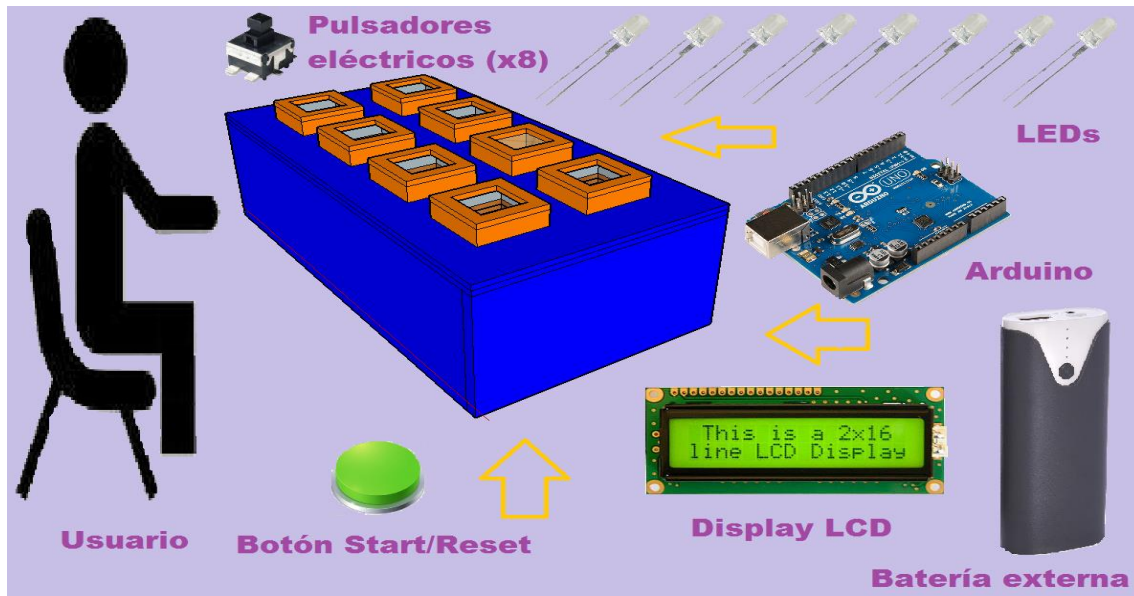


Figura 7. Elementos del sistema

2.1.1 Explicación del juego

El juego consiste en memorizar la secuencia que se ilumina con sus LEDs en cada nivel y repetirla sobre los pulsadores correspondientes. Se compone de 4 niveles con 3 rondas en cada uno y una última ronda donde se deberán presionar sólo los pulsadores pares.

El número de LEDs iluminados en cada secuencia depende del nivel en el que se encuentre el jugador, así en el nivel 1 se encenderá un solo LED por secuencia, en el nivel 2 se encenderán dos por secuencia, y así sucesivamente hasta el nivel 5, donde, atendiendo la sugerencia de los Centros, el jugador debe presionar solamente los pulsadores pares y en orden creciente. Cada nivel del 1 al 4 pasa por 3 rondas.

Cuando el jugador presiona un pulsador, el LED adherido a él se iluminará para mostrar que se ha presionado correctamente, como indica la Figura 8.



Figura 8. Juego con pulsador iluminado

El jugador no debe pulsar hasta que se ha mostrado la secuencia completa, debe memorizarla y seguidamente repetirla mediante los pulsadores. El juego quedará a la espera del turno del jugador, sin límite de tiempo. Si la secuencia es correcta, pasará a la siguiente ronda con otra secuencia aleatoria. Si se completan las 3 rondas de cada nivel correctamente, se iluminarán todos los LEDs en cascada y se pasará al siguiente nivel.

Si se produce un error, el jugador permanecerá en el nivel en el que se encuentre pero volverá a la ronda inicial. Parpadearán todos los leds como señal de que la secuencia del jugador no es correcta.

El botón de START/RESET del lado derecho puede ser pulsado en cualquier momento para empezar un nuevo juego desde la ronda inicial del nivel 1 reiniciando todos los contadores de tiempo, así como una vez finalizado el nivel 5, para comenzar otra partida.

El display LCD situado también en el lateral derecho del jugador, mostrará el nivel en el que se encuentra, así como el tiempo de juego de cada nivel. Además, al finalizar el nivel 5 mostrará el tiempo de juego total y entrará en un modo de espera de 2 minutos donde, si no se ha pulsado el botón de START/RESET, apagará la pantalla para reducir el consumo.

2.2 Diseño del sistema

Una vez conseguida una idea clara del sistema a realizar, se adjunta el diagrama de bloques (Fig. 9) de las partes integrantes de manera que cada una sea claramente diferenciada durante el resto del capítulo.

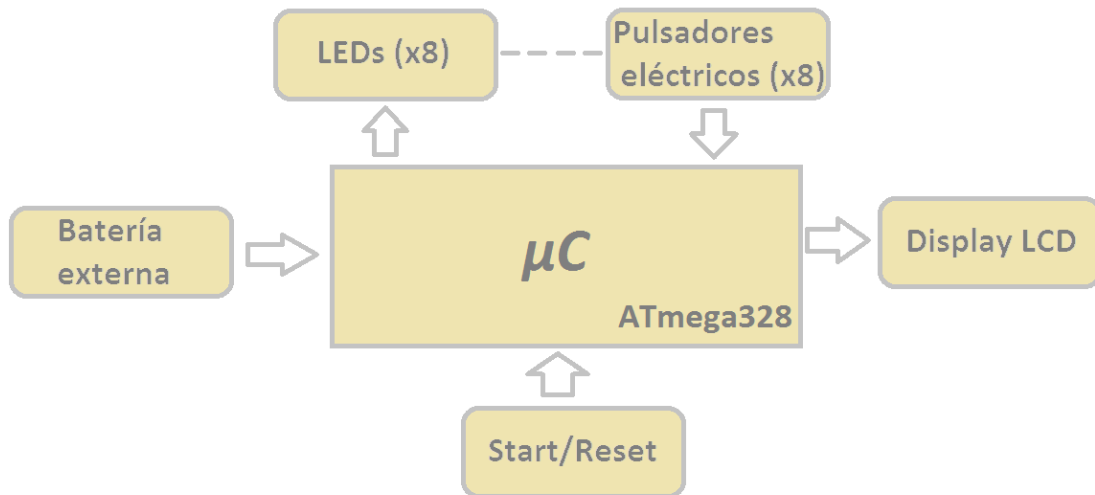


Figura 9. Diagrama de bloques

El diseño del sistema tendrá en cuenta todas las peticiones y sugerencias realizadas por los Centros colaboradores, descritas anteriormente en el apartado [1.3.1](#).



Figura 10. Juego final

El conjunto se realizó en madera de contrachapado, ligera y resistente. Es fácilmente manejable para mayor comodidad del usuario.

Los pulsadores se dispusieron en 2 filas con una separación de 4 cm entre filas y de 3 cm entre columnas para evitar que estuviesen demasiado juntos y se pulsase alguno equivocadamente, pero a su vez situados al alcance de cualquier usuario con movilidad reducida.

Sus laterales y esquinas fueron limados y repasados con una gasa con acetona para desprender los posibles restos de plástico que pudiesen causar daños.



Figura 11. Vista lateral: display y botón

El display LCD (Fig. 11) se colocó en el lateral derecho, con vistas a que no interfiriera en la atención de los jugadores y que la información mostrada sirviese eventualmente para el personal del centro.

Asimismo, el botón verde de START/RESET (Fig. 11) se colocó bajo el display para evitar pulsaciones accidentales, pero a su vez a fin de que fuese fácilmente localizable en caso de necesitar su uso.

En la parte inferior de la caja (Fig.11) se encuentran unas patillas de altura regulable para dar mayor o menor inclinación según así lo prefiera el usuario.

Las luces LED fueron colocadas en el interior de los pulsadores naranjas, bajo un plástico rígido semitransparente con los números del 1 al 8 correspondientes situados en el centro del mismo para así evitar un excesivo brillo que saliese directo hacia el usuario.

Con respecto al mínimo cableado posible, la solución que se escogió fue utilizar una batería externa que alimentara al sistema. Posteriormente se hablará de la elección de dicha batería. Con esto, ya no era necesaria la existencia de cables de por medio, evitando posibles tropiezos de los usuarios y, además, el sistema pasa a ser fácilmente transportable y su uso no está condicionado a la existencia de una toma de corriente.

Por último, en lo referente a la velocidad del juego, se programó inicialmente una velocidad estándar, que posteriormente se ajustó adecuadamente al nivel de los usuarios en las pruebas realizadas en los Centros. Esta cuestión será tratada más adelante en el apartado de [4.2 Pruebas en los centros](#).

2.3 Interfaz de uso

Se entiende por interfaz de uso aquellos elementos directamente relacionados con la acción del juego por parte del usuario o del personal de los centros.

2.3.1 Pulsadores

Los pulsadores son el elemento principal del sistema y, como ya se ha comentado en apartados anteriores, se diseñaron siguiendo los principios del Diseño para Todos. Así, deben ser fáciles de usar, accesibles, intuitivos y deben presentar la información de manera que sea fácil de percibir por el usuario, al tiempo que su pulsación suponga bajo esfuerzo físico. Esto será posible mediante la iluminación, la separación entre ellos, y la ayuda de muelles.

Alternativas de diseño

Inicialmente, una opción barajada fue la de unos pulsadores de tipo seta (Fig. 11). La forma circular de la base era un inconveniente a la hora de crear una amortiguación con los muelles y no aseguraba una pulsación correcta, ya que el pulsador eléctrico quedaría en el centro de la base dejando un espacio más reducido para los muelles que en la alternativa elegida para el diseño final.

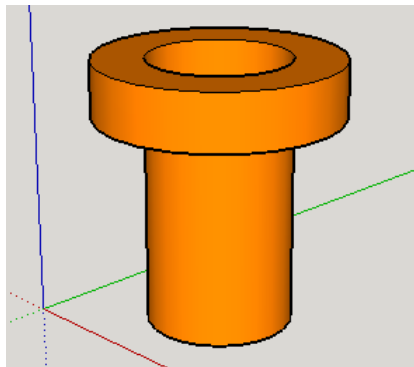


Figura 12. Pulsador descartado forma de seta

Diseño final

Como se ha comentado anteriormente, se optó por un diseño de pulsadores cuadrados de 55 mm de lado y 60 mm de alto incluyendo 4 patas cuadradas de 15 mm de lado donde se apoyarían los muelles.

Los clavos que aparecen en horizontal en las Figuras 13 y 14, además de evitar posibles movimientos en los pulsadores si se voltea la caja, fijan a todos los pulsadores a una misma altura gracias a los muelles, dejando la parte plana del pulsador a una distancia de unos 3 mm del pulsador eléctrico. En las siguientes imágenes (Fig. 12 y Fig. 13) se observa el pulsador en reposo y el pulsador mientras está siendo pulsado.

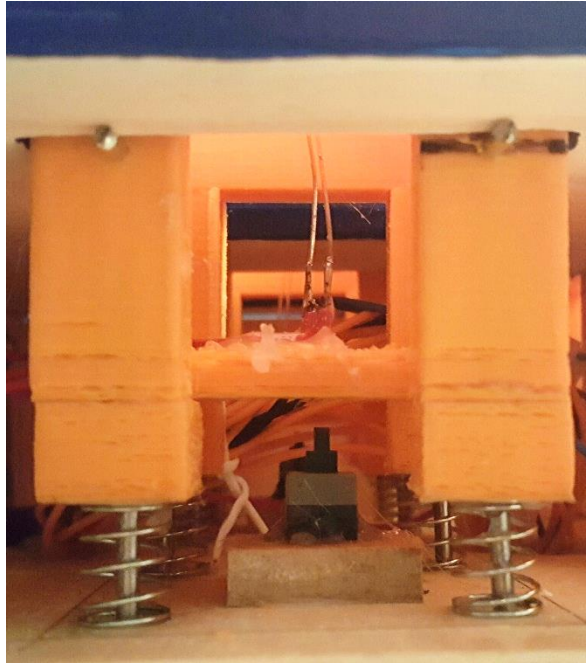


Figura 13. Pulsador en reposo

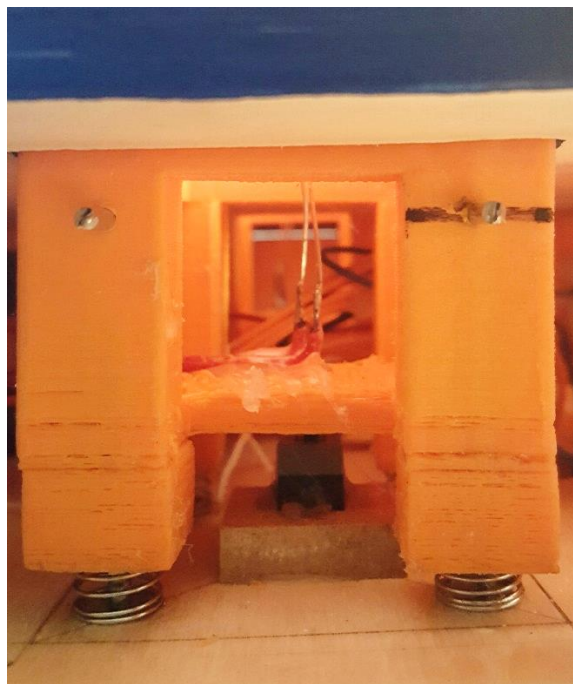


Figura 14. Pulsador presionado

Los pulsadores eléctricos utilizados son de 6 patillas. Se configuraron como entradas al microcontrolador, y se conectaron de la siguiente manera que aparece en la Figura 15⁷:

⁷ Nótese que en la simulación aparece un pulsador de 4 patillas, que funciona exactamente igual que el finalmente utilizado de 6, en este caso escogiendo las patillas centrales y las de la izquierda, o las de la derecha.

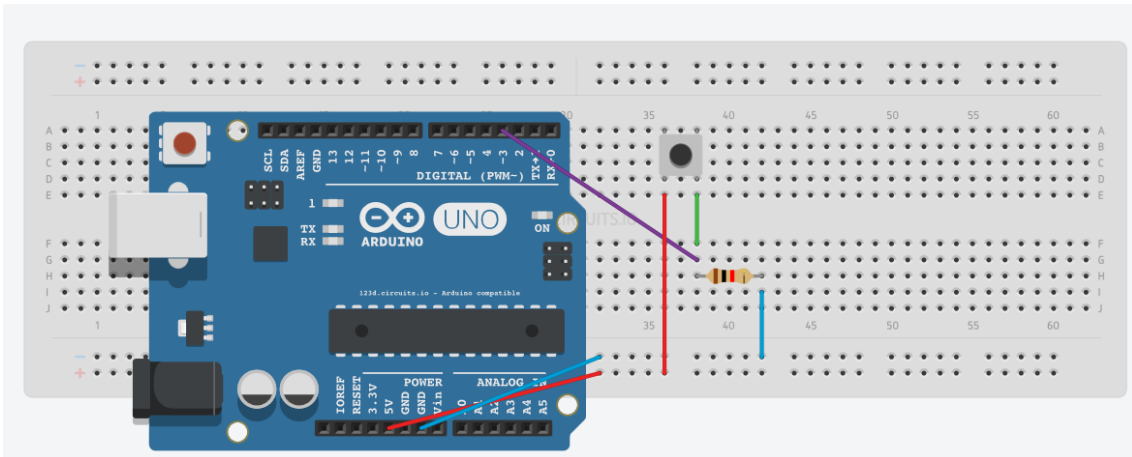


Figura 15. Conexión pulsador eléctrico

Para saber si se ha apretado el pulsador, bastará con saber si la entrada del pin correspondiente ha cambiado de voltaje.

Las patillas de cada uno de los pulsadores están conectadas internamente con las que se encuentran en la fila del lado opuesto. La conexión realizada en la Figura 15 muestra una unión de las 2 patas de la izquierda hacia la alimentación de 5V, mientras que las de la derecha irán conectadas a GND. Para no provocar un cortocircuito, se insertó en el medio una resistencia suficientemente grande, en este caso de 1KΩ. En el circuito, se podrá medir la tensión que hay entre el pulsador y la resistencia. Este modo de conexión se denomina PULL-DOWN. Así, se leerá el pin de entrada (en el caso de la Figura 15, el pin 3) para saber si está a 0V o a 5V, en cuyo caso la respuesta será 5V si está presionado (HIGH), o 0V si no lo está (LOW). Por tanto, el pin digital 3 está permanentemente unido a GND y recibirá 5V cuando sea pulsado.

Análogamente ocurrirá con el resto de pulsadores y sus pines correspondientes.

Los pines de la placa Arduino utilizados finalmente para los pulsadores se adjuntan en la siguiente Tabla 2:

Nº de Pulsador	Pin Arduino
1	3
2	4
3	5
4	6
5	7
6	15
7	16
8	17

Tabla 2. Pines de Arduino para los pulsadores

2.3.2 LEDs

Los LEDs utilizados emiten luz blanca de alta luminosidad. Tras varias pruebas de iluminación, se decidió que con un único LED de este tipo, se proporcionaba luz suficiente para una correcta percepción.

Los LEDs serán configurados como salidas del microcontrolador. Al haber 8 LEDs distintos e independientes, se buscó una solución para evitar utilizar otros 8 pines de la placa Arduino, ya que junto con los 8 ya utilizados en los pulsadores, limitaban la posibilidad de conectar más elementos.

Alternativa de diseño

Inicialmente se pensó en solucionar este problema usando un decodificador 3 a 8, donde con solo 3 entradas se obtuviesen 8 salidas, una para cada LED. Finalmente se descartó esta idea ya que, pese a ser igualmente válida a la solución final, cada LED debería llevar un transistor asociado para estabilizar la corriente, además de una resistencia, con lo cual aumentaba el cableado considerablemente y dificultaba la posterior tarea de soldadura.

Diseño final

El diseño finalmente elegido fue utilizar un registro de desplazamiento, concretamente el 74HC595, muy utilizado con Arduino y cuya librería contiene directamente funciones específicamente implementadas para trabajar con él (5).

El **74HC595** es un registro de desplazamiento de 8 bits con una entrada serie y salida paralelo, que controla ocho salidas con tan solo 3 pines mediante las entradas Clock, Latch y Data, pines 11, 12 y 14 respectivamente (ver Figuras 16 y 17).

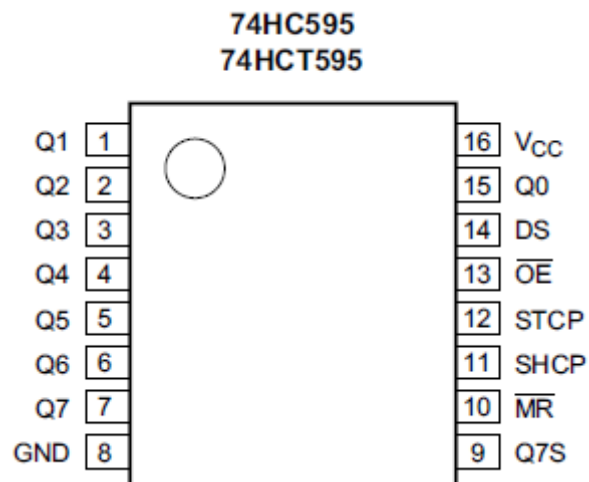


Figura 16. Esquemático del 74HC595

Symbol	Pin	Description
Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7	15, 1, 2, 3, 4, 5, 6, 7	parallel data output
GND	8	ground (0 V)
Q7S	9	serial data output
MR	10	master reset (active LOW)
SHCP	11	shift register clock input
STCP	12	storage register clock input
\overline{OE}	13	output enable input (active LOW)
DS	14	serial data input
Q0	15	parallel data output 0
V _{CC}	16	supply voltage

Figura 17. Descripción de pines del 74HC595

El registro de desplazamiento funciona de manera que espera nuevas secuencias de datos. Cuando el bit de Data cambia de estado (LOW A HIGH) y el bit de Clock genera un nuevo pulso de reloj pasando de HIGH a LOW, se graba el valor del pin de Data en la posición actual donde se encuentre el desplazamiento. Esta acción se repite 8 veces para generar un byte a la salida (Q0-Q7). Así es, por tanto, como se consigue controlar un byte de salida con sólo 3 pines del microcontrolador.

Atendiendo una vez más a las hojas de características del 74HC595, se observa que, al estar alimentado por una V_{CC} de 5V, el valor máximo de corriente I_o que puede recorrer el chip por los pines Q0-Q7 es de 35mA. Si hay 8 pines, la corriente que circule por cada uno será:

$$I_{pin} = 35mA / 8 = 4,375mA$$

Por tanto, al estar conectada cada una de estas 8 patillas con una resistencia en serie con un LED (ver Fig. 18), obtenemos lo siguiente:

-Según la hoja de características del LED, V_{LED}=3,6V.

-La corriente que circula es I_{pin} = 4,375mA.

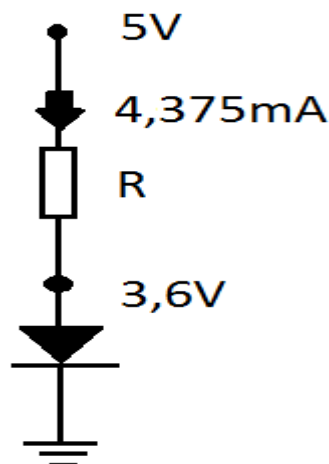


Figura 18. Esquema de conexión LED

El valor de la resistencia vendrá dado por la ley de Ohm:

$$V = I_o * R$$

De donde:

$$R = V / I_o$$

$$R = \frac{5V - 3,6V}{4,375mA} \approx 330\Omega$$

Una vez obtenido este valor, se eligieron resistencias comerciales de 330 Ω .

En la siguiente imagen (Fig. 19) se muestra las conexiones entre el registro de desplazamiento, los LEDs, sus resistencias y la placa Arduino.

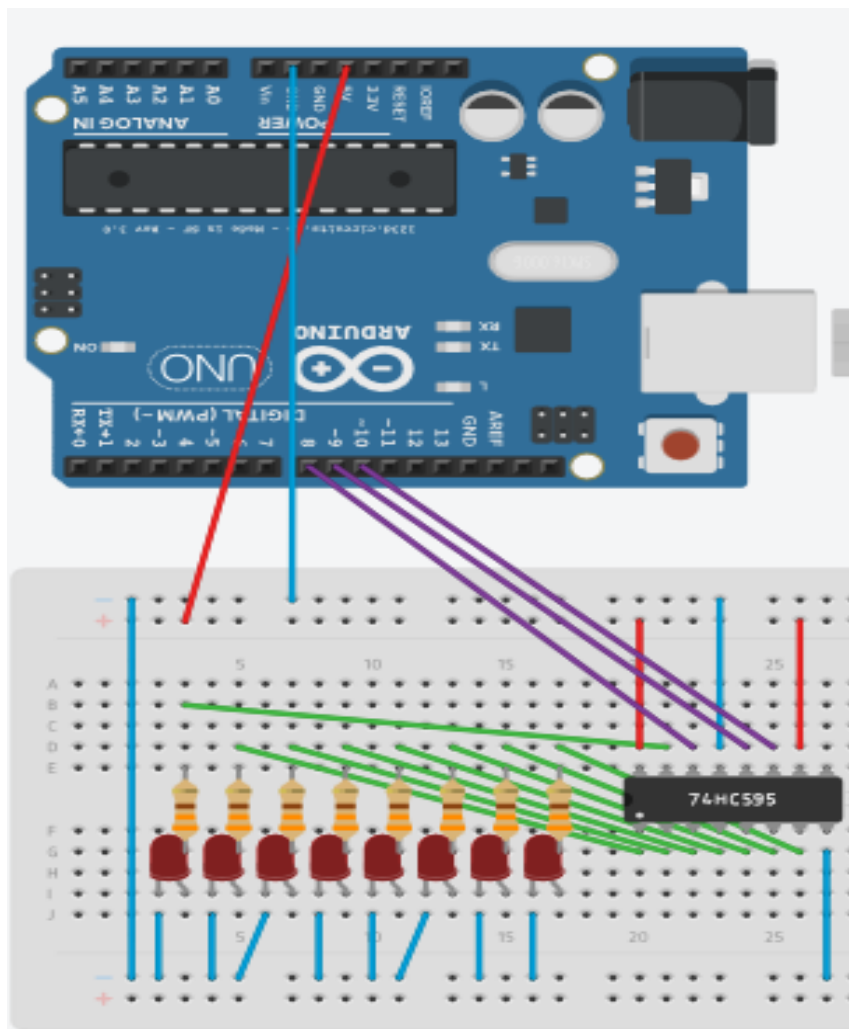


Figura 19. Conexión registro de desplazamiento y LEDs

Los pines de la placa Arduino utilizados para el registro de desplazamiento de los LEDs se adjuntan en la Tabla 3:

Pin 74HC595 LEDs	Pin Arduino
14	8
12	9
11	10

Tabla 3. Pines de Arduino para registro de desplazamiento de los LEDs

2.3.3 Pantalla LCD

El display mostrará los tiempos de juego donde se puede observar los progresos del usuario que puede ser de utilidad para el Centro.

El elegido para el sistema es una pantalla LCD de 2x16 de dimensiones totales 80x36 mm donde el área de visualización es de 64,5x16,4 mm.

Inicialmente se soldó una tira de pines en los huecos dispuestos para ello con el fin de poder utilizar el display en la placa de pruebas.

Atendiendo a sus hojas de características, la alimentación recomendada será 5V y su distribución de pines es la mostrada en la Figura 20.

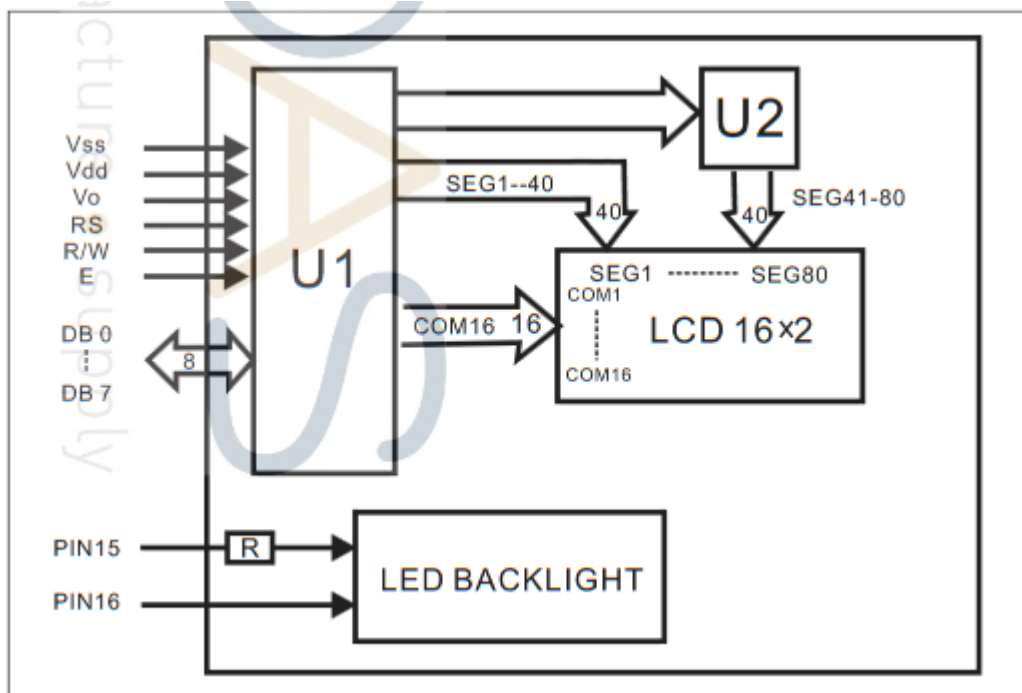


Figura 20. Extracto datasheet LCD

A continuación, se describirán brevemente dichos pines del display:

Vss: Pin de masa, GND o 0v.

Vdd: Alimentación principal de la pantalla, 5v.

Vo: Es el contraste de la pantalla. Estará conectado a un potenciómetro de 10KΩ (recomendado) para ajustar el contraste.

RS: Es el selector de registro. El microcontrolador comunica al display si quiere mostrar caracteres o si lo que quiere es enviar comandos de control como borrar la pantalla o situar el cursor en una posición determinada.

R/W: Es el pin de lectura/escritura. Como el fin es que escriba cuando se le ordene, estará conectado a GND en todo momento.

E: El pin de enable habilita la pantalla para recibir información.

D0-D7: Bus de datos de 8 bits. En este caso sólo se utilizarán los pines D4-D7 que servirán para establecer las líneas de comunicación por donde se transfieren datos.

A y K, o pines 15 y 16: Son los pines del led de la luz de fondo de la pantalla. Hablaremos de ellos más adelante.

Alternativa de diseño

En un principio, se conectaron los pines del display con la placa Arduino siguiendo la configuración de la hoja de características, tal y como se muestra en la Tabla 4:

LCD	Conexión Arduino
A	5V
K	GND
Vss	GND
Vdd	5V
Vo	Pin central del potenciómetro ⁸
RS	Pin 14 Arduino (A0) ⁹
RW	GND
E	Pin 15 Arduino (A1)
D4	Pin 16 Arduino (A2)
D5	Pin 17 Arduino (A3)
D6	Pin 18 Arduino (A4)
D7	Pin 19 Arduino (A5)

Tabla 4. Conexión inicial LCD con Arduino

Como se puede observar, siguiendo esta configuración, se están utilizando 6 pines de la placa Arduino. Podría servir como opción, pero se buscó una solución más efectiva ya que éstos, añadidos a los 8 de los pulsadores y a los 3 de los LEDs, ocupan 17 pines de los 18 disponibles.

⁸ El primer pin del potenciómetro se conectada a 5V y el tercero a GND.

⁹ Los pines analógicos A0-A5 de la placa Arduino, podrán ser configurados como digitales definiéndolos como pin 14-19.

Diseño final

Nuevamente se hizo uso del registro de desplazamiento 74HC595. Éste, además de ahorrar pines de la placa Arduino, proporcionaba una serie de ventajas adicionales:

- Liberación del pin A0, utilizado para poder leer a través de él ruido que servirá de semilla para la obtención de números aleatorios.
- Posibilidad de obtener un control individual del pin A del display que se encarga de la retroiluminación para poder apagarlo y reducir el consumo de batería.
- Simplificación del cableado a la hora de la soldadura en la placa de puntos.

Al utilizar este método, el único componente adicional que se necesita, además del propio 74HC595, es un transistor NPN. Este circuito permitirá controlar la salida del display, así como la retroiluminación. Ésta será controlada por dicho transistor NPN conectado al pin Q6 del registro de desplazamiento. Se utiliza este transistor ya que proporciona una corriente de salida adecuada.

La conexión entre el chip, el display y la placa Arduino se detalla a continuación en la Tabla 5:

74HC595	LCD	Arduino	
1	DB7	-	
2	DB6	-	
3	DB5	-	
4	DB4	-	
5	-	-	
6	-	-	Pin 2 del transistor NPN
7	RS	-	
8	-	GND	
9	-	-	
10	-	5V	
11	-	Pin 2	
12	-	Pin 11	
13	-	GND	
14	-	Pin 12	
15	E	-	
16	-	5V	
-	A	Pin 13	
-	K	-	Pin 3 del transistor NPN
-	Vss	GND	
-	Vdd	5V	
-	Vo	-	Pin 2 del potenciómetro ⁸
-	R/W	GND	

Tabla 5. Conexión final LCD con Arduino

Entre todas las conexiones realizadas, se observa que el pin A del display está conectado al pin 13 de la placa Arduino. Esta opción nos permite controlar individualmente la retroiluminación de la pantalla ya que se configuró por software que el pin 13 de Arduino fuese una salida que estuviese siempre encendida hasta la orden de apagado. Esto es muy conveniente para conseguir un ahorro de batería, ya que, una vez finalizado el nivel 5, si pasados 2 minutos no se pulsa el botón de Start, la pantalla borra los caracteres que esté mostrando en ese momento y, posteriormente, se apaga.

Hay que añadir que el pin 1 del transistor fue conectado a GND, y los pines 1 y 3 del potenciómetro fueron conectados a 5v y a GND respectivamente.

Así, gracias al registro de desplazamiento 74HC595 se ha reducido en 3 el número de pines de la placa Arduino utilizados. Además, se ha conseguido controlar la iluminación mediante el pin 13. A continuación, en la Figura 21, se muestra el diagrama de conexiones descrito en este apartado.

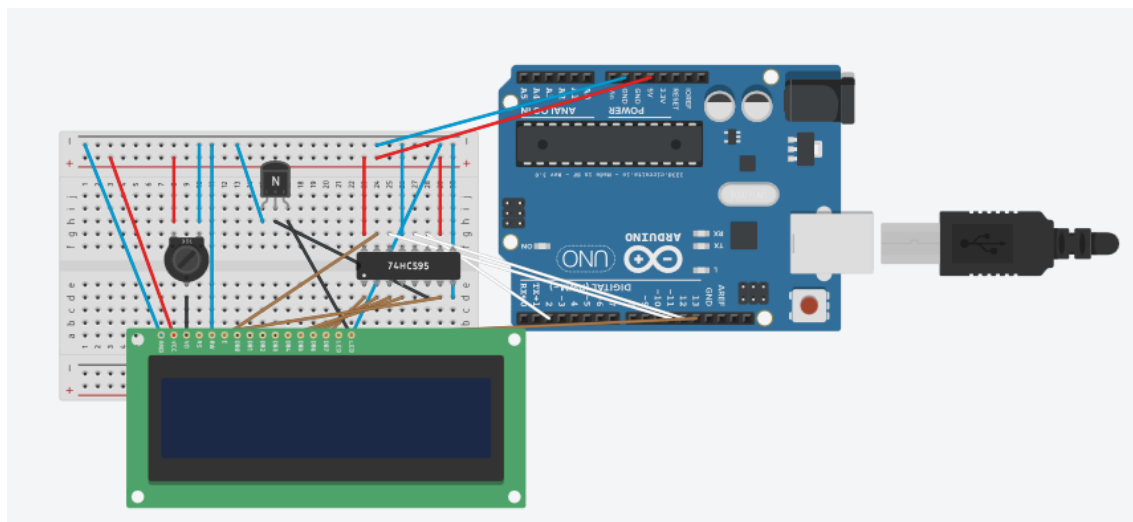


Figura 21. Cableado LCD con Arduino

Los pines de la placa Arduino utilizados finalmente para el registro de desplazamiento del display se adjuntan en la Tabla 6:

Pin 74HC595 display	Pin Arduino
14	11
12	12
11	2

Tabla 6. Pines de Arduino para registro de desplazamiento del display

2.3.3 Botón Start/Reset

Como se describirá posteriormente en el apartado [3.1.2 Funciones](#), la programación en Arduino consiste en 2 funciones principales; *setup* para definir los elementos a utilizar en la programación y sus funciones, y un *loop* (o bucle) ejecutándose, por tanto, continuamente.

No era de interés que una vez acabado el nivel 5, empezase el nivel 1 de nuevo sin consentimiento del jugador, por lo tanto la función *loop* debía quedarse en espera en vez de seguir ejecutándose.

Alternativa de diseño

Inicialmente se barajó la opción de utilizar interrupciones. En el modelo Arduino Uno es posible configurar INT0 e INT1 en los pines 2 y 3 respectivamente. Por tanto, se creó una interrupción software a través del pin 2.

El procedimiento fue configurar una variable global *flag* con valor 0 que cambiase su estado a 1 al terminar el nivel 5, de forma que dentro del *loop* existiese una condición *if* inicial que permitiese acceder a los niveles de juego si dicha variable *flag* tenía valor 0 o de lo contrario, bloquease el *loop*.

No es necesario definir el pin 2 como entrada, es suficiente con definir la interrupción en la función *setup*. A continuación, se utilizó el comando:

```
attachInterrupt( 0, CambioFlag, FALLING);
```

Los parámetros son la interrupción 0, el nombre de la función que actuará como rutina de atención a la interrupción (ISR o Callback) y el modo de disparo.

La función CambioFlag únicamente pondrá el valor del *flag* a 0 para permitir que se ejecuten las instrucciones del *loop* y volverá al punto donde se encontrase antes de ejecutar la interrupción.

```
void CambioFlag() { flag=0; }
```

El modo de disparo es *Falling* ya que la interrupción iba a estar asociada a un botón configurado como pull-up, porque el estado es normalmente HIGH y pasa a ser LOW al pulsar el botón, utilizando de esta manera el disparo con el flanco de bajada.

Aunque fue muy sencillo de realizar por software y funcionó a la perfección, una vez más se estaba utilizando un pin de la placa Arduino innecesariamente, ya que la misma placa proporciona otra solución igualmente eficaz.

Diseño final

La propia placa Arduino dispone un pin Reset, que servirá tanto para comenzar una nueva al final del nivel 5 como para volver al nivel 1 en cualquier momento¹⁰. Por ello, se añadió un botón externo conectado a dicho pin para que el usuario pueda pulsarlo cuando lo desee.

La variable *flag* seguirá cambiando su valor al finalizar el nivel 5, de tal manera que la función *loop* quedará en espera y no comenzará un nuevo juego sin consentimiento.

¹⁰ Esta opción es interesante para los usuarios a los que les cuesta más trabajo los niveles 3 y 4, de tal manera que pueden jugar tantas veces quieran en los niveles de secuencias más cortas.

En la siguiente imagen (Fig. 22) se muestra la conexión realizada. No sólo ahorra el uso de un pin extra, sino que el cableado es extremadamente sencillo.

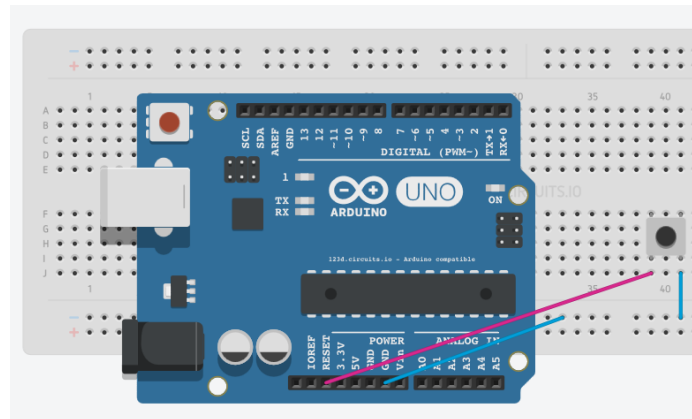


Figura 22. Conexión botón Start/Reset

2.3.4 Batería externa

Para la alimentación del sistema, se atendió a las peticiones de los Centros de utilizar una batería que estuviese integrada en el conjunto para evitar tener cables que pudiesen entorpecer a los usuarios. Además, el uso de una batería supone que el sistema puede ser usado en cualquier lugar, sin tener que estar supeditado a una toma de corriente.

La batería escogida tiene una capacidad de 2200mAh y un voltaje de salida de 5v, como se puede observar en la Figura 23.



Figura 23. Características batería externa



Figura 24. Detalle batería externa

La batería se conectará a la placa Arduino mediante el puerto USB (Fig. 24). Esto es muy conveniente porque, para utilizar ese puerto de Arduino, debe ser alimentado con 5V constantes, justo lo que proporciona dicha batería.

Para hacer una estimación de la duración de la batería, se procedió a hacer un cálculo aproximado del consumo del sistema. Para ello, interesa conocer el momento de máximo consumo. Éste será cuando todos los LEDs están iluminados al mismo tiempo y el display está a su vez encendidos, es decir, cuando el usuario ha fallado y se realiza un parpadeo de los LEDs a la vez que el display está iluminado y mostrando “Fallo”.

Por tanto, atendiendo al consumo de los elementos involucrados:

- La placa Arduino en vacío consume 46mA.
- El display LCD consume 25mA.
- Cada LED de alta luminosidad consume un máximo de 4,375mA.

Resultando en una suma de:

$$46\text{mA} + (4,375\text{mA} / \text{led} * 8\text{LEDs}) + 25\text{mA} = 106\text{mA}$$

La vida útil o la duración de una batería puede calcularse a partir de la corriente nominal de entrada de la batería y de la corriente de carga del circuito. La mayor vida útil de la batería será cuando la corriente de carga sea lo más pequeña posible, y viceversa.

El cálculo para conocer la capacidad de la batería se puede obtener mediante la siguiente fórmula matemática:

$$\text{Vida útil de la batería} = \frac{\text{Capacidad de la batería (mAh)}}{\text{Corriente de carga (mA)}} * \text{factor de corrección}$$

El factor de corrección permite tolerancias ante factores externos que pueden afectar la vida útil de la batería. Su valor es de 0,7.

Aplicando, por tanto, la anterior fórmula:

$$\text{Duración} = (2200\text{mAh})/106\text{mA} * 0,7 \approx 14 \text{ horas}$$

La batería utilizada tendrá una duración de unas 14 horas aproximadamente, en el caso de mayor consumo constante, con lo que podemos estar seguros de que durante una Jornada laboral en un Centro de Día podrá utilizarse sin problemas y recargarse durante la noche.

Finalmente, se adjunta el esquema de conexiones de todo el conjunto (Fig. 25).

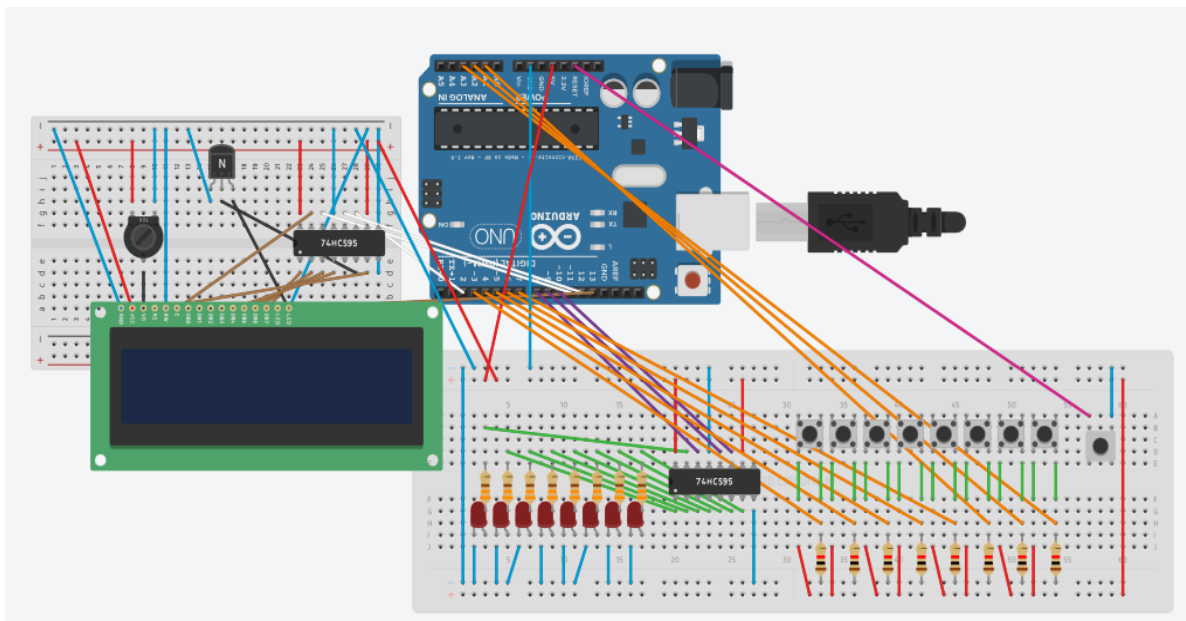


Figura 25. Esquema de conexión final

Capítulo 3. Implementación y pruebas iniciales

3.1 Software del microcontrolador

En este apartado, se tratará lo relacionado al microcontrolador que gobierna el sistema, desde su elección hasta la programación del juego y la descripción de las principales funciones programadas.

Como ya se ha comentado anteriormente en el resumen del proyecto, Arduino es la plataforma electrónica de código libre basada en software y hardware fáciles de usar. Para trabajar con ello, Arduino tiene su lenguaje de programación y su software (IDE) fácilmente instalable en cualquier ordenador.

Arduino cuenta con diversas placas, dependiendo de las características que tenga el proyecto a realizar.

3.1.1 Elección del microcontrolador

En una primera estimación del proyecto, se estudió los requerimientos que debería cumplir el microcontrolador. El elegido fue el ATmega328P, del fabricante Atmel, incluido en la placa de Arduino Uno.

Las características del microcontrolador ATmega328P (Fig. 26) son, entre otras, las siguientes:

- Tiene 14 pines digitales (0-13), de los cuales 6 pueden ser utilizados como salidas PWM de 8 bits, siendo 0 el nivel de mínima potencia, y 255 el de máxima.
- El pin 0 (TX) y el pin 1 (RX) se utilizan para transferir el programa al microcontrolador vía serie, dejando por lo tanto 12 pines digitales útiles.
- Tiene 6 entradas analógicas (A0-A5), que pueden ser programadas como I/Os digitales simplemente siguiendo la nomenclatura y definiendo los pines del 14 al 19.
- Reloj de 16 MHz.
- Conexión USB.
- Memoria Flash de 32 kBytes, de los cuales 0,5 kB son usados por bootloader.

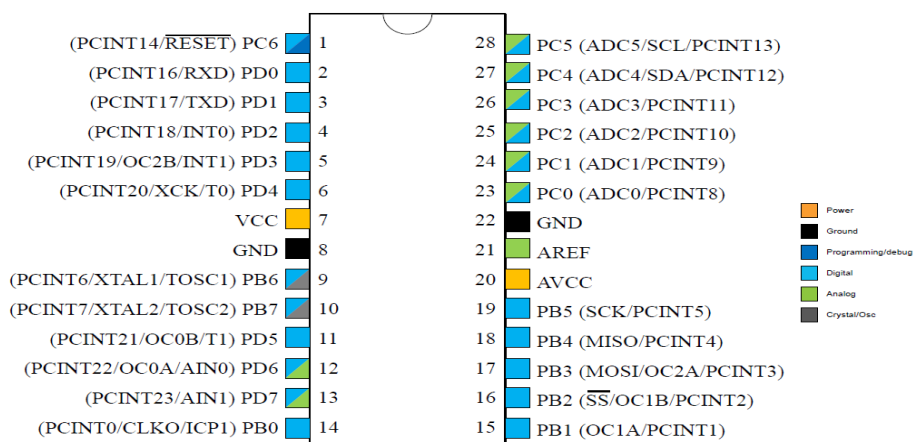


Figura 26. Descripción pines del ATmega328P

Además de todo lo anterior, la elección de dicha placa se debió a las siguientes razones:

- Cuenta con Bootloader¹¹, que hace las veces de programador.
- Tamaño y peso reducidos, 68,6mm x 53,4 mm y 25 gramos.
- El chip ATmega328P es fácilmente reemplazable en la placa de Arduino Uno, y su coste en el mercado es de aproximadamente 2 €.

3.1.2 Funciones

Siguiendo la programación en Arduino, el juego constará de dos funciones principales y otras funciones secundarias.

La primera función principal es *setup()*, y en ella se inicializan las variables, el puerto serie¹², los pinModes, se crean los mensajes de entrada, etc. Esta función sólo será llamada una vez cada vez que se encienda o se resetee la placa Arduino.

La segunda función principal es el bucle, o *loop()*. Como su nombre indica, se ejecutará continuamente las instrucciones que contenga mientras el Arduino esté encendido.

El resto de funciones fueron creadas con motivo de este juego, y se adjunta una tabla con sus características (ver Tabla 7).

En orden de aparición en el código del programa adjuntado en el Anexo:

Función	Descripción	Comentarios
<code>void clearRegisters();</code>	Pone todos los pines del registro de desplazamiento 74HC595 en LOW.	Función específica para el uso del registro de desplazamiento 74HC595. (5)
<code>void writeRegisters();</code>	Establece y muestra los registros.	Función específica para el uso del registro de desplazamiento 74HC595. (5)
<code>void setRegisterPin(int index, int value);</code>	Pone un pin individual HIGH o LOW.	Función específica para el uso del registro de desplazamiento 74HC595. (5)
<code>void juego_nuevo();</code>	Crea los vectores con las secuencias de números aleatorios de cada nivel.	
<code>void brillo_led (int led);</code>	Enciende y apaga el LED cuyo número es pasado como parámetro.	
<code>void parpadeo_fallo();</code>	Parpadean todos los LEDs.	Se utiliza cuando el jugador comete un error.

¹¹ Bootloader es un pequeño programa cargado previamente en el µc, y permite cargar el código sin necesidad de hardware adicional. Sólo está activo unos segundos cuando se resetea Arduino y después comienza el programa que está cargado en la memoria Flash.

¹² Se recomienda iniciar el puerto serie con el comando `Serial.begin(9600)` como acción de depuración.

<i>int</i> nbotones_pulsados(<i>int</i> estado_b1, <i>int</i> estado_b2, <i>int</i> estado_b3, <i>int</i> estado_b4, <i>int</i> estado_b5, <i>int</i> estado_b6, <i>int</i> estado_b7, <i>int</i> estado_b8);	Comprueba si hay más de un pulsador presionado.	Devuelve TRUE si hay más de 1 botón pulsado, de lo contrario devuelve FALSE.
<i>int</i> boton_pulsado();	Sirve para saber qué pulsador ha cambiado de estado de HIGH a LOW (cuando ha sido pulsado).	
<i>int</i> boton_cambio(<i>int</i> boton_inicial)	Igual que botón_pulsado(), pero comprueba qué botón ha cambiado de LOW a HIGH.	Permite conocer cuándo el botón ha dejado de estar pulsado.
<i>int</i> comprobar(<i>int</i> button_registered, <i>int</i> ronda_jug_n01)	Comprueba si el número de pulsador coincide con el número de LED de la secuencia correspondiente	Devuelve TRUE si coincide y FALSE de lo contrario.
<i>int</i> turno_jugador();	Comprueba si el jugador ha acertado la secuencia. Llama a todas las funciones mencionadas para que cada una compruebe su parte.	Devuelve TRUE si comprobar() devuelve TRUE y sale de la función cuando han pasado 3 rondas.
<i>void</i> contadores0();	Resetea a 0 los contadores de tiempo de los niveles y el contador de la interrupción.	
<i>void</i> contadores_total0();	Resetea a 0 el contador de tiempo total.	
<i>void</i> imprimotiempolcd();	Muestra el tiempo de cada nivel por el display.	
<i>void</i> imprimotiempolcd_total()	Muestra el tiempo total de juego por el display.	
<i>void</i> ISR_Count()	Rutina de atención a la interrupción del contador de tiempo que aumenta la variable global <i>contador</i> en 1 unidad cada 250 ms.	Se lanza cada 250ms, por tanto, cuando se haya terminado el juego y el valor del contador sea de 480 (2 min), apagará la retroiluminación de la pantalla para disminuir el consumo de batería.

Tabla 7. Descripción funciones del programa

3.1.3 Bucle principal

La función *loop()* consistirá en mostrar secuencias según el nivel en el que se encuentre el jugador. Para ello, siempre y cuando la variable *nuevo* sea 0, comprobará el nivel en el que se encuentra, iniciará los contadores de tiempo global y de cada nivel y, con ello, las rondas de

juego, donde irá mostrando las secuencias, comprobando que no hay fallos y que, una vez que haya conseguido 3 rondas correctas, pasará al siguiente nivel.

Cuando finalice correctamente el nivel 5, la variable *nuevo* tomará valor 1 para mantener el loop en espera, y, mientras no se inicie un nuevo juego, cuando hayan pasado 2 minutos, se apagará la retroiluminación del display LCD, disminuyendo el consumo de batería.

En la siguiente página se adjunta el diagrama de flujo del bucle principal (Fig. 27) donde se muestra su relación con las demás funciones.

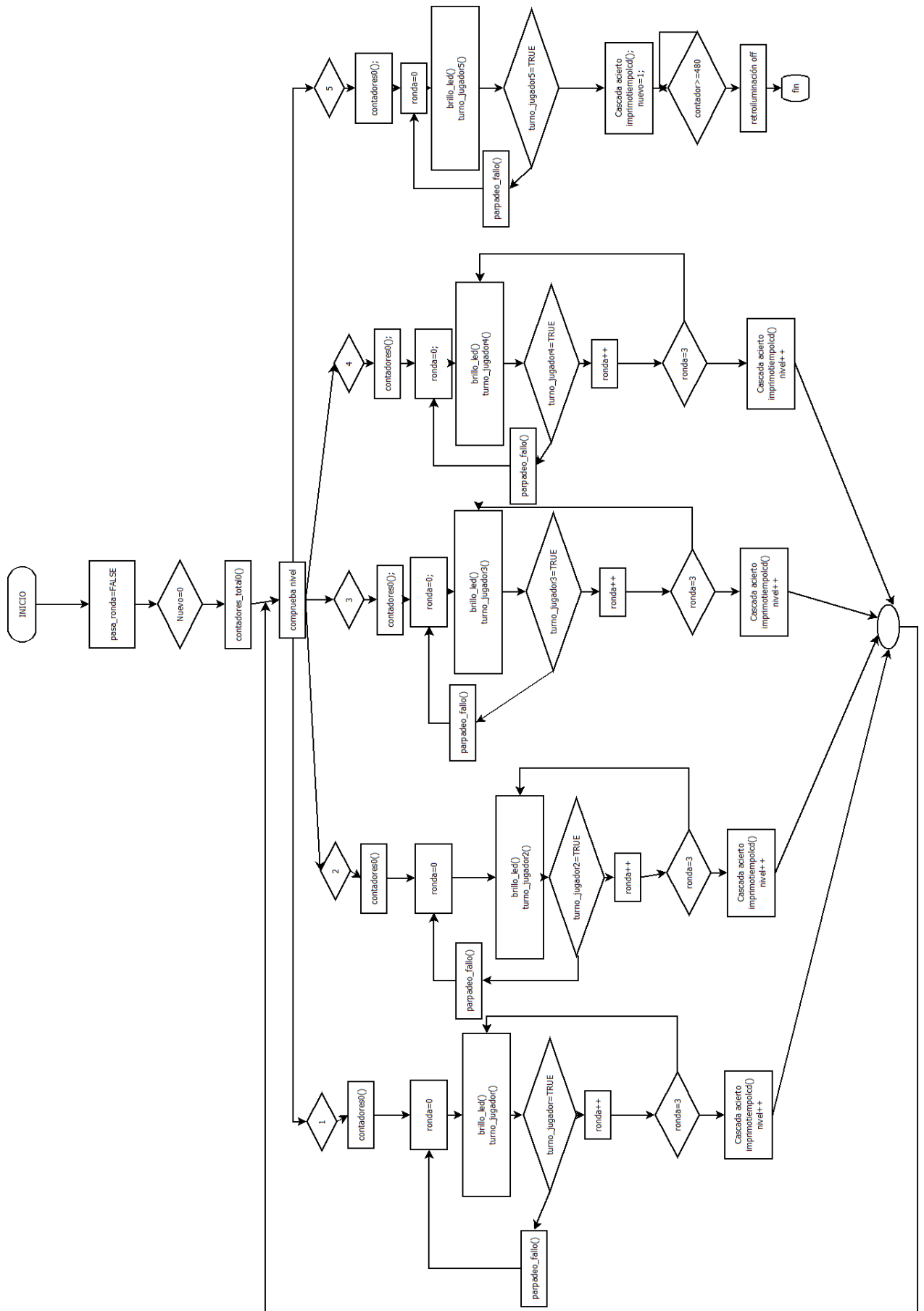


Figura 27. Diagrama de flujo función principal loop

3.2 Montaje mecánico del conjunto

3.2.1 Impresión de piezas en 3D Witbox

Al recibir especificaciones concretas por parte de los Centros (pulsadores cuadrados y de determinado tamaño), desde el principio se pensó en diseñar los pulsadores desde cero, ya que las soluciones comerciales no se ajustaban al propósito final, y las más parecidas elevaban considerablemente el coste.



Figura 28. Impresora 3D Witbox

Aprovechando el material del laboratorio 1.2.C12 del Grupo de Displays y Aplicaciones Fotónicas (GDAF) del Departamento de Tecnología Electrónica, se hizo uso de la impresora 3D Witbox de Bq (Fig. 28) que allí se encuentra.

Dicha impresora utiliza el método de Fabricación de Filamento Fundido (FFF), que consiste en ir añadiendo, capa por capa mediante el extrusor, filamento polimérico PLA de 1.75 mm fundido a 220° hasta completar la pieza deseada.

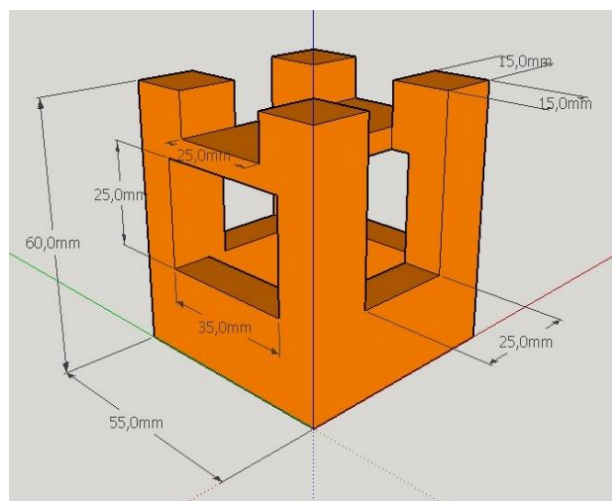


Figura 29. Diseño pulsador SketchUp

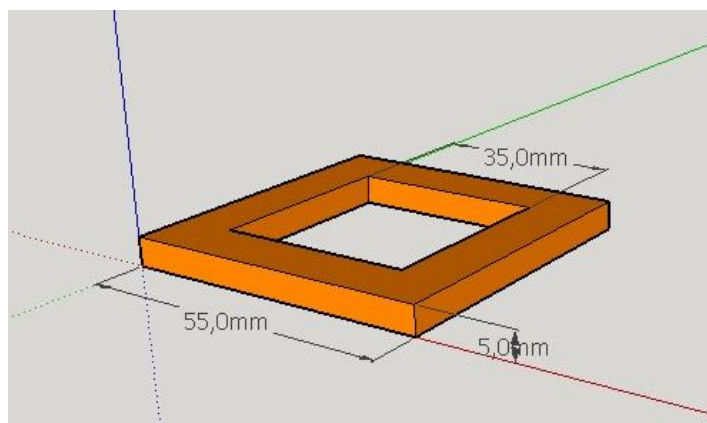


Figura 30. Diseño tapa SketchUp

Para el diseño de las piezas (Fig. 29 y Fig. 30) es necesario un software que sea compatible con la impresora 3D. En este caso, para crear los primeros modelos de los pulsadores, el elegido fue Sketch Up, debido a su facilidad de uso y a que proporcionaba un formato (.dae) compatible con el utilizado por la impresora (.gcode). Una vez concretados los pulsadores, se abrieron los archivos obtenidos en dicho software, Cura, diseñado por Ultimaker para imprimir piezas en impresoras 3D. Cura se encarga del aspecto y de la calidad de la pieza en 3D, permitiendo elegir los parámetros de impresión y dimensionando el diseño de las capas de la pieza¹³.

Para conseguir piezas impresas de buena calidad, es esencial el nivelado de la base. Esto consiste en ajustar la distancia entre la punta del extrusor y la base, para que sea la misma en todos los puntos.

Además de una correcta limpieza de la base para evitar tener impurezas que dificulten la tarea de impresión, es importante rociar ligeramente dicha base con laca para una correcta adherencia de la pieza y evitar que se mueva o desprenda durante el proceso. Asimismo, es necesario controlar que el extrusor no se encuentre obstruido por restos de filamento, así que en caso de que hubiese restos, se eliminan con un cepillo de cerdas metálicas, o calentando el extrusor a altas temperaturas.

Por último, se seleccionó una calidad de impresión alta, de 100 micras, y una velocidad de impresión recomendada de 50 mm/s, para tener una buena adhesión entre capas. En la Figura 31 se muestra el inicio de la impresión de una tapa.

¹³ Nótese que en la Figura 28 se muestra el diseño del pulsador del revés. Esto es debido a que la impresora realiza mejor la pieza de esta manera ya que consigue una mayor adhesión en la base y, posteriormente, es más fácil despegarla de la bandeja de impresión. Además, se minimiza el número de elementos flotantes y saltos sin depositar plástico.

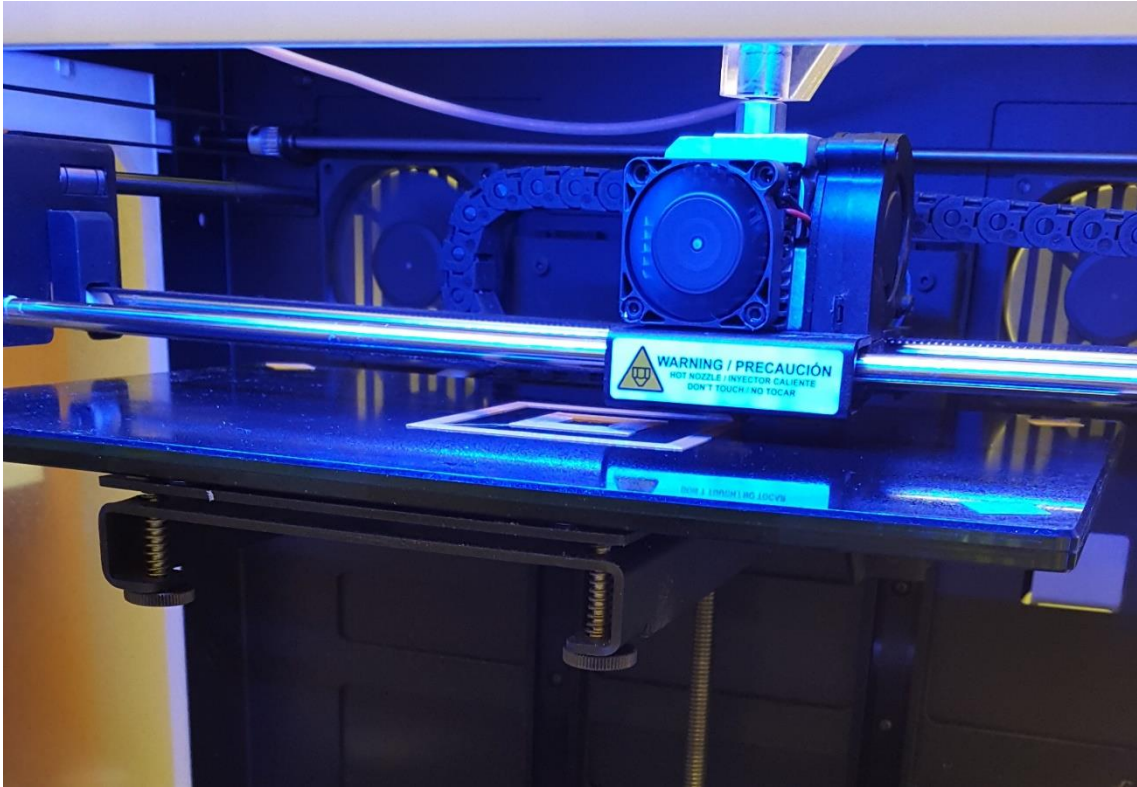


Figura 31. Detalle impresión 3D

Una vez impresos 3 modelos de pulsadores y 3 modelos de tapas, se llevaron ante usuarios reales que eligieron el diseño definitivo, descartando los que aparecen en la Figura 32 debido, uno a su gran tamaño, y otro al reducido hueco por donde se iluminaría.

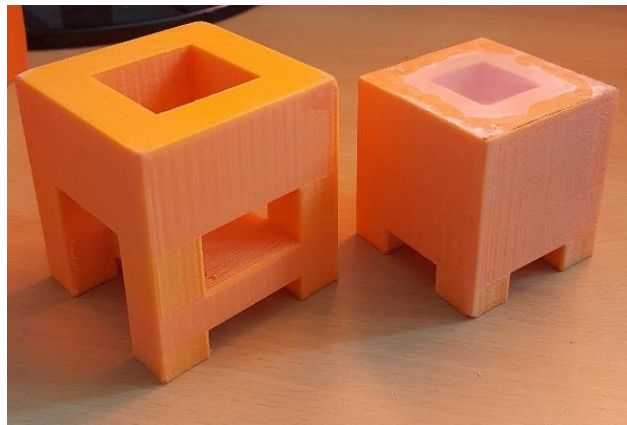


Figura 32. Piezas descartadas

A continuación, en la Figura 33 , se muestra el pulsador definitivo recién salido de la impresora Witbox con el plástico sobrante de la impresión, que se procedió a retirar, y posteriormente se limó la pieza como se aprecia en la Figura 34.

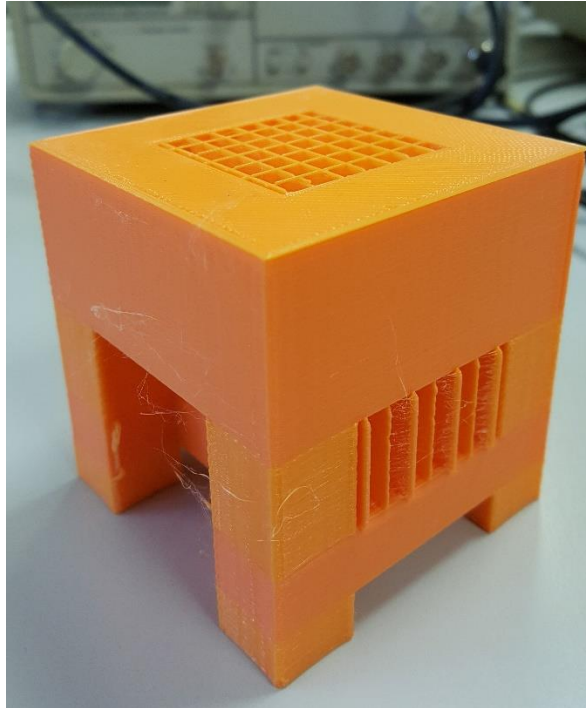


Figura 33. Pulsador final con plástico sobrante

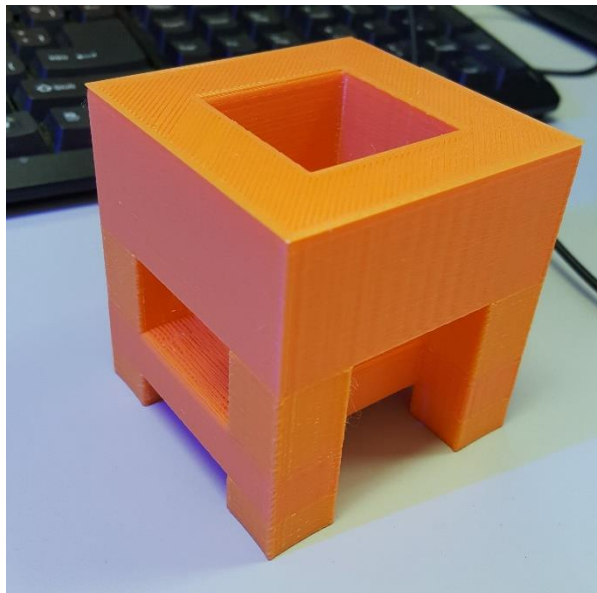


Figura 34. Pulsador final limado

Finalmente se imprimieron 7 pulsadores más con sus 7 tapas correspondientes. Las tapas servirán como protección ante el plástico rígido semitransparente donde se colocarán los números, y que sirve de difusor de la luz procedente del LED situado dentro del botón. De esta manera no se pulsará directamente sobre el plástico y se conseguirá algo de altura para el conjunto.

En la figura 35 se pueden observar todos los pulsadores finales.



Figura 35. Pulsadores finales

En la Figura 36, se observa la comparativa entre las tapas descartadas (izquierda y centro) y la finalmente elegida (derecha). Las razones de los descartes fueron, en el caso de la izquierda debido a que no eran del tamaño de los pulsadores elegidos, y en el caso de la tapa del centro, a su reducido hueco por donde saldría la luz.



Figura 36. Comparativa tapas impresas

El tiempo de impresión de cada pulsador fue de 5 horas y 30 minutos; y el de cada tapa de 35 minutos.

3.2.2 Fabricación de la caja

Pasos previos

El primer paso a realizar fue un boceto usando nuevamente el software SketchUp (Fig. 37), donde se describieron la forma de la caja, los pulsadores y el mecanismo interno de los pulsadores con los muelles.

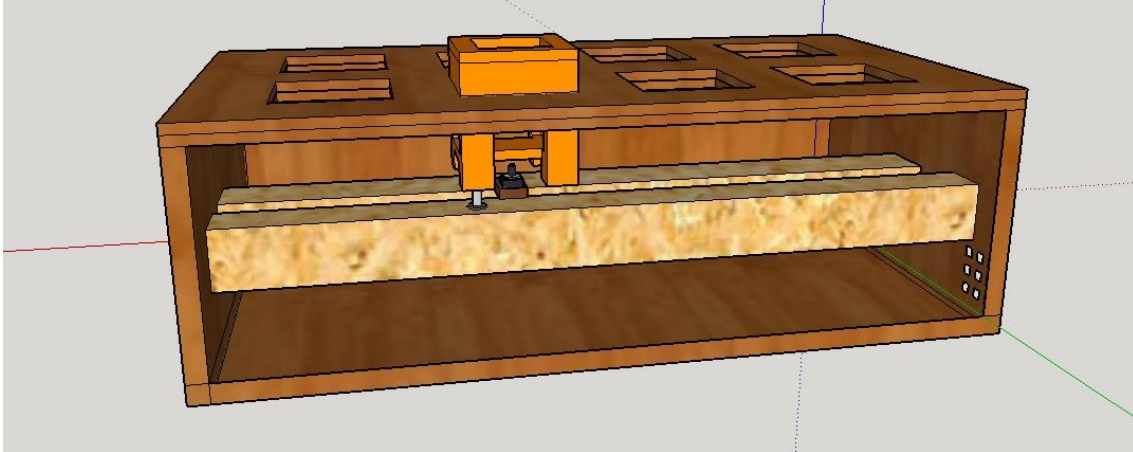


Figura 37. Diseño caja inicial SketchUp

A la hora de elegir la mejor opción para presentar el sistema a los usuarios de los centros, primeramente se pensó en una caja impresa con filamento PLA, del mismo material que los pulsadores, pero debido a las dimensiones del conjunto y al elevado tiempo de impresión, finalmente se optó por hacerla manualmente.

Tras esto, se barajaron diversas opciones de materiales para encontrar el equilibrio entre ligereza y robustez, para obtener un sistema fácilmente manejable por el usuario a la vez que consistente ante posibles golpes.

El mecanismo de accionamiento de los pulsadores se llevará a cabo con 4 muelles por pulsador, junto con unos clavos pasantes en cada pata, que servirán de guía para los muelles y ayudarán a encauzar el pulsador cuando baje. De esta manera se consigue que el pulsador tenga mayor recorrido. En la Figura 38 puede observarse el mecanismo.

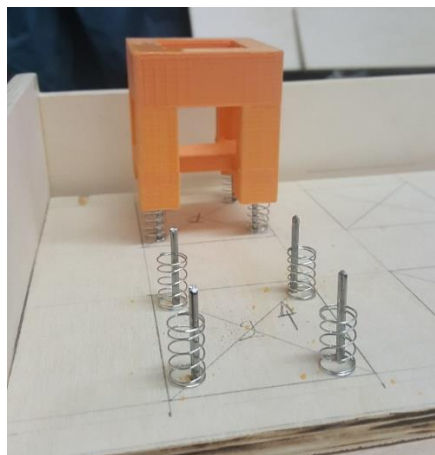


Figura 38. Detalle del mecanismo de los pulsadores

Por tanto, se realizaron unos pequeños orificios de 2 cm de profundidad en cada una de las 4 patas de los pulsadores y se introdujeron unos conductos de cobre en dichos orificios para evitar que los clavos rozasen con las posibles imperfecciones que se hubiesen formado al taladrar el plástico (Fig. 39). Los conductos de cobre fueron cortados de un tubo de 50 cm de largo e introducidos a presión con un martillo.

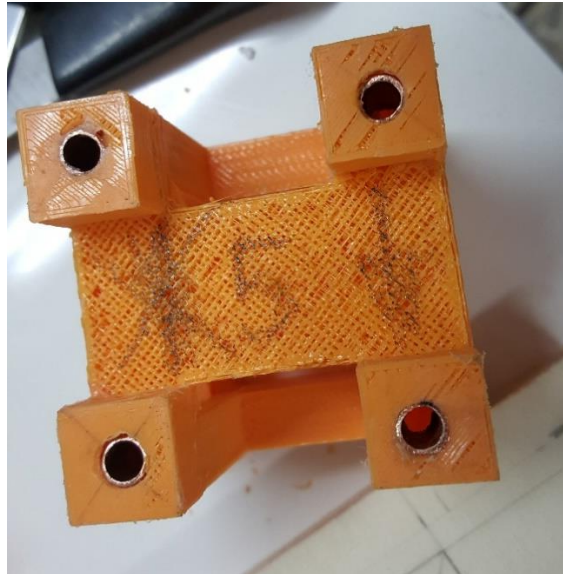


Figura 39. Detalle de los conductos en los pulsadores

Inicialmente se pensó utilizar 4 listones de madera (Fig. 37), colocados a lo ancho de la caja, como soporte para el conjunto de clavos y muelles, dejando así huecos entre ellos para poder manejar los cables de los LEDs y de los pulsadores eléctricos.

Diseño final

Finalmente, la opción elegida fue una tabla de contrachapado de 10 mm de espesor y de dimensiones 1200x600 mm, para poder cortar las diferentes piezas que formasen la caja.

La caja de madera estará formada por:

- Tapa superior de 390x200 mm, con cajeados de 5,6 cm de lado, que servirán como separación entre pulsadores, dejando 1 mm de holgura para evitar roces entre la madera y los pulsadores.

- Tapa frontal de 390x110 mm.

- Tapas laterales de 200x130 mm.

- Tapa posterior de 390x110 mm, que contará con unos tiradores, para poder abrir la caja y manejar el interior.

- Tapa inferior de 390x200 mm, que servirá como base del conjunto.

- Tabla de soporte para los pulsadores de 390x180 mm, donde irán sujetos los clavos que servirán como guías para los muelles.

Finalmente, no se llevó a cabo la idea mostrada en la Figura 37 debido a que el espacio que ofrecía para manejar el cableado no era tan ventajoso como la segunda opción que se barajó. En la Figura 40, simplificando para el caso de un pulsador, se observa que, sobre una tabla de contrachapado de 10mm de grosor, podemos conseguir también el soporte necesario para los muelles y los pulsadores eléctricos. Respecto al espacio necesario para poder pasar el cableado, se realizaron tres agujeros pasantes de 3 cm de diámetro que solucionarían esta cuestión.

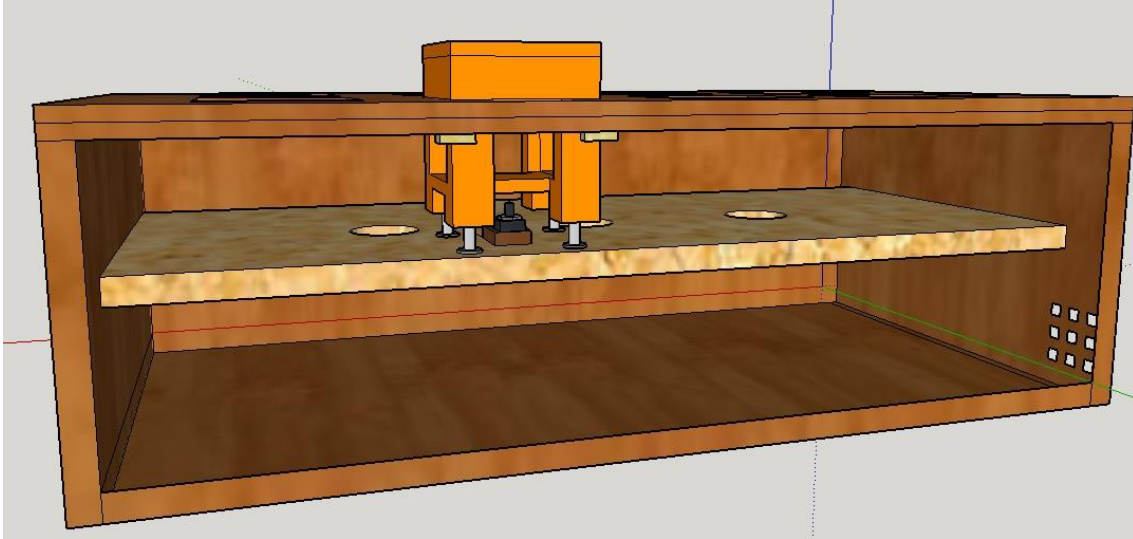


Figura 40. Diseño caja final SketchUp

A continuación, se añadieron unos pequeños tacos de madera de 10 mm de altura en el centro de cada cuadrado (Fig. 41), que servirán como soporte para los pulsadores eléctricos.

Además, se agujerearon en el mismo centro para poder encajarlos y fijar su posición con un poco de cola de contacto. La altura a la que se fijó la tabla soporte de los pulsadores fue de 6 cm con respecto a la base, suficiente para colocar posteriormente la placa Arduino, la placa de puntos y el cableado correspondiente.



Figura 41. Vista de la tabla de apoyo

Una vez limadas todas las tapas, se montaron para comprobar que encajaban correctamente. Era necesario que la caja pudiera abrirse para manejar el interior, por lo que se le colocó un tirador a la tapa trasera junto con unos imanes, que facilitan un buen cierre. A su vez, a la tapa superior se le colocaron unos pequeños clavos en los laterales que permitiesen abrir la caja por la parte de arriba por si era necesario manipular los pulsadores. En las Figuras 42 y 43 se observan las vistas frontal y trasera respectivamente.



Figura 42. Vista frontal caja contrachapado



Figura 43. Vista posterior caja contrachapado

En la tapa lateral derecha se realizó un cajeadado de 80x25 mm para poder colocar el display LCD que informará de los niveles y sus tiempos. Además, bajo éste se realizó otro agujero con una pequeña broca para ubicar el botón START/RESET.

En la tapa lateral izquierda se colocó un agujero de ventilación con una rejilla, la cual se fijó internamente con 4 clavos pequeños (Fig. 44).

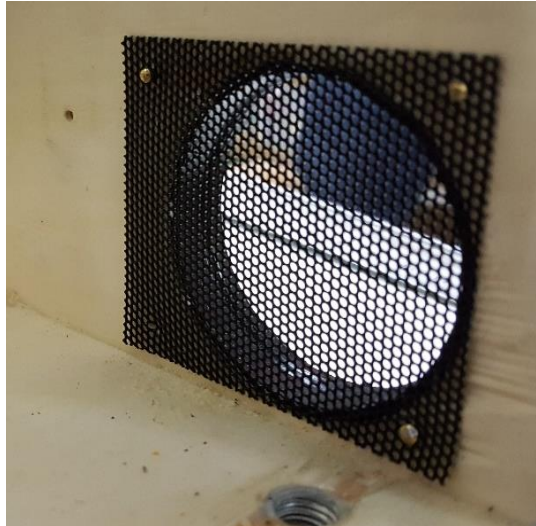


Figura 44. Vista interior rejilla de ventilación

Los pulsadores, por su parte, se pegaron con cola de contacto de modo que quedó el conjunto pulsador-plástico semitransparente-tapa. Además, se numeró cada uno en la parte interna del plástico, como fue solicitado por los centros colaboradores.

Una vez hecho esto, se pegó cada led en la parte interna de los pulsadores, con la ayuda de una pistola de cola termofusible, a una altura suficiente para permitir una buena percepción de la iluminación por parte de los usuarios.



Figura 45. Detalle pulsador con clavos de sujeción

Para poder conseguir que todos quedasen a la misma altura, se colocaron unos clavos fijadores (Fig. 45) en cada pulsador que, además, evitaban que saliesen de sus ubicaciones si la caja era volteada.

También se colocaron unas patillas en la base de la caja con el fin de dar inclinación al conjunto y que fuese más cómodo para utilizar en una mesa. Dichas patillas son regulables para poder controlar el grado de inclinación deseado (Fig. 46).

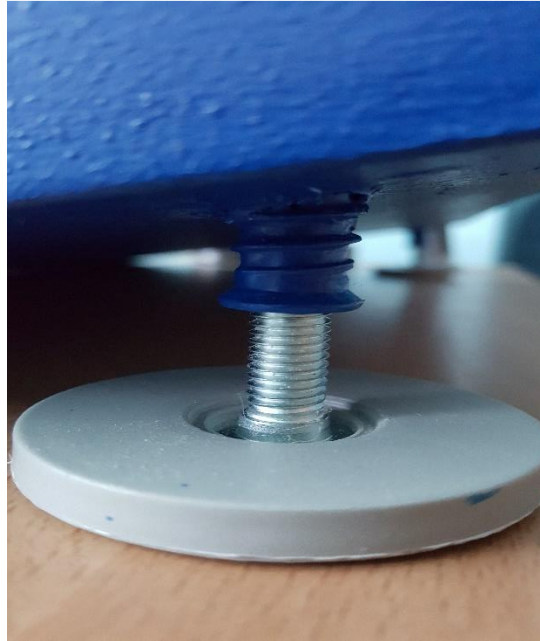


Figura 46. Detalle de patillas regulables

Por último, se aplicó una imprimación de madera en todas las superficies exteriores de la caja, con el fin de preparar el contrachapado para la pintura y posteriormente aplicar la pintura de color azul marino (Fig. 47). Además se han incorporado unos marcos en cada pulsador con una tira de cromado plateado, cortados a medida, con un fin meramente estético.



Figura 47. Pintura Acualux e imprimación Gesso

En la Figura 48 se puede apreciar la caja ya pintada y terminada.

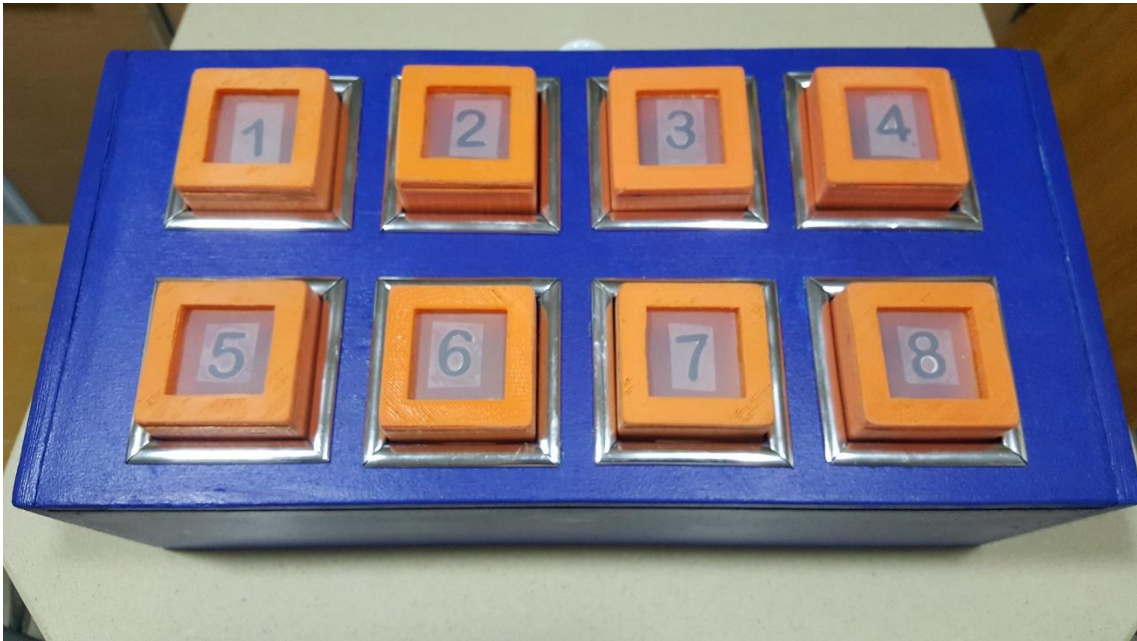


Figura 48. Vista superior de la caja terminada

3.3 Implementación electrónica

3.3.1 Placa de pruebas

Las primeras pruebas del circuito electrónico se realizaron con 3 únicos LEDs y 3 pulsadores (ver Fig. 49), para reducir la programación y comprobar que el programa reconocía correctamente las pulsaciones y no ocurrían rebotes.

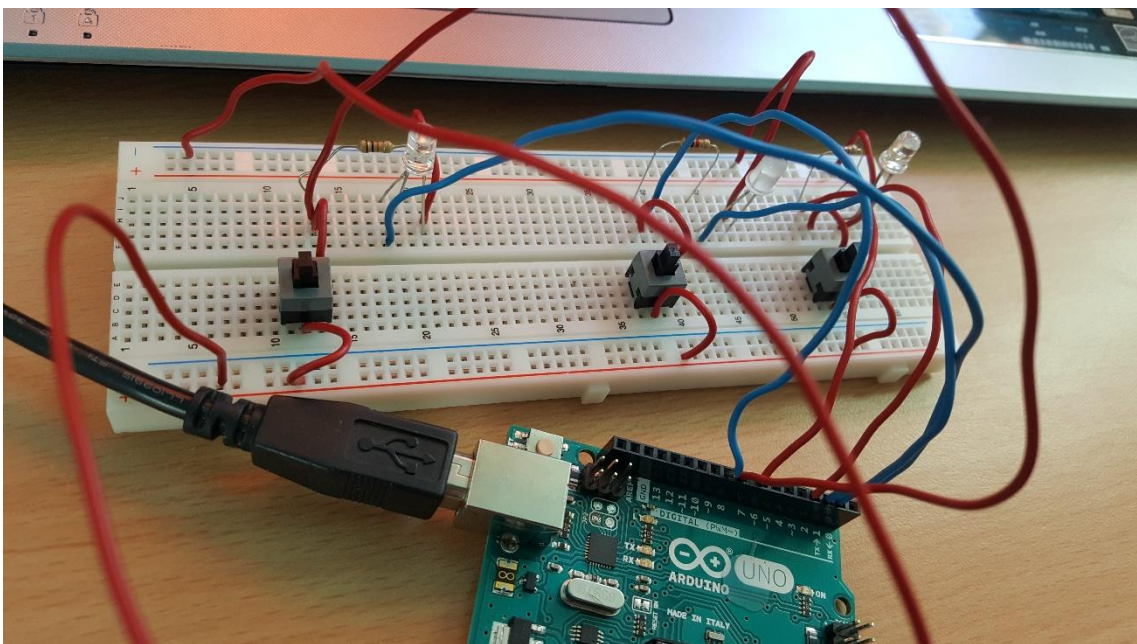


Figura 49. Placa de pruebas inicial

Las pruebas con el registro de desplazamiento 74HC595 fueron exitosas tanto en el caso de los LEDs, como en el del display LCD. En las Figuras 50 y 51 se muestra el grado de iluminación que se alcanzó en el caso de los LEDs.

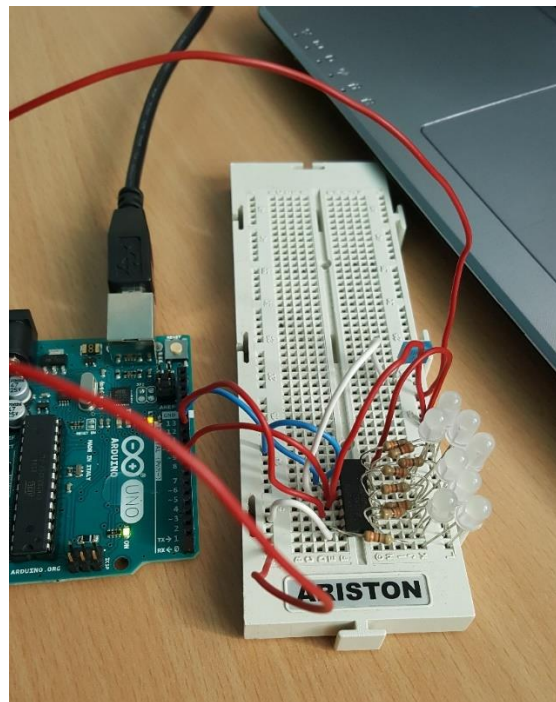


Figura 50. Prueba 74HC595 con LEDs apagados

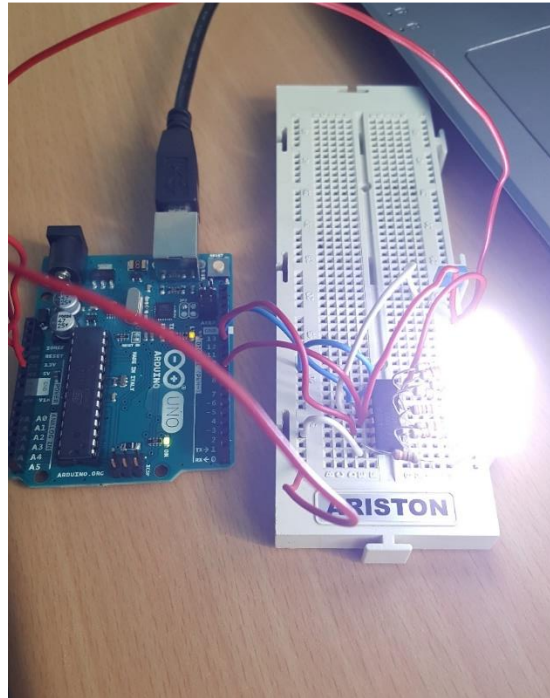


Figura 51. Prueba 74HC595 con LEDs iluminados

Para la programación del display LCD con el registro de desplazamiento, se utilizó una librería especialmente diseñada para este tipo de displays con el 74HC595 (6).

3.3.2 Placa de puntos

Una vez añadidos todos los LEDs y los pulsadores en la placa de pruebas y configurados en el sketch de Arduino, se procedió a soldar todos los componentes en una placa de puntos.

Primeramente se soldaron los componentes básicos (Fig. 52) como resistencias, cables conectores, registros de desplazamiento, componentes del display LCD (transistor NPN y potenciómetro regulador del contraste) y unas tiras de pines para realizar las pruebas de conexión de los LEDs y de los pulsadores. Se adjunta vista posterior de la placa de puntos soldada en la Figura 53.

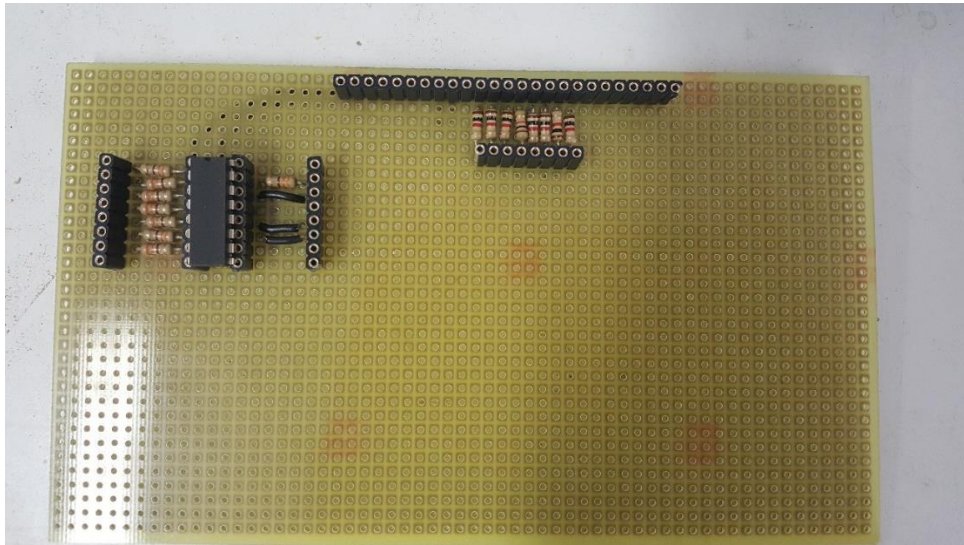


Figura 52. Placa de puntos inicial

Una vez comprobado el correcto funcionamiento, se procedió a desoldar las tiras de pines hembra donde encajaban los cables de los LEDs y de los pulsadores eléctricos. Para ello se utilizó una bomba de succión de soldadura hecha de aleación de aluminio para retirar el estaño sobrante, previamente calentado con la ayuda del soldador eléctrico.

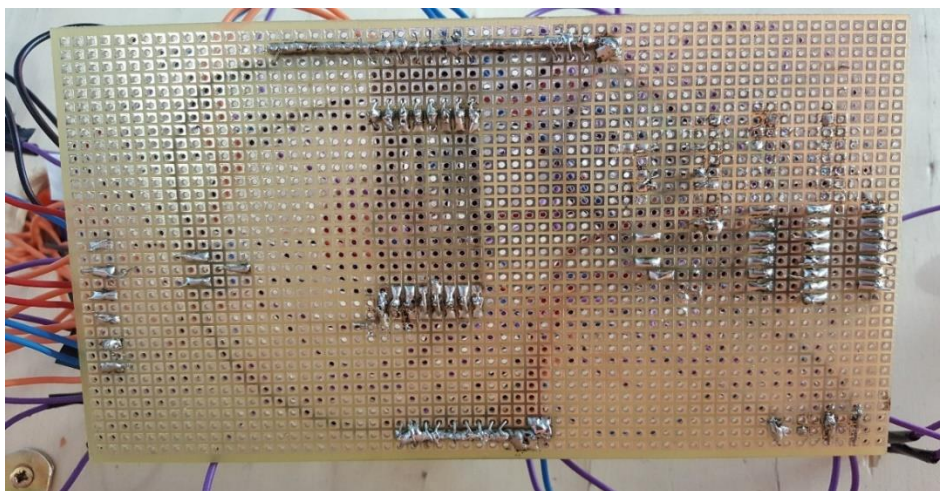


Figura 53. Soldadura placa de puntos

Finalmente, se soldaron el resto de los cables provenientes de los LEDs y de los pulsadores eléctricos a la placa de puntos (Fig. 54) y se comprobó una vez más su funcionamiento para, posteriormente, fijar la placa de puntos en la parte baja del sistema colocando unos pequeños botones de silicona en la base.

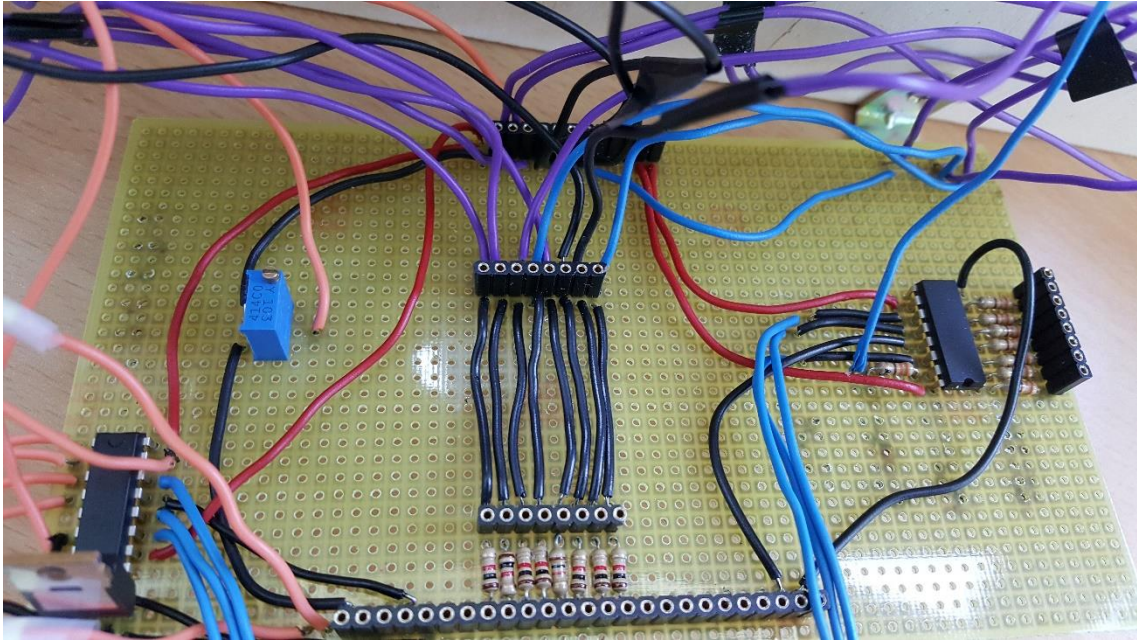


Figura 54. Placa de puntos final

Cabe añadir que, para fijar los cables que conectaban la placa de puntos con la placa Arduino, se aplicó un poco de cola termofusible, fácilmente eliminable en caso de necesidad, pero que proporcionaba la suficiente fijación para evitar desconexiones si la caja sufría algún golpe.

Capítulo 4. Resultados experimentales e implementación final

4.1 Pruebas desarrolladas

Cuando el conjunto estuvo soldado, se procedió a etiquetar los cables que van desde los elementos (pulsadores eléctricos, LEDs y display) a la placa de puntos. Cada cable lleva el nombre del lugar al que va conectado y su posición. En la Figura 55 se muestran los lugares de conexión antes de desoldar las tiras de pines hembra y soldar los cables directamente a la placa de puntos.

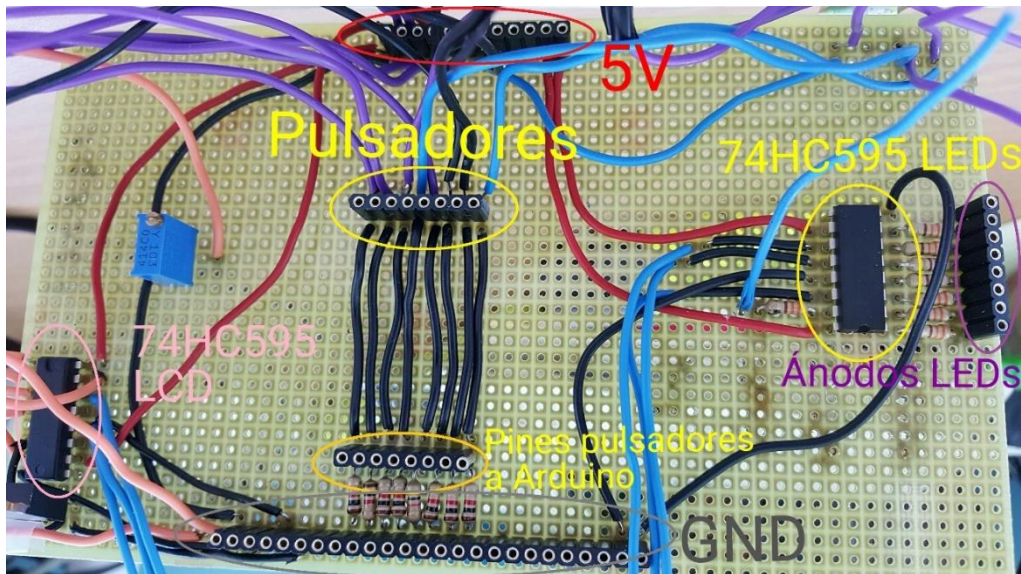


Figura 55. Descripción elementos placa de puntos

Posteriormente se realizó otra prueba de funcionamiento. Se comprobó la luminosidad de los LEDs y se amplió el tiempo que pasaba entre LEDs de la misma secuencia, ya que, al ir demasiado rápido y presionar un determinado pulsador, no daba tiempo a ver cuál se había iluminado a continuación.

Al término de realizar los ajustes y comprobar que todo funcionaba correctamente, se fijó con tornillos la balda que sirve de sujeción para los pulsadores (Fig. 56).

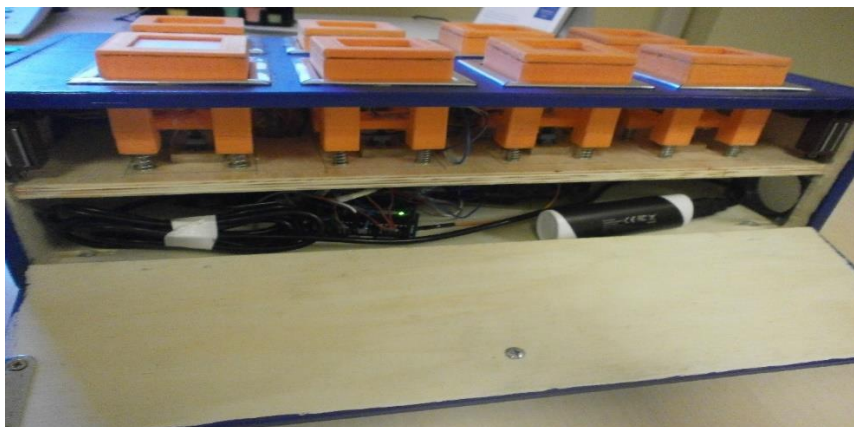


Figura 56. Interior de la caja con tabla fijada

4.2 Pruebas en los Centros

Una vez comprobado el correcto funcionamiento del sistema, se contactó una vez más con los centros para concertar una visita en la que los usuarios finales probasen el sistema.

Así, el 14 de Septiembre de 2016, se realizó la primera prueba en el Centro Dr. Salgado Alba. Primeramente se mostró el funcionamiento a la directora, Mónica Sánchez, a terapeutas y demás personal del centro.

Se acordó modificar la duración de los LEDs durante las secuencias alargando un poco más el tiempo que permanecían iluminados, de modo que los usuarios pudiesen percibir la iluminación sin problemas.

Para ello, se modificó una pequeña parte del código de Arduino por ordenador, como se observa en la Figura 57.



Figura 57. Pruebas Centro Dr. Salgado Alba

Una vez supervisado de nuevo, se llevó el sistema para ser probado en dos grupos de personas con diferentes grados de deterioro cognitivo, comenzando por el más leve.

Los dos grupos, compuestos por 4 y 9 personas, utilizaron el sistema durante una media de 10 minutos por usuario.

Una de las mayores dificultades encontradas en el primer grupo fue el entendimiento del juego, ya que no esperaban a que la secuencia se mostrase completa y presionaban los pulsadores antes de tiempo. Los usuarios de este primer grupo terminaron, tras una segunda ronda, entendiendo el funcionamiento y pudiendo manejar el sistema por ellos mismos.

El segundo grupo encontró más dificultad en esta tarea, pero al cabo de un tiempo terminaron comprendiendo también el funcionamiento, aunque quedándose en los niveles

iniciales. Esto no supuso un problema ya que, puesto que querían seguir jugando, al presionar el botón de Reset podían hacerlo en los niveles en los que se sintieran más cómodos.

Otro punto a tratar fue la explicación del parpadeo de fallo y la cascada de acierto que muestra el paso al siguiente nivel. Los usuarios, al estar centrando su atención en el sistema, encontraban confuso cuando erraban en la secuencia y se realizaba el parpadeo de fallo. Pero una vez más, este punto es achacable a que se debía al primer uso y, pasado un tiempo, lograron distinguir ambas acciones.



Figura 58. Pruebas con usuarios Centro Dr. Salgado Alba. Conforme a la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal, los usuarios han dado su consentimiento para aparecer en esta fotografía.

En la Figura 58 se observa a dos usuarias del primer grupo que accedieron a ser fotografiados para aparecer en la presente memoria.

Ambos grupos al término de las pruebas mostraron su interés en jugar más veces.

“El SIEC nos parece muy adecuado como herramienta para incorporar en las actividades cognitivas que se realizan, no como mero juego lúdico que proporciona entretenimiento, sino como herramienta que facilita la estimulación de capacidades cognitivas”.

Mónica Sánchez, directora del Centro de Día Doctor Salgado Alba.

El 19 de Septiembre de 2016, se realizó la prueba del sistema con usuarios del Centro de Día Municipal Peñagrande. Una vez más, se procedió mostrando primero el resultado final a la directora, Ainhoa Hernández, así como a la terapeuta del centro, Patricia Vega.

En esta ocasión, para no entorpecer el horario de actividades del centro, se probó el sistema únicamente en 3 personas diferentes durante una media de 15 minutos cada uno. En la Figura 59 se muestra una de ellas.

Inicialmente, tras atender a la explicación del juego por parte de la terapeuta, los usuarios volvían a presionar los pulsadores antes de tiempo. Esto se debía, según las profesionales del centro, a una falta de atención. Tras varias rondas, consiguieron llegar al nivel final mediante la repetición en voz alta de los números de las secuencias, lo cual centraba su atención y mejoraba su capacidad de memorización.



Figura 59. Prueba usuaria Centro de Día Peñagrande

Al igual que en el centro anterior, los usuarios se mostraron interesados en el juego y expresaron su deseo de seguir utilizándolo.

“A través de un juego dinámico, como el SIEC, los usuarios/as trabajan su capacidad de atención y memoria, de manera lúdica y siendo posible, además, verificar sus progresos a través del control del tiempo”.

Ainhoa Hernández, directora del Centro de Día Municipal Peñagrande.

A todo lo anterior hay que añadir que, una de las principales características del juego, como ya se sabe, es la iluminación. Los resultados en las pruebas indican que los usuarios perciben la luz emitida por los LEDs correctamente incluso en habitaciones muy iluminadas como fueron las utilizadas durante las pruebas.

Al término de las pruebas se procedió a realizar una encuesta de satisfacción basada en los Principios del Diseño para Todos a los 3 usuarios acerca de la utilidad del sistema, con una valoración de **1 (nada de acuerdo) a 5 (muy de acuerdo)**. Asimismo, se realizó una valoración

por parte del Centro Dr. Salgado Alba, por medio de su directora. Una copia de dicho modelo de encuesta se adjunta en el Anexo de la presente memoria.

Los resultados se muestran en la Tabla 8:

	1	2	3	4	5
1. El usuario ha comprendido rápidamente el manejo del juego.			50%		50%
2. Los tamaños de los botones y su distribución son adecuados.					100%
3. Se perciben bien las luces cuando se iluminan.				50%	50%
4. El juego ha resultado entretenido y el usuario desea jugar más.				25%	75%
5. El usuario se ha cansado físicamente o se ha hecho daño.	100%				
6. El usuario está interesado en utilizar más juegos como éste.					100%

Tabla 8. Resultados encuestas

4.3 Presupuesto

A continuación se dividen los elementos utilizados en el proyecto en parte mecánica y parte electrónica.

4.3.1 Parte Mecánica

<i>Elemento</i>	<i>Descripción</i>	<i>Precio Unitario</i>	<i>Cantidad</i>	<i>Precio Total</i>
1	Bobina de polímero PLA de 1.75 mm de diámetro. 1kg	19,90 €	1	19,90 €
2	Pintura Acualux azul marino, 100 ml.	2,95 €	1	2,95 €
3	Imprimación Gesso para madera	2,65 €	1	2,65 €

4	Plástico rígido semitransparente 14x20 cm	1,20 €	1	1,20 €
5	Tira de cromado color plata 100x1 cm	2,50 €	1	2,50 €
6	Patillas pie nivelador	1,20 €	2	2,40 €
7	Pomo tirador blanco	0,70 €	1	0,70 €
8	Pestillos magnéticos color marrón (pack de 4 unidades)	2,30 €	1	2,30 €
9	Pack de 50 clavos punta cabeza plana para guía de muelles 3x35 mm	3,10 €	1	3,10 €
10	Pack de 50 clavos finos para tope pulsadores 1,5x20 mm	2,00 €	1	2,00 €
11	Tablón de contrachapado 1.200 x 600 x 10 mm	9,90 €	1	9,90 €
12	Tornillo avellanado de 4 cm, cabeza de estrella	0,16 €	14	2,24 €
13	Pack 12 barras de pegamento termofusible (8x100 mm)	3,65 €	1	3,65 €
14	Pegamento instantáneo Axton 5g	1,99 €	1	1,99 €
15	Cinta aislante negra	1,00 €	1	1,00 €
16	Tubo de cobre de 6mm de diámetro, 50 cm largo	2,70 €	1	2,70 €
17	Acetona 1L	3,95 €	1	3,95 €
18	Laca fijadora 400 ml	1,50 €	1	1,50 €
19	Muelles de compresión 15 mm largo x 10 mm diámetro exterior, paso 4,29 mm	0,40 €	32	12,8 €
Total parte mecánica				79,48 €

Tabla 9. Materiales parte mecánica

4.3.2 Parte Electrónica

<i>Elemento</i>	<i>Descripción</i>	<i>Precio Unitario</i>	<i>Cantidad</i>	<i>Precio Total</i>
1	Placa Arduino UNO REV3	24,29 €	1	24,29 €
2	LED blanco alta luminosidad	1,30 €	8	10,40 €
3	Display LCD Alfanumérico 2x16	9,08 €	1	9,08 €
4	Transistor NPN	0,55 €	1	0,55 €
5	Pulsador pequeño montaje superficial	0,75 €	8	6,00 €
6	Resistencias 1/4W diferentes valores	0,02 €	20	0,40 €
7	Potenciómetro trimmer 10K	2,32 €	1	2,32 €
8	Registro de desplazamiento 74HC595	0,44 €	2	0,88 €
9	Tira de 36 pines hembra	2,19 €	3	6,57 €
10	Cable unipolar 5m	0,95 €	3	2,85 €
11	Placa de puntos	7,66 €	1	7,66 €
12	Batería externa 2200mAh	8,00 €	1	8,00 €
Total parte electrónica				79,00 €

Tabla 10. Materiales parte electrónica

4.3.3 Total

El valor total de los elementos de ambas partes será, por tanto 158,48€.

Tanto en la Tabla 9 como en la Tabla 10 no se incluyen los honorarios por tratarse de una entrega académica. Para cumplir con la formalidad que debe tener un presupuesto, se incluyen a continuación los honorarios de la ingeniera del proyecto, coste que no se cargará a los centros colaboradores (ver Tabla 11).

Tomando como referencia el salario por hora de un ingeniero superior en proyectos de la Universidad Carlos III de Madrid: 20,5€/hora.

Además, se añaden las cargas económicas correspondientes:

Seguridad Social: 27%

Desempleo: 1,5%

Obteniendo,

Coste hora total: 26,3425€/hora

Tarea	Nº de horas	Coste/Hora	Coste total
Planificación	20	26,3425€	526,85€
Desarrollo del concepto	40	26,3425€	1053,7€
Diseño del sistema	100	26,3425€	2634,25€
Diseño detallado	180	26,3425€	4741,65€
Pruebas y afinación	80	26,3425€	2107,4€
Producción de transición	6	26,3425€	158,055€
Tiempo de realización de la Memoria	80	26,3425€	2107,4€

Tabla 11. Desglose coste del proyecto

TOTAL: 13329,305€

Capítulo 5. Conclusiones y líneas futuras

5.1 Conclusiones

Uno de los grandes objetivos cumplidos de este proyecto fue haber podido conectar mediante el SIEC a dos grupos de personas opuestos, y que ambos salgan beneficiados. Como ya se ha mencionado anteriormente, los usuarios/as del Centro Dr. Salgado Alba, son personas que sufren algún tipo de demencia o un deterioro cognitivo en fase leve/leve-moderado; mientras que el grupo del Centro de Día Peñagrande no tiene deterioro cognitivo, o si lo tiene es en fase leve, menor que los anteriores. El punto de unión entre estos dos grupos es el trabajo con el nivel de atención, ya que es la propia atención lo que facilitará la retención de las secuencias y, por tanto, el trabajo de la memoria.

El SIEC resulta útil y beneficioso a grupos de personas con características opuestas, cada uno trabajando a su nivel. Además, cada usuario tiene su propio espacio individual de juego, lo cual refuerza este punto de atención focalizada mientras se utiliza un sistema lúdico.

El coste de los materiales del proyecto es considerablemente más reducido que las soluciones de características similares que se encuentran actualmente en el mercado, como tablets u ordenadores con software especializado. Por otro lado, la adaptación conseguida con los materiales y el diseño realizado es siempre superior a cualquier app utilizada con dispositivos comerciales de propósito general.

El haber tenido la oportunidad de aplicar el Diseño para todos en un proyecto real creado desde una idea básica, no hace sino provocar una motivación para proyectos futuros. Así, se reafirma la idea de la necesidad de que todos los proyectos ingenieriles que de ahora en adelante se realicen, sean útiles y accesibles para todas las personas. Además, el poder comprobar de primera mano cómo los usuarios finales disfrutaban del sistema hace que todo el trabajo merezca la pena.

Por tanto, al término de este proyecto se puede concluir que se han cumplido todos los objetivos.

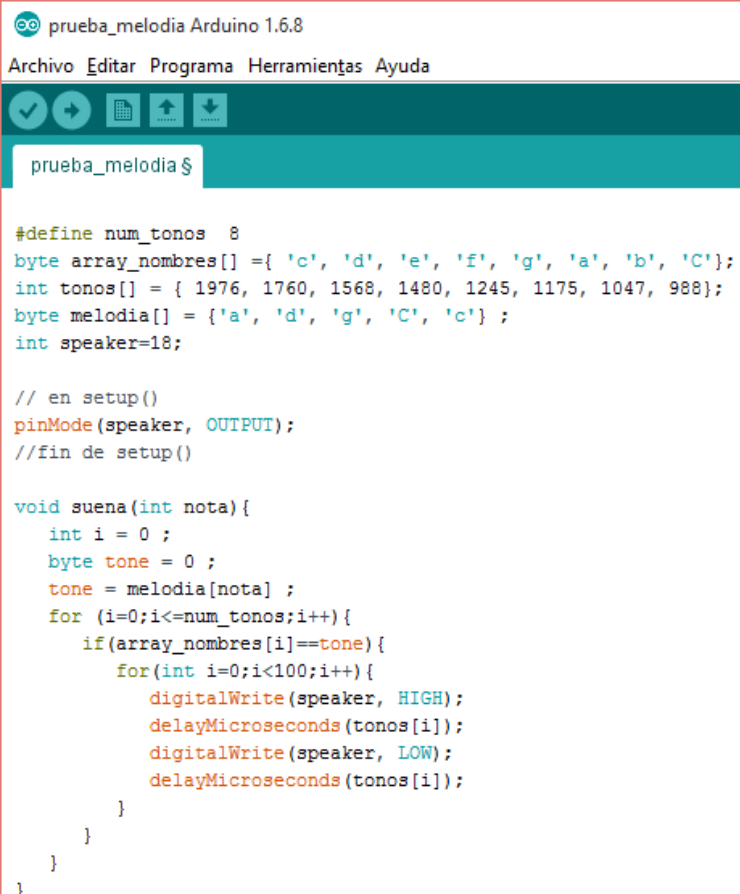
5.2 Líneas futuras

La gran ventaja de haber utilizado la plataforma Arduino en este proyecto es la facilidad para modificar el código de programación y variar las características del juego.

Una de las opciones barajadas al inicio del diseño del proyecto fue la introducción de un pequeño altavoz por donde se emitiesen sonidos tanto al pasar de nivel, como en los casos de fallo. Fue un punto controvertido, porque por un lado, evitaría la confusión que se generó inicialmente en el Centro Dr. Salgado Alba al confundir el parpadeo y la cascada con nuevas secuencias; pero por otro lado, se producen más fallos que aciertos, por lo que un sonido de fallo podría desmotivar a los jugadores. Incluso se estimó que un sonido de acierto desviaría el

elemento más importante del trabajo, su atención y, además, resultaría un inconveniente al utilizar el sistema en una sala con más personas realizando otras actividades.

La programación, sin embargo, fue implementada inicialmente en Arduino (7). En la Figura 60, se muestra parte del código con la función suena(), que posteriormente debería ser introducida en los aciertos. Análogamente, se añadiría suena2() con otras notas diferentes para ser introducida en los fallos.



```
prueba_melodia Arduino 1.6.8
Archivo Editar Programa Herramientas Ayuda
prueba_melodia $

#define num_tonos 8
byte array_nombres[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
int tonos[] = { 1976, 1760, 1568, 1480, 1245, 1175, 1047, 988 };
byte melodia[] = { 'a', 'd', 'g', 'C', 'c' };
int speaker=18;

// en setup()
pinMode(speaker, OUTPUT);
//fin de setup()

void suena(int nota){
  int i = 0 ;
  byte tone = 0 ;
  tone = melodia[nota] ;
  for (i=0;i<=num_tonos;i++){
    if(array_nombres[i]==tone){
      for(int i=0;i<100;i++){
        digitalWrite(speaker, HIGH);
        delayMicroseconds(tonos[i]);
        digitalWrite(speaker, LOW);
        delayMicroseconds(tonos[i]);
      }
    }
  }
}
```

Figura 60. Programación sonido

Además, para evitar posibles desconexiones, la circuitería podría implementarse en una placa PCB, reduciendo considerablemente todo el cableado.

Otra posible mejora podría consistir en introducir un interruptor para seleccionar una velocidad mayor o menor de aparición de las secuencias de LEDs, y así adaptarse a las diferentes habilidades de los jugadores. Esto se podría conseguir utilizando un potenciómetro cuyo valor lleve a un valor de caída de tensión regulable que se lea por el convertidor A/D de la placa Arduino y conforme un nivel de velocidad proporcional al valor leído. Un pin analógico (A0-A5) podría dar solución a esta mejora del sistema. Las entradas analógicas son aquellas que pueden variar su valor a lo largo del tiempo. En Arduino pueden ir desde 0V-5V tomando hasta 1024 valores distintos por lo que resultaría útil, por tanto, utilizar un potenciómetro.

Bibliografía

1. **Aspinall, Adam.** *www.mirror.co.uk.* [En línea] 24 de Julio de 2016.
<http://www.mirror.co.uk/tech/playing-online-games-can-halve-8485786>.
2. **The Center for Universal Design.** *Principles for Universal Design.* s.l. : NC State University, 1997.
3. **Ceapat/Imsero.** *www.imsero.es.* [En línea] 2013.
4. **Sánchez, Mónica.** *Sobre el Centro de Día Doctor Salgado Alba.* Madrid, 16 de Septiembre de 2016.
5. **Luke, Dr.** *www.bildr.org.* [En línea] 8 de Febrero de 2011.
6. **Parish, Chris.** Controlling an LCD Display with a Shift Register . *www.cjpatish.blogspot.com.* [En línea] 16 de Enero de 2010.
7. **Igoe, Tom.** *www.arduino.cc.* [En línea] 21 de Enero de 2010.
<https://www.arduino.cc/en/Tutorial/toneMelody>.
8. *Administración de operaciones. Producción y cadena de suministros.* Chase, Jacobs, Aquilano, 12e (2009). Ed. Mc Graw-Hill
9. <http://www.electrontools.com/Home/WP/2016/03/09/registro-de-desplazamiento-74hc595/>
Último acceso: 12/08/ 2016
10. <https://www.arduino.cc/en/Tutorial/ShiftOut>
Último acceso: 14/08/ 2016
11. <http://www.prometec.net/indice-tutoriales>
Último acceso: 26/08/ 2016
12. <https://circuits.io/>
Último acceso: 14/09/2016
13. <https://www.tomsplanner.es/>
Último acceso: 20/09/2016

Créditos Fotográficos

CC disacc.es – Figura 1.

CC Costelo – Figuras 2 y 3.

CC Rehacom – Figuras 4 y 5.

Índice de acrónimos

GND - Ground

LCD – Liquid Crystal Display

LED - Light-Emitting Diode

PLA - Políácido Láctico

PCB – Printed Circuit Board

Anexos

[A.1 Manual de usuario](#)

SIEC

Sistema Interactivo de Estimulación Cognitiva



Manual de Usuario

Sara García Fernández

Universidad Carlos III de Madrid

Índice

1. Contenido	77
2. El juego	78
3. Alimentación.....	81
4. Posibles problemas y soluciones.....	82
Anexo: Conexiones.....	83

I. Contenido

El sistema (Ilustración I) consta de una caja rectangular que contiene lo siguiente:

- 8 pulsadores numerados.
- 8 luces LEDs de color blanco.
- Una pantalla LCD que funciona a modo de display de tiempos.
- Un botón adicional de Start/Reset.
- Una batería externa recargable.
- Unas patillas reguladoras de altura.



Ilustración I. Vista superior de la caja del juego

2. El juego

El juego consiste en memorizar la secuencia que se ilumina con sus LEDs en cada nivel y repetirla sobre los pulsadores correspondientes. Se compone de 4 niveles con 3 rondas en cada uno y una última ronda donde se deberán presionar sólo los pulsadores pares.

El número de LEDs iluminados en cada secuencia depende del nivel en el que se encuentre el jugador, así en el nivel 1 se encenderá un solo LED por secuencia, en el nivel 2 se encenderán dos por secuencia, y así sucesivamente hasta el nivel 5, donde el jugador debe presionar solamente los pulsadores pares y en orden creciente. Cada nivel del 1 al 4 pasa por 3 rondas. Se adjunta un resumen en la Tabla 1.

Nivel	Rondas	LEDs por ronda
1	3	1
2	3	2
3	3	3
4	3	4
5	1	4

Tabla 12. Resumen niveles y secuencias del juego

Cuando el jugador presiona un pulsador, el LED adherido a él se iluminará para mostrar que se ha presionado correctamente, como indica la Ilustración 2.

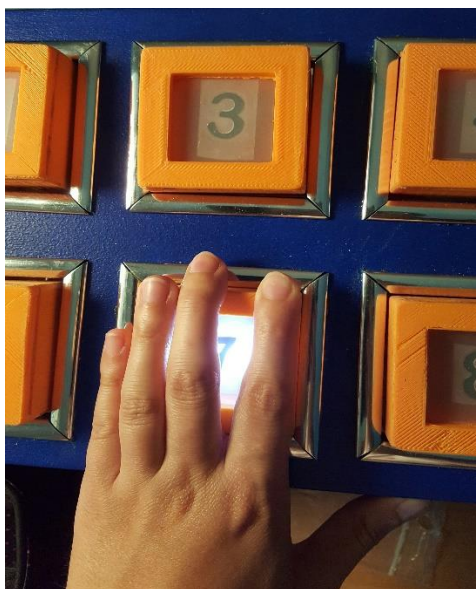


Ilustración 2. Ejemplo de pulsación

El jugador no debe pulsar hasta que se ha mostrado la secuencia completa, debe memorizarla y seguidamente repetirla mediante los pulsadores.

El juego quedará a la espera del turno del jugador, sin límite de tiempo. Si la secuencia es correcta, pasará a la siguiente ronda con otra secuencia aleatoria. Si se completan las 3 rondas de cada nivel correctamente, se iluminarán todos los LEDs en cascada y se pasará al siguiente nivel.

¡Atención! Al avanzar de nivel, el juego realizará una secuencia en cascada, mientras que si se produce un fallo, los LEDs parpadearán ¡3 veces!

Si se produce un error, el jugador permanecerá en el nivel en el que se encuentre pero volverá a la ronda inicial. Parpadearán todos los leds como señal de que la secuencia del jugador no es correcta.

La Ilustración 3 muestra el display y el botón Start/Reset situados en el lateral derecho.



Ilustración 3. Display y botón Start/Reset

El display permitirá al jugador conocer el tiempo en el que ha realizado cada nivel, y el juego total.

Para comenzar un nuevo juego, el jugador solamente deberá pulsar el botón de Start verde que se encuentra bajo el display. Al conectar la batería, no es necesario pulsarlo ya que el juego comienza automáticamente.

Dicho botón también servirá para resetear el sistema y comenzar un nuevo juego cuando el jugador lo desee.



Ilustración 4. Usuario jugando con el sistema

3. Alimentación

Para evitar cableado exterior, el SIEC no necesita depender de una toma de corriente. Puede utilizarlo en cualquier lugar que desee, basta con conectar la batería adjunta en el juego.

Para ello, con ayuda del tirador blanco, abra la puerta trasera del sistema y conéctela al cable USB negro. Después, sitúe la batería en el lateral y cierre la puerta trasera como indica la Ilustración 5.



Ilustración 5. Pasos para la conexión de la batería

El juego estará encendido y listo para usarse desde el momento que conecte la batería. El sistema contiene un modo de ahorro de energía si no se presiona el botón de juego nuevo una vez pasados 2 minutos después de finalizar el último nivel.

4. Posibles problemas y soluciones

1- Se ha apagado la pantalla y al presionar los botones no se enciende nada.

Pruebe a resetear el sistema con el botón verde.

2- El sistema sigue sin encenderse una vez pulsado el botón Reset.

Es posible que la batería se haya agotado. Conéctela mediante un cargador con USB a la corriente durante al menos un par de horas.

3- El juego funciona pero la pantalla no muestra nada.

Compruebe que las conexiones de los cables en la Ilustración 6 del Anexo son las correctas.

4- El sistema parpadea constantemente sin necesidad de presionar ningún botón.

Es posible que se haya desconectado algún cable de un pulsador. Compruebe las conexiones con el Anexo.

5- Un LED permanece constantemente encendido.

Se necesita comprobar la conexión de los cables de los LEDs. Acuda al Anexo de este manual.

Anexo: Conexiones

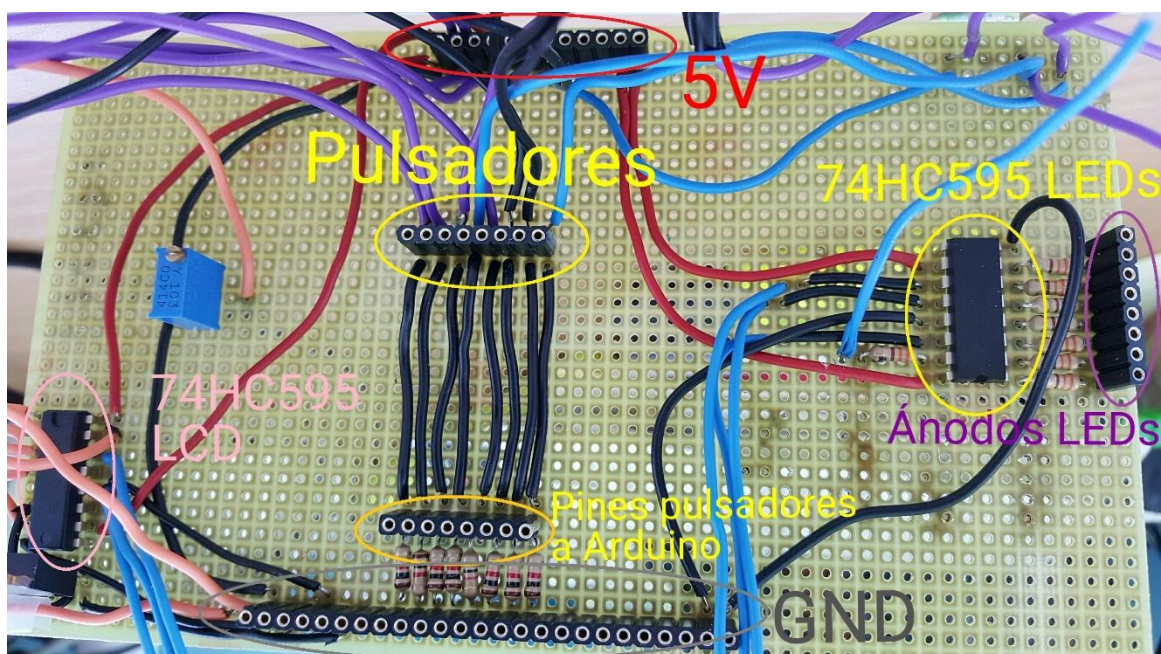


Ilustración 6. Conexiones placa de puntos

Cableado LEDs:

Cada LED dispone de 2 cables de colores, uno marcado con una cinta negra y el otro sin marcar. El primero de cada LED deberá soldarse donde se indica “Ánodos LEDs” en color morado. Los demás cables no marcados deberán ir soldados donde pone GND.

Cableado Pulsadores:

Asimismo, cada pulsador cuenta con 2 cables, donde uno de ellos también está marcado por una cinta negra. Estos cables deberán soldarse donde se indica en rojo 5V independientemente del número de pulsador. Los demás cables deberán soldarse, por orden de derecha a izquierda donde indica “Pulsadores”.

Así mismo, los cables que parten desde “Pines pulsadores a Arduino” deberán estar bien conectados a la placa Arduino, concretamente, y por este orden de izquierda a derecha, a los pines: 17, 16, 15, 7, 6, 5, 4 y 3.

Cableado Display:

Los cables azules que parten de donde indica “74HC595 LCD” en color rosa, están etiquetados con pequeños carteles numerados con “2” “11” y “12”. Deberán estar bien conectados en dichos pines de Arduino.

En caso de problema continuado, contactar con la autora de este manual.

A.2 Encuesta de satisfacción

La siguiente encuesta tiene como objetivo comprobar la utilidad del Sistema Interactivo de Estimulación Cognitiva creado.

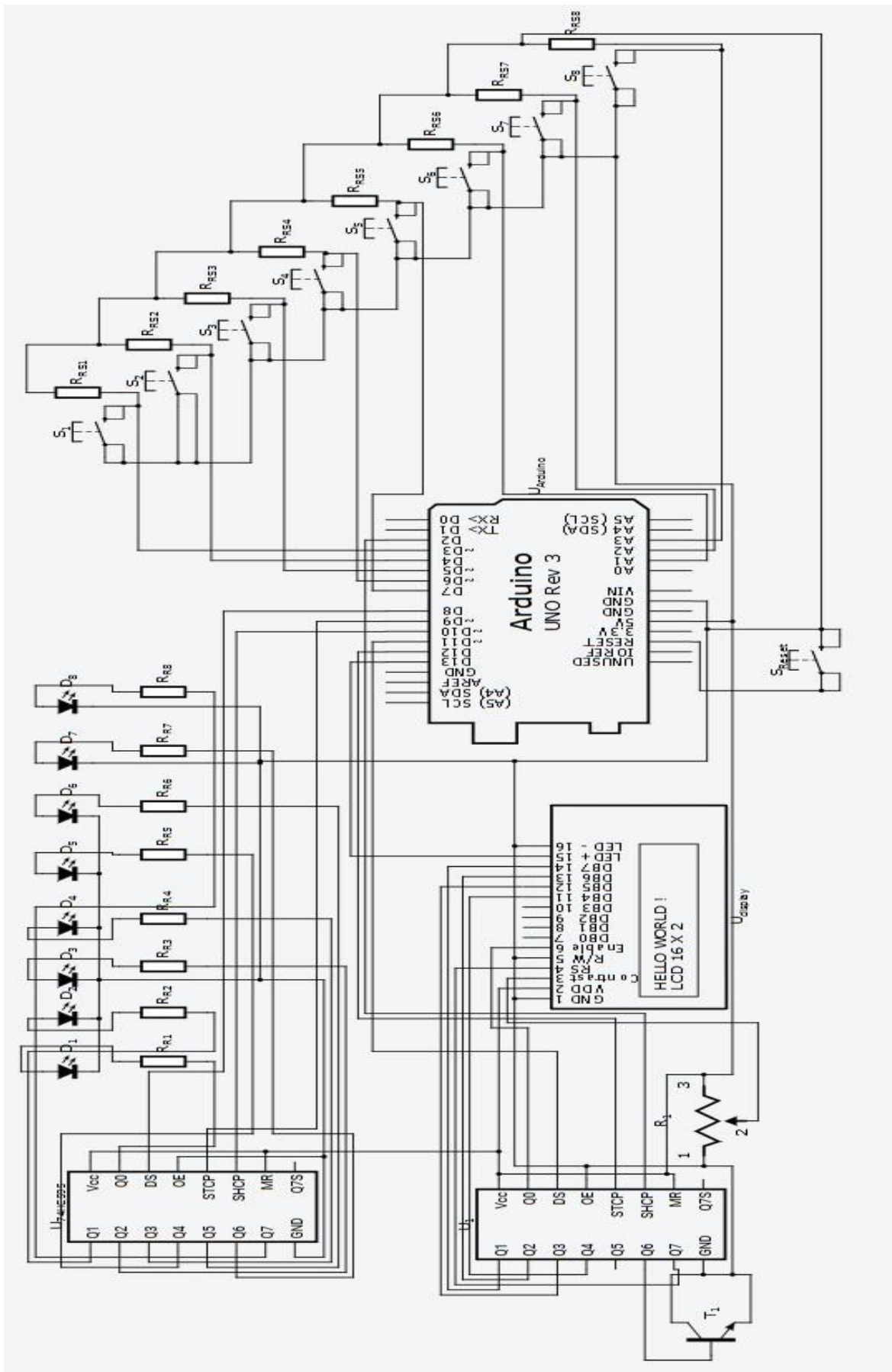
Dicha encuesta es totalmente anónima y servirá exclusivamente para fines académicos.

A continuación, responda a las siguientes preguntas valorando del 1 al 5, siendo **1 nada de acuerdo**, y **5 muy de acuerdo**.

Pregunta	Escala				
	Nada de acuerdo				Muy de acuerdo
1. He comprendido rápidamente el manejo del juego.	1	2	3	4	5
2. El tamaño de los botones y su distribución son adecuados para mí.	1	2	3	4	5
3. Percibo bien las luces cuando se iluminan.	1	2	3	4	5
4. El juego me ha resultado entretenido y quiero jugar más.	1	2	3	4	5
5. Me he cansado físicamente jugando o me he hecho daño.	1	2	3	4	5
6. Me gustaría utilizar más juegos como éste.	1	2	3	4	5

¡Muchas gracias por su participación!

A.3 Esquemático del circuito



A.4 Programa del microcontrolador

A continuación se muestra parte del código final. Las funciones propias *turno_jugador()* de cada nivel se han suprimido para simplificar el código.

```
#include <stdio.h>
#include <stdlib.h>
#include <LiquidCrystal595.h> //librería para el registro de desplazamiento en el display
#include <TimerOne.h> //librería para el contador de tiempo
#define rebote_n 5
#define num_leds_total 8
#define num_leds_actual 8
#define boton8 17 // pin analógico (A3)
#define boton7 16 // pin analógico (A2)
#define boton6 15 // pin analógico (A1)
#define boton5 7
#define boton4 6
#define boton3 5
#define boton2 4
#define boton1 3
#define number_of_74hc595s 1
#define numOfRegisterPins number_of_74hc595s * 8
typedef enum{ TRUE, FALSE, ERROR } eBoolean;
int vector_leds[num_leds_total]; // vector de LEDs
int vector_botones[num_leds_total]; // vector de botones
int nivel1[3], nivel2[6], nivel3[9], nivel4[12];
int nivel5[4]={1,3,5,7}; //LEDs pares
int ronda, nuevo, contador=0;
int nivel=1;
int ronda_jug_n01, ronda_jug_n02, ronda_jug_n03, ronda_jug_n04, ronda_jug_n05 = 0 ;
int retro=13;
int m, mu=0,md=0;
int s, su=0,sd=0;
int l, lu=0,ld=0,lc=0;
int mt, mut=0,mdt=0;
int st, sut=0,sdt=0;
int lt, lut=0,ldt=0,lct=0;
long int tiempo,inicio, inicio_total, tiempo_total;
LiquidCrystal595 lcd(11,12,2);
int SER_Pin = 8; int RCLK_Pin = 9; int SRCLK_Pin = 10;
boolean registers[numOfRegisterPins];

void setup(){
  Serial.begin(9600) ; // Se inicializa la conexión serie para posible debugging
  randomSeed(analogRead(0)); //lee ruido para crear la semilla de los números aleatorios
  pinMode(retro, OUTPUT); //se configura el pin de retroiluminación
```

```

digitalWrite(retro, HIGH);
Timer1.initialize(250000); // Dispara cada 250 ms para el contador de tiempo
Timer1.attachInterrupt(ISR_Count); //se define la interrupción que contará los 2 minutos
lcd.begin(16, 2); //se inicializa el display lcd, colocando el cursor en la posición a escribir
lcd.setCursor(2,0);
lcd.print("Nuevo juego ");
//se configuran los pines del registro de desplazamiento que controla los LEDs como salidas
pinMode(SER_Pin, OUTPUT);
pinMode(RCLK_Pin, OUTPUT);
pinMode(SRCLK_Pin, OUTPUT);
//resetea todos los pines del registro
clearRegisters();
writeRegisters();
//Configura los leds y los asigna en el vector_leds
for (int i = 0 ; i < num_leds_total ; i++){ vector_leds[i]=i;}
vector_botones[0] = boton1 ; // Asigna los pulsadores al vector de botones
vector_botones[1] = boton2 ;
vector_botones[2] = boton3 ;
vector_botones[3] = boton4 ;
vector_botones[4] = boton5 ;
vector_botones[5] = boton6 ;
vector_botones[6] = boton7 ;
vector_botones[7] = boton8 ;
// configura los pulsadores como entradas
for (int i = 0 ; i < num_leds_total ; i++){ pinMode( vector_botones[i], INPUT ) ; }
//Cascada de inicio del juego
for(int i=0; i<num_leds_actual; i++){
setRegisterPin(vector_leds[i], HIGH); //el pin 0 del registro es el LED1, el pin 1 es el LED2, etc.
writeRegisters();
delay(200);
setRegisterPin(vector_leds[i], LOW);
writeRegisters();
delay(200);
}
delay(1000);
juego_nuevo() ;
}

```

```

void clearRegisters(){ //Pone todos los pines del registro a LOW
for(int i = num_of_register_pins - 1; i >= 0; i--){
    registers[i] = LOW;
    setRegisterPin(vector_leds[i],LOW);
    writeRegisters();
}
}

```

```

void writeRegisters(){ //establece y muestra los registros
digitalWrite(RCLK_Pin, LOW);
for(int i = numOfRegisterPins - 1; i >= 0; i--){
digitalWrite(SRCLK_Pin, LOW);
int val = registers[i];
digitalWrite(SER_Pin, val);
digitalWrite(SRCLK_Pin, HIGH);
}
digitalWrite(RCLK_Pin, HIGH);
}

void setRegisterPin(int index, int value){ //pone un pin individual HIGH o LOW
registers[index] = value;
}

void juego_nuevo(){ //Crea las secuencias de los 5 niveles
ronda=0;
randomSeed(analogRead(0));
for(int i=0; i<3; i++){ nivel1[i]=random(0,num_leds_actual); }
for(int i=0; i<6; i++){ nivel2[i]=random(0,num_leds_actual); }
for(int i=0; i<9; i++){ nivel3[i]=random(0,num_leds_actual); }
for(int i=0; i<12; i++){ nivel4[i]=random(0,num_leds_actual); }
}

void brillo_led(int led){ //Se enciende y se apaga un LED individual
setRegisterPin(vector_leds[led], HIGH);
writeRegisters();
delay(500); //duración encendido
setRegisterPin(vector_leds[led], LOW);
writeRegisters();
delay(500); //duración apagado
}

void parpadeo_fallo(){ //PARPADEAN TODOS
for (int i = 0 ; i < num_leds_total ; i++){
setRegisterPin( vector_leds[i], HIGH ); writeRegisters(); }
delay(200) ;
for (int i = 0 ; i < num_leds_total ; i++){
setRegisterPin( vector_leds[i], LOW ) ; writeRegisters(); }
delay(200) ;
}

int nbotones_pulsados( int estado_b1, int estado_b2, int estado_b3, int estado_b4, int
estado_b5, int estado_b6, int estado_b7, int estado_b8){

```



```

int valor = TRUE ;
int numbotones = 0 ;
if ( estado_b1 == LOW ){ numbotones++ ; }
if ( estado_b2 == LOW ){ numbotones++ ; }
if ( estado_b3 == LOW ){ numbotones++ ; }
if ( estado_b4 == LOW ){ numbotones++ ; }
if ( estado_b5 == LOW ){ numbotones++ ; }
if ( estado_b6 == LOW ){ numbotones++ ; }
if ( estado_b7 == LOW ){ numbotones++ ; }
if ( estado_b8 == LOW ){ numbotones++ ; }
if ( numbotones > 1 ){ valor = TRUE ; }
else{ valor = FALSE ; }
return valor ;
}

```

```

int boton_pulsado(){
int rebote = ultimo_boton = boton_actual = 0 ;
int b1 = b2= b3= b4= b5= b6= b7= b8= HIGH ;
int varias_pulsaciones = FALSE ;
while (rebote != rebote_n){
    b1 = digitalRead(vector_botones[0]); b2 = digitalRead(vector_botones[1]);
    b3 = digitalRead(vector_botones[2]); b4 = digitalRead(vector_botones[3]);
    b5 = digitalRead(vector_botones[4]); b6 = digitalRead(vector_botones[5]);
    b7 = digitalRead(vector_botones[6]); b8 = digitalRead(vector_botones[7]);
    varias_pulsaciones = nbotones_pulsados( b1, b2, b3, b4, b5, b6, b7, b8 ) ;
    if (varias_pulsaciones==TRUE){ return -1 ; }
    if (LOW == b1){ boton_actual = 1 ; }
    else if (LOW == b2){ boton_actual = 2 ; }
    else if (LOW == b3){ boton_actual = 3 ; }
    else if (LOW == b4){ boton_actual = 4 ; }
    else if (LOW == b5){ boton_actual = 5 ; }
    else if (LOW == b6){ boton_actual = 6 ; }
    else if (LOW == b7){ boton_actual = 7 ; }
    else if (LOW == b8){ boton_actual = 8 ; }
    else{ boton_actual = 0 ; }
    if ((ultimo_boton == boton_actual) && (boton_actual != 0 )){ rebote++ ;}
    else{ ultimo_boton = boton_actual ; rebote = 0 ; }
}
return ultimo_boton;
}

```

```

int boton_cambio(int boton_inicial){
int b1 = b2 = b3 = b4 = b5 = b6 = b7 = b8 LOW ;
int valor = boton_inicial ;
int boton_liberado = rebote = 0 ;

```

```

int varias_pulsaciones = TRUE ;
int leído = 0;
while ( rebote != rebote_n ){
    b1 = digitalRead( boton1 ); b2 = digitalRead( boton2 );
    b3 = digitalRead( boton3 ); b4 = digitalRead( boton4 );
    b5 = digitalRead( boton5 ); b6 = digitalRead( boton6 );
    b7 = digitalRead( boton7 ); b8 = digitalRead( boton8 );
    varias_pulsaciones = nbotones_pulsados( b1, b2, b3, b4, b5, b6, b7, b8 );
    if(varias_pulsaciones == TRUE){ return -1 ; }
    switch (boton_inicial){
        case 1: leído = b1 ; break ;
        case 2: leído = b2 ; break ;
        case 3: leído = b3 ; break ;
        case 4: leído = b4 ; break ;
        case 5: leído = b5 ; break ;
        case 6: leído = b6 ; break ;
        case 7: leído = b7 ; break ;
        case 8: leído = b8 ; break ;
        default: return -1 ; break ;
    }
    if (leído == HIGH){ rebote++ ; }
}
return valor ;
}

```

```

int comprobar(int boton_detectado, int ronda){
    int valor = TRUE ;
    switch(nivel){
        case 1:
            if ( (boton_detectado -1) == nivel1[ronda]){ valor = TRUE ; }
            else{ valor = FALSE ; }
            break;
        case 2:
            if ((boton_detectado -1) == nivel2[ronda]){ valor = TRUE ; }
            else{ valor = FALSE ; }
            break;
        case 3:
            if ((boton_detectado -1) == nivel3[ronda]){ valor = TRUE ; }
            else{ valor = FALSE ; }
            break;
        case 4:
            if ((boton_detectado -1) == nivel4[ronda]){ valor = TRUE ; }
            else{ valor = FALSE ; }
            break;
        case 5:

```

```

        if ((boton_detectado -1) == nivel5[ronda]){ valor = TRUE ; }
        else{ valor = FALSE ; }
        break;
    }
return valor ;
}

int turno_jugador(){
    int button = 0 ;
    int valor = -1 ;
    while ((ronda_jug_n01 <= ronda) &&(valor != FALSE)){
        button = boton_pulsado() ;
        if (button == -1 ){ return FALSE ; }
        setRegisterPin(vector_leds[(button-1)], HIGH) ;
        writeRegisters();
        button = boton_cambio(button) ;
        if(button == -1){
            setRegisterPin(vector_leds[(button-1)], LOW);
            writeRegisters();
            return FALSE;
        }
        setRegisterPin(vector_leds[(button-1)], LOW);
        writeRegisters();
        if (TRUE == comprobar(button, ronda_jug_n01)){ ronda_jug_n01++ ; }
        else{ valor = FALSE ; }
    }
    if((ronda_jug_n01 - 1) == ronda){ valor = TRUE ; }
    else{ valor = FALSE ; }
    return valor ;
}

void contadores0(){
    contador=0; //para evitar overflow
    tiempo=0;
    m=mu=md=s=su=sd=l=lu=ld=lc=0;
}

void contadores_total0(){
    tiempo_total=0;
    mt=mut=mdt=st=sut=sdt=lt=lut=ldt=lct=0;
}

void imprimotiempolcd(){
    tiempo=millis()-inicio;
    m=(tiempo/1000)/60; mu=m%10; md=(m-mu)/10;
    s=(tiempo/1000)%60; su=s%10; sd=(s-su)/10;
}

```

```

l=(tiempo%1000); lu=l%10; ld=((l-lu)/10)%10;
lc=(l-(ld*10)-lu)/100;
lcd.setCursor(7,1); lcd.print(md);
lcd.setCursor(8,1); lcd.print(mu);
lcd.setCursor(9,1); lcd.print(":");
lcd.setCursor(10,1); lcd.print(sd);
lcd.setCursor(11,1); lcd.print(su);
lcd.setCursor(12,1); lcd.print(":");
lcd.setCursor(13,1); lcd.print(lc);
lcd.setCursor(14,1); lcd.print(ld);
lcd.setCursor(15,1); lcd.print(lu);
}

void imprimotiempolcd_total(){
    tiempo_total=millis()-inicio_total;
    mt=(tiempo_total/1000)/60; mut=mt%10; mdt=(mt-mut)/10;
    st=(tiempo_total/1000)%60; sut=st%10; sdt=(st-sut)/10;
    lt=(tiempo_total%1000); lut=lt%10; ldt=((lt-lut)/10)%10;
    lct=(lt-(ldt*10)-lut)/100;
    lcd.setCursor(4,1); lcd.print(mdt);
    lcd.setCursor(5,1); lcd.print(mut);
    lcd.setCursor(6,1); lcd.print(":");
    lcd.setCursor(7,1); lcd.print(sdt);
    lcd.setCursor(8,1); lcd.print(sut);
    lcd.setCursor(9,1); lcd.print(":");
    lcd.setCursor(10,1); lcd.print(lct);
    lcd.setCursor(11,1); lcd.print(ldt);
    lcd.setCursor(12,1); lcd.print(lut);
}

void ISR_Count(){ contador++; }

void loop(){
    int i, j=0 ;
    int pasa_ronda = FALSE ;
    if (nuevo==0){
        juego_nuevo();
        delay(1000);
        contadores_total0(); contadores0();
        switch(nivel){
            case 1:
                inicio=millis();
                inicio_total=millis();
                lcd.clear(); lcd.setCursor(3,0); lcd.print("Nivel 1");
                for(i=j; i<3; i++){ brillo_led(nivel1[i]);

```

```

pasa_ronda=turno_jugador();
if ( TRUE == pasa_ronda ){ //Acierto
    ronda++;
    if(ronda==3){ nivel++; ronda=0; ronda_jug_n01=0; j=0;
        for (i=0 ; i < 2 ; i++){
            for (int j = 0 ; j < num_leds_total ; j++){
                setRegisterPin(vector_leds[j], HIGH);
                writeRegisters();
                delay(100) ;
            }
            for (int j = 0 ; j < num_leds_total ; j++){
                setRegisterPin(vector_leds[j], LOW);
                writeRegisters();
                delay(100) ;
            }
        }
        lcd.setCursor(0,1); lcd.print("Tiempo:");
        imprimotiempo(); imprimotiempolcd();
    }
}
else{
    lcd.clear(); lcd.setCursor(3,0); lcd.print("Fallo");
    for (i=0 ; i<3 ; i++){ parpadeo_fallo() ;}
    pasa_ronda = FALSE ;
    ronda_jug_n01=0; ronda=0; nivel=1; j=0;
    juego_nuevo() ;
}
delay(1000);
}
break;

case 2:
    contadores0();
    inicio=millis();
    lcd.clear(); lcd.setCursor(3,0); lcd.print("Nivel 2");
    for(i=0; i<3; i++){
        brillo_led(nivel2[j]);
        delay(200); //Velocidad muestra de secuencia
        brillo_led(nivel2[j+1]);
        pasa_ronda=turno_jugador2();
        if ( TRUE == pasa_ronda ){
            ronda++;
            j=j+2;

```

```

        if(ronda==3){ nivel++; ronda=0; ronda_jug_n02=0;j=0;
            for (i=0 ; i < 2 ; i++){
                for (int j = 0 ; j < num_leds_total ; j++){
                    setRegisterPin(vector_leds[j], HIGH);
                    writeRegisters();
                    delay(100) ;
                }
                for (int j = 0 ; j < num_leds_total ; j++){
                    setRegisterPin(vector_leds[j], LOW);
                    writeRegisters();
                    delay(100) ;
                }
            }
            lcd.setCursor(0,1); lcd.print("Tiempo:");
            imprimotiempo(); imprimotiempolcd();
        } //fin ronda 3
    }
    else{
        lcd.clear(); lcd.setCursor(3,0); lcd.print("Fallo");
        for (i=0 ; i<3 ; i++){parpadeo_fallo() ;}
        pasa_ronda = FALSE ;
        ronda_jug_n02=0; ronda=0; nivel=2; j=0;
        juego_nuevo() ;
        break;
    }
    delay(1000);
}
break;

case 3:
    contadores0();
    inicio=millis();
    lcd.clear(); lcd.setCursor(3,0); lcd.print("Nivel 3");
    for(i=0; i<3; i++){
        brillo_led(nivel3[j]);
        delay(200);
        brillo_led(nivel3[j+1]);
        delay(200);
        brillo_led(nivel3[j+2]);
        pasa_ronda=turno_jugador3();
        if ( TRUE == pasa_ronda ){
            ronda++; j=j+3;
            if(ronda==3){ nivel++; ronda=0; ronda_jug_n03=0;
                for (i=0 ; i < 2 ; i++){
                    for (int j = 0 ; j < num_leds_total ; j++){

```

```

        setRegisterPin(vector_leds[j], HIGH);
        writeRegisters();
        delay(100);
    }
    for (int j = 0 ; j < num_leds_total ; j++){
        setRegisterPin(vector_leds[j], LOW);
        writeRegisters();
        delay(100);
    }
}
lcd.setCursor(0,1); lcd.print("Tiempo:");
imprimotiempo(); imprimotiempolcd();
}
}
else{
    lcd.clear(); lcd.setCursor(3,0); lcd.print("Fallo");
    for (i=0 ; i<3 ; i++){parpadeo_fallo() ;}
    pasa_ronda = FALSE ;
    ronda_jug_n03=0; ronda=0; j=0; nivel=3;
    juego_nuevo() ;
}
delay(1000);
}
break;

case 4:
    contadores0(); inicio=millis();
    lcd.clear(); lcd.setCursor(3,0); lcd.print("Nivel 4");
    for(i=0; i<3; i++){
        brillo_led(nivel4[j]); delay(200);
        brillo_led(nivel4[j+1]); delay(200);
        brillo_led(nivel4[j+2]); delay(200);
        brillo_led(nivel4[j+3]);
        pasa_ronda=turno_jugador4();
        if ( TRUE == pasa_ronda ){

            ronda++; j=j+4;
            if(ronda==3){
                nivel++; ronda=0; ronda_jug_n04=0;
                for (i=0 ; i < 2 ; i++){
                    for (int j = 0 ; j < num_leds_total ; j++){
                        setRegisterPin(vector_leds[j], HIGH);
                        writeRegisters();
                        delay(100);
                    }
                }
            }
        }
    }
}

```



```

        for (int j = 0 ; j < num_leds_total ; j++){
            setRegisterPin(vector_leds[j], LOW);
            writeRegisters();
            delay(100) ;
        }
    }
    lcd.setCursor(0,1); lcd.print("Tiempo:");
    imprimotiempo(); imprimotiempolcd();
}
}
else{
    lcd.clear(); lcd.setCursor(3,0); lcd.print("Fallo");
    for (i=0 ; i<3 ; i++){parpadeo_fallo() ;}
    pasa_ronda = FALSE ;
    ronda_jug_n04=0;
    ronda=0; nivel=4; j=0;
    juego_nuevo() ;
}
delay(1000);
}
break;

```

case 5:

```

    contadores0(); inicio=millis();
    lcd.clear(); lcd.setCursor(1,0); lcd.print("Nivel 5: Pares");
    for(int k=0; k<4; k++){ brillo_led(nivel5[k]); delay(200); }
    pasa_ronda=turno_jugador5();
    if ( TRUE == pasa_ronda ){
        ronda=0; ronda_jug_n05=0;
        nuevo=1; contador=0;
        Timer1.initialize(250000);
        interrupts();
        for (i=0 ; i < 2 ; i++){
            for (int j = 0 ; j < num_leds_total ; j++){
                setRegisterPin(vector_leds[j], HIGH);
                writeRegisters();
                delay(100) ;
            }
            for (int j = 0 ; j < num_leds_total ; j++){
                setRegisterPin(vector_leds[j], LOW);
                writeRegisters();
                delay(100) ;
            }
        }
    }
}

```

```

        delay(1000);
        lcd.setCursor(0,1); lcd.print("Tiempo:");
        imprimotiempolcd();
    }
    else{
        lcd.clear(); lcd.setCursor(3,0); lcd.print("Fallo");
        for (i=0 ; i<3 ; i++){parpadeo_fallo() ;}
        pasa_ronda = FALSE ;
        ronda_jug_n05=0; ronda=0;
        nivel=5;
    }
    delay(2000);
    lcd.clear(); lcd.setCursor(2,0); lcd.print("Tiempo total");
    imprimotiempolcd_total();
    break;
}
}
if((nuevo==1)&&(contador>=480)){ //2min de espera y si no se ha pulsado el botón de start
    contador=0;          //se apaga la retroiluminación
    lcd.clear();
    digitalWrite(retro, LOW);
}
}
}

```

A.5 Hojas de características (Datasheets)

LEDs blancos:

http://www.farnell.com/datasheets/709104.pdf?_ga=1.101891713.1433225393.1458932474

Pulsadores eléctricos:

http://www.farnell.com/datasheets/1839888.pdf?_ga=1.59924013.1433225393.1458932474

Display LCD:

http://www.farnell.com/datasheets/2021752.pdf?_ga=1.262304652.1433225393.1458932474

Registro de desplazamiento 74HC595:

http://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf

Arduino Uno:

http://www.farnell.com/datasheets/1682209.pdf?_ga=1.96073628.1433225393.1458932474