



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Diseño del firmware del panel electrónico para la enseñanza de operaciones matemáticas básicas a niños con capacidad visual especial.

TRABAJO DE TITULACIÓN.

AUTORES: Figueroa Agurto, Cinthya Adriana
Valarezo Armijos, Jhuly Raquel

DIRECTOR: Jaramillo Pacheco, Jorge Luis, Ing.

LOJA-ECUADOR

2016



Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

Septiembre, 2016

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Ingeniero.

Jorge Luis Jaramillo Pacheco.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: "Diseño del firmware del panel electrónico para la enseñanza de operaciones matemáticas básicas a niños con capacidad visual especial", realizado por Figueroa Agurto Cinthya Adriana y Valarezo Armijos Jhuly Raquel ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, marzo de 2016

f)

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Nosotras, Figueroa Agurto Cinthya Adriana y Valarezo Armijos Jhuly Raquel declaramos ser las autoras del presente trabajo de titulación: "Diseño del firmware del panel electrónico para la enseñanza de operaciones matemáticas básicas a niños con capacidad visual especial", de la Titulación de Ingeniería en Electrónica y Telecomunicaciones siendo el Ingeniero Jorge Luis Jaramillo Pacheco Director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: "Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad"

f

Autor: Figueroa Agurto Cinthya Adriana

Cédula: 1105665192

f

Autor: Valarezo Armijos Jhuly Raquel

Cédula: 115008484

DEDICATORIA

El presente trabajo dedico especialmente a Dios, por brindarme diariamente la sabiduría para cumplir mis metas, especialmente por permitirme terminar con éxito mi Titulación Universitaria y guiarme por el camino correcto para realizarlo. A mis Padres, Héctor y Rosita, por darme su amor y apoyo incondicional durante toda mi vida, especialmente en esta etapa de vida universitaria. A mis hermanos que de una u otra forma han sido un pilar fundamental en mi formación. Y a mis compañeros, y más que todo amigos, con los que compartí momentos únicos y la vivencia de muchas anécdotas a lo largo de mi formación académica.

Cinthya Adriana

A mis padres, José y Raquel, por estar siempre a mi lado, ayudándome en todas las etapas de mi vida, por guiarme, y ser ese apoyo incondicional que me alentaba a seguir adelante. A mis hermanos, José, Jenny y Carlos, por su apoyo y sus cuidados y sobre todo su amor. A mi sobrina, Paula porque desde que llego a nuestras vidas nos ha llenado de alegrías. A mis sobrinos, José Antonio e Ismael, que llenan de felicidad mi vida. A esa persona especial que ha sido una parte indispensable en mi vida. Y sin duda a mis amigas, amigos y compañeros con los que he compartido inolvidables momentos.

Jhuly Raquel

AGRADECIMIENTO

Queremos agradecer principalmente a Dios, por permitirnos culminar con éxito nuestra formación profesional.

A nuestros Padres, por brindarnos siempre su amor y apoyo incondicional en toda nuestra formación universitaria, por haber confiado y creído en nuestra capacidad para cumplir nuestra meta. De igual manera a nuestros hermanos y familiares, que de una u otra forma contribuyeron positivamente en nuestra formación durante estos años universitarios.

Así mismo a nuestros compañeros que con el paso de los años llegaron a convertirse en grandes amigos, gracias por todo lo que convivimos en las aulas, y por las experiencias que se convirtieron en muchas anécdotas inolvidables, y sobre todo, por su ayuda y apoyo durante estos cinco años de vida universitaria.

A nuestro Director de Tesis, el Ingeniero Jorge Luis Jaramillo, porque a más de ser un guía, para cumplir exitosamente con la culminación de este trabajo de Fin de Titulación, fue un gran Docente y amigo que siempre supo brindarnos sus conocimientos y ayuda desinteresada. Y finalmente queremos agradecer a todos los Docentes de la Titulación por haber dedicado su tiempo en nuestra formación profesional.

Cinthyra Adriana y Jhuly Raquel

ÍNDICE DE CONTENIDOS

| | |
|---|------|
| APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN..... | ii |
| DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS..... | iii |
| DEDICATORIA | iv |
| AGRADECIMIENTO | v |
| ÍNDICE DE CONTENIDOS..... | v |
| LISTA DE FIGURAS | viii |
| LISTA DE TABLAS..... | x |
| RESÚMEN..... | 1 |
| ABSTRACT | 2 |
| INTRODUCCIÓN..... | 3 |
| CAPÍTULO I..... | 4 |
| 1. A MANERA DE RESUMEN SOBRE LA OPTIMIZACIÓN DEL PANEL ORIGINAL | 4 |
| 1.1.Sobre la necesidad de optimizar el diseño original..... | 5 |
| 1.2. Requerimientos generales para el diseño del nuevo panel..... | 10 |
| 1.3.Diseño a detalle de las fichas a utilizar en el panel | 13 |
| 1.4.Programación de la ingeniería de detalle del panel..... | 14 |
| CAPITULO II | 15 |
| 2. DISEÑO E IMPLEMENTACIÓN DEL FIRMWARE | 15 |
| 2.1.Metodología de desarrollo del firmware..... | 16 |
| 2.2.Fase de definición de requisitos de diseño | 16 |
| 2.3.Fase de diseño lógico..... | 17 |
| 2.3.1. Identificación de fichas | 17 |
| 2.3.2. Ejecución de operaciones | 22 |
| 2.3.3. Comprobación de respuesta..... | 22 |
| 2.4.Selección de la plataforma tecnológica a utilizar..... | 24 |

| | |
|---|----|
| 2.5.Fase de implementación del firmware | 24 |
| 2.5.1. Codificación | 25 |
| 2.5.2. Pruebas | 25 |
| 2.6.Fases de verificación y mantenimiento | 27 |
| CONCLUSIONES | 28 |
| BIBLIOGRAFÍA | 30 |
| ANEXOS | 31 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1.1 Versión original del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales | 5 |
| Figura 1. 2 Primera opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales..... | 6 |
| Figura 1.3. Segunda opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales..... | 6 |
| Figura 1.4. Tercera opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales..... | 7 |
| Figura 1.5. Cuarta opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales..... | 7 |
| Figura 1.6. Quinta opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales..... | 8 |
| Figura 1.7. Sexta opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales..... | 9 |
| Figura 1.8. Tablero del panel electrónico optimizado..... | 10 |
| Figura 1.9. Vista superior y lateral izquierda del tablero..... | 11 |
| Figura 1.10. Vista general de una ficha del tablero..... | 11 |
| Figura 1.11 Vista lateral derecha de una pieza LEGO 4 x 2 | 12 |
| Figura 1. 12. Vista inferior de una pieza de LEGO 4 x 2 | 12 |
| Figura 1.13. Vista lateral izquierda de una pieza de LEGO 4 x 2..... | 12 |
| Figura 1.14. Vista general del tablero y de las fichas..... | 13 |
| Figura 1.15. Vista de planta de ficha seleccionada..... | 14 |
| Figura 2.1. Fases del método de desarrollo de software en cascada..... | 16 |
| Figura 2.2. Asignación de variables a las posiciones posibles de las fichas en el panel..... | 17 |
| Figura 2.3. Esquema de funcionamiento de la solución basada en sensores infrarrojos para la identificación de fichas..... | 18 |
| Figura. 2.4. Flujograma de las operaciones lógicas en el firmware para el reconocimiento de la ubicación de la ficha..... | 20 |
| Figura. 2.5 Flujograma de las operaciones lógicas en el firmware para el reconocimiento del valor numérico de la ficha..... | 21 |

| | |
|---|----|
| Figura. 2.6. Flujograma de las operaciones lógicas en el firmware para la función de ejecución de operaciones. | 22 |
| Figura. 2.7. Flujograma de las operaciones lógicas en el firmware para la función presentación de respuesta. | 23 |
| Figura. 2.8. Rutina de generación de tonos musicales..... | 23 |
| Figura. 2.9 Protocolo de respuesta correcta para la identificación de fichas y realización de operación. | 26 |
| Figura. 2.10. Protocolo de respuesta incorrecta para la identificación de fichas y realización de operación..... | 26 |
| Figura. 2.11. Señalización de una respuesta correcta en el bloque de señalización luminosa. | 27 |
| Figura. 2.12. Señalización de una respuesta incorrecta en el blouqe de señalización luminosa. | 27 |

LISTA DE TABLAS

| | |
|---|----|
| Tabla 1.1 Análisis comparativo de las opciones presentadas. | 9 |
| Tabla 2.1. Asignación de variables a las posiciones de las fichas en tablero. | 18 |
| Tabla 2.2. Equivalente decimal y binario de los estados de los sensores infrarrojos utilizados para la identificación de fichas. | 19 |
| Tabla 2.3. Número de entradas y salidas necesarias para la operación del firmware. | 24 |
| Tabla 2.4. Características técnicas del microcontrolador Arduino Mega 2560. | 24 |
| Tabla 2.5. Asignación de los pines de la plataforma Arduino a cada una de las variables identificadas en el diseño lógico del firmware. | 25 |

RESÚMEN

Entre septiembre de 2013 y abril de 2015, en la Universidad Técnica Particular de Loja, equipos de trabajo conformados en la Sección de Electrónica y Telecomunicaciones del Departamento de Ciencias de la Computación y Electrónica, apoyaron a personal de la Sección de Diseño, Proyectos Arquitectónicos y Urbanismo del Departamento de Arquitectura y Artes, en el diseño, implementación y optimización de un panel electrónico para la enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales. La evaluación de desempeño de un primer panel desarrollado, mostró correspondencia con las especificaciones funcionales pero encontró también problemas operativos. Con la intención de superar los problemas, se planteó la necesidad de optimizar el panel, incluyendo una etapa de diseño e implementación de firmware que soporte el reconocimiento de fichas, la operatividad matemática y la interacción con el usuario. El firmware se desarrolló aplicando una metodología en cascada, utilizando los recursos de software y hardware de Arduino.

PALABRAS CLAVES: *enseñanza de matemáticas, enseñanza de matemáticas a niños con capacidad visual especial, paneles electrónico, LEGO, Arduino MEGA.*

ABSTRACT

During September 2013 and April 2015, in the Technical University of Loja, teams formed in the Electronics and Telecommunications sections of Department of the Computer Science and Electronics, supported to staff of the Design, Architectural projects and Urban sections of the Architecture and Arts department, in the design, implementation and optimization of an electronic panel for the teaching of basic mathematical operations to children with special visual capabilities. The evaluation of performance the first developed panel, showed correspondence with the functional specifications but also found operational problems. With the intention of overcoming the problems, raised the need to optimize the panel, including a stage of design and implementation of firmware that supports recognition of chips, mathematical operability and interaction with the user. The firmware was developed using a methodology in waterfall, using the resources of software and hardware of Arduino.

KEYWORDS: math teaching, teaching basic math to children with special visual capacity, electronic panels, LEGO, Arduino MEGA.

INTRODUCCIÓN

En septiembre de 2013, personal de la Sección de Electrónica y Telecomunicaciones del Departamento de Ciencias de la Computación y Electrónica, y de la Sección de Diseño, Proyectos Arquitectónicos y Urbanismo del Departamento de Arquitectura y Artes, de la Universidad Técnica Particular de Loja, decidieron colaborar en el diseño e implementación de un panel electrónico para la enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales. Como resultado, se desarrolló el panel electrónico que se muestra en la Figura.1.1, y que se describe en [1].

La evaluación de desempeño del panel desarrollado mostró que se cumplía con las especificaciones funcionales, pero se presentaba problemas operativos, especialmente en relación a la manipulación de las fichas. En este contexto, en marzo 2014 se acordó optimizar el panel, obteniendo los resultados descritos en trabajos previos [2], [3].

En Abril del 2015, como resultado de un trabajo de titulación en Ingeniería en Electrónica y Telecomunicaciones en la UTPL [4], se presentó el diseño e implementación del bloque de identificación electrónica de fichas tipo LEGO. Luego, dando continuidad a la programación del proyecto, se inició la fase de diseño del firmware del panel electrónico. En este documento se recoge los resultados obtenidos en esta fase.

En el primer capítulo se presenta, a manera de resumen, los resultados del análisis de diversas opciones tecnológicas para la optimización del panel original, y se muestra los resultados obtenidos en la fase de conceptualización de un nuevo panel, y el tipo de fichas a utilizar.

En el segundo capítulo se describe los resultados obtenidos en la etapa de diseño e implementación del firmware para el funcionamiento lógico del prototipo electrónico para la enseñanza de matemáticas básicas a niños con capacidad visual especial.

CAPÍTULO I

1. A MANERA DE RESUMEN SOBRE LA OPTIMIZACIÓN DEL PANEL ORIGINAL

1.1. Sobre la necesidad de optimizar el diseño original

Debido a los problemas presentados en el panel implementado (ver Figura 1.1), se identificaron necesidades de optimización: disminución del volumen y peso de las fichas, integración de identificadores decimales y código Braille, reducción del desgaste de contactos en el circuito impreso, y diversificación en el uso de colores [1].

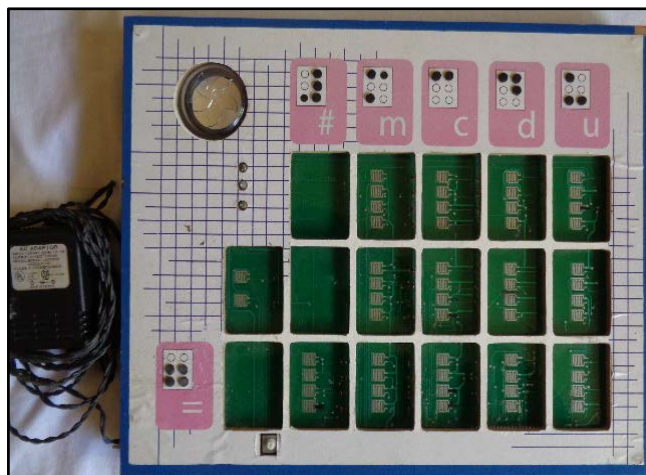


Figura 1.1 Versión original del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales

Fuente: [1]

Conocidas las necesidades de optimización, se formularon 6 diferentes propuestas tecnológicas. En la **primera opción** se propuso conservar las características de funcionamiento del panel original e incorporar un switch de encendido y apagado. El usuario ingresa los operandos presionando los pulsadores ubicados en el tablero, tal como se muestra en la Figura 1.2. Luego, selecciona el operador matemático, e indica la respuesta expresada en unidades, decenas y centenas. Al oprimir el operador “igual” y la tecla “comprobar”, se escucha el mensaje “correcto” para una operación válida, o “incorrecto” para una operación errónea. El usuario puede resetear los pulsadores, para ejecutar la operación nuevamente.

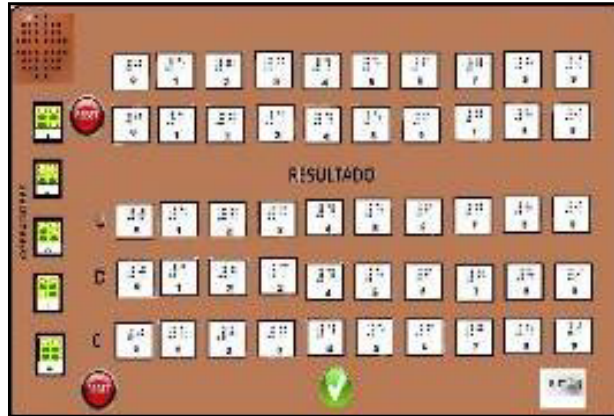


Figura 1. 2 Primera opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales.

Fuente: Autores.

En la **segunda opción**, se pretende llamar la atención de los niños para facilitar el aprendizaje, por lo que se utiliza un panel electrónico integrado a un oso de peluche. El panel opera de forma similar a lo descrito en el apartado anterior (ver Figura 1.3).

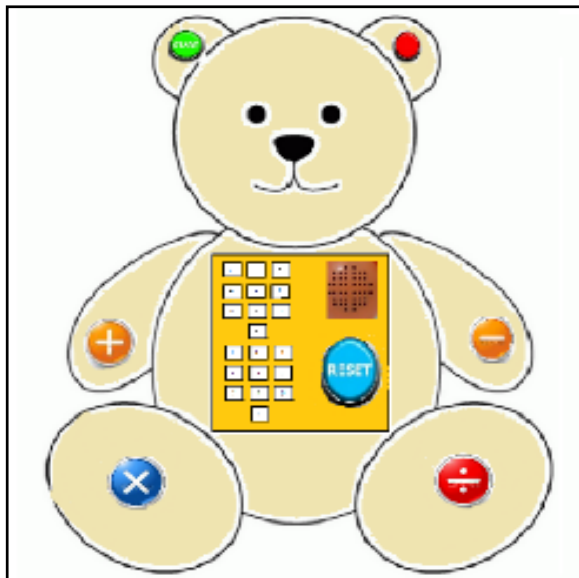


Figura 1.3. Segunda opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales.

Fuente: Autores.

En la **tercera opción** se presenta un paradigma diferente al panel original, en el que dos jugadores participan identificándose cada uno de ellos con el color verde o azul respectivamente. Se utiliza un tablero con pulsadores (que contienen números entre uno y veinte), y tres dados (dos para los operandos y uno para las operaciones). El tablero y los

dados están identificados con código braille. El jugador lanza los dados. El procedimiento y el resultado se ingresa en el tablero, y, se verifica la correcta ejecución (ver Figura 1.4).



Figura 1.4. Tercera opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales.

Fuente: Autores.

En la **cuarta opción**, para iniciar se presiona el botón "inicio", y el panel da a conocer, mediante audio, un número aleatorio. El usuario identifica dos operandos y una operación, que se ingresan al panel presionando los pulsadores del tablero. Si los operandos y la operación ingresada son correctos, se escuchará "correcto", caso contrario se indicará "incorrecto", y se generará un nuevo número aleatorio (ver Figura 1.5).



Figura 1.5. Cuarta opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales.

Fuente: Autores.

La **quinta opción** se basa en el funcionamiento del juego “sudoku”, adaptándolo con numeración Braille (ver Figura 1.6). Este juego utiliza una tabla de 9x9, compuesta por bloques denominados también subtablas de 3x3. Algunas celdas contienen números dados. El juego tiene por objetivo completar las celdas vacías, de forma tal que cada columna, fila, y región contenga números del 1 al 9 una sola vez. El tablero contendrá los botones “empezar” y “evaluar”. La rutina de comprobación terminará con un mensaje de “correcto” o “incorrecto”.

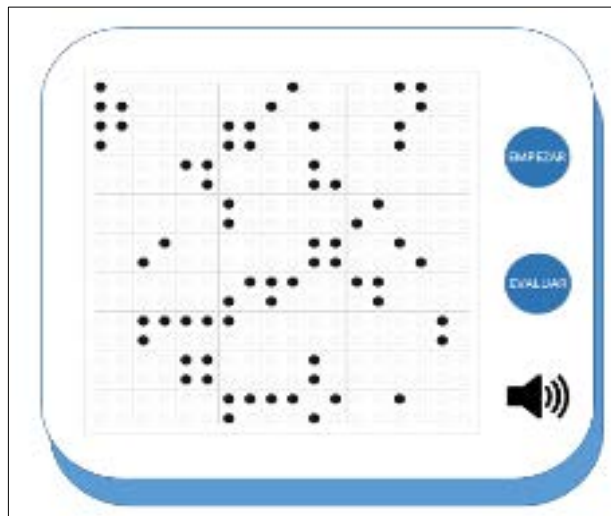


Figura 1.6. Quinta opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales.

Fuente: Autores.

En la **sexta opción** se adapta fichas tipo LEGO al panel electrónico, con lo que se pretende reducir el desgaste prematuro de los contactos en los circuitos impresos. Se utiliza un tablero y diversas fichas (números y operaciones matemáticas). Las fichas se colocan en el tablero en la lógica: primer operando, segundo operando, operación, solución, y, comprobación (ver Figura 1.7).

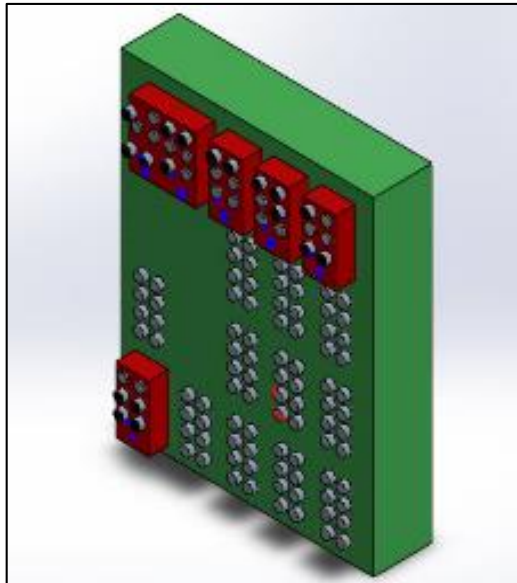


Figura 1.7. Sexta opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales.

Fuente: Autores.

Las opciones señaladas se analizaron identificando potenciales inconvenientes de orden mecánico, y limitaciones en el número de cifras significativas de operandos y resultados. La Tabla 1.1 muestra las ventajas y desventajas de cada una de las opciones. Como resultado se seleccionó la **sexta opción**, considerando sobre todo la ampliación de las potencialidades en la enseñanza de las matemáticas que esta brinda.

Tabla 1.1 Análisis comparativo de las opciones presentadas.

| Opción | Ventajas | Desventajas |
|--------|---|--|
| 1 | <ul style="list-style-type: none"> • Elementos no móviles • Se puede verificar que pulsadores están activados | <ul style="list-style-type: none"> • No se puede activar 2 pulsadores simultáneamente • Desgaste prematuro • Sistema mecánico complejo • Gran tamaño |
| 2 | <ul style="list-style-type: none"> • Elementos no móviles | <ul style="list-style-type: none"> • Desgaste prematuro • Sistema mecánico complejo |
| 3 | <ul style="list-style-type: none"> • Desgaste limitado | <ul style="list-style-type: none"> • Operación compleja |
| 4 | <ul style="list-style-type: none"> • Desgaste limitado | <ul style="list-style-type: none"> • Operación compleja |
| 5 | <ul style="list-style-type: none"> • Desgaste limitado | <ul style="list-style-type: none"> • Sistema complejo • Operación compleja |
| 6 | <ul style="list-style-type: none"> • Desgaste limitado • Manipulación intuitiva | <ul style="list-style-type: none"> • Operación compleja |

Fuente: Autores

Las fichas correspondientes a los operandos y al resultado, podrán ocupar exclusivamente posiciones predeterminadas en el tablero. La Figura 1.9 muestra el diseño del tablero, que corresponde al estándar de una pieza tipo LEGO.

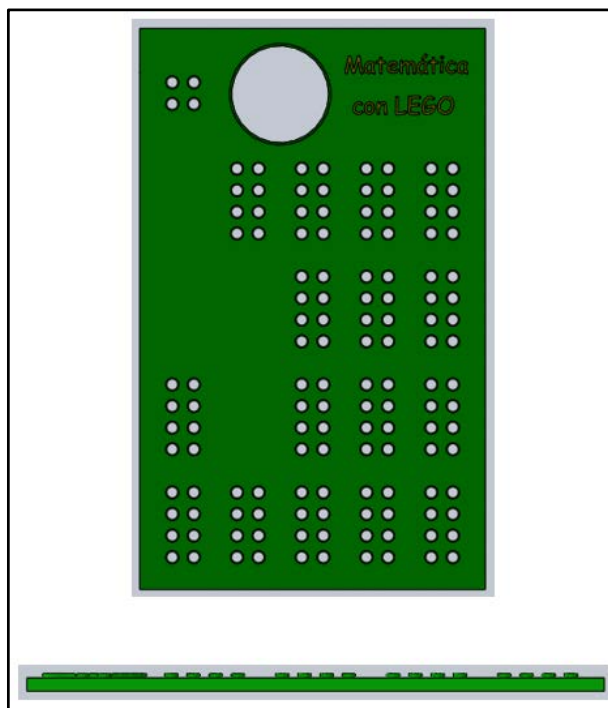


Figura 1.9. Vista superior y lateral izquierda del tablero.

Fuente: Autores

En cada ficha de los operandos, se imprimirá el valor absoluto de la misma (de 0 a 9), en Braille (en la parte superior de la ficha) y en números arábigos (en la parte inferior de la ficha), tal como lo muestra la Figura 1.10.

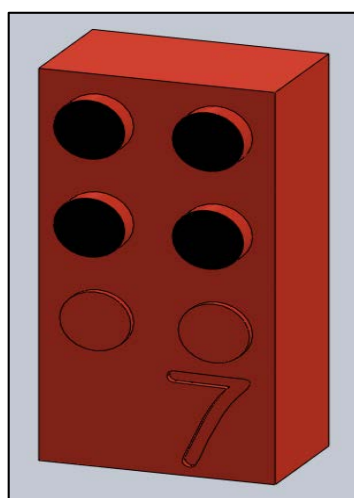


Figura 1.10. Vista general de una ficha del tablero.

Fuente: Autores

La geometría de cada una de las fichas corresponde al estándar de una pieza LEGO (ver Figura 1.11 a la 1.13).

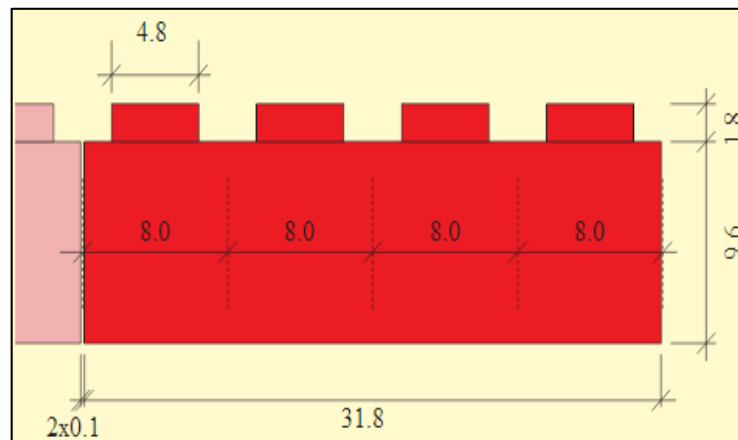


Figura 1.11 Vista lateral derecha de una pieza LEGO 4 x 2
Fuente: [7]

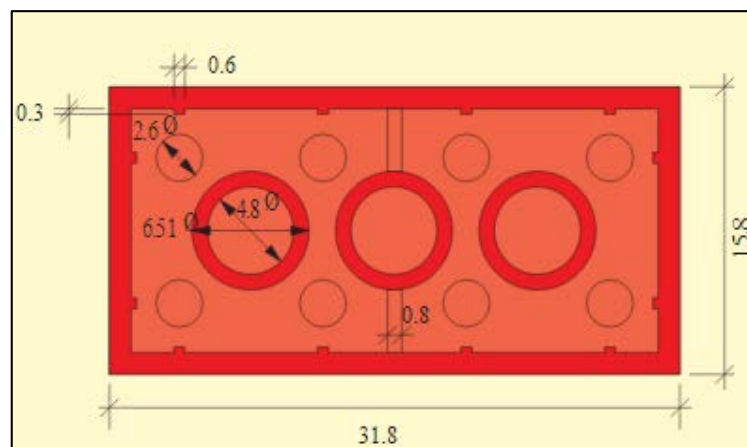


Figura 1.12. Vista inferior de una pieza de LEGO 4 x 2
Fuente: [7]

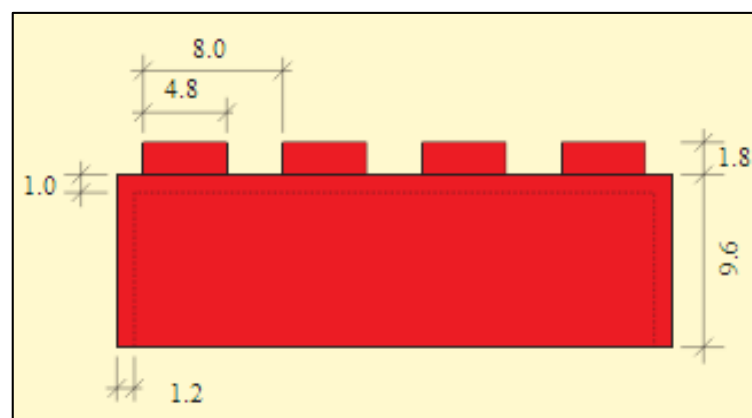


Figura 1.13. Vista lateral izquierda de una pieza de LEGO 4 x 2
Fuente: [7]

Adicionalmente, se diseñó fichas para las operaciones matemáticas básicas, y, para el reconocimiento de las posiciones. La Figura 1.14 muestra el ensamblaje de las fichas sobre el tablero.

En otro contexto, al beneficio que brinda las fichas tipo LEGO (múltiples combinaciones y las distintas estructuras que se logran crear con un grupo básico de piezas), se suma el desarrollo de habilidades motrices y la adquisición de conceptos espaciales, como volumen, tamaño y formas geométricas [8]. Los infantes se sienten atraídos por la idea de aprender divirtiéndose, condición totalmente válida para niños con capacidad visual especial, esta opción se propone adaptar fichas tipo lego en el panel electrónico. Al utilizar las fichas tipo lego se pretende reducir el desgaste prematuro de los contactos en los circuitos impresos en el bloque de control, de tal forma que el panel tenga durabilidad.

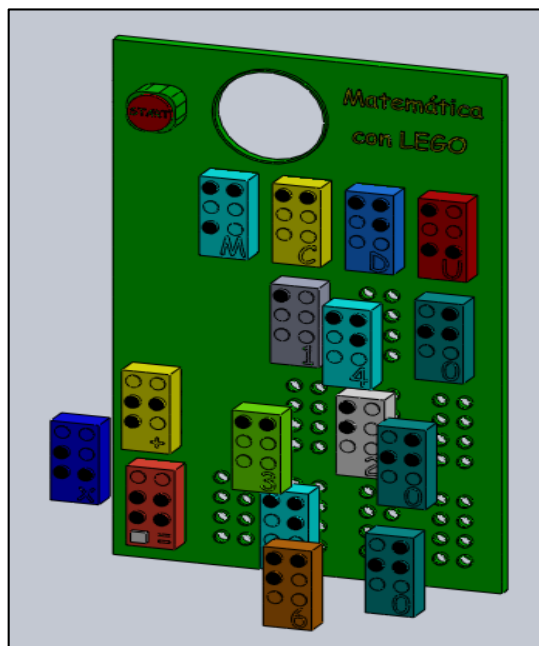


Figura 1.14. Vista general del tablero y de las fichas.
Fuente: Autores

1.3. Diseño a detalle de las fichas a utilizar en el panel

Considerando el grado de desarrollo de la capacidad táctil en el grupo beneficiario del proyecto, y, los requerimientos de estética planteados por el equipo de trabajo, se decidió plantear al menos 4 posibles diseños de fichas, de entre las cuales se escogería la opción óptima.

Debido a la disponibilidad de software y equipos, se decidió imprimir las fichas en 3D, utilizando los recursos del aplicativo Solidworks [9], y, de la impresora 3D tipo Makerbot Replicator 2X [10].

Para seleccionar la opción óptima del diseño de las fichas a utilizar en el panel, se decidió someter las impresiones al análisis del grupo beneficiario del proyecto. Se trabajó con los niños del cuarto año de educación básica del Instituto para Ciegos “Byron Eguiguren”. Este grupo está conformado por 6 menores, de los cuales 3 tienen capacidad visual especial. A cada uno de los niños se le pidió manipular las diferentes fichas impresas, recogiendo sus impresiones. Adicionalmente, se solicitó una apreciación a la tutora del grupo, una docente con 15 años de experiencia en el Instituto. Como resultado se pudo inferir que la opción óptima está representada en una ficha con los cilindros significativos para la codificación Braille con perfil redondeado, y, con el número arábigo impreso en bajo relieve (ver Figura 1.15).

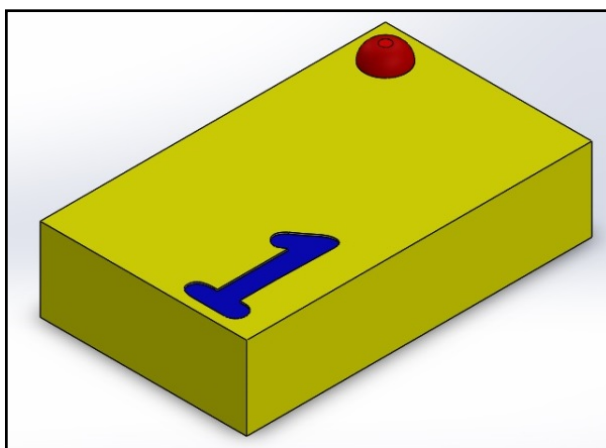


Figura 1.15. Vista de planta de ficha seleccionada.

Fuente: Autores

1.4. Programación de la ingeniería de detalle del panel

Establecida la conceptualización y el diseño de fichas tipo LEGO del panel, se decidió proseguir con el diseño e implementación de un prototipo del panel electrónico. Para este fin, en septiembre de 2014 se conformaron cuatro grupos de trabajo, encargándoles el diseño e implementación del panel, el diseño e implementación del bloque de identificación electrónica de fichas, el diseño e implementación del algoritmo de control del panel, y, el análisis de resultados.

CAPÍTULO II

2. DISEÑO E IMPLEMENTACIÓN DEL FIRMWARE

2.1. Metodología de desarrollo del firmware

En función de la adaptabilidad a las exigencias de este proyecto, para diseñar el firmware del panel se seleccionó el método de desarrollo de software en cascada [11], conformado por cinco fases (ver Figura 2.1). La primera, se relaciona a la definición de requerimientos. La segunda, corresponde al diseño lógico del firmware. La tercera, representa la implementación del diseño, esto es, la codificación y pruebas de funcionalidad. La cuarta fase, corresponde a la verificación de la integración y funcionalidad de todo el sistema. Y, la quinta corresponde al mantenimiento.

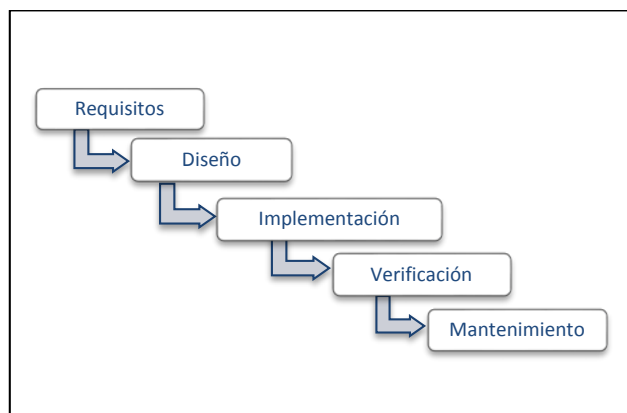


Figura 2.1. Fases del método de desarrollo de software en cascada.

Fuente: [11].

2.2. Fase de definición de requisitos de diseño

Los investigadores de la Sección de Diseño, Proyectos Arquitectónicos y Urbanismo del Departamento de Arquitectura y Artes de la UTP, establecieron los requerimientos de operatividad del panel. Éste deberá realizar las cuatro operaciones matemáticas básicas (suma, resta, multiplicación y división), empleando un máximo de dos operandos a la vez. En el caso de las operaciones de suma, resta y división, los operandos podrán contener cifras significativas hasta las centenas. Para el caso de la multiplicación se introduce la restricción a 2 posiciones (decenas) para uno de los operandos.

Finalmente, se requiere que la respuesta de las operaciones (correcta o incorrecta) se entregue en un formato adecuado a las capacidades del usuario, priorizando el uso de audio y de señales luminosas.

Por cuanto el firmware coordina el funcionamiento del panel, éste hereda los requerimientos descritos anteriormente.

2.3. Fase de diseño lógico.

El diseño lógico del firmware deberá resolver tres problemas a saber: identificación de fichas, ejecución de operaciones, y comprobación de respuesta.

2.3.1. Identificación de fichas.

El diseño lógico de la función de identificación de fichas incluye las tareas de identificar el valor numérico de la ficha, establecer la posición que ocupa la ficha en el operando (unidad, decena o centena); y, para el caso de las fichas operaciones, identificar la operación matemática requerida.

Para la identificación de fichas se propuso asignar variables (de la A a la K,) a las posiciones que una ficha podría ocupar en el panel (Ver Fig. 2.2 y Tabla 2.1), y emplear una solución basada en sensores infrarrojos descrita en [4].

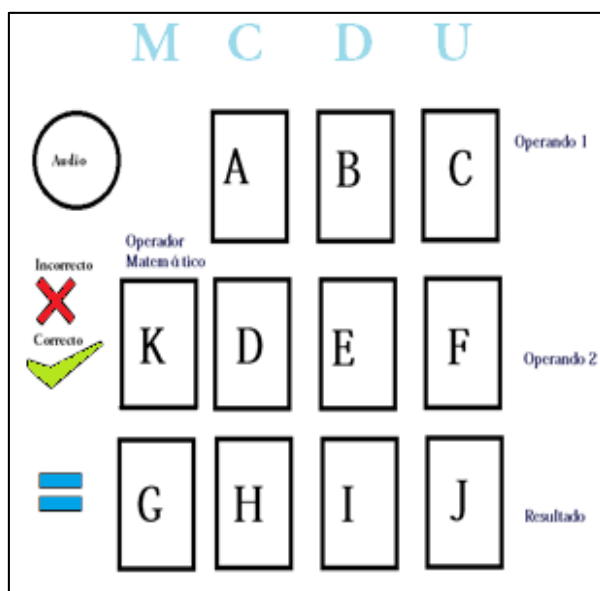


Figura 2.2. Asignación de variables a las posiciones posibles de las fichas en el panel.

Fuente: [4]

Tabla 2.1. Asignación de variables a las posiciones de las fichas en tablero.

| Variable | Descripción |
|------------------------------|---------------------|
| Bloque del operando 1 | |
| A | Centena |
| B | Decena |
| C | Unidad |
| Bloque del operando 2 | |
| D | Centena |
| E | Decena |
| F | Unidad |
| Bloque del resultado | |
| G | Unidad de Mil |
| H | Centena |
| I | Decena |
| J | Unidad |
| Operador | |
| K | Operador matemático |

Fuente: [4]

La Fig.2.3 muestra el concepto de operación de la solución, mientras que la Tabla 2.2 resume la codificación del valor numérico de la ficha [4].

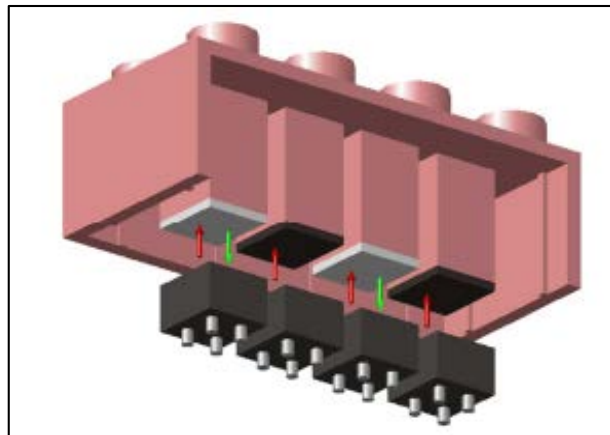


Figura 2.3. Esquema de funcionamiento de la solución basada en sensores infrarrojos para la identificación de fichas

Fuente: [4]

Tabla 2.2. Equivalente decimal y binario de los estados de los sensores infrarrojos utilizados para la identificación de fichas.

| Valor binario | Equivalente decimal |
|--|---------------------|
| Variables A, B, C, D, E, F, G, H, I, J | |
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |
| 0100 | 4 |
| 0101 | 5 |
| 0110 | 6 |
| 0111 | 7 |
| 1000 | 8 |
| 1001 | 9 |
| 1010 | Error en la lectura |
| 1011 | Error en la lectura |
| 1100 | Error en la lectura |
| 1101 | Error en la lectura |
| 1110 | Error en la lectura |
| 1111 | Sin ficha |
| Variable K | |
| 000 | + |
| 001 | - |
| 010 | X |
| 011 | ÷ |
| 1XX | Sin ficha |

Fuente: [4]

El flujograma mostrado en la Fig. 2.4 explica la organización de las operaciones lógicas a nivel de firmware para el reconocimiento de la ubicación de la ficha. Durante el proceso de reconocimiento, se identifican posibles errores: posición de la unidad sin ficha, ausencia de fichas en el bloque de resultado, y, ficha ubicada en forma incorrecta.

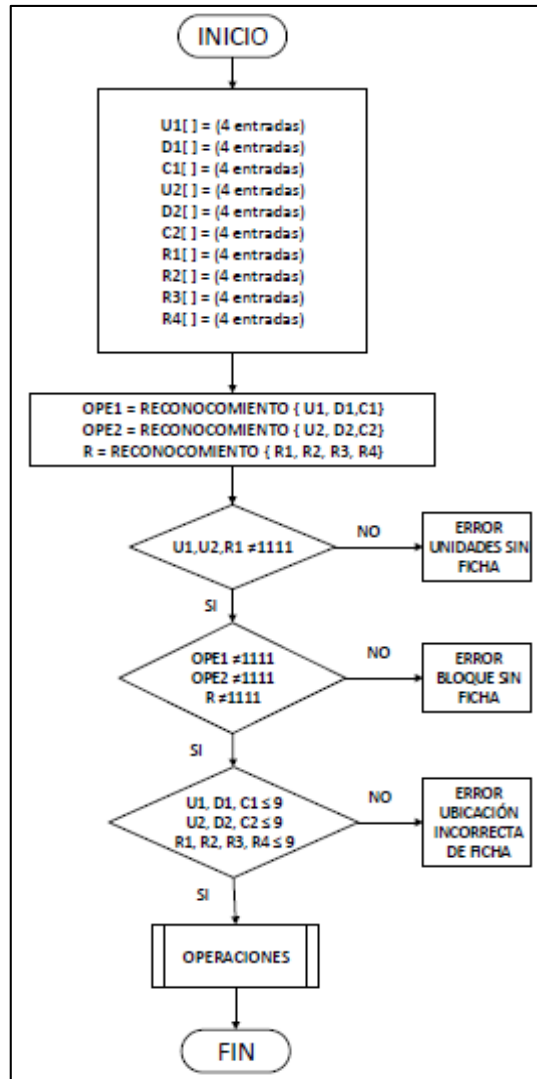


Figura. 2.4. Flujograma de las operaciones lógicas en el firmware para el reconocimiento de la ubicación de la ficha.
Fuente: [Autores]

Bajo el concepto de posición de la unidad sin ficha, se entiende el caso en el que no se colocó fichas numéricas en la posición de las unidades, tanto en los bloques de operando como en el de resultado. El error ausencia de fichas en el bloque de resultado se refiere al hecho de que no se colocó fichas en este bloque, por lo que no es posible completar la operación requerida ante la ausencia de un valor para comprobar la respuesta. El error ficha ubicada en forma incorrecta se refiere a que se colocó la ficha numérica de forma tal, que el encabezado señalado con la notación braille se encuentra hacia abajo, lo que puede conducir a que el sistema registre un valor numérico mayor a 9, con lo que se imposibilita la realización de la operación.

El flujograma mostrado en la Fig. 2.5 describe la organización de las operaciones lógicas a nivel de firmware para el reconocimiento del valor numérico de la ficha.

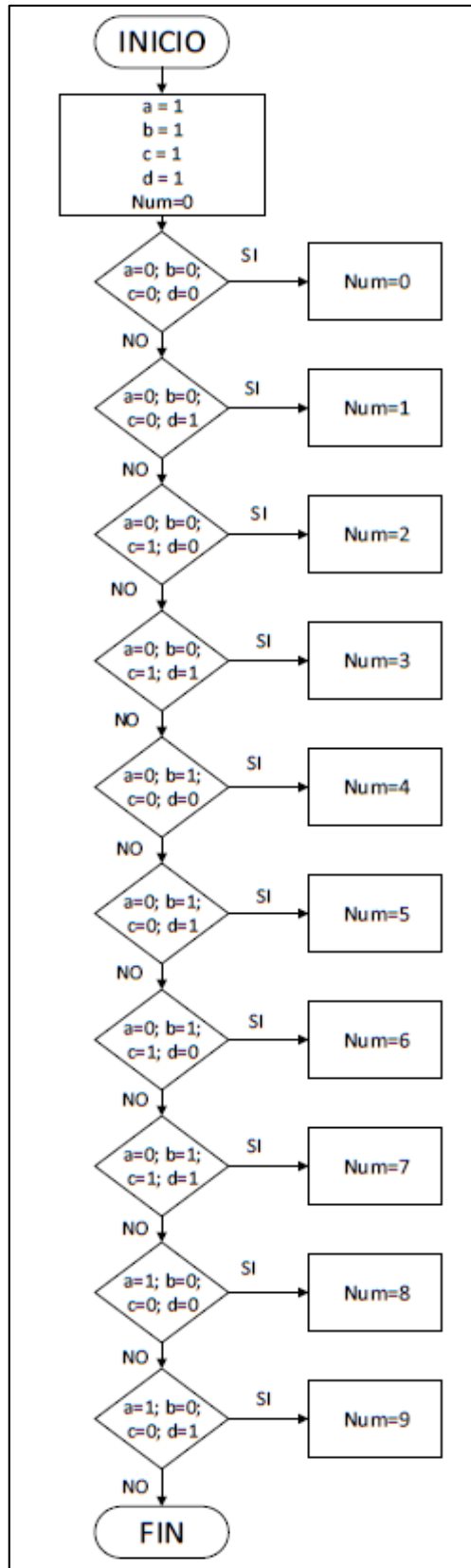


Figura. 2.5 Flujograma de las operaciones lógicas en el firmware para el reconocimiento del valor numérico de la ficha.

Fuente: Autores

2.3.2. Ejecución de operaciones

La Fig. 2.6 muestra el flujograma de las operaciones lógicas en el firmware para la función para ejecución de las operaciones matemáticas básicas (suma, resta, multiplicación y división). El proceso empieza con la identificación de la operación a realizar (según la combinación binaria ingresada) y termina con la ejecución de la operación correspondiente.

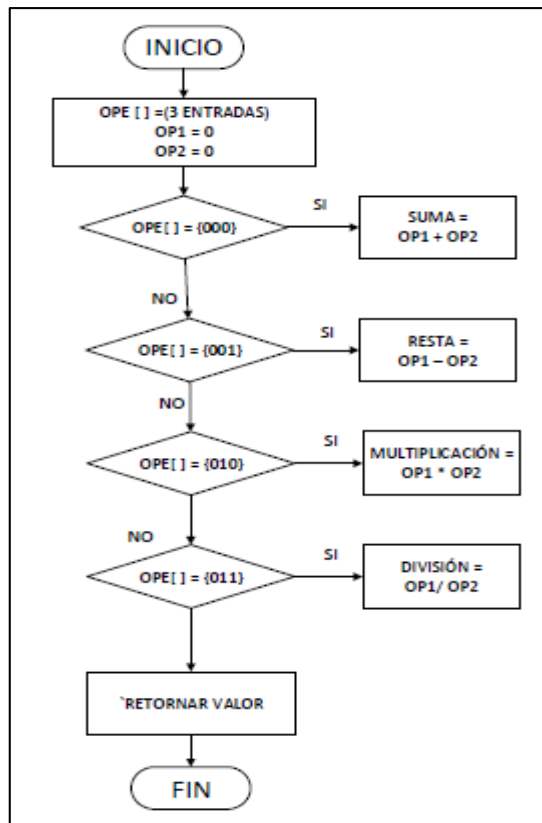


Figura. 2.6. Flujograma de las operaciones lógicas en el firmware para la función de ejecución de operaciones.

Fuente: Autores

2.3.3. Comprobación de respuesta

Al diseñar la función de comprobación de respuesta (ver Fig. 2.7), el equipo de trabajo decidió utilizar señales luminosas y de audio. Las señales luminosas serán presentadas a través de 2 arreglos LED, mientras que el audio se entregará a través de un parlante de baja potencia, empleando tonos musicales generados en frecuencias del rango audible para el ser humano [12] mediante la rutina mostrada en la Fig.2.8.

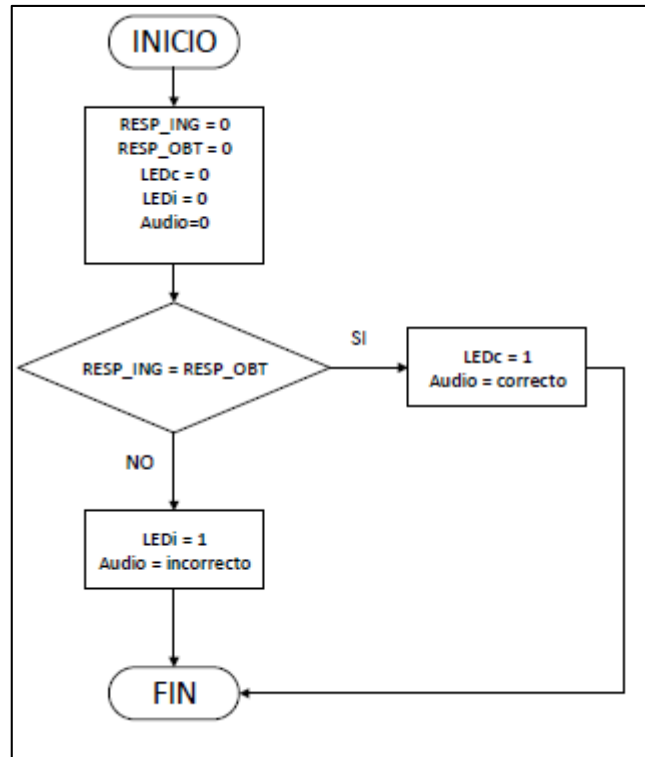


Figura. 2.7. Flujograma de las operaciones lógicas en el firmware para la función presentación de respuesta.

Fuente: Autores

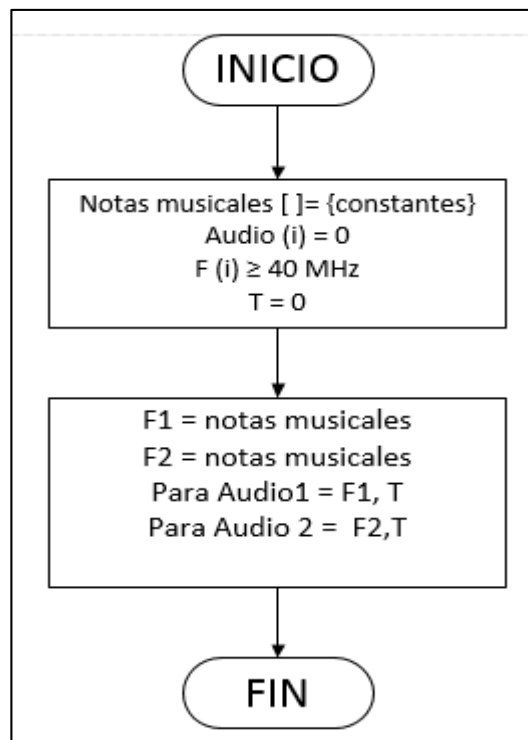


Figura. 2.8. Rutina de generación de tonos musicales.

Fuente: Autores

2.4. Selección de la plataforma tecnológica a utilizar

Al finalizar la fase de diseño lógico del firmware se seleccionó la plataforma tecnológica a utilizar, considerando criterios tales como la naturaleza open source del hardware y software a utilizar, la existencia de las entradas y salidas suficientes en el hardware (ver Tabla 2.3), y el conocimiento previo del lenguaje de programación.

Tabla 2.3. Número de entradas y salidas necesarias para la operación del firmware.

| Variables | Cantidad |
|---------------------|-----------|
| Entradas operando 1 | 12 |
| Entradas operando 2 | 12 |
| Entradas resultado | 16 |
| Entradas operador | 4 |
| Salidas | 3 |
| Total | 47 |

Fuente: Autores

Como resultado del análisis efectuado, se seleccionó la plataforma Arduino Mega 2560 [13], cuyas características técnicas se detallan en la Tabla 2.4.

Tabla 2.4. Características técnicas del Microcontrolador Arduino Mega 2560

| Característica técnica | Valor |
|-------------------------------------|--------------------------|
| Voltaje de operación, V | 5 |
| Voltaje de entrada (recomendado), V | 7-12 |
| Voltaje entrada (límite), V | 6-20 |
| Pines Digitales I/O | 54 (15 salidas para PWM) |
| Pines de entrada análoga | 16 |
| Corriente DC por Pin I/O, mA | 20 |
| Corriente DC para Pin 3.3V, mA | 50 |
| Memoria Flash, KB | 256 |
| SRAM, KB | 8 |
| EEPROM, KB | 4 |
| Velocidad de reloj, MHz | 16 |

Fuente: [13]

2.5. Fase de implementación del firmware

La fase de implementación del firmware se dividió en 2 subetapas: codificación y pruebas.

2.5.1. Codificación

Para iniciar la codificación, se asignó a cada uno de los pines del Arduino, una de las variables creadas en el diseño lógico (ver Tabla 2.5). La codificación se realizó en base a los flujogramas descritos en los apartados anteriores, utilizando el software Arduino, basado en lenguaje de programación C++.

Tabla 2.5. Asignación de los pines de la plataforma Arduino a cada una de las variables identificadas en el diseño lógico del firmware.

| Descripción (Variable) | Numeración de PIN |
|-------------------------------|-------------------|
| Operando 1 | |
| Unidad 1 (C) | 2,3,4,5 |
| Decena (B) | 6,7,8,9 |
| Centena (A) | 10,11,12,13 |
| Operando 2 | |
| Unidad (F) | 14,15,16,17 |
| Decena (E) | 18,19,20,21 |
| Centena (D) | 22,23,24,25 |
| Resultado | |
| Unidad (J) | 26,27,28,29 |
| Decena (I) | 30,31,32,33 |
| Centena (H) | 34,35,36,37 |
| Unidad de Mil (G) | 38,39,40,41 |
| Operador matemático | |
| Operador (K) | 42,43,44 |
| Salidas | |
| Igual | 45 |
| Arreglo de LEDs (Correcto) | 52 |
| Arreglo de LEDs (Incorrecto) | 53 |
| Salida de audio | 50 |

Fuente: Autores

2.5.2. Pruebas

En esta subetapa se verificó el desempeño del firmware en la identificación de fichas, la ejecución de las operaciones ingresadas, y la comprobación de respuestas.

Inicialmente se estableció una comunicación serial entre el hardware de programación y un computador. Luego, se comprobó la identificación de fichas, y la ejecución de la operación matemática ingresada, utilizando los protocolos mostrados en las Fig. 2.9 y 2.10.

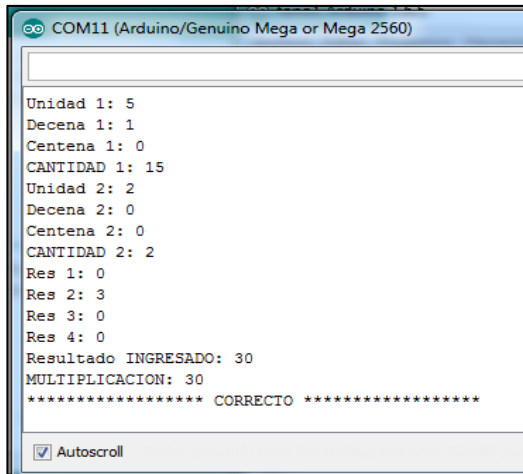


Figura. 2.9 Protocolo de respuesta correcta para la identificación de fichas y realización de operación.

Fuente: Autores

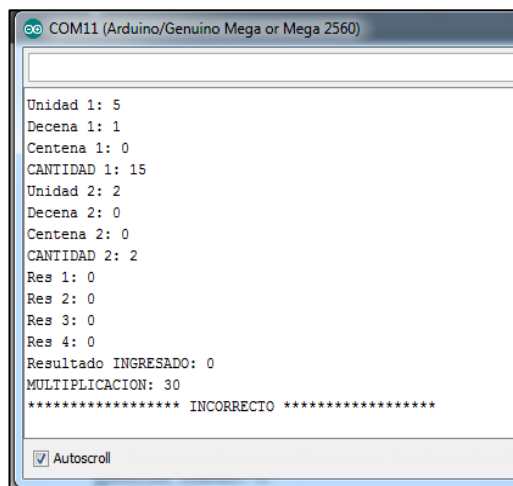


Figura. 2.10. Protocolo de respuesta incorrecta para la identificación de fichas y realización de operación.

Fuente: Autores

La prueba se completó con la validación del funcionamiento del bloque de señalización luminosa (ver Fig. 2.11 y 2.12). En total se ejecutó 40 pruebas.



Figura. 2.11. Señalización de una respuesta correcta en el bloque de señalización luminosa.

Fuente: Autores

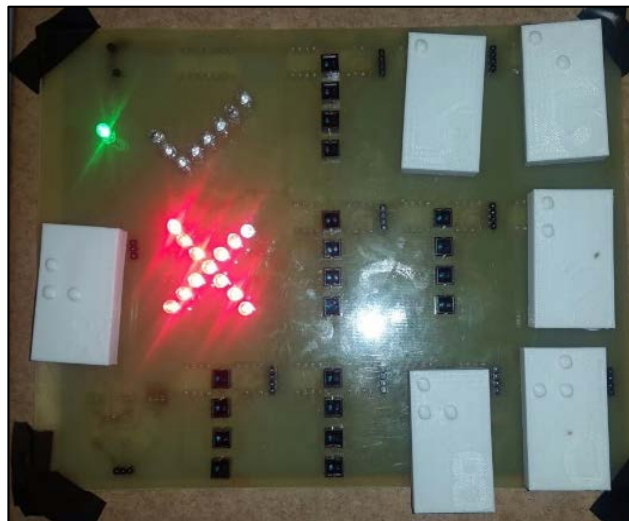


Figura. 2.12. Señalización de una respuesta incorrecta en el bloque de señalización luminosa.

Fuente: Autores

Una vez completado el diseño del case del panel, se ensamblaron todos los elementos y se realizó una segunda subetapa de pruebas. Los resultados mostraron inconsistencias imputables al apareamiento de sombras sobre los sensores infrarrojos, por lo cual fue necesario rediseñar el case.

2.6. Fases de verificación y mantenimiento

Estas etapas se encuentran pendientes de ejecución. La no disponibilidad de recursos no ha posibilitado la conclusión del case final.

CONCLUSIONES

- Se analizó 6 opciones de optimización del panel electrónico implementado, identificando como óptima a la que propone la adopción de fichas tipo LEGO, considerando sus ventajas y desventajas, pero sobre todo la ampliación de las potencialidades de enseñanza de matemáticas.
- Considerando el grado de desarrollo de la capacidad táctil en el grupo beneficiario del proyecto, y, los requerimientos de estética planteados por el equipo de trabajo, se decidió plantear al menos 4 posibles diseños de fichas tipo LEGO a utilizar en el panel (operandos, operadores, y, referencias de posición), de entre las cuales se escogería la opción óptima.
- La elección del hardware para la implementación del firmware, fue el microprocesador ARDUINO MEGA 2560, tomando como requisito principal que debe satisfacer el número de entradas y salidas necesarias.
- El panel electrónico fue diseñado para realizar operaciones sólo con números enteros, considerando que cuando se realice la suma, resta y división los dos operandos puede utilizar cantidades hasta el orden de las centenas, pero para la multiplicación solo se puede utilizar operandos que su resultado no sea mayor a las unidades de mil.
- El case debe ser diseñado de tal manera que este no cause obstrucción entre la ficha y el sensor, ya que esto ocasiona una lectura incorrecta de la ficha.
- Para realizar la comprobación de identificación de las fichas en el firmware fue necesario hacer una comunicación serial entre el microprocesador y el computador.

RECOMENDACIONES

- Debido a limitaciones presupuestarias y de tiempo no fue posible completar la implementación y validación del firmware diseñado, por lo que se recomienda su implementación futura de cara a cerrar el proyecto.
- Completada la implementación y validación, se deberá atender también la elaboración de un presupuesto real de construcción en serie de los paneles electrónicos diseñados.

BIBLIOGRAFÍA

- [1] Diseño del Material Didáctico Matemático y Discapacidad Visual en niños de Educación Básica [en línea]. Disponible en <http://memorias.utpl.edu.ec/sites/default/files/documentacion/arte2013/utpl-diseno-material-didactico-matematico.pdf>
- [2] Bermeo. Figueroa. Lima. Ochoa. Peña. Romero. Jaramillo, «Análisis de opciones de optimización de un panel electrónico diseñado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales,» Loja, 2014.
- [3] Bermeo. Figueroa. Lima. Ochoa. Peña. Romero. Jaramillo, «Diseño de un panel electrónico para la enseñanza de operaciones matemáticas básicas a niños con capacidad visual especial, basado en la utilización de fichas tipo LEGO,» Loja, 2014.
- [4] Jaramillo. Jaramillo, «Diseño del bloque de identificación electrónica de fichas tipo LEGO en un panel diseñado para la enseñanza de operaciones matemáticas básicas a niños con capacidades visuales diferentes» Loja, 2015.
- [5] “Los niños ciegos y su educación”, [en línea]. Consultado 2014-03-11 URL: http://catarina.udlap.mx/u_dl_a/tales/documentos/lda/arteaga_j_g/capitulo2.pdf
- [6] FERNÁNDEZ DEL CAMPO, José Enrique. “Braille y Matemática” [en línea]. Consultado 2014-03-04. URL: http://sid.usal.es/idocs/f8/fdo10191/braille_y_matematica.pdf
- [7] “Dimensiones de LEGO” [en línea]. Consultado 2014-03-15 URL: <http://www.robertcailliau.eu/Lego/Dimensions/zMeasurements-en.xhtml>
- [8] “Educación con LEGO” [en línea]. Consultado 2014-03-15. URL: <http://education.lego.com/en-us/lego-education-product-database/preschool/9027-lego-duplo-brick-set>
- [9] Software de diseño CAD en 3D, SOLIDWORKS, sitio web, [en línea], consultado 2014-03-11. URL: <http://www.solidworks.com/>
- [10] MakerBot Replicator 2 Desktop 3D Printer, sitio web, [en línea], consultado 2014-04-23. URL: <http://store.makerbot.com/replicator2x>
- [11] Modelos y metodologías para el desarrollo de software [en línea] Consultado 2015-10-03. URL: <http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>
- [12] Pérez, Constantino, «AUDICIÓN HUMANA», [en línea]. Consultado 2015-10-25 URL: <http://personales.unican.es/perezvr/pdf/La%20Audicion%20Humana.pdf>.
- [13] Arduino MEGA 2560, [en línea]. Consultado 2015-09-11 URL: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

ANEXOS

Diseño del firmware de un panel electrónico para la enseñanza de operaciones matemáticas básicas a niños con capacidad visual especial, basado en la utilización de fichas tipo LEGO

Cinthy Figueroa^{#1}, Jhuly Valarezo^{#1}, Jorge Luis Jaramillo^{#2}

^{#1} Profesional en formación de la Titulación de IET, Universidad Técnica Particular de Loja

^{#2} Docente de la SET del DCCE, Universidad Técnica Particular de Loja

Loja, Ecuador

2015

¹cafigueroa1@utpl.edu.ec, ¹jrvalarezo@utpl.edu.ec, ²jorgeluis@utpl.edu.ec

Resumen— En el presente documento se describen los resultados obtenidos en la etapa de diseño del firmware a utilizar en un panel electrónico para la enseñanza de matemáticas básicas a niños con capacidad visual especial, basado en la utilización de fichas tipo LEGO.

Palabras claves— enseñanza de matemáticas, enseñanza de matemáticas a niños con capacidad visual especial, paneles electrónico, LEGO, Arduino MEGA.

I. INTRODUCCIÓN

En septiembre de 2013, personal de la Sección de Electrónica y Telecomunicaciones del Departamento de Ciencias de la Computación y Electrónica, y de la Sección de Diseño, Proyectos Arquitectónicos y Urbanismo del Departamento de Arquitectura y Artes, de la Universidad Técnica Particular de Loja, decidieron colaborar en el diseño e implementación de un panel electrónico para la enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales. Como resultado, se desarrolló el panel electrónico que se muestra en la Fig.1, y que se describe en [1].

La evaluación de desempeño del panel desarrollado mostró que se cumplía con las especificaciones funcionales, pero se presentaban problemas operativos, especialmente en relación a la manipulación de las fichas. En este contexto, en marzo 2014 se acordó optimizar el panel, obteniendo los resultados descritos en trabajos previos [2], [3].

En Abril del 2015, como resultado de un trabajo de titulación en Ingeniería en Electrónica y Telecomunicaciones en la UTPL [4], se presentó el diseño e implementación del bloque de identificación electrónica de fichas tipo LEGO. Luego, dando continuidad a la programación del proyecto, se inició la fase de diseño del firmware del panel electrónico. En este documento se recoge los resultados obtenidos en esta fase.

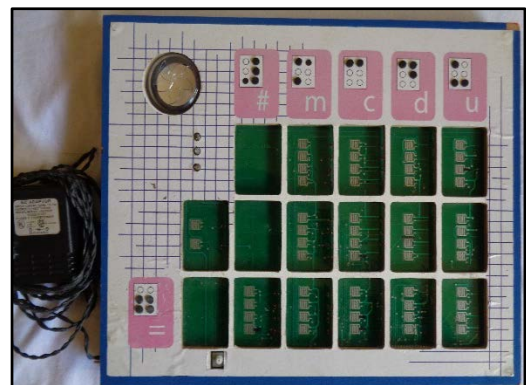


Fig. 1. Versión original del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales [1].

II. SOBRE LA OPTIMIZACIÓN DEL PANEL ORIGINAL

A. Sobre la necesidad de optimizar el diseño original

Debido a los problemas presentados en el panel implementado se identificaron necesidades de optimización: disminución del volumen y peso de las fichas, integración de identificadores decimales y código Braille, reducción del desgaste de contactos en el circuito impreso, y diversificación en el uso de colores [1].

Conocidas las necesidades de optimización, se formularon 6 diferentes propuestas tecnológicas. En la **primera opción** se propuso conservar las características de funcionamiento del panel original e incorporar un switch de encendido y apagado. El usuario ingresa los operandos presionando los pulsadores ubicados en el tablero, tal como se muestra en la Fig.2. Luego, selecciona el operador matemático, e indica la respuesta expresada en unidades, decenas y centenas. Al oprimir el operador “igual” y la tecla “comprobar”, se escucha el mensaje “correcto” para una operación válida, o “incorrecto” para una operación errónea. El usuario puede resetear los pulsadores, para ejecutar la operación nuevamente.

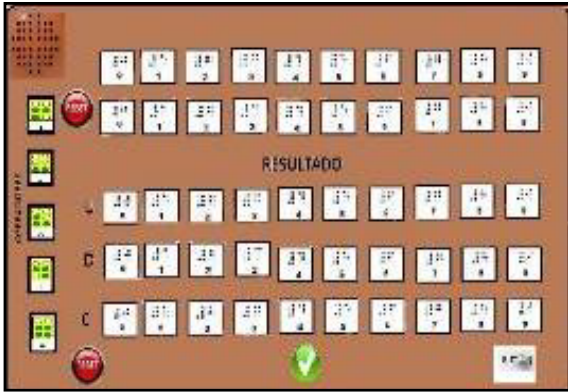


Fig. 2. Primera opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales. [Autores].

En la **segunda opción**, se pretende llamar la atención de los niños para facilitar el aprendizaje, por lo que se utiliza un panel electrónico integrado a un oso de peluche. El panel opera de forma similar a lo descrito en el apartado anterior (ver Fig.3).

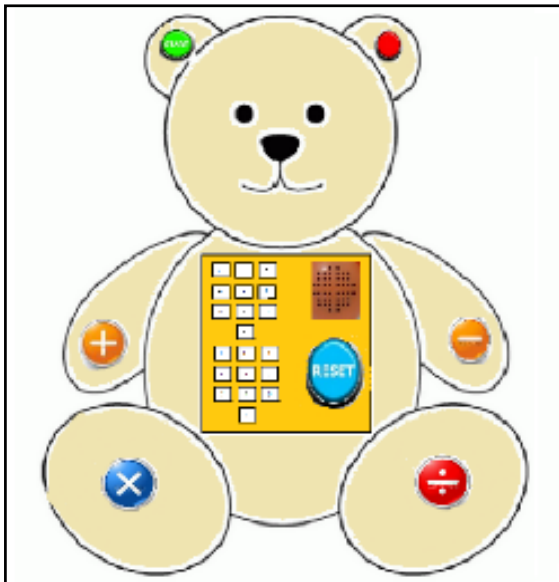


Fig. 3. Segunda opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales. [Autores].

En la **tercera opción** se presenta un paradigma diferente al panel original, en el que dos jugadores participan identificándose cada uno de ellos con el color verde o azul respectivamente. Se utiliza un tablero con pulsadores (que contienen números entre uno y veinte), y tres dados (dos para los operandos y uno para las operaciones). El tablero y los dados están identificados con código braille. El jugador lanza los dados. El procedimiento y el resultado se ingresa en el tablero, y se verifica la correcta ejecución (ver Fig.4).



Fig.4. Tercera opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales. [Autores].

En la **cuarta opción**, para iniciar se presiona el botón "inicio", y el panel da a conocer, mediante audio, un número aleatorio. El usuario identifica dos operandos y una operación, que se ingresan al panel presionando los pulsadores del tablero. Si los operandos y la operación ingresada son correctos, se escuchará "correcto", caso contrario se indicará "incorrecto", y se generará un nuevo número aleatorio (ver Fig. 5).



Fig. 5. Cuarta opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales. [Autores].

La **quinta opción** se basa en el funcionamiento del juego "sudoku", adaptándolo con numeración Braille (ver Fig. 6). Este juego utiliza una tabla de 9x9, compuesta por bloques denominados también subtablas de 3x3. Algunas celdas contienen números dados. El juego tiene por objetivo completar las celdas vacías, de forma tal que cada columna, fila, y región contenga números del 1 al 9 una sola vez. El tablero contendrá los botones "empezar" y "evaluar". La rutina de comprobación terminará con un mensaje de "correcto" o "incorrecto".

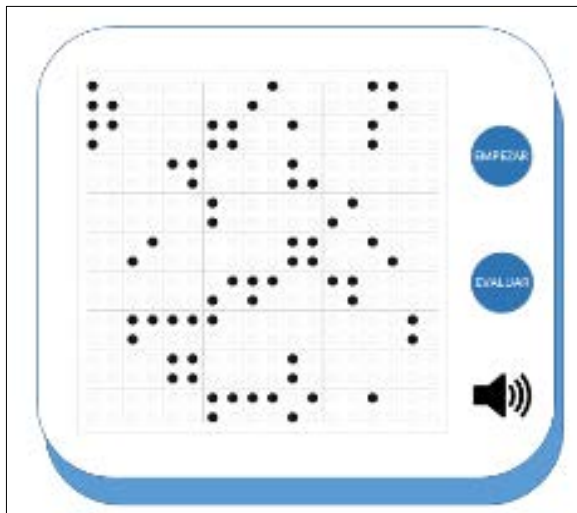


Fig. 6. Quinta opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales. [Autores].

En la **sexta opción** se adapta fichas tipo LEGO al panel electrónico, con lo que se pretende reducir el desgaste prematuro de los contactos en los circuitos impresos. Se utiliza un tablero y diversas fichas (números y operaciones matemáticas). Las fichas se colocan en el tablero en la lógica: primer operando, segundo operando, operación, solución, y, comprobación (ver Fig. 7).

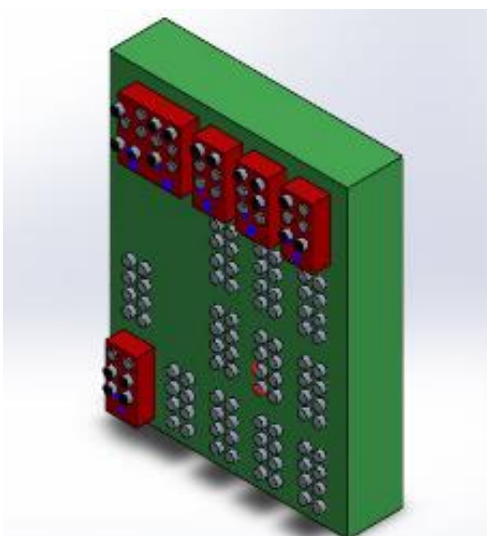


Fig. 7. Sexta opción de optimización del panel electrónico implementado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales. [Autores].

Las opciones señaladas se analizaron identificando potenciales inconvenientes de orden mecánico, y limitaciones en el número de cifras significativas de operandos y resultados. La Tabla 1 muestra las ventajas y desventajas de cada una de las opciones. Como resultado se seleccionó la sexta opción, considerando sobre todo la ampliación de las potencialidades en la enseñanza de las matemáticas que esta brinda.

Tabla 1.

Análisis comparativo de las opciones presentadas. [Autores]

| Opción | Ventajas | Desventajas |
|--------|---|--|
| 1 | <ul style="list-style-type: none"> Elementos no móviles Se puede verificar que pulsadores están activados | <ul style="list-style-type: none"> No se puede activar 2 pulsadores simultáneamente Desgaste prematuro Sistema mecánico complejo Gran tamaño |
| 2 | <ul style="list-style-type: none"> Elementos no móviles | <ul style="list-style-type: none"> Desgaste prematuro Sistema mecánico complejo |
| 3 | <ul style="list-style-type: none"> Desgaste limitado | <ul style="list-style-type: none"> Operación compleja |
| 4 | <ul style="list-style-type: none"> Desgaste limitado | <ul style="list-style-type: none"> Operación compleja |
| 5 | <ul style="list-style-type: none"> Desgaste limitado | <ul style="list-style-type: none"> Sistema complejo Operación compleja |
| 6 | <ul style="list-style-type: none"> Desgaste limitado Manipulación intuitiva | <ul style="list-style-type: none"> Operación compleja |

B. Requerimientos generales para el diseño del nuevo panel

En el grupo beneficiario de este proyecto, existen menores con ceguera parcial por lo que se consideró incluir en el diseño colores vivos. Además, se decidió incluir elementos estimulantes al tacto (codificación Braille) y al oído (sonidos “respuesta correcta” y “respuesta incorrecta”) [5], [6].

Al igual que en la versión original, el panel permitirá ejecutar las cuatro operaciones básicas (suma, resta, multiplicación y división). El panel constará de un tablero con una disposición de 5 columnas y 4 filas, y de una sección de encendido y sonorización, (ver Fig. 8).

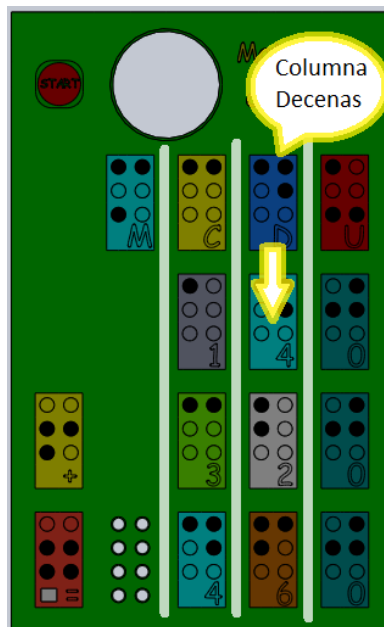


Fig. 8. Tablero del panel electrónico optimizado. [Autores]

Desde la derecha, las columnas del tablero permitirán ubicar las fichas correspondientes a unidades, decenas, centena, y, miles. En la primera fila, en cada columna, permanecerán fijas las fichas que señalan el valor posicional de la columna.

En la última columna, a la izquierda del tablero, se ubicarán las fichas correspondientes a la operación

aritmética a realizar (tercera fila desde arriba del tablero). En la posición quinta columna y cuarta fila, permanecerá fija la ficha de resultado de operación, construida sobre un pulsador.

En la sección de encendido y sonorización, ubicada en la parte superior del tablero, se encontrará el botón de inicio (START) y un parlante, que servirá para comunicar lo acertado o no de la operación efectuada.

Las fichas correspondientes a los operandos y al resultado, podrán ocupar exclusivamente posiciones predeterminadas en el tablero. La Fig.9 muestra el diseño del tablero, que corresponde al estándar de una pieza tipo LEGO.

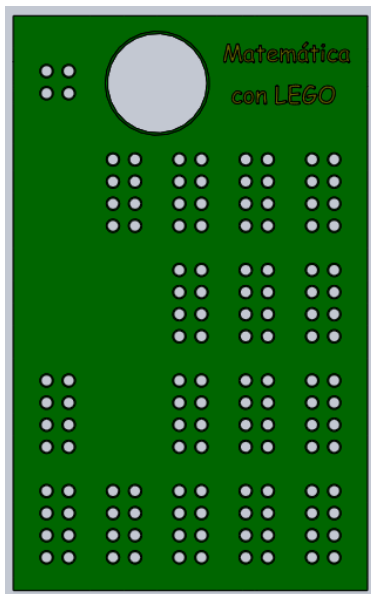


Fig. 9. Vista superior y lateral izquierda del tablero. [Autores]

En cada ficha de los operandos, se imprimirá el valor absoluto de la misma (de 0 a 9), en Braille (en la parte superior de la ficha) y en números arábigos (en la parte inferior de la ficha), tal como lo muestra la Fig.10.

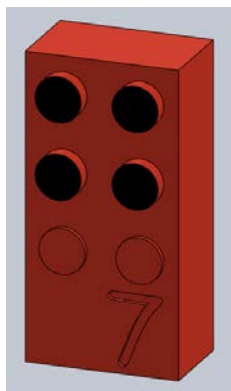


Fig. 10. Vista general de una ficha del tablero. [Autores]

La geometría de cada una de las fichas corresponde al estándar de una pieza LEGO (ver Fig. 11 a la 13).

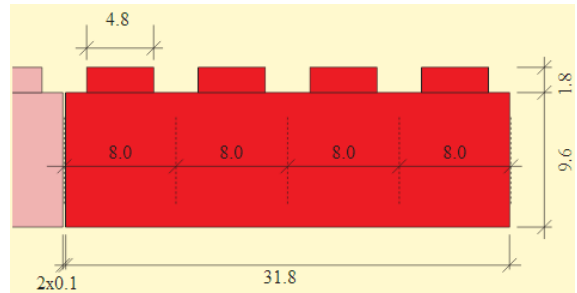


Fig. 11. Vista lateral derecha de una pieza LEGO 4 x 2, [7]

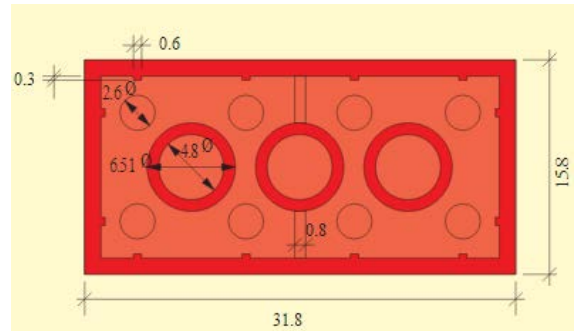


Fig. 12. Vista inferior de una pieza de LEGO 4 x 2, [7]

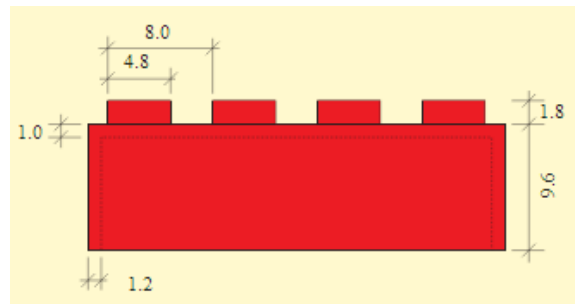


Fig. 13. Vista lateral izquierda de una pieza de LEGO 4 x 2, [7]

Adicionalmente, se diseñó fichas para las operaciones matemáticas básicas, y, para el reconocimiento de las posiciones. La Fig.14 muestra el ensamblaje de las fichas sobre el tablero.

En otro contexto, al beneficio que brinda las fichas tipo LEGO (múltiples combinaciones y las distintas estructuras que se logran crear con un grupo básico de piezas), se suma el desarrollo de habilidades motrices y la adquisición de conceptos espaciales, como volumen, tamaño y formas geométricas [8]. Los infantes se sienten atraídos por la idea de aprender divirtiéndose, condición totalmente válida para niños con capacidad visual especial, esta opción se propone adaptar fichas tipo lego en el panel electrónico. Al utilizar las fichas tipo lego se pretende reducir el desgaste prematuro de los contactos en los circuitos impresos en el bloque de control, de tal forma que el panel tenga durabilidad.

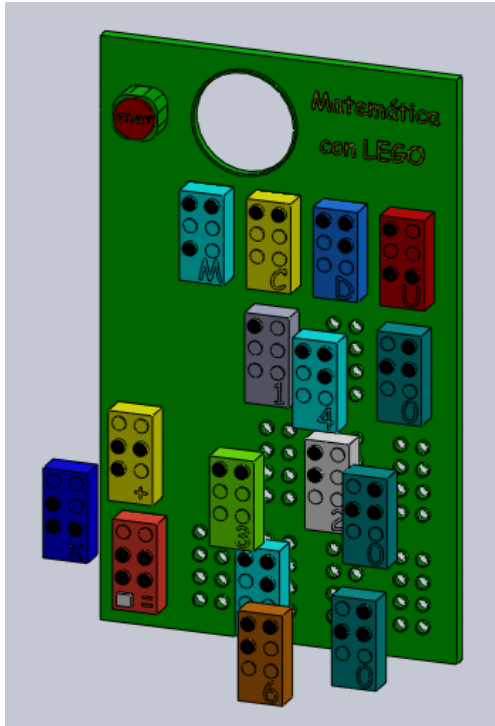


Fig. 14. Vista general del tablero y de las fichas. [Autores]

C. Diseño a detalle de las fichas a utilizar en el panel

Considerando el grado de desarrollo de la capacidad táctil en el grupo beneficiario del proyecto, y, los requerimientos de estética planteados por el equipo de trabajo, se decidió plantear al menos 4 posibles diseños de fichas, de entre las cuales se escogería la opción óptima.

Debido a la disponibilidad de software y equipos, se decidió imprimir las fichas en 3D, utilizando los recursos del aplicativo Solidworks [9], y, de la impresora 3D tipo Makerbot Replicator 2X [10].

Para seleccionar la opción óptima del diseño de las fichas a utilizar en el panel, se decidió someter las impresiones al análisis del grupo beneficiario del proyecto. Se trabajó con los niños del cuarto año de educación básica del Instituto para Ciegos "Byron Eguiguren". Este grupo está conformado por 6 menores, de los cuales 3 tienen capacidad visual especial. A cada uno de los niños se le pidió manipular las diferentes fichas impresas, recogiendo sus impresiones. Adicionalmente, se solicitó una apreciación a la tutora del grupo, una docente con 15 años de experiencia en el Instituto. Como resultado se pudo inferir que la opción óptima está representada en una ficha con los cilindros significativos para la codificación Braille con perfil redondeado, y, con el número arábigo impreso en bajo relieve (ver Fig.15).

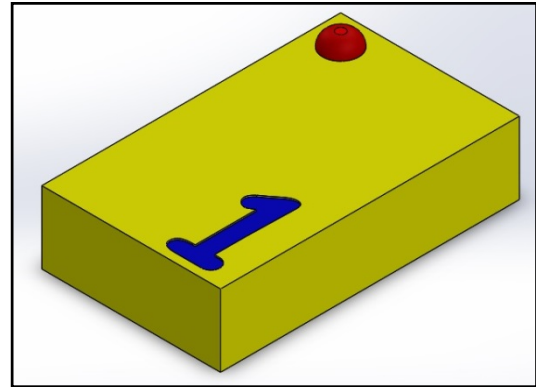


Fig. 15. Vista de planta de ficha seleccionada. [Autores].

D. Programación de la ingeniería de detalle del panel

Establecida la conceptualización y el diseño de fichas tipo LEGO del panel, se decidió proseguir con el diseño e implementación de un prototipo del panel electrónico. Para este fin, en septiembre de 2014 se conformaron cuatro grupos de trabajo, encargándoles el diseño e implementación del panel, el diseño e implementación del bloque de identificación electrónica de fichas, el diseño e implementación del algoritmo de control del panel, y, el análisis de resultados.

III. DISEÑO E IMPLEMENTACIÓN DEL FIRMWARE

A. Metodología de desarrollo del firmware

En función de la adaptabilidad a las exigencias de este proyecto, para diseñar el firmware del panel se seleccionó el método de desarrollo de software en cascada [11], conformado por cinco fases (ver Fig.16). La primera, se relaciona a la definición de requerimientos. La segunda, corresponde al diseño lógico del firmware. La tercera, representa la implementación del diseño, esto es, la codificación y pruebas de funcionalidad. La cuarta fase, corresponde a la verificación de la integración y funcionalidad de todo el sistema. Y, la quinta corresponde al mantenimiento.

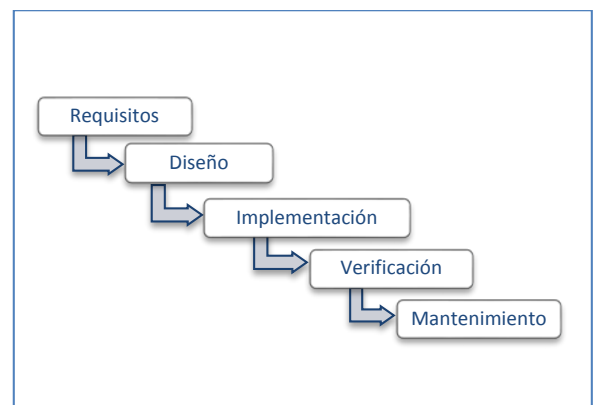


Fig 16. Fases del método de desarrollo de software en cascada, [11]

B. Fase de definición de requisitos de diseño

Los investigadores de la Sección de Diseño, Proyectos Arquitectónicos y Urbanismo del Departamento de Arquitectura y Artes de la UTPL, establecieron los requerimientos de operatividad del panel. Éste deberá

realizar las cuatro operaciones matemáticas básicas (suma, resta, multiplicación y división), empleando un máximo de dos operandos a la vez. En el caso de las operaciones de suma, resta y división, los operandos podrán contener cifras significativas hasta las centenas. Para el caso de la multiplicación se introduce la restricción a 2 posiciones (decenas) para uno de los operandos.

Finalmente, se requiere que la respuesta de las operaciones (correcta o incorrecta) se entregue en un formato adecuado a las capacidades del usuario, priorizando el uso de audio y de señales luminosas.

Por cuanto el firmware coordina el funcionamiento del panel, éste hereda los requerimientos descritos anteriormente.

C. Fase de diseño lógico

El diseño lógico del firmware deberá resolver tres problemas a saber: identificación de fichas, ejecución de operaciones, y comprobación de respuesta.

Identificación de fichas

El diseño lógico de la función de identificación de fichas incluye las tareas de identificar el valor numérico de la ficha, establecer la posición que ocupa la ficha en el operando (unidad, decena o centena); y, para el caso de las fichas operaciones, identificar la operación matemática requerida.

Para la identificación de fichas se propuso asignar variables (de la A a la K,) a las posiciones que una ficha podría ocupar en el panel (Ver Fig.17 y Tabla 2), y emplear una solución basada en sensores infrarrojos descrita en [4]. La Fig.18 muestra el concepto de operación de la solución, mientras que la Tabla 3 resume la codificación del valor numérico de la ficha [4].

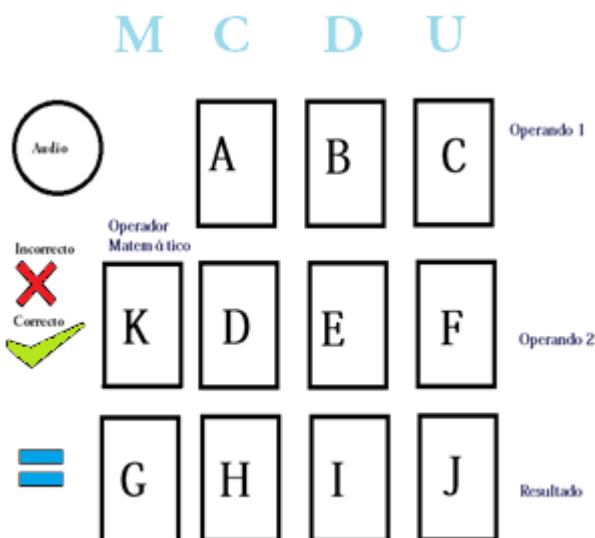


Fig 17. Asignación de variables a las posiciones posibles de las fichas en el panel [4]

Tabla 2.

Asignación de variables a las posiciones de las fichas en el tablero. [4]

| Variable | Descripción |
|------------------------------|---------------------|
| Bloque del operando 1 | |
| A | Centena |
| B | Decena |
| C | Unidad |
| Bloque del operando 2 | |
| D | Centena |
| E | Decena |
| F | Unidad |
| Bloque del resultado | |
| G | Unidad de Mil |
| H | Centena |
| I | Decena |
| J | Unidad |
| Operador | |
| K | Operador matemático |

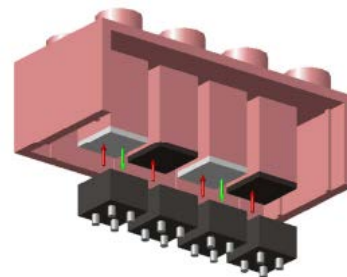


Fig 18. Esquema de funcionamiento de la solución basada en sensores infrarrojos para la identificación de fichas [4].

Tabla 3.

Equivalente decimal y binario de los estados de los sensores infrarrojos utilizados para la identificación de fichas [4]

| Valor binario | Equivalente decimal |
|--|---------------------|
| Variables A, B, C, D, E, F, G, H, I, J | |
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |
| 0100 | 4 |
| 0101 | 5 |
| 0110 | 6 |
| 0111 | 7 |
| 1000 | 8 |
| 1001 | 9 |
| 1010 | Error en la lectura |
| 1011 | Error en la lectura |
| 1100 | Error en la lectura |
| 1101 | Error en la lectura |
| 1110 | Error en la lectura |
| 1111 | Sin ficha |
| Variable K | |
| 000 | + |
| 001 | - |
| 010 | X |
| 011 | ÷ |
| 1XX | Sin ficha |

El flujograma mostrado en la Fig. 19 explica la organización de las operaciones lógicas a nivel de firmware para el reconocimiento de la ubicación de la ficha. Durante el proceso de reconocimiento, se identifican posibles errores: posición de la unidad sin ficha, ausencia de fichas en el bloque de resultado, y, ficha ubicada en forma incorrecta.

Bajo el concepto de posición de la unidad sin ficha, se entiende el caso en el que no se colocó fichas numéricas en la posición de las unidades, tanto en los bloques de operando como en el de resultado. El error ausencia de fichas en el bloque de resultado se refiere al hecho de que no se colocó fichas en este bloque, por lo que no es posible completar la operación requerida ante la ausencia de un valor para comprobar la respuesta. El error ficha ubicada en forma incorrecta se refiere a que se colocó la ficha numérica de forma tal, que el encabezado señalado con la notación braille se encuentra hacia abajo, lo que puede conducir a que el sistema registre un valor numérico mayor a 9, con lo que se imposibilita la realización de la operación.

El flujograma mostrado en la Fig. 20 describe la organización de las operaciones lógicas a nivel de firmware para el reconocimiento del valor numérico de la ficha.

Ejecución de operaciones

La Fig. 21 muestra el flujograma de las operaciones lógicas en el firmware para la función para ejecución de las operaciones matemáticas básicas (suma, resta, multiplicación y división). El proceso empieza con la identificación de la operación a realizar (según la combinación binaria ingresada) y termina con la ejecución de la operación correspondiente.

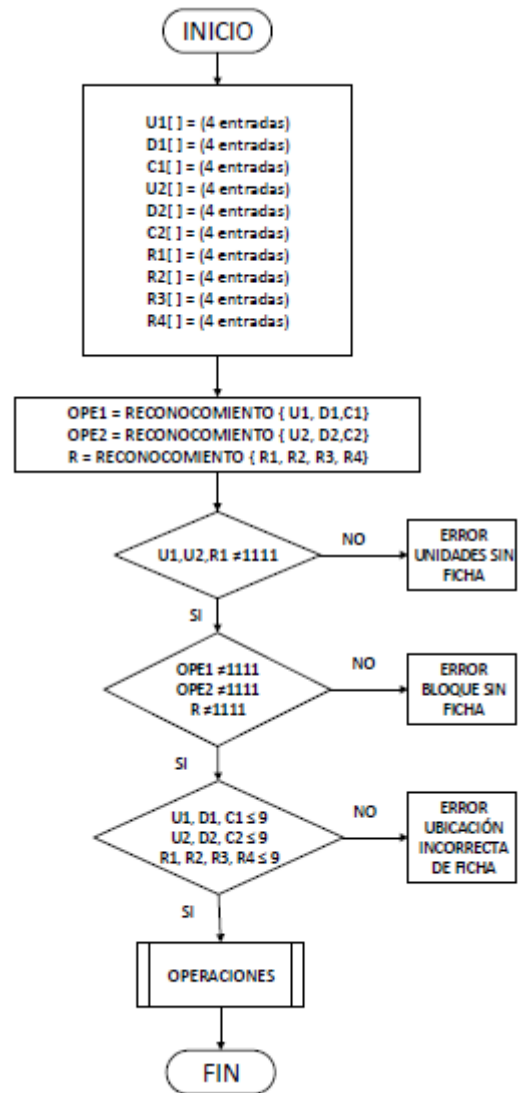


Fig. 19. Flujograma de las operaciones lógicas en el firmware para el reconocimiento de la ubicación de la ficha. [Autores]

Comprobación de respuesta

Al diseñar la función de comprobación de respuesta (ver Fig. 22), el equipo de trabajo decidió utilizar señales luminosas y de audio. Las señales luminosas serán presentadas a través de 2 arreglos LED, mientras que el audio se entregará a través de un parlante de baja potencia, empleando tonos musicales generados en frecuencias del rango audible para el ser humano [12] mediante la rutina mostrada en la Fig.23.

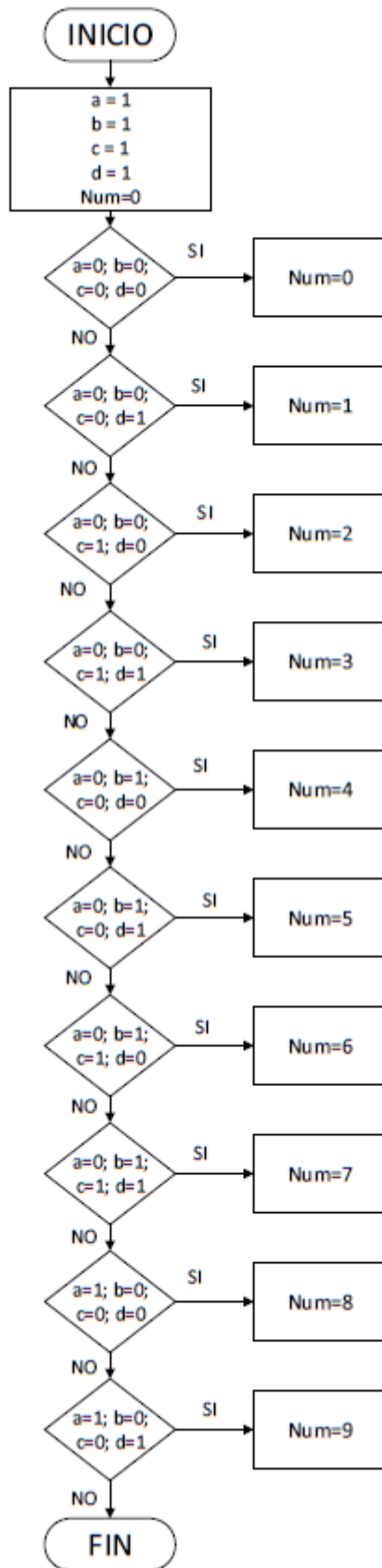


Fig. 20 Flujo de las operaciones lógicas en el firmware para el reconocimiento del valor numérico de la ficha. [Autores]

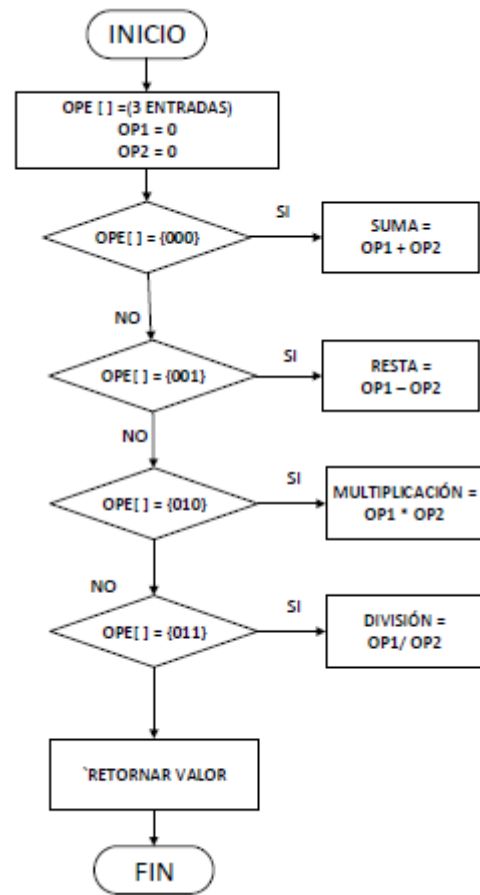


Fig. 21. Flujo de las operaciones lógicas en el firmware para la función de ejecución de operaciones. [Autores]

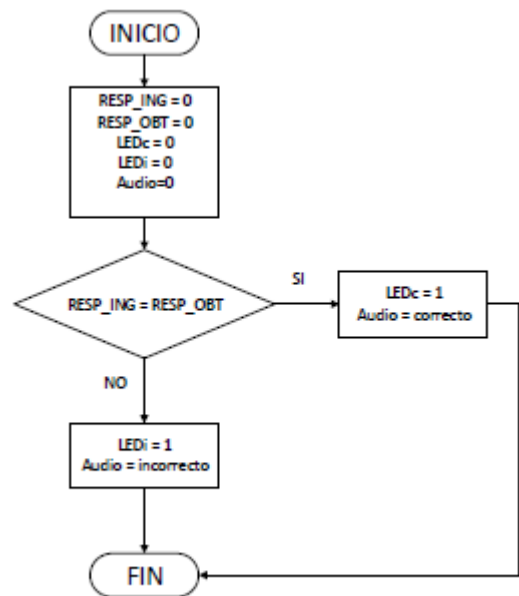


Fig. 22. Flujo de las operaciones lógicas en el firmware para la función presentación de respuesta. [Autores]

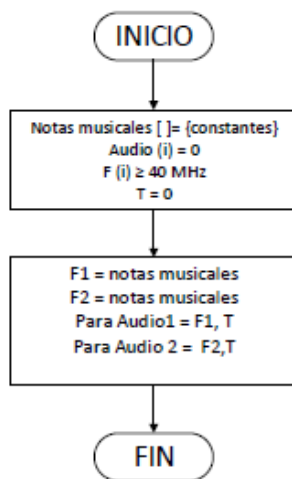


Fig. 23. Rutina de generación de tonos musicales. [Autores]

D. Selección de la plataforma tecnológica a utilizar

Al finalizar la fase de diseño lógico del firmware se seleccionó la plataforma tecnológica a utilizar, considerando criterios tales como la naturaleza open source del hardware y software a utilizar, la existencia de las entradas y salidas suficientes en el hardware (ver Tabla 4), y el conocimiento previo del lenguaje de programación.

Tabla 4.

Número de entradas y salidas necesarias para la operación del firmware. [Autores]

| Variabes | Cantidad |
|---------------------|-----------|
| Entradas operando 1 | 12 |
| Entradas operando2 | 12 |
| Entradas resultado | 16 |
| Entradas operador | 4 |
| Salidas | 3 |
| Total | 47 |

Como resultado del análisis efectuado, se seleccionó la plataforma Arduino Mega 2560 [13], cuyas características técnicas se detallan en la Tabla 5.

Tabla 5.

Características técnicas del microcontrolador Arduino Mega 2560 [13]

| Característica técnica | Valor |
|-------------------------------------|--------------------------|
| Voltaje de operación, V | 5 |
| Voltaje de entrada (recomendado), V | 7-12 |
| Voltaje entrada (límite), V | 6-20 |
| Pines Digitales I/O | 54 (15 salidas para PWM) |
| Pines de entrada análoga | 16 |
| Corriente DC por Pin I/O, mA | 20 |
| Corriente DC para Pin 3.3V, mA | 50 |
| Memoria Flash, KB | 256 |
| SRAM, KB | 8 |
| EEPROM, KB | 4 |
| Velocidad de reloj, MHz | 16 |

E. Fase de implementación del firmware

La fase de implementación del firmware se dividió en 2 subetapas: codificación y pruebas.

Codificación

Para iniciar la codificación, se asignó a cada uno de los pines del Arduino, una de las variables creadas en el diseño lógico (ver Tabla 6). La codificación se realizó en base a los flujogramas descritos en los apartados anteriores, utilizando el software Arduino, basado en lenguaje de programación C++.

Tabla 6.

Asignación de los pines de la plataforma Arduino a cada una de las variables identificadas en el diseño lógico del firmware [Autores]

| Descripción (Variable) | Numeración de PIN |
|-------------------------------|-------------------|
| Operando 1 | |
| Unidad 1 (C) | 2,3,4,5 |
| Decena (B) | 6,7,8,9 |
| Centena (A) | 10,11,12,13 |
| Operando 2 | |
| Unidad (F) | 14,15,16,17 |
| Decena (E) | 18,19,20,21 |
| Centena (D) | 22,23,24,25 |
| Resultado | |
| Unidad (J) | 26,27,28,29 |
| Decena (I) | 30,31,32,33 |
| Centena (H) | 34,35,36,37 |
| Unidad de Mil (G) | 38,39,40,41 |
| Operador matemático | |
| Operador (K) | 42,43,44 |
| Salidas | |
| Igual | 45 |
| Arreglo de LEDs (Correcto) | 52 |
| Arreglo de LEDs (Incorrecto) | 53 |
| Salida de audio | 50 |

Pruebas

En esta subetapa se verificó el desempeño del firmware en la identificación de fichas, la ejecución de las operaciones ingresadas, y la comprobación de respuestas.

Inicialmente se estableció una comunicación serial entre el hardware de programación y un computador. Luego, se comprobó la identificación de fichas, y la ejecución de la operación matemática ingresada, utilizando los protocolos mostrados en las Fig. 24 y 25. La prueba se completó con la validación del funcionamiento del bloque de señalización luminosa (ver Fig. 26 y 27). En total se ejecutó 40 pruebas.

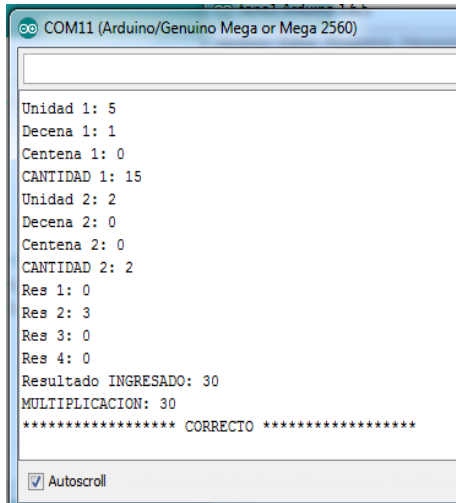


Fig. 24. Protocolo de respuesta correcta para la identificación de fichas y realización de operación. [Fotografía de los autores]

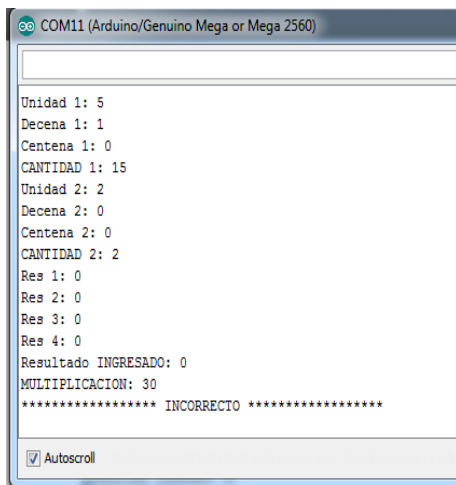


Fig. 25. Protocolo de respuesta incorrecta para la identificación de fichas y realización de operación. [Fotografía de los autores]

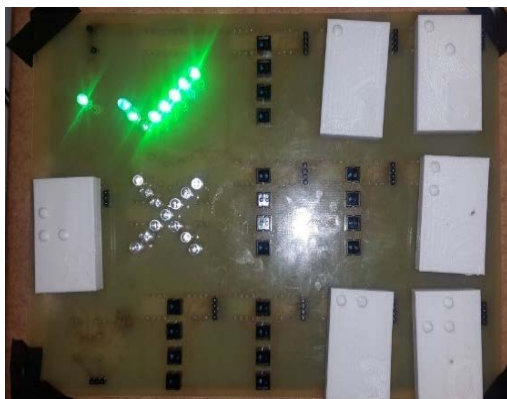


Fig. 26. Señalización de una respuesta correcta en el bloque de señalización luminosa [Fotografía de los autores]

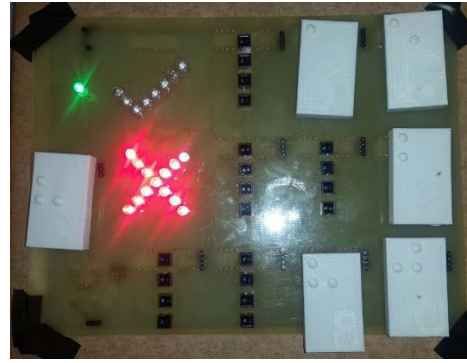


Fig. 27. Señalización de una respuesta incorrecta en el bloque de señalización luminosa [Fotografía de los autores]

Una vez completado el diseño del case del panel, se ensamblaron todos los elementos y se realizó una segunda subetapa de pruebas. Los resultados mostraron inconsistencias imputables al apareamiento de sombras sobre los sensores infrarrojos, por lo cual fue necesario rediseñar el case.

F. Fases de verificación y mantenimiento

Estas etapas se encuentran pendientes de ejecución. La no disponibilidad de recursos no ha posibilitado la conclusión del case final.

IV. CONCLUSIONES

- Se analizó 6 opciones de optimización del panel electrónico implementado, identificando como óptima a la que propone la adopción de fichas tipo LEGO, considerando sus ventajas y desventajas, pero sobre todo la ampliación de las potencialidades de enseñanza de matemáticas.
- Considerando el grado de desarrollo de la capacidad táctil en el grupo beneficiario del proyecto, y, los requerimientos de estética planteados por el equipo de trabajo, se decidió plantear al menos 4 posibles diseños de fichas tipo LEGO a utilizar en el panel (operandos, operadores, y, referencias de posición), de entre las cuales se escogería la opción óptima.
- La elección del hardware para la implementación del firmware, fue el microprocesador ARDUINO MEGA 2560, tomando como requisito principal que debe satisfacer el número de entradas y salidas necesarias.
- El panel electrónico fue diseñado para realizar operaciones sólo con números enteros, considerando que cuando se realice la suma, resta y división los dos operandos puede utilizar cantidades hasta el orden de las centenas, pero para la multiplicación solo se puede utilizar operandos que su resultado no sea mayor a las unidades de mil.
- El case debe ser diseñado de tal manera que este no cause obstrucción entre la ficha y el sensor, ya que esto ocasiona una lectura incorrecta de la ficha.
- Para realizar la comprobación de identificación de las fichas en el firmware fue necesario hacer una comunicación serial entre el microprocesador y el computador

V. REFERENCIAS

- [1] Diseño del Material Didáctico Matemático y Discapacidad Visual en niños de Educación Básica [en línea]. Disponible en <http://memorias.utpl.edu.ec/sites/default/files/documentacion/artes2013/utpl-diseno-material-didactico-matematico.pdf>
- [2] Bermeo. Figueroa. Lima. Ochoa. Peña. Romero. Jaramillo, «Análisis de opciones de optimización de un panel electrónico diseñado para enseñanza de operaciones matemáticas básicas a niños con capacidades visuales especiales,» Loja, 2014.
- [3] Bermeo. Figueroa. Lima. Ochoa. Peña. Romero. Jaramillo, «Diseño de un panel electrónico para la enseñanza de operaciones matemáticas básicas a niños con capacidad visual especial, basado en la utilización de fichas tipo LEGO,» Loja, 2014.
- [4] Jaramillo. Jaramillo, «Diseño del bloque de identificación electrónica de fichas tipo LEGO en un panel diseñado para la enseñanza de operaciones matemáticas básicas a niños con capacidades visuales diferentes» Loja, 2015.
- [5] “Los niños ciegos y su educación”, [en línea]. Consultado 2014-03-11 URL: http://catarina.udlap.mx/u_dl_a/tales/documentos/lda/arteaga_j_g/capitulo2.pdf
- [6] FERNÁNDEZ DEL CAMPO, José Enrique. “Braille y Matemática” [en línea]. Consultado 2014-03-04. URL: http://sid.usal.es/idoscs/f8/fdo10191/braille_y_matematica.pdf
- [7] “Dimensiones de LEGO” [en línea]. Consultado 2014-03-15 URL: <http://www.robertcailliau.eu/Lego/Dimensions/zMeasurements-en.xhtml>
- [8] “Educación con LEGO” [en línea]. Consultado 2014-03-15. URL: <http://education.lego.com/en-us/lego-education-product-database/preschool/9027-lego-duplo-brick-set>
- [9] Software de diseño CAD en 3D, SOLIDWORKS, sitio web, [en línea], consultado 2014-03-11. URL: <http://www.solidworks.com/>
- [10] MakerBot Replicator 2 Desktop 3D Printer, sitio web, [en línea], consultado 2014-04-23. URL: <http://store.makerbot.com/replicator2x>
- [11] Modelos y metodologías para el desarrollo de software [en línea] Consultado 2015-10-03. URL: <http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>
- [12] Pérez, Constantino, «AUDICIÓN HUMANA», [en línea]. Consultado 2015-10-25 URL: <http://personales.unican.es/perezvr/pdf/La%20Audicion%20Humana.pdf>.
- [13] Arduino MEGA 2560, [en línea]. Consultado 2015-09-11 URL: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

FIRMWARE DEL PROTOTIPO DEL PANEL ELECTRÓNICO

```
//- DECLARACIÓN GLOBAL DE VARIABLES PARA LA
//- RESPECTIVA LECTURA DE LOS PINES DE LAS FICHAS

// OPERADOR 1
const int U1[] = {2, 3, 4, 5}, D1[] = {6, 7, 8, 9}, C1[] = {10, 11, 12, 13};
// OPERADOR 2
const int U2[] = {14, 15, 16, 17}, D2[] = {18, 19, 20, 21}, C2[] = {22, 23, 24, 25};
// RESULTADO
const int R1[] = {26, 27, 28, 29}, R2[] = {30, 31, 32, 33}, R3[] = {34, 35, 36, 37}, R4[] =
{38, 39, 40, 41};
// ALMACENAMIENTO
int result, numU1[4], nU1, numD1[4], nD1, numC1[4], nC1, numU2[4], nU2, numD2[4], nD2,
numC2[4], nC2, numR1[4], nR1, numR2[4], nR2, numR3[4], nR3, numR4[4], nR4;
// SIGNO DE OPERACION
const int OP[] = {42, 43, 44};
// IGUAL
const int I = 45;
int Ig;
// RESULTADO'???'
int oper[3], r = 0;
// PRESENTACION DE RESPUESTA
int LEDc = 46, LEDi = 47, f = 0;

// Declaración de variables para VECTORES CON LAS diferentes frecuencia de sonidos
int spk = 50;
int c[5] = {100, 262, 523, 1046, 2093}; // frecuencias 4 octavas de Do
int cs[5] = {59, 277, 554, 1108, 2217}; // Do#
int d[5] = {147, 294, 587, 1175, 2349}; // Re
int ds[5] = {156, 311, 622, 1244, 2489}; // Re#
int e[5] = {165, 330, 659, 1319, 2637}; // Mi
int fl[5] = {175, 349, 698, 1397, 2794}; // Fa
int fs[5] = {185, 370, 740, 1480, 2960}; // Fa#
int g[5] = {196, 392, 784, 1568, 3136}; // Sol
int gs[5] = {208, 415, 831, 1661, 3322}; // Sol#
int a[5] = {220, 440, 880, 1760, 3520}; // La
int as[5] = {233, 466, 932, 1866, 3729}; // La#
int b[5] = {247, 494, 988, 1976, 3951}; // Si

//- DECLARACIONES GENERALES PARA LA COMUNICACION SERIAL
//- Y DECLARACION DE LOS RESPECTIVOS PINES COMO ENTRADA O SALIDAS

void setup() {
  Serial.begin(9600);
  pinMode(LEDc, OUTPUT);
  pinMode(LEDi, OUTPUT);
  pinMode(I, INPUT);
  for (int i = 0; i < 4; i++) {
    pinMode(U1[i], INPUT);
    pinMode(D1[i], INPUT);
    pinMode(C1[i], INPUT);
    pinMode(U2[i], INPUT);
    pinMode(D2[i], INPUT);
    pinMode(C2[i], INPUT);
    pinMode(R1[i], INPUT);
    pinMode(R2[i], INPUT);
    pinMode(R3[i], INPUT);
    pinMode(R4[i], INPUT);
  }
  // ESTABLECE LOS PINES DE LAS FICHA DE OPERACIÓN COMO ENTRADA
  for (int i = 0; i < 3; i++) {
    pinMode(OP[i], INPUT);
  }
}

//- FUNCIÓN PARA EMITIR SONIDO PARA RESPUESTA CORRECTA
void nota(int frec, int t)
{
  tone(spk, frec);
}
```

```

    delay(t);
}

// - FUNCIÓN PRINCIPAL DEL PROGRAMA
void loop() {
    int sw1 = 1, sw2 = 1, sw3 = 1;
    Ig = digitalRead(I);

    // LECTURA DE LOS PINES CORRESPONDIENTES A LAS FICHAS DE CANTIDADES
    for (int i = 0; i < 4; i++) {
        numU1[i] = digitalRead(U1[i]);
        numD1[i] = digitalRead(D1[i]);
        numC1[i] = digitalRead(C1[i]);
        numU2[i] = digitalRead(U2[i]);
        numD2[i] = digitalRead(D2[i]);
        numC2[i] = digitalRead(C2[i]);
        numR1[i] = digitalRead(R1[i]);
        numR2[i] = digitalRead(R2[i]);
        numR3[i] = digitalRead(R3[i]);
        numR4[i] = digitalRead(R4[i]);
    }

    // LECTURA DE LOS PINES CORRESPONDIENTES A LA FICHA DEL OPERADOR
    for (int i = 0; i < 3; i++) {
        oper[i] = digitalRead(OP[i]);
    }
    if (Ig == 1) {

        //          BLOQUE DEL OPERADOR 1
        // CONVERSION DE LOS VALORES BINARIOS A DECIMALES,
        nU1 = binToDec(numU1, 4);
        nD1 = binToDec(numD1, 4);
        nC1 = binToDec(numC1, 4);
        // INEXISTENCIA DE FICHAS EN LAS DIFERENTES POSICIONES (U,D,C)
        // SIN FICHA EN UNIDAD
        if ((nU1 == 16) && ((nD1 != 16) || (nC1 != 16))) {
            sw1 = 0;
        }
        // SIN FICHA EN DECENA
        if ((nD1 == 16) && ((nU1 != 16) || (nC1 != 16))) {
            sw1 = 0;
        }
        // SIN FICHA EN CENTENA
        if ((nU1 != 16) && (nD1 != 16) && (nC1 == 16)) {
            nC1 = 0;
            sw1 = 1;
        }
        // SIN FICHA EN DECENA Y CENTENA
        if ((nU1 != 16) && (nD1 == 16) && (nC1 == 16)) {
            nD1 = 0;
            nC1 = 0;
            sw1 = 1;
        }
    }

    // PRESENTACION EN PANTALLA DE LAS CANTIDADES EN U, D, y C
    // Y TODA LA CANTIDAD DEL OPERADOR
    Serial.print("Unidad 1: ");
    Serial.println(nU1);
    Serial.print("Decena 1: ");
    Serial.println(nD1);
    Serial.print("Centena 1: ");
    Serial.println(nC1);
    int cantidad1 = nC1 * 100 + nD1 * 10 + nU1;
    Serial.print("CANTIDAD 1: ");
    Serial.println(cantidad1);

    //          BLOQUE DEL OPERADOR 2
    // CONVERSION DE LOS VALORES BINARIOS A DECIMALES,
    nU2 = binToDec(numU2, 4);
    nD2 = binToDec(numD2, 4);
    nC2 = binToDec(numC2, 4);
    // INEXISTENCIA DE FICHAS EN LAS DIFERENTES POSICIONES (U,D,C)

```

```

// SIN FICHA EN UNIDAD
if ((nU2 == 16) && ((nD2 != 16) || (nC2 != 16))) {
    sw2 = 0;
}
// SIN FICHA EN DECENA
if ((nD2 == 16) && ((nU2 != 16) || (nC2 != 16))) {
    sw2 = 0;
}
// SIN FICHA EN CENTENA
if ((nU2 != 16) && (nD2 != 16) && (nC2 == 16)) {
    nC2 = 0;
    sw2 = 1;
}
// SIN FICHA EN DECENA Y CENTENA
if ((nU2 != 16) && (nD2 == 16) && (nC2 == 16)) {
    nD2 = 0;
    nC2 = 0;
    sw2 = 1;
}

// PRESENTACION EN PANTALLA DE LAS CANTIDADES EN U, D, y C
//Y TODA LA CANTIDAD DEL OPERADOR
Serial.print("Unidad 2: ");
Serial.println(nU2);
Serial.print("Decena 2: ");
Serial.println(nD2);
Serial.print("Centena 2: ");
Serial.println(nC2);
int cantidad2 = nC2 * 100 + nD2 * 10 + nU2;
Serial.print("CANTIDAD 2: ");
Serial.println(cantidad2);

// BLOQUE DE RESULTADO
// CONVERSION DE LOS VALORES BINARIOS A DECIMALES,
nR1 = binToDec(numR1, 4);
nR2 = binToDec(numR2, 4);
nR3 = binToDec(numR3, 4);
nR4 = binToDec(numR4, 4);
// SIN FICHA EN DECENA, CENTENA Y UNIDAD DE MIL
if ((nR1 != 16) && (nR2 == 16) && (nR3 == 16) && (nR4 == 16)) {
    nR2 = 0;
    nR3 = 0;
    nR4 = 0;
    sw3 = 1;
}
// SIN FICHA EN CENTENA Y UNIDAD DE MIL
if ((nR1 != 16) && (nR2 != 16) && (nR3 == 16) && (nR4 == 16)) {
    nR3 = 0;
    nR4 = 0;
    sw3 = 1;
}
// SIN FICHA EN UNIDAD DE MIL
if ((nR1 != 16) && (nR2 != 16) && (nR3 != 16) && (nR4 == 16)) {
    nR4 = 0;
    sw3 = 1;
}
// SIN FICHA EN UNIDAD
if ((nR1 == 16) && ((nR2 != 16) || (nR3 != 16) || (nR4 != 16))) {
    sw3 = 0;
}
// SIN FICHA EN DECENA
if ((nR2 == 16) && ((nR1 != 16) || (nR3 != 16) || (nR4 != 16))) {
    sw3 = 0;
}
// SIN FICHA EN CENTENA
if ((nR3 == 16) && ((nR1 != 16) || (nR2 != 16) || (nR4 != 16))) {
    sw3 = 0;
}
// PRESENTACION EN PANTALLA DE LAS CANTIDADES EN U, D, y C
// Y TODA LA CANTIDAD DEL RESULTADO
Serial.print("Res 1: ");
Serial.println(nR1);

```



```

Serial.print("Res 2: ");
Serial.println(nR2);
Serial.print("Res 3: ");
Serial.println(nR3);
Serial.print("Res 4: ");
Serial.println(nR4);
int tot1 = nR2 * 10 + nR1;
int tot2 = nR4 * 1000 + nR3 * 100;
int total = tot1 + tot2;
Serial.print("Resultado INGRESADO: ");
Serial.println(total);

// LECTURA DE LOS VALORES DE LA FICHA EN EL OPERADOR MATEMATICA
// SIN FICHA
if ((oper[0] == 1) && (oper[1] == 1) && (oper[2] == 1)) {
  f = 1;
}
// CON FICHA
else {
  f = 0;
}

// LECTURA DE UN VALOR ERRONEO EN LAS FICHAS
// Y PRESENTACIÓN EN PANTALLA
if ((nU1 == 66) || (nD1 == 66) || (nC1 == 66) || (nU2 == 66) || (nD2 == 66) || (nC2 ==
66) || (nR1 == 66) || (nR2 == 66) || (nR3 == 66) || (nR4 == 66)) {
  Serial.println("***** ERROR EN FICHA *****");
  digitalWrite(LEDc, 0);
  digitalWrite(LEDi, 0);
}
// INEXISTENCIA DE FICHAS EN OPERADOR 1, OPERADOR 2, O RESULTADO
//Y PRESENTACION EN PANTALLA
else if ((nU1 == 16) && (nD1 == 16) && (nC1 == 16)) || ((nU2 == 16) && (nD2 == 16) &&
(nC2 == 16)) || ((nR1 == 16) && (nR2 == 16) && (nR3 == 16) && (nR4 == 16)) {
  Serial.println("***** BLOQUE ENTERO SIN FICHA *****");
  digitalWrite(LEDc, 0);
  digitalWrite(LEDi, 0);
}
// COMPARACION DE LOS VALORES CORRECTOS INGRESADOS
// Y PRESENTACION DE RESPUESTA EN PANTALLA, ACTIVACIÓN
// DE LEDS Y PARLANTE
else if ((sw1 == 1) && (sw2 == 1) && (sw3 == 1) && (f == 0)) {
  r = operacion(oper[0], oper[1], oper[2], cantidad1, cantidad2);
  Serial.println(r);
  // COMPARACIÓN
  if (r == total) {
    Serial.println("***** CORRECTO *****");
    digitalWrite(LEDc, 1);
    digitalWrite(LEDi, 0);
    // GENERACION DEL SONIDO PARA RESPUESTA INCORRECTA
    nota (g[3], 500);
    nota (g[3], 300);
    nota (g[2], 200);
    nota (b[2], 300);
    noTone(sp);
    delay (1000);
  }
  //GENERACION DE SEÑAL LUMINOSA PARA RESPUESTA INCORRECTA
  else {
    Serial.println("***** INCORRECTO *****");
    digitalWrite(LEDi, 1);
    digitalWrite(LEDc, 0);
    tone (50, 200, 1000 );
    delay (1000);
    noTone(sp);
    delay (1000);
  }
}
//GENERACION DE SEÑAL LUMINOSA PARA RESPUESTA CORRECTA
else {
  Serial.println("***** SIN FICHA *****");
  digitalWrite(LEDc, 0);
}

```

```

        digitalWrite(LEDi, 0);
    }
}
delay(2000);
}
// FUNCIÓN DE CONVERSION DE BINARIO A DECIMAL
int binToDec (int vect [], int tam) {
    int resul = 0;
    for (int i = 0; i < tam; i++) {
        resul *= 2;
        resul += vect[i];
        if ((resul > 9) && (resul <= 14)) {
            resul = 66;
        }
        else if (resul == 15)
        { resul = 16;
        }
    }
    return resul;
}

// FUNCION DE CONVERSION DE BINARIO A LA OPERACION MATEMÁTICA AGNADA
int operacion(int x, int y, int z, int s, int t) {
    // SUMA
    if ((x == 0) && (y == 0) && (z == 0)) {
        Serial.print("SUMA: ");
        return s + t;
    }
    // RESTA
    if (((x == 0) && (y == 0) && (z == 1)) || ((x == 1) && (y == 0) && (z == 0))) {
        Serial.print("RESTA: ");
        return s - t;
    }
    // MULTIPLICACIÓN
    if ((x == 0) && (y == 1) && (z == 0)) {
        Serial.print("MULTIPLICACION: ");
        return s * t;
    }
    // DIVISIÓN
    if (((x == 0) && (y == 1) && (z == 1)) || ((x == 1) && (y == 1) && (z == 0))) {
        Serial.print("DIVISION: ");
        return s / t;
    }
}
}

```