

¡Contemos!

Lorenzana Soul, Torres, Miguel, and Herrerías, José Pablo

Abstract—Este reporte contiene la metodología y procedimientos que se siguieron con la intención de desarrollar un juego que funcione como herramienta de estimulación para ayudar a las personas con discapacidad a continuar su aprendizaje de una manera divertida. Nos enfocamos en las personas que tienen dificultad de memoria o concentración, ya que esta discapacidad incluye limitaciones o dificultades para aprender una tarea o prestar atención por un tiempo determinado, resultando en que tengan limitaciones para recordar información o realizar actividades en la vida diaria. A través del juego, se implementa el desarrollo cognitivo lo que estimula la comprensión y la concentración

Index Terms—Desarrollo cognitivo, discapacidad, herramienta de estimulación, juego.

- Lorenzana Soul. Estudiante del Instituto Tecnológico y de Estudios Superiores de Monterrey. E-mail: A01338826@itesm.mx
- Torres Miguel. Estudiante del Instituto Tecnológico y de Estudios Superiores de Monterrey. E-mail: A01168868@itesm.mx
- Herrerías, José Pablo. Estudiante del Instituto Tecnológico y de Estudios Superiores de Monterrey. E-mail: A01377189@itesm.mx

1 INTRODUCCIÓN

El objetivo de este proyecto es desarrollar un juego con la finalidad de ayudar a las personas que cuenten con una discapacidad física o mental a que continúen con su aprendizaje de una manera amena.

De acuerdo con la Organización Mundial de la Salud al 2020, alrededor de 1,000 millones de personas en todo el mundo cuentan con alguna discapacidad, esto corresponde al 15% de la población mundial y de las cuales 190 millones cuentan con al menos una dificultad en su funcionamiento y requieren servicios de asistencia de manera frecuente, desafortunadamente este número va en aumento debido al incremento de enfermedades crónicas. [1](Rúa, N. E. G., Moreno-Angarita, M., Toro, M. L., Ochoa, D. A. R., & Sánchez, D. C. S. 2020)

Según el INEGI, reportó en el Censo de Población y Vivienda 2020 que en México hay 6, 179, 890 personas con algún tipo de discapacidad, esto representa al 4.9% de la población total del país. [2] (Rodríguez Pérez, R. E., García Alvarado, F. D. J. 2020).

Es importante mencionar que el INEGI identifica a las personas con discapacidad como aquellas que cuentan con alguna dificultad para llevar a cabo actividades básicas como lo son:

- Ver
- Escuchar
- Caminar
- Recordar
- Concentrarse
- Comunicarse
- Realizar su cuidado personal

A lo largo de este proyecto, nosotros buscamos enfocarnos en aquellas personas que cuentan con dificultad para recordar o concentrarse, ya que esta discapacidad incluye las limitaciones o dificultades para aprender una tarea o para poner atención por un tiempo determinado, lo que los lleva a tener limitaciones para

recordar información o realizar actividades en la vida cotidiana.

Se le denomina como estimulación cognitiva al conjunto de técnicas y estrategias que buscan optimizar la eficacia del funcionamiento de las distintas capacidades y funciones cognitivas, a través de serie de actividades concretas que se estructuran en lo que denominan “programas de entrenamiento cognitivo” [3] (Betancur-Caro, M. L., Molina, D. A., & Cañizales-Romaña, L. Y. 2016)

Como justificación, a lo largo del tiempo se ha visto que las formas de aprendizaje han cambiado, actualmente se ha demostrado que las herramientas más efectivas para que los niños aprendan y desarrollen habilidades es a través del juego, algunos de los beneficios que brinda el aprendizaje a través del juego son[4](Navarro, G. M. 2017)

- Desarrollo cognitivo: El juego es capaz de estimular la imaginación y la creatividad
- Mejor comprensión y capacidad de concentración: Para jugar es necesario comprender la mecánica del juego y a la vez, el interés en formar parte del juego permite que el jugador se mantenga pendiente de los elementos que involucra el juego, por lo que su nivel de concentración aumenta.
- Fomentar la capacidad de expresarse en público: Debido a las diversas situaciones que se dan en el juego, esto genera que el jugador exprese su punto de vistas sobre cómo resolver cada reto que se presente, así se acostumbra a expresar, contrastar y defender sus opiniones de manera clara.

2 PROCEDIMIENTOS Y RESULTADOS

2.1 Procedimientos

Este juego, consta de un tablero que estará compuesto de botones, un display y varios leds. De esta forma, se presentará un mensaje en el display pidiendo al usuario que identifique algún número en específico. Si el usuario oprime el número correcto, aparecerá un mensaje de felicitación en el display y se encenderán los leds con motivo de celebración, en caso contrario, se le pedirá al usuario que lo siga intentando hasta que encuentre el número correcto.

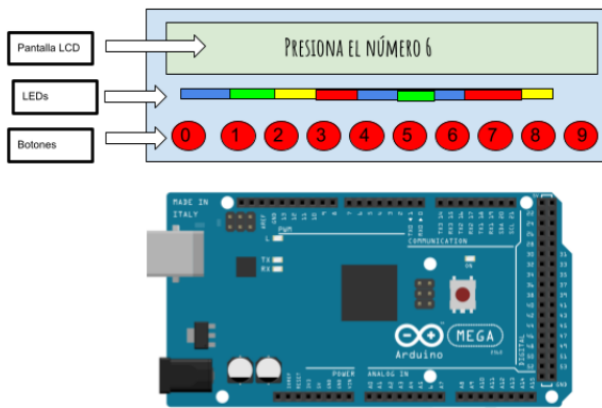


Fig. 1. Prototipo inicial del proyecto

Este juego pretende que el usuario sea capaz de pensar y mejorar su razonamiento de una forma mucho más amena, dinámica y divertida empleando diferentes colores y texturas para que su aprendizaje no sea monótono y poco a poco pueda ir mejorando.

Con la intención de clarificar y especificar la comunicación y el comportamiento de un sistema mediante la interacción con los usuarios, decidimos elaborar un diagrama de caso de uso, que resultó en lo siguiente:



Fig. 2. Diagrama de uso del proyecto

Así mismo se realizó un diagrama de actividad, el cual es un tipo de diagrama de flujo que tiene la intención de dividir en secciones para delegar el trabajo y dividir así los subprocesos entre procesos.

DIAGRAMA DE ACTIVIDAD



Fig. 3. Diagrama de actividad del proyecto

Una vez que se tuvo en mente la idea del juego que se quería desarrollar y la problemática que se buscaba resolver, decidimos enumerar los procesos con los que contaba nuestro proyecto, resultando nos lo siguiente:

- **Alimentación del microcontrolador** - El microcontrolador es alimentado por medio del usb de la computadora.
- **Máquina de estados** - Con la máquina de estados se diseñará una rutina que el usuario tiene que seguir para ir avanzando hasta llegar al final de cada nivel.
- **Detección del botón oprimido** - El usuario pulsa el botón de la opción que cree correcta.
- **Mensaje mostrado en el display** - El display muestra cual es el número que debe de identificar el usuario.
- **Secuencia de Leds** - Una secuencia de Leds enciende para indicar que el usuario está correcto.
- **Casos de fallas** - En caso que se presenten fallas al momento de ejecutar el juego, se tendrá un botón que haga reset al sistema.
- **Pruebas finales** - Una vez que todo el Hardware y Software estén completos y funcionales se realizarán las pruebas finales para analizar el juego de forma completa.

Una vez que se tuvieron en cuenta los procesos mínimos que tenía que tener nuestro proyecto, se definió la lista de hardware, esto con la finalidad de proyectar una idea de qué elementos tenía que tener nuestra simulación en el entorno de Proteus, los elementos que consideramos esenciales fueron los siguientes:

❖ **Arduino ATmega 328P**

Para el desarrollo de este proyecto se necesita una tarjeta de desarrollo con un microcontrolador que cuente con al menos **30** pines de entrada/salida, La corriente de salida para los pines es de 40 mA, por lo que podemos conectar hasta 3 LEDs de 15mA de cada color en cada pin, se necesitarán por lo menos 12 pines para conectar grupos de 2 LEDs a la tarjeta.

- ❖ 12 Leds
- ❖ 23 Resistencias
- ❖ 12 Botones

La corriente de lectura para cada push button es de 0.5mA, por lo que en total se podrían consumir hasta 4mA para los 8 push buttons. Por medio del botón de nivel, se puede seleccionar el nivel de dificultad, entonces bien, para el primer nivel solo se le pide al usuario que identifique las unidades, en el segundo nivel se busca reforzar el aprendizaje de las decenas y en el tercer nivel el de las centenas

❖ **Display LCD**

La pantalla LCD de 16x2 se comunica con el microcontrolador por medio de I2C, tiene un consumo de corriente máxima de 25 mA y puede ser alimentada por 5V, por lo que se encuentra dentro de los parámetros especificados por la tarjeta de desarrollo.

- ❖ Cables
- ❖ Memoria SD

La finalidad de hacer uso de una memoria SD es para poder guardar los datos de cada sesión y así poder observar la evolución del paciente a tratar

Por ello, se hará uso del ATMEGA328 con 8 bits, esto es porque los componentes que se van a emplear para este proyecto no requiere que sean de una gran resolución , ya que la gran mayoría son botones con dos estados. De esta forma se decidió que para esta aplicación lo mejor sería este microcontrolador.

Como se mencionó anteriormente, el software a utilizar a lo largo de la elaboración de este proyecto es Proteus, pues nos permite integrar una extensa variedad de

sistemas, conexiones y entornos de programación que nos permiten diseñar un ambiente muy parecido a la realidad, aunado a lo anterior, Proteus es una aplicación que se puede utilizar para realizar proyectos de construcción de equipos electrónicos en varias etapas: el diseño, programación y construcción de placas de circuito impreso, así como la simulación de todo el dispositivo.

En el caso que se deseara llevar este proyecto a un entorno físico, para generar la PCB, en Proteus se pueden generar los archivos GERBER, estos archivos en la actualidad son necesarios para los fabricantes de circuitos impresos pues dentro de estos archivos encuentran las especificaciones para cada capa que constituye el circuito impreso, y debe contener gráficamente todos los archivos que se quieren imprimir en la capa.

La siguiente lista contiene los recursos que se necesitan para poder construir la electrónica del tablero:, por tanto son considerados como requerimientos del sistema

- 1 Fuente CA a CC de 12V 1A
- 1 Arduino UNO
- 1 Pantalla LCD 2x16 I2C
- 11 Push buttons
- 12 Resistencias 330
- 12 LED's
- 1 memoria SD
- 1 switch

Teniendo en cuenta los materiales y las características con las que contaría nuestro proyecto, decidimos establece el ciclo de vida del mismo, sabemos que el ciclo de vida está compuesto por:



Fig. 4. Diagrama del ciclo de vida de un proyecto

Siendo nuestro primer paso analizar y diseñar el juego con las características que consideramos pertinentes.

Una vez que se identificaron los elementos que componen a nuestro proyecto, evaluamos el costo del proyecto de la siguiente manera:

ELEMENTO	CANTIDAD	PRECIO x UNIDAD	PRECIO TOTAL
Arduino ATmega 328p	1	\$303.00 mxn	\$303.00 mxn
LEDs	12	\$1.00 mxn	\$12.00 mxn
Resistencias	23	\$3.00 mxn	\$69.00 mxn
Botones	12	\$5 mxn	\$60.00 mxn
Display LCD	1	\$52.00 mxn	\$52.00 mxn
Módulo memoria SD	1	\$37.50	\$37.50
Memoria SD 32G	1	\$100.00	\$100.00
		Total	\$633.50 mxn

Fig. 5. Costo aproximado del proyecto

Posteriormente para el desarrollo de hardware se elaboró un plan de trabajo con el tiempo estimado que tomaría resolver cada una de las tareas así como los recursos que se necesitarían en cada una de ellas

Actividad	Tiempo	Recursos
Diagrama de flujo	1 día	Computadora
Máquina de estados	3 días	Arduino UNO Resistencias LEDs
Detección de botón oprimido	3	Arduino UNO Resistencias LEDs
Mensajes en pantalla	3	Arduino UNO LCD
Secuencia de LEDs	3	Arduino UNO Resistencias LEDs
Integración total	5	Arduino UNO Resistencias LEDs LCD
Pruebas Finales	2	Proyecto completo

Fig. 6. Plan de trabajo del proyecto

Esto nos permitió avanzar en el ciclo de vida y lograr la integración del proyecto, lo que posteriormente nos permitió realizar pruebas para asegurarnos que la entrega contaba con las especificaciones idealizadas en un principio.

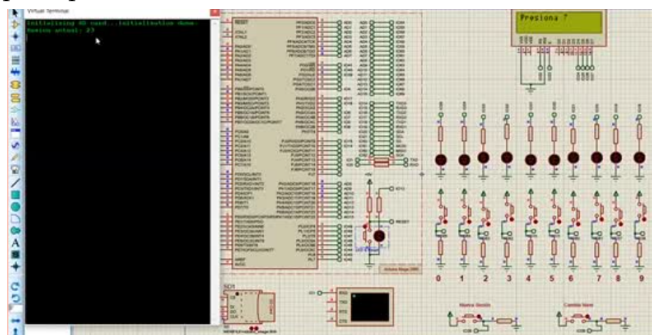


Fig. 7. Esquemático del proyecto

Este juego pretende que el usuario sea capaz de pensar y mejorar su razonamiento de una forma mucho más amena, dinámica y divertida empleando diferentes colores y texturas para que su aprendizaje no sea monótono y poco a poco pueda ir mejorando; sin embargo, este juego puede ser modificado a futuro para resolver alguna otra problemática, no obstante a lo largo del proyecto pudimos observar ciertos hallazgos de desempeño con los que podía contar nuestro proyecto. Inicialmente consideramos los siguientes puntos clave:

- Botones
 - Consideramos que una mejora dentro de nuestro sistema embebido puede ser implementar botones de un tamaño considerable pues creemos que entre más grande sean las dimensiones del botón, es más cómodo para el usuario. Actualmente hacemos uso de unos botones con 10mm de diámetro; sin embargo consideramos que unos botones de 30mm podrían ser más cómodos para el usuario
- Bocina
 - El incluir una bocina al juego puede ayudar a los usuarios que no puedan leer o les cueste trabajo leer desde una pantalla, gracias a la bocina es posible escuchar el número y de ese modo poder identificarlo de forma mucho más sencilla.
- Comunicación vía Bluetooth
 - En lugar de almacenar la información de las sesiones en una tarjeta SD y que tenga que quitarse del dispositivo para leer los datos en una computadora, se podría integrar un módulo Bluetooth para enviar los datos a un dispositivo móvil y que este sea capaz de leer los datos y mostrarlos en una APP.
- Baterías recargables
 - Para poder hacer que el dispositivo sea inalámbrico, se pueden integrar baterías recargables Li-ION junto con el módulo de carga TP4056 que permite cargar las baterías con un cable USB a micro USB y tiene incluida la electrónica de protección ante sobrecarga.

Sin embargo, lo anterior nos obligó a fijarnos objetivos de desempeño, los cuales fueron:

- Tiempo de respuesta
 - Los tiempos de respuesta de nuestro sistema embebido pueden ser mejorados ya que actualmente para detectar la opresión del botón se tiene un rango con la finalidad de que el sistema no lo cuente como una doble opresión o cómo ruido.
- Costos
 - En la parte de electrónica se podrían reducir los costos si se usan componentes de montaje superficial SMD ya que nuestro proyecto no requiere implementación de electrónica para procesamiento adicional al microcontrolador. En cuanto a la carcasa del dispositivo, podría usarse MDF para reducir los costos en lugar de plástico o algún material más costoso.
- Tamaño
 - Los botones son los que limitan el tamaño de nuestro dispositivo, en este caso se usarían botones tipo “arcade” que tienen un diámetro de 10mm cada uno. La aplicación de nuestro dispositivo no requiere que su tamaño sea pequeño, es preferible que tenga un tamaño de 30mm para que los usuarios puedan interactuar con él cómodamente.

Para verlo de manera más clara, desarrollamos el siguiente diagrama de desempeño:

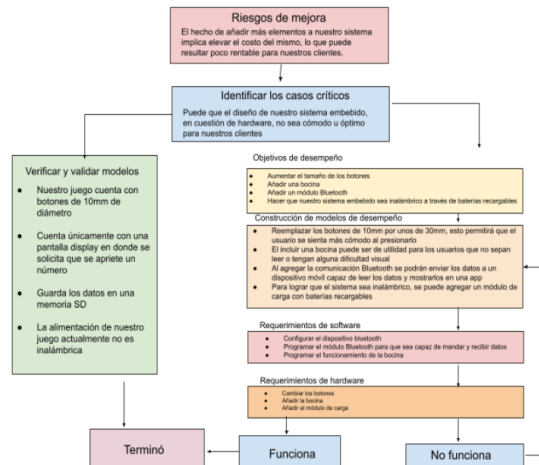


Fig. 8. Diagrama de desempeño del proyecto

Por eso, en cuanto a modelos de diseño, consideramos que el modelo de procesos unificados es el más adecuado para la realización de este proyecto, ya que este va checando que cada elemento del proyecto vaya funcionando además nos permite realizar modificaciones a lo largo del proceso sin sufrir retrasos

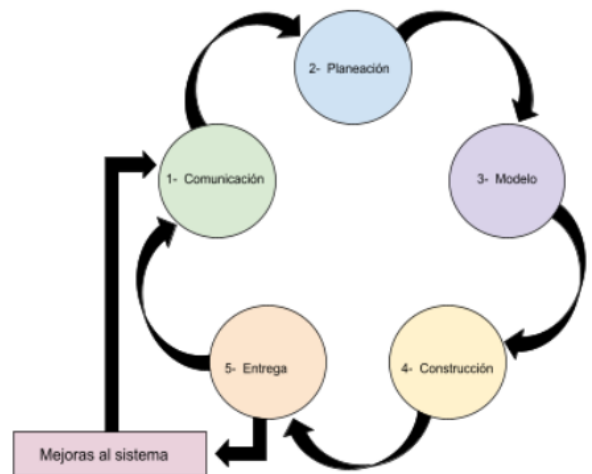


Fig. 9. Diagrama del modelo de procesos unificados

A lo largo de este proyecto, como equipo consideramos que el más adecuado para el desarrollo del mismo sería el *proceso iterativo*, pues este consta de las 5 fases esenciales en todo proceso; sin embargo, su distintivo es que la comunicación es vital y constante, de esta manera, nos podemos enfocar en la solución de posibles fallas e ir adaptando el modelo para que funcione en óptimas condiciones y bajo circunstancias poco favorables

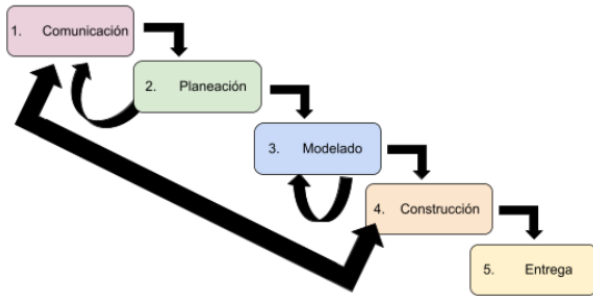


Fig. 10. Diagrama del proceso iterativo.

- Problema al compartir el archivo .pdsprj : El problema que se ocasionó fue que si algún miembro del equipo quería compartir su avance relacionado al proyecto, y únicamente mandaba el archivo con terminación .pdsprj, cuando los demás integrantes del equipo intentaban descargarlo, el archivo se perdía y se generaban carpetas aleatorias con contenido nulo. Para evitar lo anterior, se decidió que al compartir algún avance se realizará un zip del proyecto entero para evitar así modificaciones o alteraciones.
- Problema al leer la memoria SD: conectamos una memoria SD en nuestro esquemático; sin embargo esta no funcionaba correctamente al final se tuvo que usar el SDK de arduino para generar y compilar los archivos .hex y posteriormente importarlos en el microcontrolador de proteus y adicionalmente se modificó la frecuencia del microcontrolador a 16MHz

Por el contrario, para obtener una idea del código que debemos implementar para que nuestro proyecto funcione de manera óptima con lo ya antes mencionado, elaboramos un diagrama de secuencia.



Fig. 11. Diagrama de secuencia del proyecto.

Esto nos permitió establecer nuestras tareas y prioridades dentro del proyecto, siendo:

1. Generar el número aleatorio en el display
2. Leer tecla oprimida
3. Prender leds y bocina
4. Secuencia de leds
5. Guardar datos de la sesión
6. Mostrar datos de las sesiones
7. Reset
 - Es una interrupción por lo que es considerada de prioridad alta

Afortunadamente, los problemas que obtuvimos durante el proyecto fueron mínimos y se resumen en:

2.2 Diagrama de flujo

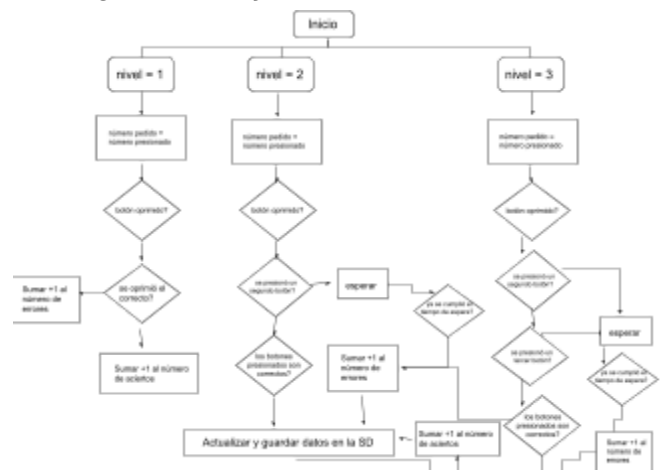


Fig. 12. Diagrama de flujo del proyecto.

Como se mencionó anteriormente, en nuestro proyecto se tienen 3 niveles de dificultad, con la finalidad de que los usuarios que hagan uso de nuestro juego puedan repasar las unidades, decenas y centenas.

Por medio de un switch se puede seleccionar el nivel de dificultad, entonces bien, para el primer nivel solo se le pide al usuario que identifique las unidades, en el caso de que el botón que apriete el usuario no corresponda con el solicitado, se contará como un error mientras que si es correcto, contará como acierto.

En el segundo nivel se busca reforzar el aprendizaje de las decenas, por tanto, es necesario presionar dos botones, en caso de que el usuario únicamente presione un botón en vez de dos, se le dará una tolerancia al usuario, una vez cumplida esta tolerancia y si no se presiona un segundo botón, se tomará como un resultado incorrecto.

Lo mismo sucede para el nivel tres donde se busca repasar el tema de las centenas.

2.3 Código, explicación y evidencias de la simulación

Inicialmente, se asignaron pines a cada uno de los componentes que componen nuestro proyecto, entre estos:

- Botones
- Memoria SD
- Leds
- Pantalla LCD

Así mismo, se declararon algunas funciones que serán de utilidad más adelante para el código

Cuando el juego se prende, aparece un mensaje de bienvenida, y el juego es capaz de iniciar una sesión con un número aleatorio gracias a una función, gracias a esta función, siempre que empiece el juego se nos pedirá un número diferente sin importar el grado de dificultad del mismo.

Así mismo, en caso de que el valor que insertemos corresponda con el valor solicitado, se desplegará una felicitación de manera aleatoria, gracias a que se creó una función que lleva por nombre “felicidades” la cual contiene diversas felicitaciones, y en caso de que nuestro resultado sea incorrecto se desplegará el mensaje “Otra vez” con la finalidad de que el usuario se motive a realizar un segundo intento

```
#include <LiquidCrystal.h>
#include <SPI.h>
#include <SD.h>

File myFile;
const int chipSelect = 53;
int limiteNivel=3;
long ran=0;
int intentos=0;
int incorrectos=0;
int correctos=0;
int sesion=0;
int nivel=1;
int randSeq=1;
int porcentaje=0;
char numA[20];
String numToInt="";
int numAint=0;
String sesionStr="";
char temp='A';
String resultados="";
char resultadosTemp='A';
int ranSup[6];
int usuarioSup[6];
int ite=0; //Itera en ranSup

//-----LCD-----
const int rs = 22, en = 23, d4 = 24, d5 = 25, d6 = 26, d7 = 27;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
//-----BOTONES-----
const int botonCero = 65;
const int botonUno = 64;
const int botonDos = 63;
const int botonTres = 62;
const int botonCuatro = 10;
const int botonCinco = 11;
const int botonSeis = 60;
const int botonSiete = 61;
const int botonOcho = 37;
const int botonNueve = 36;
const int botonSesion = 38;
const int botonNivel = 39;
```

```
//-----ESTADOD DEL BOTON
int BP=0; //Boton presionado
int buttonState0 = 0;
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
int buttonState5 = 0;
int buttonState6 = 0;
int buttonState7 = 0;
int buttonState8 = 0;
int buttonState9 = 0;
int buttonStateSesion = 0;
int buttonStateNivel = 0;

//-----LEDS-----
const int ledPin0 = 35;
const int ledPin1 = 34;
const int ledPin2 = 33;
const int ledPin3 = 32;
const int ledPin4 = 31;
const int ledPin5 = 30;
const int ledPin6 = 21;
const int ledPin7 = 20;
const int ledPin8 = 19;
const int ledPin9 = 18;

//
void setup(){
  Serial.begin(9600);
  while (!Serial) {
  }
  Serial.print("Initializing SD card...");
  if (!SD.begin(chipSelect)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
  lcd.begin(16, 2);
  lcd.print("BIENVENIDO");
  pinMode(ledPin0, OUTPUT);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
  pinMode(ledPin5, OUTPUT);
  pinMode(ledPin6, OUTPUT);
  pinMode(ledPin7, OUTPUT);
  pinMode(ledPin8, OUTPUT);
  pinMode(ledPin9, OUTPUT);
  pinMode(botonCero, INPUT);
  pinMode(botonUno, INPUT);
  pinMode(botonDos, INPUT);
  pinMode(botonTres, INPUT);
  pinMode(botonCuatro, INPUT);
  pinMode(botonCinco, INPUT);
  pinMode(botonSeis, INPUT);
  pinMode(botonSiete, INPUT);
  pinMode(botonOcho, INPUT);
  pinMode(botonNueve, INPUT);
  pinMode(botonSesion, INPUT);
  pinMode(botonNivel, INPUT);
```

```

    lcd.setCursor(0, 0);
    lcd.print("Arduotive");
    lcd.setCursor(0, 1);
    lcd.print("Arduo");
    lcd.setCursor(0,0);
    delay(200);

    generaRandSup();
    ite=0;
    getSesion();
}

void loop(){
    BP=obtenBoton();
    if(BP!=666){
        if(BP==ranSup[ite]){
            ++correctos;
            usuarioSup[ite]=BP;

            if(ite>=nivel-1){
                felicidades();
                ite=0;
                generaRandSup();
            }else{
                ite++;
                imprimeRandSup();
            }
            guardaSD();

        }else if(BP!=20 && BP!=30){
            ++incorrectos;
            lcd.clear();
            lcd.print("Otra Vez");
            delay(1000);
            imprimeRandSup();
            guardaSD();
        }
    }

    if(BP==20){
        nuevaSesion();
        lcd.clear();
        lcd.print("Sesion #");
        lcd.print(sesion);
        delay(2000);
        generaRandSup();
    }

    if(BP==30){
        cambiaNivel();
        lcd.clear();
        lcd.print("Nivel #");
        lcd.print(nivel);
        delay(2000);
        generaRandSup();
    }
}

void generaRandSup(){
    for(int i=0;i<nivel;i++){
        ranSup[i]=random(10);
    }
    for(int i=nivel;i<6;i++){
        ranSup[i]=10;
    }
    lcd.clear();
    lcd.print("Presiona ");
    for(int i=0;i<nivel;i++){
        lcd.print(ranSup[i]);
    }
}

void imprimeRandSup(){
    lcd.clear();
    lcd.print("Presiona ");
    for(int i=0;i<nivel;i++){
        lcd.print(ranSup[i]);
    }
    lcd.setCursor(0, 1);
    for(int i=0;i<ite;i++){
        lcd.print(usuarioSup[i]);
    }
    lcd.setCursor(0, 0);
}

```



```

void cambiaNivel(){
if(nivel<limiteNivel){
    ++nivel;
}else if(nivel==limiteNivel){
    nivel=1;
}else{
    nivel=1;
}
}

void nuevaSesion(){
    correctos=0;
    incorrectos=0;
    getSesion();
    SD.remove("sesion.txt");
    myFile = SD.open("sesion.txt", FILE_WRITE);
if (myFile) {
    sesion=sesion+1;
    myFile.println(sesion);
    myFile.close();
} else {
    Serial.println("error opening test.txt");
}
Serial.print("Nueva sesion: ");
Serial.println(sesion);
}

void getSesion(){
    myFile = SD.open("sesion.txt");
if (myFile) {
while (myFile.available()) {
    temp=myFile.read();
    sesionStr+=temp;
    //Serial.write(myFile.read());
}
    myFile.close();
} else {
    Serial.println("error opening sesion.txt");
}
    sesion=sesionStr.toInt();
    Serial.print("Sesion actual: ");
    Serial.println(sesion);
    sesionStr="";
}

```

```

void guardaSD(){
    myFile = SD.open("juego.txt");
if (myFile) {
    resultados="";
while (myFile.available()) {
    resultadosTemp=myFile.read();
    resultados+=resultadosTemp;
}
    //Serial.println(resultados);
    myFile.close();
} else {
    Serial.println("error opening juego.txt");
}
    String indexOfTemp="Sesion #";
    indexOfTemp+=sesion;
    int index=resultados.indexOf(indexOfTemp);
    intentos=correctos+incorrectos;
if (intentos>0){
    porcentaje=(correctos*100)/intentos;
}else{
    porcentaje=0;
}
if (index != -1){
    //Serial.println("SI HAY");
    indexOfTemp="";
for (int i=0; i<index;i++){
    indexOfTemp+=resultados[i];
}
    resultados=indexOfTemp;
    resultados+="Sesion #";resultados+=sesion;
    resultados+="\r\nNivel: ";resultados+=nivel;
    resultados+=" C/I: ";resultados+=porcentaje;resultados+=" %";
    resultados+=" Intentos: ";resultados+=intentos;
    resultados+=" Correcto: ";resultados+=correctos;
    resultados+=" Incorrecto: ";resultados+=incorrectos;
}else{
    //Serial.println("NO HAY");
    resultados+="\r\nSesion #";resultados+=sesion;
    resultados+="\r\nNivel: ";resultados+=nivel;
    resultados+=" C/I: ";resultados+=porcentaje;resultados+=" %";
    resultados+=" Intentos: ";resultados+=intentos;
    resultados+=" Correcto: ";resultados+=correctos;
    resultados+=" Incorrecto: ";resultados+=incorrectos;
}
    SD.remove("juego.txt");
    myFile = SD.open("juego.txt", FILE_WRITE);
if (myFile) {
    myFile.print(resultados);
    myFile.close();
} else {
    Serial.println("error opening juego.txt");
}
    myFile.close();
    myFile = SD.open("juego.txt");
if (myFile) {
while (myFile.available()) {
    Serial.write(myFile.read());
}
    myFile.close();
} else {
    Serial.println("error opening juego.txt");
}
}

void generaRand(){
    ran=random(10);
    lcd.clear();
    lcd.print("Presiona ");
    lcd.print(ran);
}

```

```

int obtenBoton(){
    int returnB=666;
    buttonState0 = digitalRead(botonCero);
    buttonState1 = digitalRead(botonUno);
    buttonState2 = digitalRead(botonDos);
    buttonState3 = digitalRead(botonTres);
    buttonState4 = digitalRead(botonCuatro);
    buttonState5 = digitalRead(botonCinco);
    buttonState6 = digitalRead(botonSeis);
    buttonState7 = digitalRead(botonSiete);
    buttonState8 = digitalRead(botonOcho);
    buttonState9 = digitalRead(botonNueve);
    buttonStateSesion = digitalRead(botonSesion);
    buttonStateNivel = digitalRead(botonNivel);

    if(buttonStateSesion == HIGH){
        returnB=20;
    }

    if(buttonStateNivel == HIGH){
        returnB=30;
    }

    if(buttonState0 == HIGH){
        digitalWrite(ledPin0,HIGH);
        returnB=0;
    }

    if(buttonState1 == HIGH){
        digitalWrite(ledPin1,HIGH);
        returnB=1;
    }

    if(buttonState2 == HIGH){
        digitalWrite(ledPin2,HIGH);
        returnB=2;
    }

```

```

    if(buttonState3 == HIGH){
        digitalWrite(ledPin3,HIGH);
        returnB=3;
    }

    if(buttonState4 == HIGH){
        digitalWrite(ledPin4,HIGH);
        returnB=4;
    }

    if(buttonState5 == HIGH){
        digitalWrite(ledPin5,HIGH);
        returnB=5;
    }

    if(buttonState6 == HIGH){
        digitalWrite(ledPin6,HIGH);
        returnB=6;
    }

    if(buttonState7 == HIGH){
        digitalWrite(ledPin7,HIGH);
        returnB=7;
    }

    if(buttonState8 == HIGH){
        digitalWrite(ledPin8,HIGH);
        returnB=8;
    }

    if(buttonState9 == HIGH){
        digitalWrite(ledPin9,HIGH);
        returnB=9;
    }

```

```

digitalWrite(ledPin0,LOW);
digitalWrite(ledPin1,LOW);
digitalWrite(ledPin2,LOW);
digitalWrite(ledPin3,LOW);
digitalWrite(ledPin4,LOW);
digitalWrite(ledPin5,LOW);
digitalWrite(ledPin6,LOW);
digitalWrite(ledPin7,LOW);
digitalWrite(ledPin8,LOW);
digitalWrite(ledPin9,LOW);
delay(250);
return returnB;
}

```

```

void sequence1(){
    int vuelta;
    for( vuelta = 0; vuelta < 3; vuelta = vuelta + 1 ){
        digitalWrite(ledPin0,HIGH);
        delay(25);
        digitalWrite(ledPin1,HIGH);
        delay(25);
        digitalWrite(ledPin2,HIGH);
        delay(50);
        digitalWrite(ledPin3,HIGH);
        delay(50);
        digitalWrite(ledPin4,HIGH);
        delay(75);
        digitalWrite(ledPin5,HIGH);
        delay(75);
        digitalWrite(ledPin6,HIGH);
        delay(100);
        digitalWrite(ledPin7,HIGH);
        delay(100);
        digitalWrite(ledPin8,HIGH);
        delay(125);
    }
}

```

```

digitalWrite(ledPin9,HIGH);
    delay(125);
digitalWrite(ledPin9,LOW);
    delay(25);
digitalWrite(ledPin8,LOW);
    delay(25);
digitalWrite(ledPin7,LOW);
    delay(50);
digitalWrite(ledPin6,LOW);
    delay(50);
digitalWrite(ledPin5,LOW);
    delay(75);
digitalWrite(ledPin4,LOW);
    delay(75);
digitalWrite(ledPin3,LOW);
    delay(100);
digitalWrite(ledPin2,LOW);
    delay(100);
digitalWrite(ledPin1,LOW);
    delay(125);
digitalWrite(ledPin0,LOW);
    delay(125);

}
}

```

```

void sequence2(){
    int vuelta;
    for( vuelta = 0; vuelta < 3; vuelta = vuelta + 1 ){
        digitalWrite(ledPin0,HIGH);
        delay(25);
        digitalWrite(ledPin9,HIGH);
        delay(25);
        digitalWrite(ledPin1,HIGH);
        delay(50);
        digitalWrite(ledPin8,HIGH);
        delay(50);
        digitalWrite(ledPin2,HIGH);
        delay(75);
        digitalWrite(ledPin7,HIGH);
        delay(75);
        digitalWrite(ledPin3,HIGH);
        delay(100);
        digitalWrite(ledPin6,HIGH);
        delay(100);
        digitalWrite(ledPin4,HIGH);
        delay(125);
        digitalWrite(ledPin5,HIGH);
        delay(125);
        digitalWrite(ledPin0,LOW);
        delay(25);
        digitalWrite(ledPin9,LOW);
        delay(25);
        digitalWrite(ledPin1,LOW);
        delay(50);
        digitalWrite(ledPin8,LOW);
        delay(50);
        digitalWrite(ledPin2,LOW);
        delay(75);
        digitalWrite(ledPin7,LOW);
        delay(75);
        digitalWrite(ledPin3,LOW);
        delay(100);
    }
}

```

```

        digitalWrite(ledPin6,LOW);
        delay(100);
        digitalWrite(ledPin4,LOW);
        delay(125);
        digitalWrite(ledPin5,LOW);
        delay(125);
    }
}

```

```

void sequence3(){
    int vuelta;
    for( vuelta = 0; vuelta < 3; vuelta = vuelta + 1 ){
        digitalWrite(ledPin0,HIGH);
        delay(25);
        digitalWrite(ledPin2,HIGH);
        delay(25);
        digitalWrite(ledPin4,HIGH);
        delay(50);
        digitalWrite(ledPin6,HIGH);
        delay(50);
        digitalWrite(ledPin8,HIGH);
        delay(75);
        digitalWrite(ledPin1,HIGH);
        delay(75);
        digitalWrite(ledPin3,HIGH);
        delay(100);
        digitalWrite(ledPin5,HIGH);
        delay(100);
        digitalWrite(ledPin7,HIGH);
        delay(125);
        digitalWrite(ledPin9,HIGH);
        delay(125);
        digitalWrite(ledPin0,LOW);
        delay(25);
        digitalWrite(ledPin2,LOW);
        delay(25);
        digitalWrite(ledPin4,LOW);
        delay(50);
    }
}

```

```

        digitalWrite(ledPin6,LOW);
        delay(50);
        digitalWrite(ledPin8,LOW);
        delay(75);
        digitalWrite(ledPin1,LOW);
        delay(75);
        digitalWrite(ledPin3,LOW);
        delay(100);
        digitalWrite(ledPin5,LOW);
        delay(100);
        digitalWrite(ledPin7,LOW);
        delay(125);
        digitalWrite(ledPin9,LOW);
        delay(125);
    }
}

```

```

void felicidades(){
    lcd.clear();
    int rando=random(1,5);
    switch(rando){
        case 1:
            lcd.print("Muy bien!");
            break;
        case 2:
            lcd.print("Bien Hecho!");
            break;
        case 3:
            lcd.print("Excelente");
            break;
        case 4:
            lcd.print("Genial!");
            break;
        case 5:
            lcd.print("Buen Trabajo!");
            break;
    }
    randSeq=random(1,3);
    if(randSeq==1){
        sequence1();
    }
    if(randSeq==2){
        sequence2();
    }
    if(randSeq==3){
        sequence3();
    }
}

```

Una vez comprendido lo anterior, pusimos a prueba el código con la finalidad de comprobar que contaba con todas las características y especificaciones pensadas en un inicio.

Como lo mencionamos anteriormente, uno de nuestros más grandes problemas a lo largo de la realización del proyecto fue hacer funcionar la memoria SD, por ello, decidimos corroborar que esto funcionara. Entonces bien, decidimos asegurarnos que nuestro proyecto era capaz de crear y guardar sesiones con información importante tal como:

- Nivel de dificultad
- Errores
- Aciertos
- Porcentaje de evolución
 - Es un porcentaje que representa errores contra aciertos

Sesion #11			
Nivel: 3	Intentos: 53	Incorrecto: 41	Correcto: 222
Sesion #12			
Nivel: 2	Intentos: 43	Incorrecto: 14	Correcto: 32
Sesion #13			
Nivel: 4	Intentos: 234	Incorrecto: 23443	Correcto: 32

Como lo podemos observar en la figura anterior, la memoria SD

guarda los valores obtenidos a través del juego en un documento con extensión .txt con la finalidad de que se pueda evaluar el avance o evolución del usuario.

3 CONCLUSION

En conclusión, al ser un juego que tiene como finalidad ayudar a las personas que tengan alguna discapacidad este puede expandirse en muchos aspectos, desde el hecho que se puede agregar una bocina para que los usuarios que no puedan leer puedan escuchar las instrucciones, la temática puede ser diferente, en este caso fue identificar números, pero de igual forma se puede expandir a reconocer sonidos de animales, identificación de animales, de acciones, colores. Las temáticas que se manejan pueden ser muy variadas y personalizadas para que cubran las necesidades que se requiere. De igual forma al momento de desplegar los resultados una mejora que se puede considerar es que estos mismo se pueden exportar a Excel, de esa forma se pueden hacer diferentes procedimientos para el análisis de estos datos y que no solamente se quede en un juego, más bien que sea una herramienta que sirva para poder evaluar el desempeño de los usuarios de una manera divertida e interactiva para ellos. El objetivo propuesto en un principio se logró cumplir de forma correcta, a pesar que se ocupó un simulador para desarrollar este proyecto se pudo lograr todo lo propuesto de forma correcta. El proyecto fue trabajado y analizado desde diferentes puntos de vista, tanto funcional como económico.

LINK DE LA PRESENTACIÓN:

https://www.canva.com/design/DAEFD3AIMRS/SHARE/PRVIEW?TOKEN=HVAV2BSLQIDXWAGEBELOUQ&ROLE=EDITOR&UTM_CONTENT=DAEFD3AIMRS&UTM_CAMPAIGN=DESIGNSHARE&UTM_MEDIUM=LINK&UTM_SOURCE=SHAREBUTTON

REFERENCES

- [1] Rúa, N. E. G., Moreno-Angarita, M., Toro, M. L., Ochoa, D. A. R., & Sánchez, D. C. S. (2020). Personas con discapacidad, salud laboral y capacidades en las leyes del 2011-2020. Revista Ciencia y Cuidado, 17(2), 116-128.
- [2] Rodríguez Pérez, R. E., García Alvarado, F. D. J. (2020). Desigualdad salarial entre trabajadores con y sin discapacidad en México, ¿Discriminación o menor productividad?. Ensayos. Revista de economía, 39(2), 217-245.
- [3] Betancur-Caro, M. L., Molina, D. A., & Cañizales-Romaña, L. Y. (2016). Entrenamiento cognitivo de las funciones ejecutivas en la edad escolar. Revista Latinoamericana de Ciencias Sociales, Niñez y Juventud.
- [4] Navarro, G. M. (2017). Tecnologías y nuevas tendencias en educación: aprender jugando. El caso de Kahoot. Opción: Revista de Ciencias Humanas y Sociales, (83), 252-277.