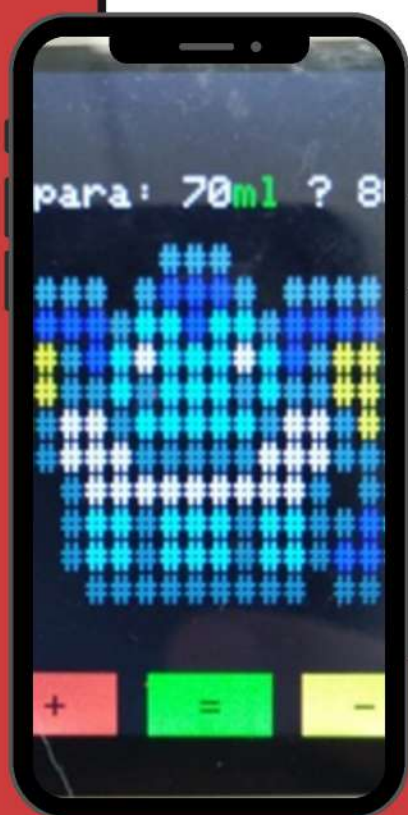




MATERIAL DIDÁCTICO ÁLGEBRA

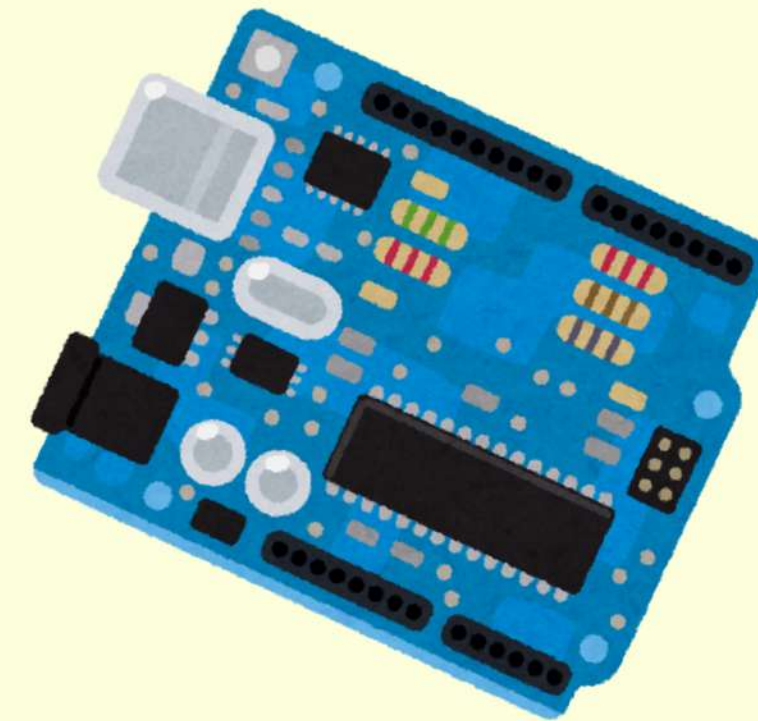
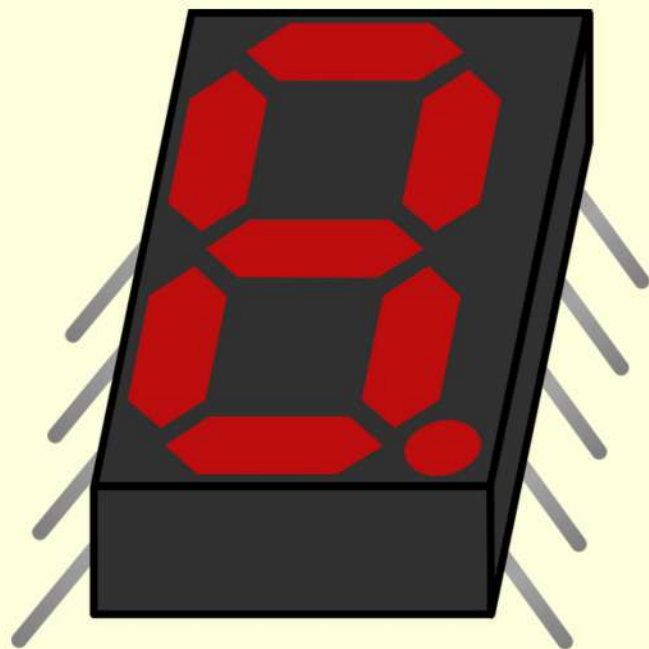
Por Mauricio García Galindo.





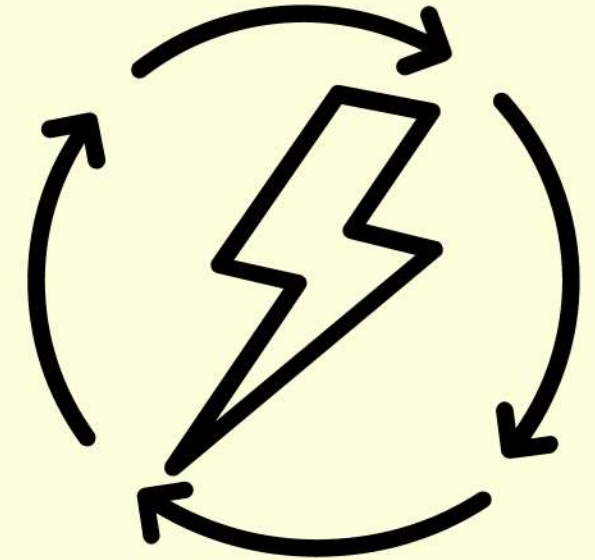
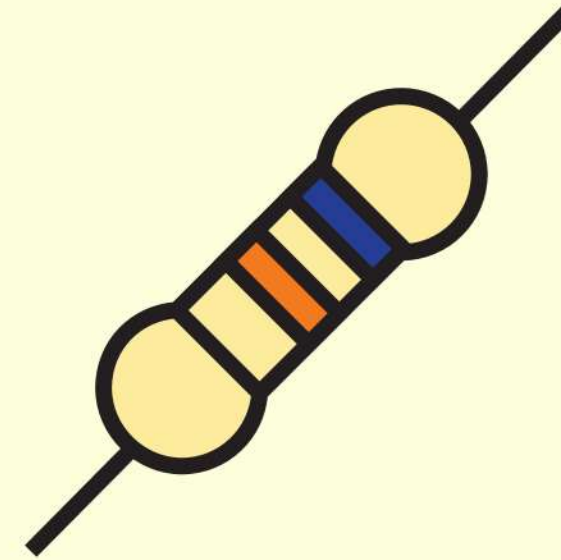
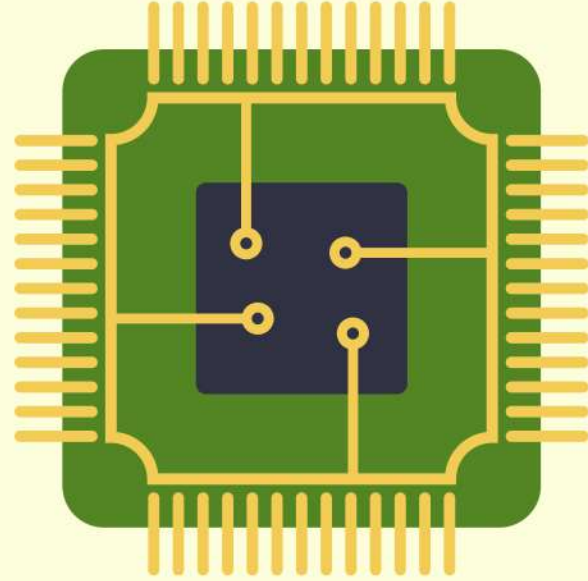
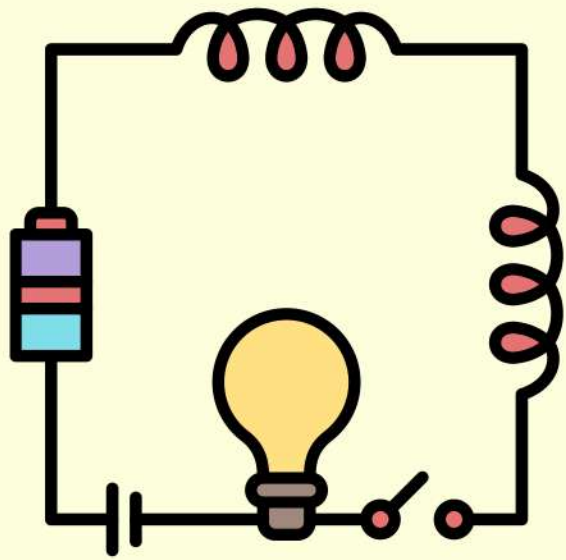
Resumen

Se documentó la **creación de un material didáctico** elaborado **con componentes electrónicos** y su aplicación en un salón de 5° grado de primaria en la escuela Anexa Pablo Cantú Villarreal. **La investigación necesaria en el área de programación se fundamentó en un paradigma funcional**, con un enfoque en **“que” se está haciendo y no en “como” se está haciendo**. Se concluyó que **este tipo de material puede ser funcional en el proceso de aprendizaje significativo de los niños**.



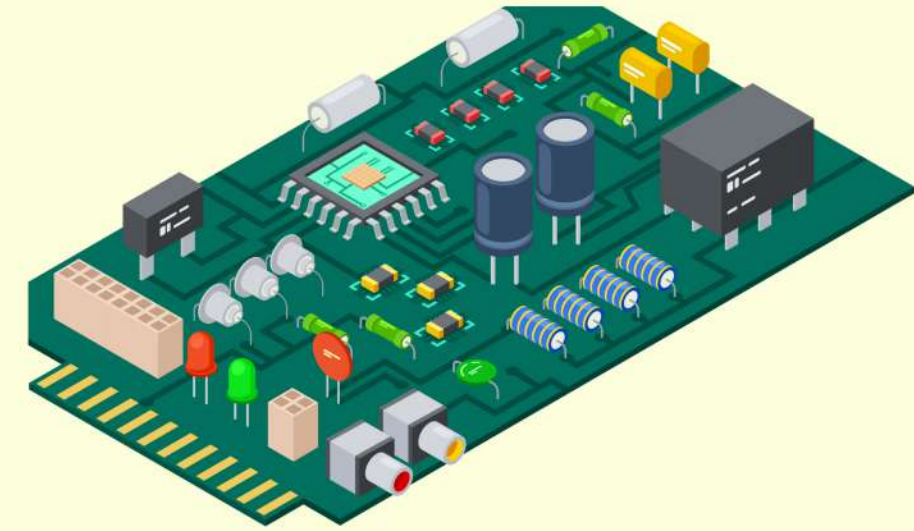
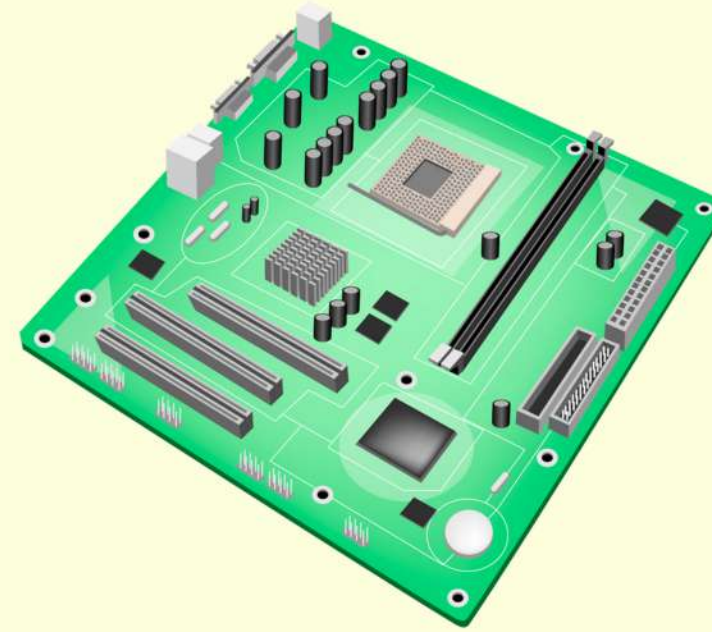
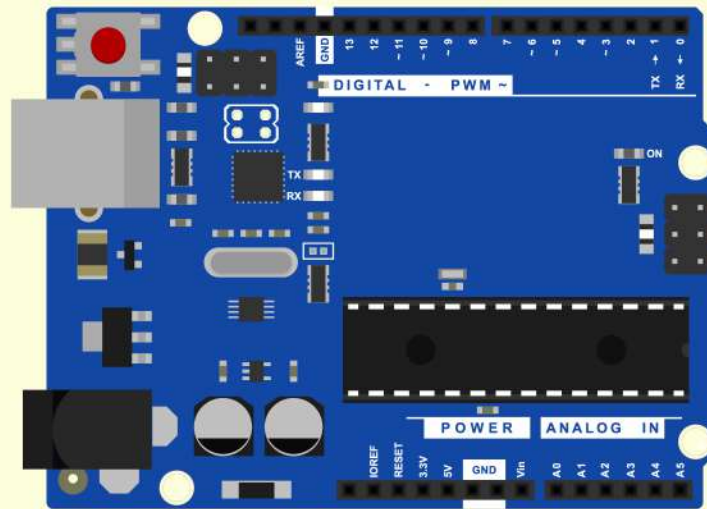
Palabras clave

- **Display de 7 segmentos:** Componente integrado por siete segmentos que se pueden encender o apagar individualmente.
- **Código:** Estructuran el lenguaje de programación, que a su vez es el encargado de garantizar el correcto funcionamiento de las aplicaciones o programas que permiten una buena comunicación entre el usuario y la computadora.
- **Arduino UNO:** Es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc.



Palabras clave

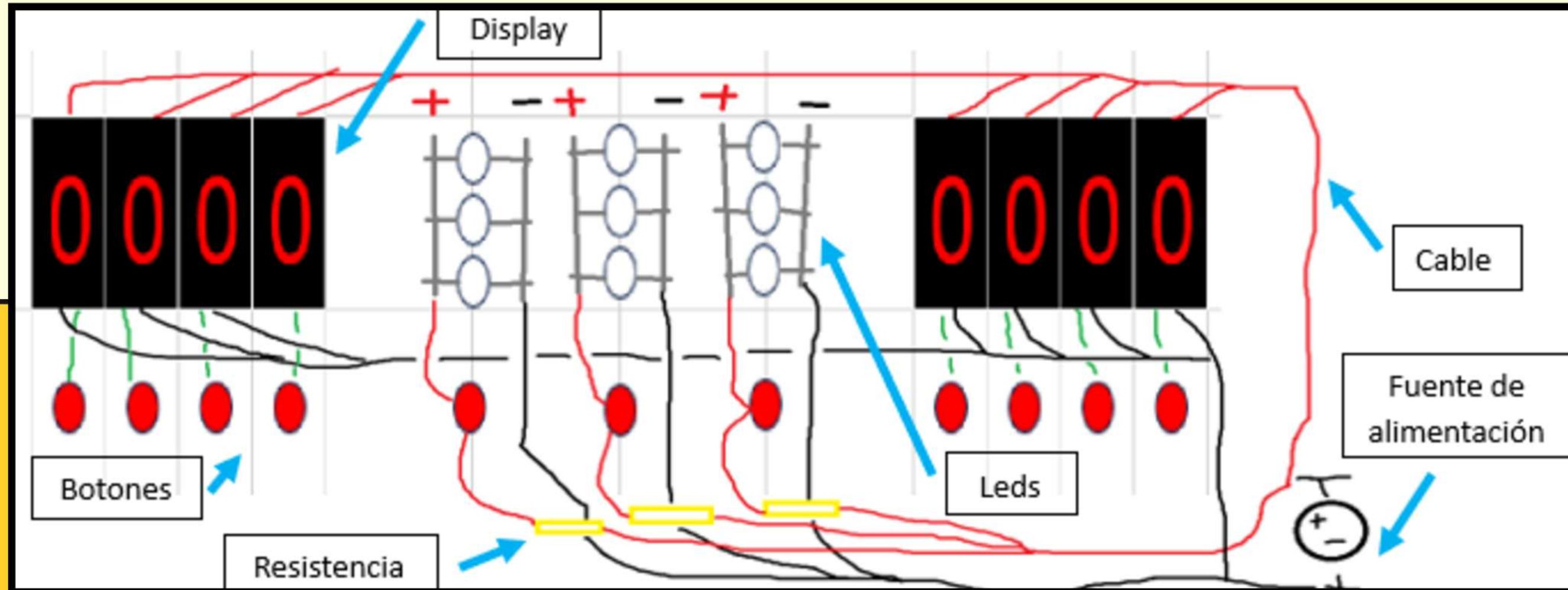
- **Diagrama:** Representación pictórica de un circuito eléctrico.
- **Chips:** Semiconductor que contiene múltiples circuitos integrados con los que se realizan numerosas funciones en computadoras y dispositivos electrónicos.
- **Led:** Diodo semiconductor que emite luz cuando se le aplica tensión.
- **Resistencia:** Es la oposición al flujo de corriente en un circuito eléctrico.
- **Voltaje:** Indicador de la capacidad de mover la electricidad.



Palabras clave

- **Tarjetas electrónicas con microprocesadores de bajo consumo programables:** Son dispositivos que cuentan con un microcontrolador (microchip) programable que puede ejecutar diferentes instrucciones y, por lo tanto, son adecuadas para crear todo tipo de dispositivos 'inteligentes', incluidos robots.

Primer diseño

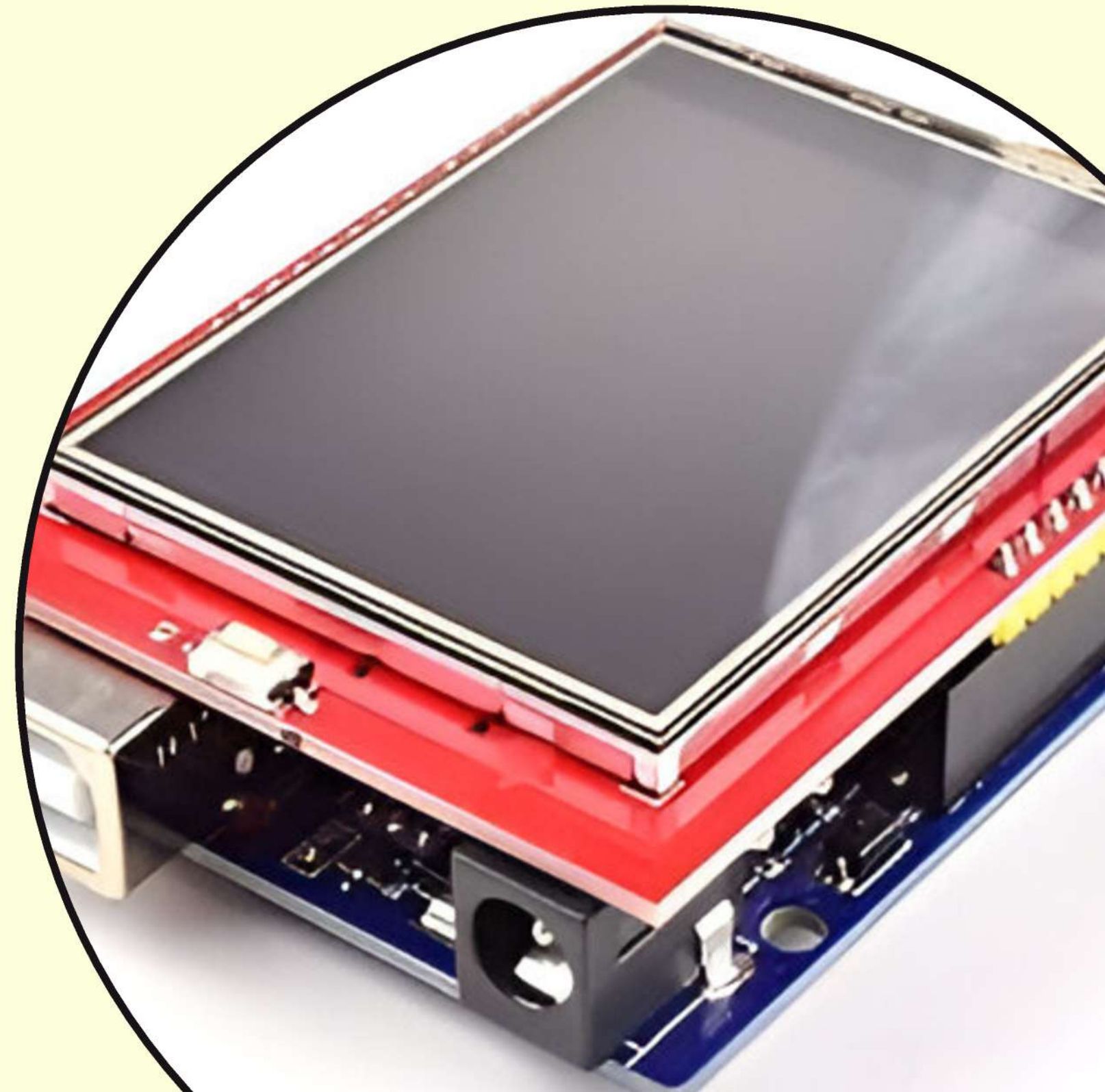


Esta idea **tenía muchos errores eléctricos**, por lo cual fue rápidamente descartada.

Segundo diseño

Se recurrió a la utilización de **tarjetas programables de bajo consumo.**

.....
Se utilizó una **pantalla táctil** y comenzó la elaboración de un **código** para darle vida al material didáctico.



ChatGPT

Código

Rediseño

```
// Draw the buckets at the bottom
const int16_t bucketWidth = 60;
const int16_t bucketHeight = 30;
const int16_t bucketStartY = screenHeight - bucketHeight;

// Calculate bucket spacing and positions
int16_t bucketGap = (screenWidth - (3 * bucketWidth)) / 4; // Gap between bucket
tft.fillRect(bucketGap, bucketStartY, bucketWidth, bucketHeight, RED);
tft.fillRect(2 * bucketGap + bucketWidth, bucketStartY, bucketWidth, bucketHeight, RED);
tft.fillRect(3 * bucketGap + 2 * bucketWidth, bucketStartY, bucketWidth, bucketHeight, RED);

// Label the buckets
tft.setCursor(bucketGap + 10, bucketStartY + 5);
tft.print("~");
tft.setCursor(2 * bucketGap + bucketWidth + 10, bucketStartY + 5);
tft.print("=");
tft.setCursor(3 * bucketGap + 2 * bucketWidth + 10, bucketStartY + 5);
tft.print("-");
}

// Function to generate a new math problem
void generateProblem() {
    num1 = random(1, 100);
    num2 = random(1, 100);
    expectedBucket = random(0, 3); // 0 = '+', 1 = '=', 2 = '-'
}

void setup() {
    Serial.begin(9600); // Initialize serial communication
    tft.reset();
    int16_t id = tft.readID();
    tft.begin(id); // Initialize the display
    randomSeed(analogRead(A0));

    generateProblem();
    drawGameField();
}

void loop() {
    TSPoint p = getTouch();

    if (p.x > MINPRESSURE && p.x < MAXPRESSURE) { // Valid touch pressure
        int16_t x = map(p.x, TS_LEFT, TS_RT, 0, screenWidth);
        int16_t y = map(p.y, TS_TOP, TS_BOT, screenHeight);

        if (y >= screenHeight - 40) { // Touch is in the bucket area
            const int16_t bucketWidth = 60;
            int16_t bucketGap = (screenWidth - (3 * bucketWidth)) / 4;

            if (x >= bucketGap && x < bucketGap + bucketWidth) { // "+" bucket
                if (expectedBucket == 0) {
```

```
// Calibration constants for the touchscreen
const int XP = 6, XM = A2, YP = A1, YM = 7; // Pins for touchscreen
const int TS_LEFT = 175, TS_RT = 903, TS_TOP = 938, TS_BOT = 185; // Calibration boundaries
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300); // Pressure threshold

// Touch pressure thresholds
const int MINPRESSURE = 200;
const int MAXPRESSURE = 1000;

// Screen dimensions
const int screenWidth = 240;
const int screenHeight = 320;

// Colors
const uint16_t BLACK = 0x0000;
const uint16_t WHITE = 0xFFFF;
const uint16_t RED = 0xF800;
const uint16_t GREEN = 0x07E0;
const uint16_t YELLOW = 0xFFE0;
const uint16_t LIGHT_BLUE = 0x07FF;
const uint16_t BLUE = 0x001F;
const uint16_t BROWN = 0x964B00;
const uint16_t GRAY = 0x2d2b2b;

// TFT display
MCUFRIEND_kbv tft;

// Game variables
int score = 0;
int level = 0;
int num1, num2; // Numbers to compare
int expectedBucket; // 0 = '+', 1 = '=', 2 = '-'

// Function to get touch input
TSPoint getTouch() {
```


Soldadura

Elaboré un **diagrama** para darle una **autonomía** al **dispositivo diseñado**.

.....

Vídeo de la
elaboración



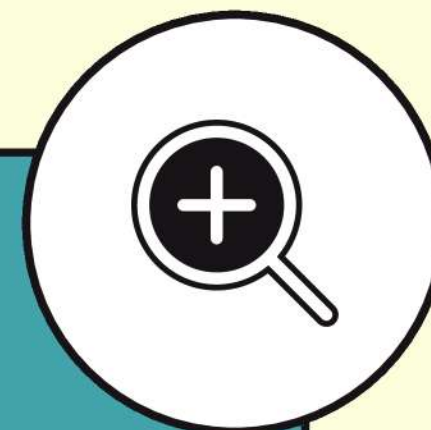


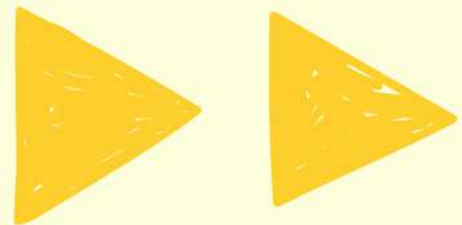
Resultado final



.....

Video de las
instrucciones





Link a la documentación



https://drive.google.com/file/d/1cpl_kAg7EgGOCzF0g-zmD3XHEl6XIsjp/view?usp=sharing

