

Carpeta de Campo

Creador de la carpeta: Miguel Betancourt

Integrantes de grupo: Miguel Betancourt, Sofia Amarilla y Ramiro

Año y Division: 4ta 1era

Proyecto: BMO, una consola de juego basado en la caricatura de hora de aventura

Fecha de entrega de proyecto: 1/12/23

25/9/23

Se definen los grupos para el proyecto y se define la fecha de entrega del proyecto.

Grupo numero 6 nos fue asignado y está conformado por:

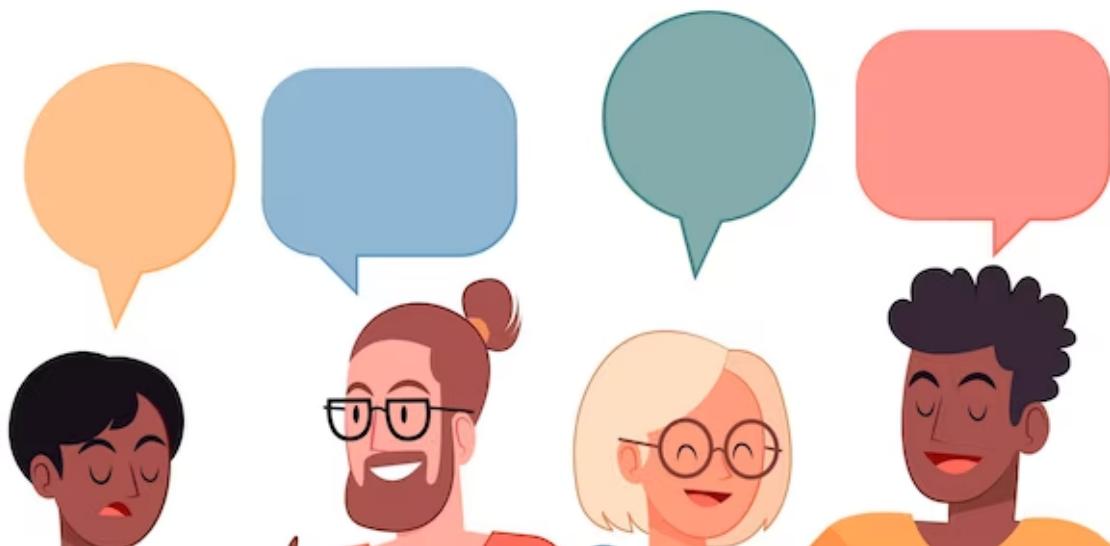
Miguel Betancourt

Sofia Amarilla

Ramiro Garcia

Al primer dia no hicimos prácticamente nada ya que no teníamos una idea sólida pero sin embargo nos pusimos a chalar y ver que podrías hacer para nuestro proyecto final.

Esta dia creo que fue solo charla.



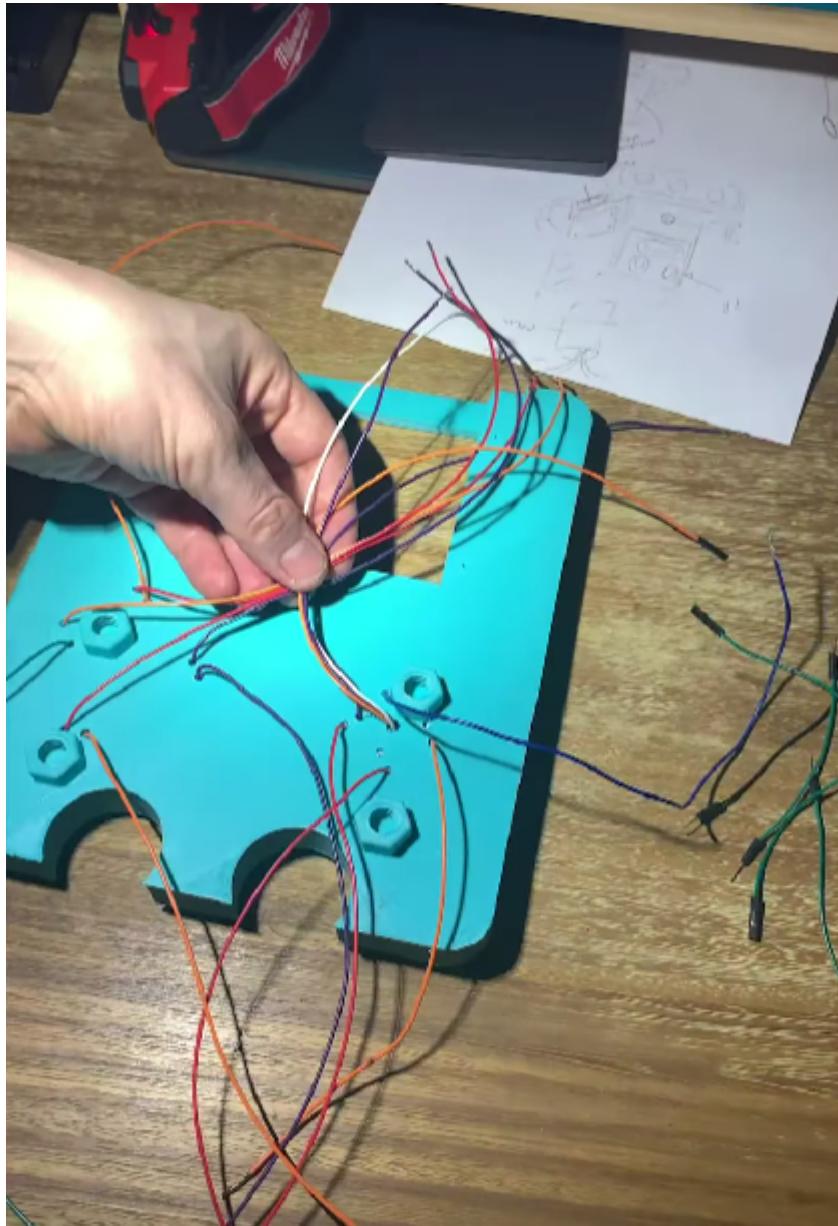
1/9/23

Se proponen las ideas para definir nuestro proyecto con mis compañeros (Sofia y Ramiro)

Ideas:

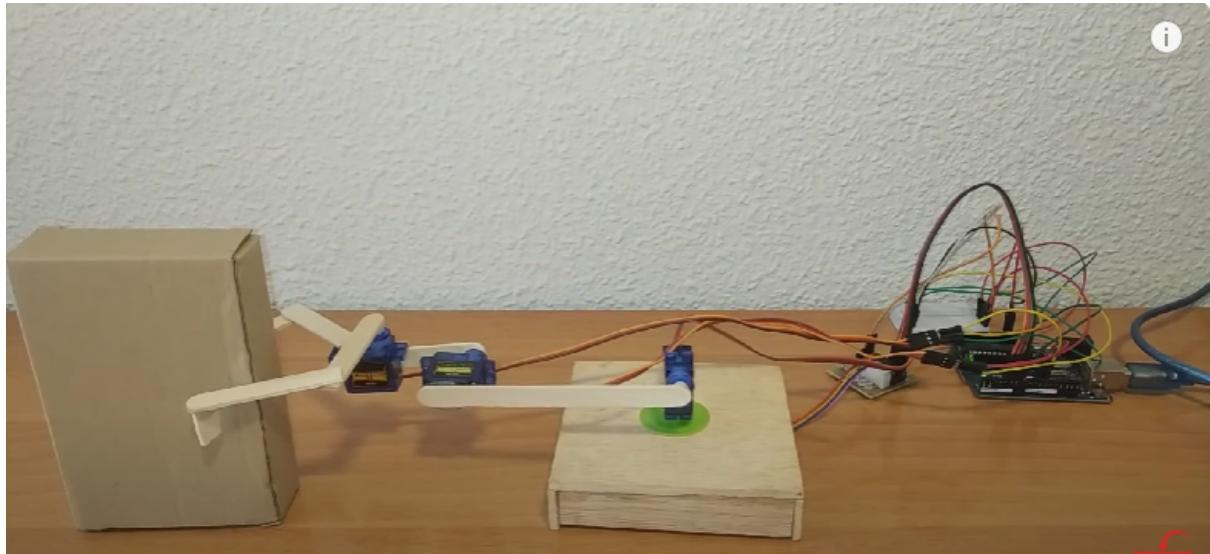
Sofia: BMO de hora de aventura.

<https://youtube.com/shorts/ChsXRkhA9n8?si=oGI0wAZAFCKEwBB9>



Yo: Propongo un brazo de madera que pueda mover sus dedos.

<https://youtu.be/JBI7gwf7ORU?si=oZ9RwapnK5ii-5SX>



Este video nos proporcionaba los datos que necesitábamos.

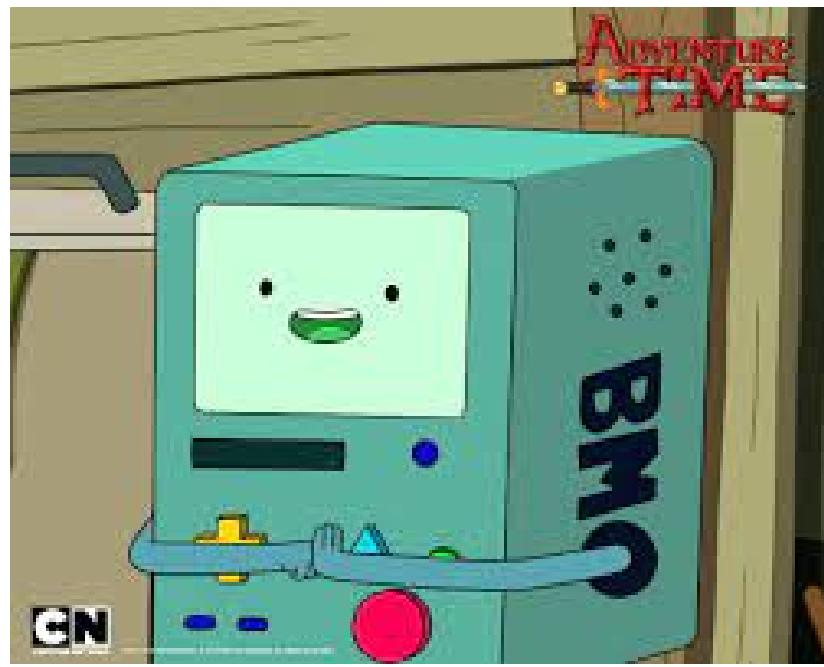
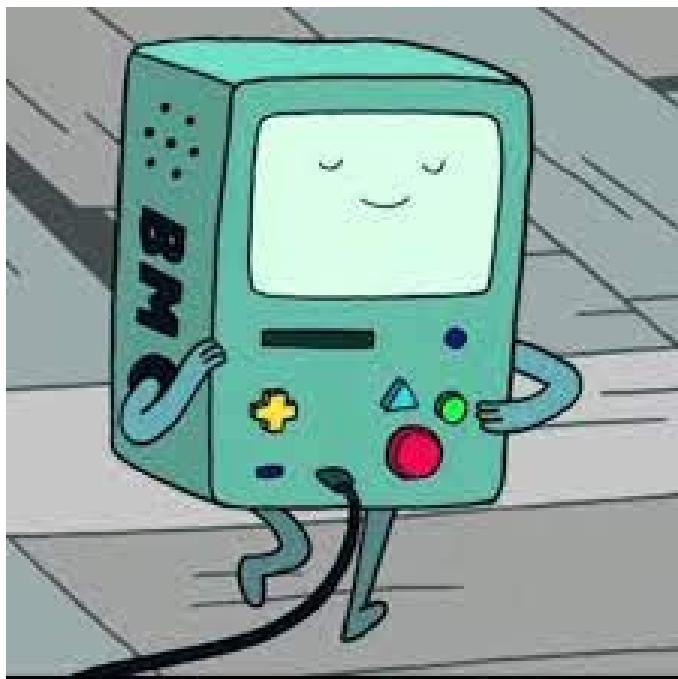
►CÓDIGO DE PROGRAMACIÓN + ESQUEMA DE CONEXIONES:
<https://www.robotuno.com/proyecto-bra...>

Mientras que en la idea que presentó mi compañera Sofía es algo muy difícil de hacer ya que no hay ejemplos y no nos proporciona algo con seguridad, pero al parecer el profe le gusto su idea ya que es algo original al mi parecer,

8/9/23

Se define el proyecto que se va a realizar (BMO), una caricatura de hora de aventura, este en si era una consola de juego. Usaremos el modelo es decir, su diseño de presentación como en la caricatura, El BMO cumplirá ciertas funciones, ya que el usuario en si podrá interactuar con el BMO, y tendrá juegos incluidos, como la serpiente, tetris etc. Su diseño exterior se hará por impresión 3D y nos ponemos

de acuerdo qué juegos va a tener incluido. desarrollamos ideas y lo que podría tener el BMO.



Investigamos que juegos podemos incluir, ya que tenemos esa idea, hacer como un arcade de juegos.

Juegos que podríamos incluir:

<https://proyectosinteresantes.com/10-juegos-arduino-lcd/>

- 5 Arduino Juego de la Serpiente con Display LCD
 - 5.1 Sketch para crear el juego de la Serpiente con Arduino
- 6 Arduino Juego de Tetris y Display LCD
 - 6.1 Sketch para crear el juego de Tetris con Arduino
- 7 Arduino Juego tipo Runner y Display LCD
 - 7.1 Sketch para crear con Arduino un juego tipo runner
- 8 Arduino Juego del Ahorcado y Display LCD
 - 8.1 Sketch para crear el juego del Ahorcado con Arduino
- 9 Arduino Juego de PacMan y Display LCD
 - 9.1 Sketch para crear el juego con Arduino de Pacman
- 10 Arduino Juego tipo Arkanoid y Display LCD
 - 10.1 Sketch para recrear un juego Arkanoid con Arduino
- 11 Arduino Juego de Helicóptero y Display LCD
 - 11.1 Sketch para crear el juego del Helicóptero con Arduino
- 12 Arduino Juego de Carreras y Display LCD
 - 12.1 Sketch para crear un juego de carreras con Arduino
- 13 Arduino Juego de Pacman Rider y Display LCD
 - 13.1 Sketch para crear un Pacman Rider en Arduino
- 14 Arduino Juego Truck Lane y Display LCD
 - 14.1 Sketch del juego con Arduino Truck Lane

15/9/2:

Definimos qué elementos podremos usar para elaborar nuestro BMO

10 push-buttons / 7 / 8 Usaremos los botones para hacer acciones en el BMO, como hacer que el usuario elija una opción o darle movimiento en el juego al que se esté jugando



6x6MM Tactile PCB Switch

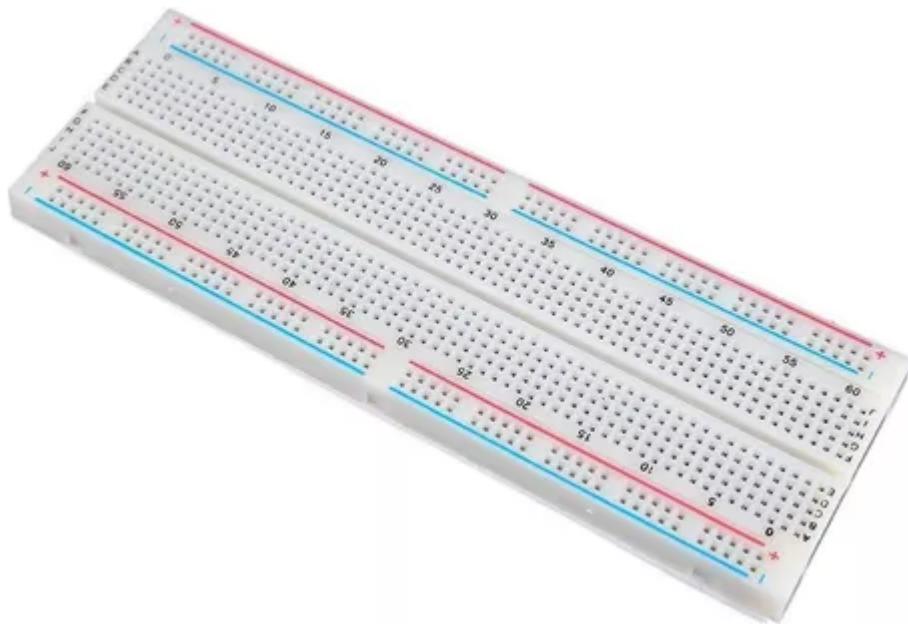
1 buzzer

usaremos el buzzer para incluirle sonidos, para que sea más divertido, puede ser música de fondo o cuando se vaya a ejecutar una acción por ejemplo morir o conseguir algo.



Protopboard

Usaremos una plataborda para conectar los cables y tener mayor facilidad a la hora de conectar. Básicamente se usara para estarías organizados a la hora de cambiar y modificar.

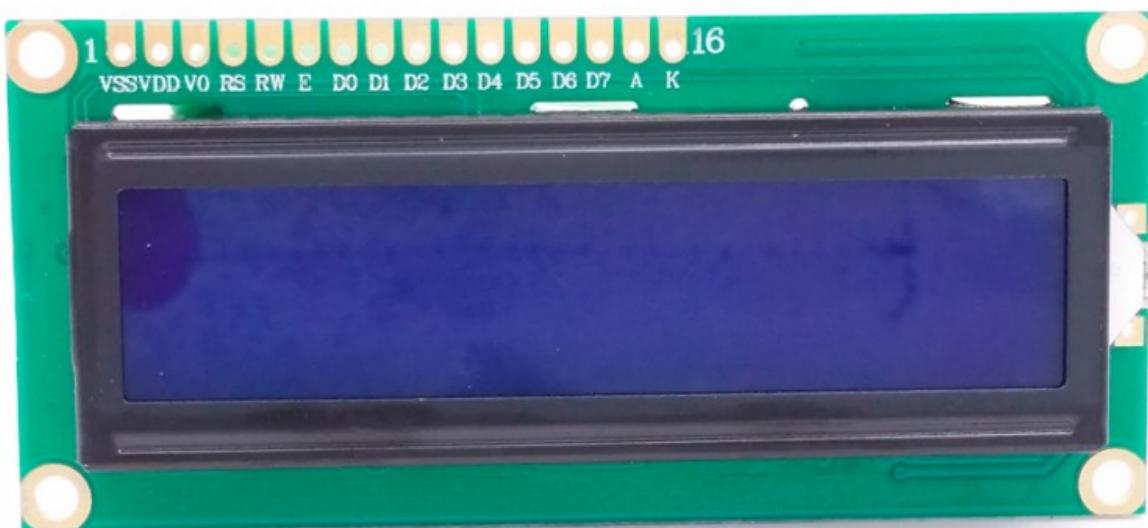


pero en este caso usaremos una pequeña ya que tenemos muy poco espacio para colocarla, ya que la carcaza del BMO es muy estrecha y la mayor parte se la lleva el arduino, en conjunto de los otros componente.



Display un ejemplo de que es lo que podriamos usar:

Se podria usar este display para visualizar el juego y las opciones que estemos haciendo. Aun estamos viendo que display usaremos si uno grande o una pequeño ya que ambas quedarian bien., esto aun no esta definido ha que estamos comparando precios y viendo que display quedaría bien.



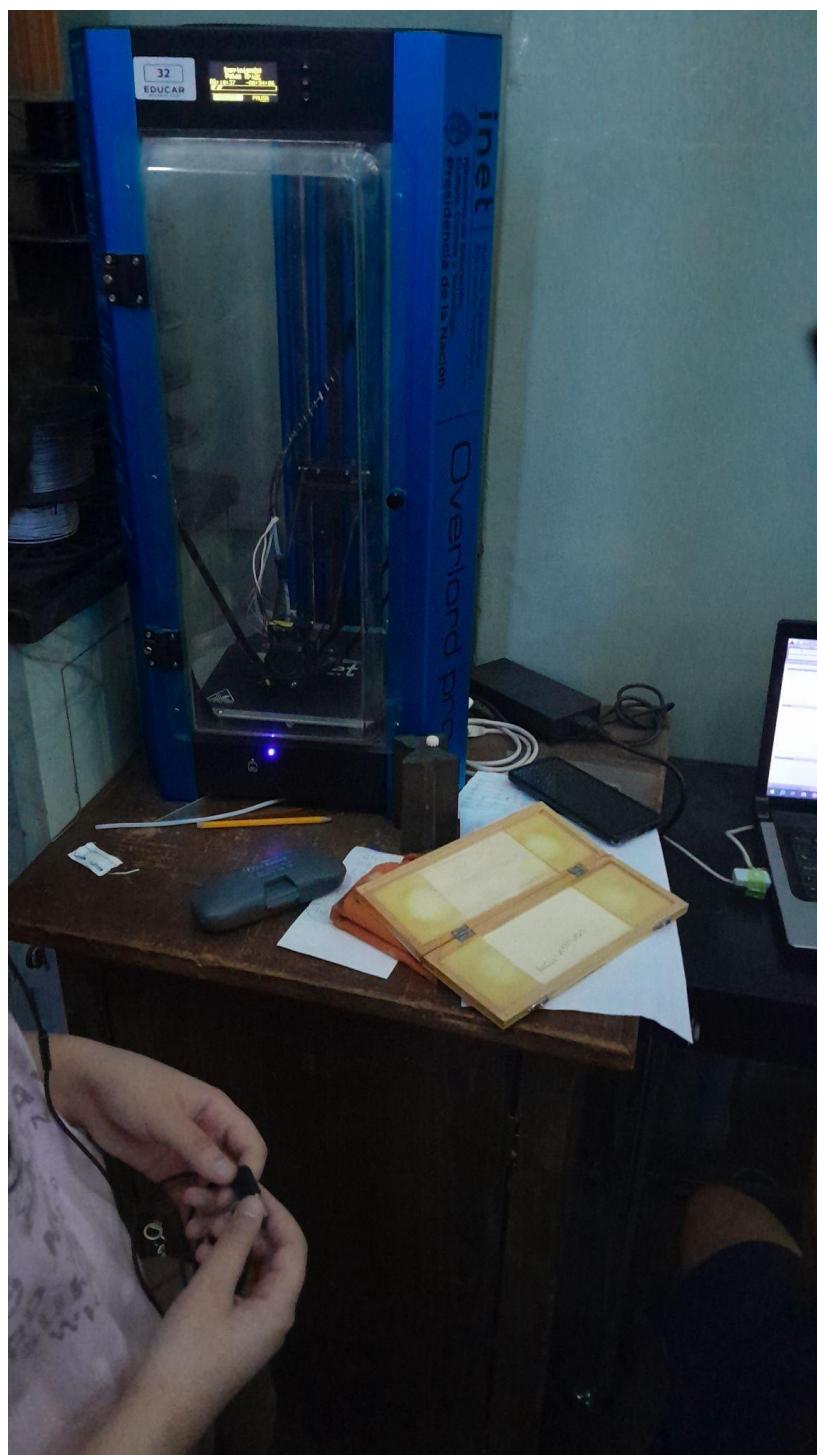
Arduino:

Este es el arduino que usaremos en todo el proyecto. Lo usaremos para controlar y alimentar los componentes y tomar decisiones de acuerdo al programa descargado e interactuar con el mundo físico gracias a los botones que usaremos



maquina 3D:

Usaremos la máquina 3D para imprimir la carcasa del BMO, se usaran medidas específicas para ello y teniendo encuentre cuánto mide los componentes, reduciendo los espacio para ello.



16/09/23 - 22/09/23

Luego de varias horas de búsqueda, el display indicado para el BMO de la placa arduino, es la Pantalla Lcd 12864. Buscando precios y comparando encontramos estos dos.

[Pantalla Lcd 12864 Impresora 3d Controlador + Adaptador](#)

[Rampas](#)



Display Lcd Grafico 12864 Backlight Azul 128x64 Arduino Ubot



Mientras que el primero es más barato viene con un controlador para impresoras 3d, el profe nos ayudará a desoldarlo y quedarnos solamente con el display. Mientras que el segundo tiene un precio más caro pero viene solamente con el display. Claramente conviene el primero. pero siempre hay que tener otra opcion por las dudas.

Los displays que encontramos tienen una resolución de 128 x 64, el tamaño de 93 mm x 70 mm



sofía: Al crear una grandiosa presentación siendo que esperaba un rotundo “no” de parte de nuestro mentor, busqué razones por la que sería una buena idea obtener específicamente este display y encontré un modelo 3d del tamaño indicado para el proyecto.

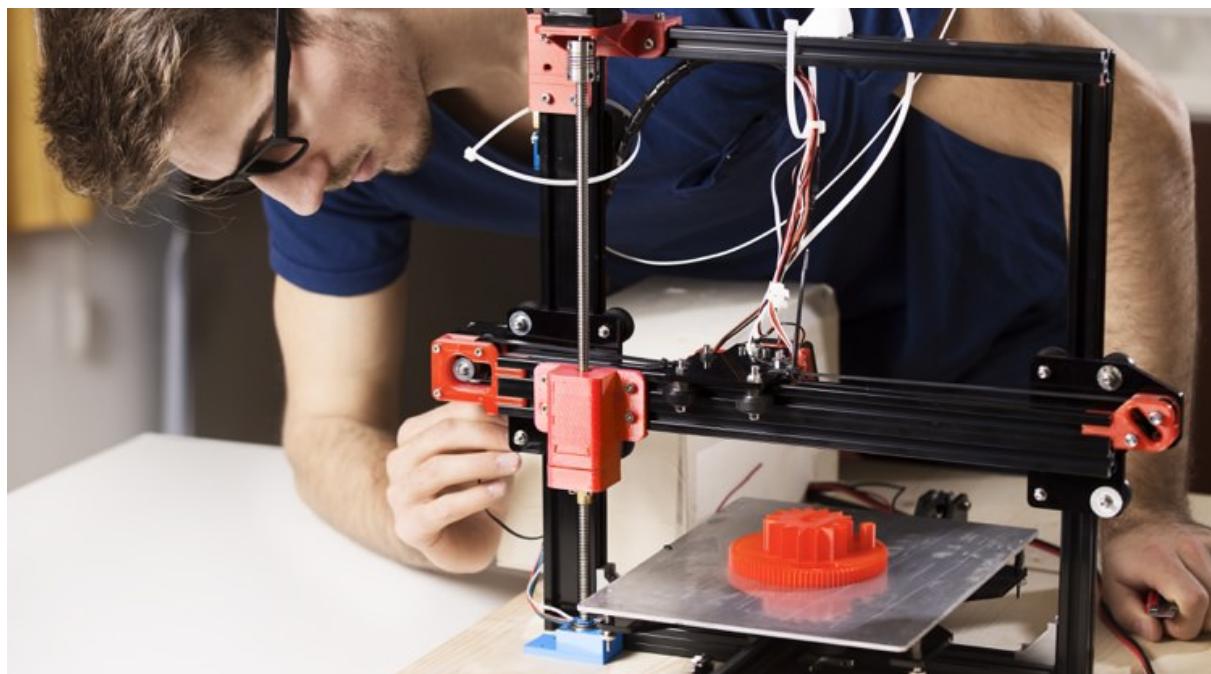
[CAJA RASPBERRY PI 4 - BMO \(ADVENTURE TIME\)](#)

El problema de este modelo es la modificación para componentes de arduino, y específicamente para el arduino uno. Y si existe la posibilidad, poder cambiarlo al tamaño indicado del display, siendo que es algo más grande y cuadrado.



- El mono

Hoy conocimos al mono, aparecer es quien se encarga de la maquina 3d, el es quien nos ayudara a imprimir la carcasa del BMO en 3D, esto tarda un tiempo (horas o dias) ya que debemos tomar las medidas. Y la maquina es pequena asi que tardaria un poco.



29/09/2023

El mono parte 2

En el día de hoy, luego de educación física, nos juntamos con "El Mono" a discutir el diseño del BMO, el cual le pareció fácil y ya tenemos las medidas del largo, ancho, donde irían los parlantes y en resumen, las medidas principales. Aunque todavía falta la parte trasera, los botones y quizás la función del disco, que probablemente no sucederá por falta de pines.

Pero debido a la cantidad de pulsadores que se deberán utilizar es muy posible que tengamos que hacer algunos cambios con las medidas, en especial el ancho.

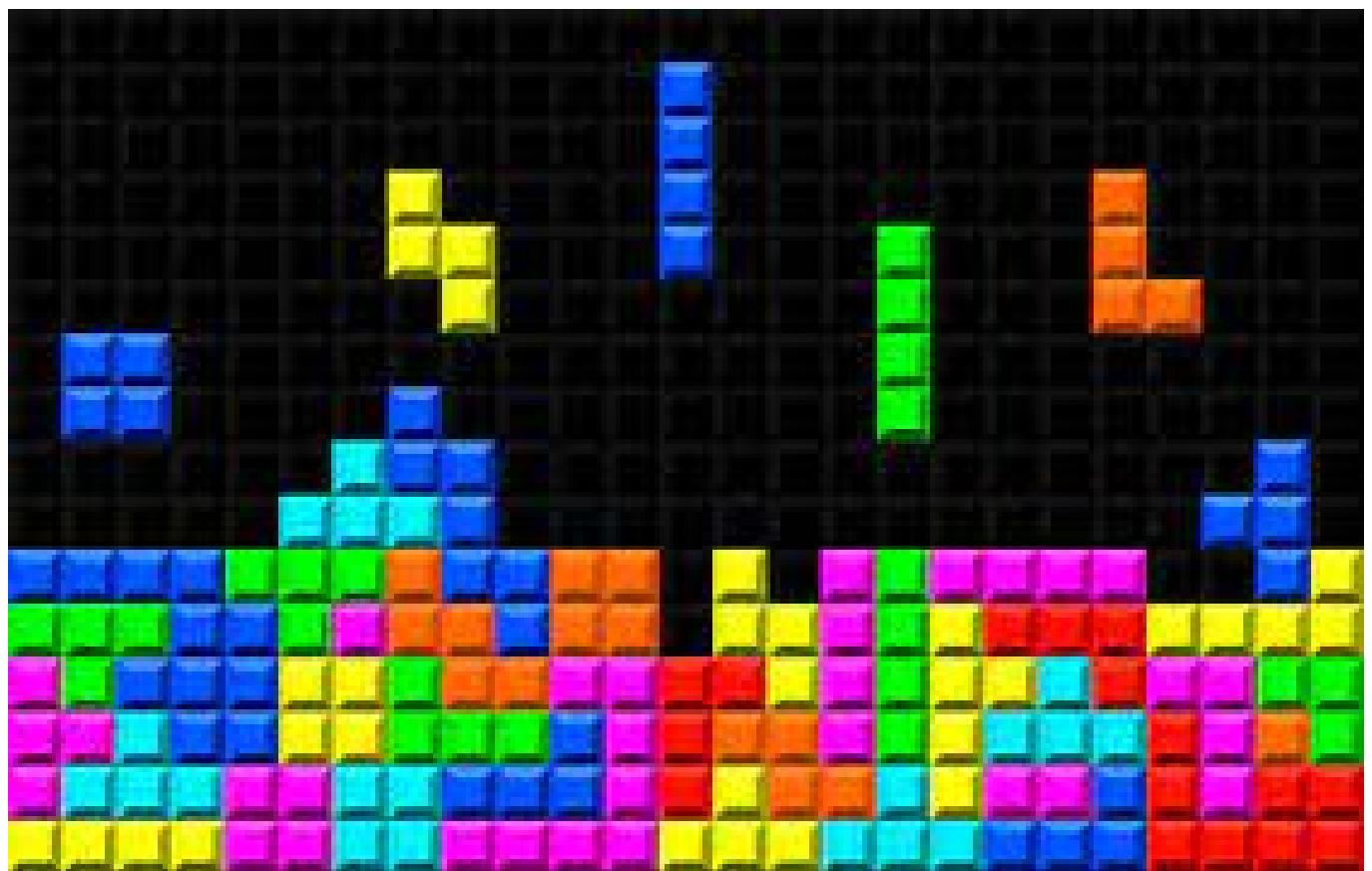
Aún tenemos muchos problemas con la cantidad de espacio que ocupan los componentes, aunque lo único que nos preocupa es el tipo de tornillos que tendremos que usar, porque eso cambiaría mucho el grosor del BMO, "El Mono" dijo que todavía no nos preocupemos por eso, pero habría que tener una idea de dónde se deberán posicionar.

Por ahora lo que arreglamos es que tenga una medida de 100 mm de ancho y 150 mm de largo, con una profundidad de 600 mm. Los

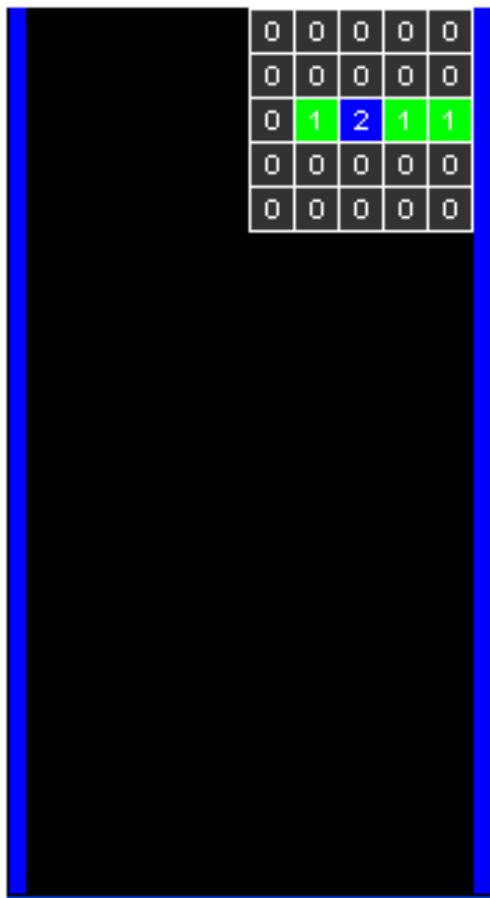
brazos de BMO tendrán un radio de 10 mm, al igual que las piernas que se encontrarán sentadas para que el robot tenga una base estable.

Luego el profesor explicó brevemente cuál será la lógica y una base de lo que sería, cómo dice él, “El Tetrix”. Debido al tiempo limitado que tenemos ,dudamos en que lleguemos ya que es muy complicado de programarlo.

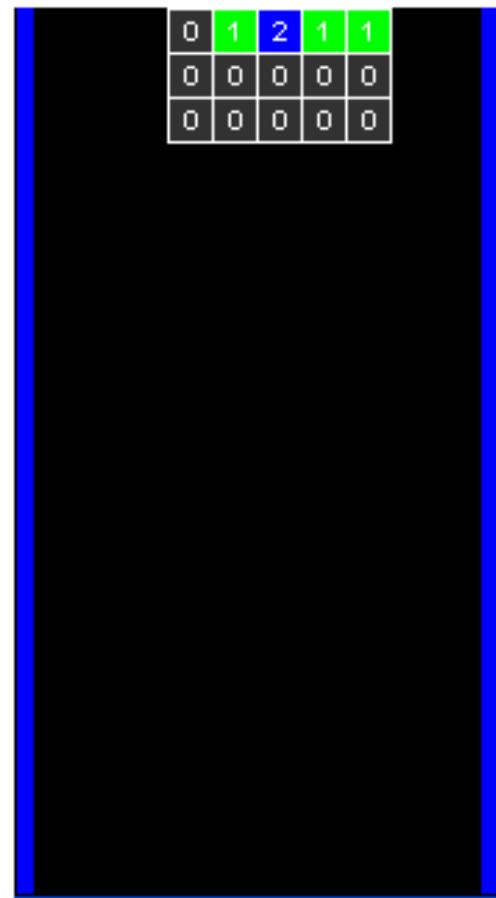
Mi compañera Sofía tendrá que investigar a fondo la lógica del tetris, que de forma muy resumida sería, la creación de las piezas, sus movimientos y sus matrices, además del “diseño” del tablero (un pequeño rectángulo).



Wrong initial position



Good initial position (-2, -2)



Segundo, la caída de las piezas, esto significa, el apagado y prendido de los píxeles mientras descienden, tendrán un tiempo determinado y tendría que ser sincronizado.

Tercero, que se reconozca si ya hay una pieza para que se amontone y no se desborde o traspase.

Cuarto, detectar en el caso de que una fila se complete y que se elimine, bajando todas las demás hileras.

Algunos problemas en este juego es la matriz cuando se encuentra cerca del tablero, cómo detectará y si destrozará por completo el recuadro.

Mientras que otro problema es la detección de otras piezas, siendo que muchas veces se pueden juntar y crear un bug en la partida.



1/09/23

hoy investigue algunos juegos que podríamos utilizar.

el código de juego: En este juego, los botones están numerados del 1 al 10, y el usuario debe presionar el botón correspondiente al número que desea ingresar. La pantalla LCD mostrará un mensaje indicando que el usuario debe adivinar el número. Después de que el usuario haya ingresado su respuesta, se mostrará un mensaje en la pantalla LCD y se reproducirá un tono en el buzzer para proporcionar retroalimentación.

```
#include <LiquidCrystal_I2C.h>

const int buzzerPin = 9; // Pin para el buzzer
const int buttonPins[] = {2, 3, 4, 5, 6, 7, 8, 10, 11, 12}; // Pines de los
botones
const int answer = 7; // Número que el usuario debe adivinar
const int lcdAddress = 0x27; // Dirección I2C de la pantalla LCD

LiquidCrystal_I2C lcd(lcdAddress, 16, 2);

void setup() {
    pinMode(buzzerPin, OUTPUT);

    for (int i = 0; i < 10; i++) {
        pinMode(buttonPins[i], INPUT_PULLUP);
    }

    lcd.begin(16, 2);
    lcd.print("Adivina el numero");
}

void loop() {
    // Espera a que se presione un botón
    int userGuess = waitForButtonPress();

    // Compara la respuesta del usuario
```

```
if (userGuess == answer) {
    lcd.clear();
    lcd.print("¡Correcto!");
    tone(buzzerPin, 1000, 500);
    delay(2000);
} else {
    lcd.clear();
    lcd.print("Incorrecto. Intentalo");
    tone(buzzerPin, 500, 500);
    delay(2000);
    lcd.clear();
}
}

int waitForButtonPress() {
    int userGuess = 0;

    while (true) {
        for (int i = 0; i < 10; i++) {
            if (digitalRead(buttonPins[i]) == LOW) {
                userGuess = i + 1; // Los botones están numerados del 1 al 10
                lcd.clear();
                lcd.print("Tu respuesta: " + String(userGuess));
                delay(1000); // Espera para evitar múltiples lecturas del mismo
botón
                return userGuess;
            }
        }
    }
}
```

Simon Says: En este juego, la secuencia inicial se genera aleatoriamente y se muestra en la pantalla LCD. Luego, el usuario debe repetir la secuencia presionando los botones correspondientes. Si el usuario completa la secuencia correctamente, se emite un tono y se incrementa la longitud de la secuencia para el siguiente nivel. Si el usuario comete un error, se emite un tono diferente y el juego termina.

```
#include <LiquidCrystal_I2C.h>
```

```
const int buzzerPin = 9; // Pin para el buzzer  
const int buttonPins[] = {2, 3, 4, 5, 6, 7, 8, 10, 11, 12}; // Pines de los  
botones
```

```
const int numButtons = 10;  
const int lcdAddress = 0x27; // Dirección I2C de la pantalla LCD
```

```
LiquidCrystal_I2C lcd(lcdAddress, 16, 2);
```

```
int sequence[10];  
int userSequence[10];  
int sequenceLength = 4; // Longitud inicial de la secuencia
```

```
void setup() {  
    pinMode(buzzerPin, OUTPUT);  
  
    for (int i = 0; i < numButtons; i++) {  
        pinMode(buttonPins[i], INPUT_PULLUP);
```

```
}
```

```
Lcd.begin(16, 2);  
Lcd.print("Simon Says");  
delay(2000);  
Lcd.clear();  
}
```

```
void loop() {
```

```
    generateSequence();  
    displaySequence();
```

```
    for (int i = 0; i < sequenceLength; i++) {  
        waitForButtonPress(sequence[i]);  
        delay(500); // Pausa entre botones en la secuencia  
    }
```

```
    getUserInput();
```

```
    if (checkUserInput()) {  
        Lcd.clear();  
        Lcd.print("¡Correcto!");  
        tone(buzzerPin, 1000, 500);  
    } else {  
        Lcd.clear();  
        Lcd.print("Incorrecto. Fin del juego");  
        tone(buzzerPin, 500, 1000);  
    }
```

```
delay(2000);
lcd.clear();
}

void generateSequence() {
    for (int i = 0; i < sequenceLength; i++) {
        sequence[i] = random(1, numButtons + 1);
    }
}

void displaySequence() {
    lcd.clear();
    lcd.print("Secuencia:");
    for (int i = 0; i < sequenceLength; i++) {
        lcd.setCursor(0, 1);
        lcd.print(sequence[i]);
        delay(1000);
        lcd.clear();
        delay(500);
    }
}

void waitForButtonPress(int button) {
    while (digitalRead(buttonPins[button - 1]) == HIGH) {
        // Espera a que se presione el botón
    }
}
```

```
tone(buzzerPin, 1000, 200);
delay(200);
noTone(buzzerPin);
}

void getUserInput() {
    lcd.clear();
    lcd.print("Tu turno:");

    for (int i = 0; i < sequenceLength; i++) {
        userSequence[i] = waitForAnyButtonPress();
        delay(500);
    }
}

int waitForAnyButtonPress() {
    while (true) {
        for (int i = 0; i < numButtons; i++) {
            if (digitalRead(buttonPins[i]) == LOW) {
                tone(buzzerPin, 1000, 200);
                delay(200);
                noTone(buzzerPin);
                return i + 1;
            }
        }
    }
}
```

```
bool checkUserInput() {  
    for (int i = 0; i < sequenceLength; i++) {  
        if (userSequence[i] != sequence[i]) {  
            return false;  
        }  
    }  
    return true;  
}
```

Whac-A-Mole: Este juego genera aleatoriamente un "topo" en uno de los botones y lo muestra en la pantalla LCD. El jugador debe golpear el topo presionando el botón correspondiente antes de que desaparezca. Si el jugador golpea al topo, obtiene un punto; de lo contrario, el topo se escapa y no se obtiene puntaje.

```
#include <LiquidCrystal_I2C.h>  
  
const int buzzerPin = 9; // Pin para el buzzer  
const int buttonPins[] = {2, 3, 4, 5, 6, 7, 8, 10, 11, 12}; // Pines de los botones  
const int numButtons = 10;  
const int lcdAddress = 0x27; // Dirección I2C de la pantalla LCD  
  
LiquidCrystal_I2C lcd(lcdAddress, 16, 2);  
  
int moleInterval = 1000; // Intervalo entre la aparición de topos en milisegundos  
int score = 0;
```

```
void setup() {
    pinMode(buzzerPin, OUTPUT);

    for (int i = 0; i < numButtons; i++) {
        pinMode(buttonPins[i], INPUT_PULLUP);
    }

    lcd.begin(16, 2);
    lcd.print("Whac-A-Mole");
    delay(2000);
    lcd.clear();
}

void loop() {
    lcd.clear();
    lcd.print("Puntuacion: " + String(score));

    int moleButton = random(1, numButtons + 1);
    lightUpMole(moleButton);

    delay(moleInterval);

    if (digitalRead(buttonPins[moleButton - 1]) == LOW) {
        // Usuario golpeó al topo
        score++;
        lcd.clear();
        lcd.print("¡Golpe!");
        tone(buzzerPin, 1000, 200);
    }
}
```

```
    delay(500);
} else {
    lcd.clear();
    lcd.print("¡Se escapó!");
    tone(buzzerPin, 500, 200);
    delay(500);
}

noTone(buzzerPin);
}

void lightUpMole(int moleButton) {
    lcd.clear();
    lcd.print("¡Topo en el " + String(moleButton) + "!");
    digitalWrite(buttonPins[moleButton - 1], HIGH); // Enciende el
LED del topo
}

void turnOffButtons() {
    for (int i = 0; i < numButtons; i++) {
        digitalWrite(buttonPins[i], LOW); // Apaga todos los LEDs
    }
}
```

Adivinar o perder: es un juego de adivinanza numérica. En este juego, el Arduino generará un número aleatorio y el usuario deberá adivinarlo presionando botones para seleccionar dígitos y luego presionar otro botón para confirmar la elección.

```
#include <LiquidCrystal_I2C.h>
```

```
const int buzzerPin = 9; // Pin para el buzzer  
const int buttonPins[] = {2, 3, 4, 5, 6, 7, 8, 10, 11, 12}; // Pines de los botones
```

```
const int confirmButtonPin = 13; // Botón para confirmar el número  
const int lcdAddress = 0x27; // Dirección I2C de la pantalla LCD
```

```
LiquidCrystal_I2C lcd(lcdAddress, 16, 2);
```

```
int answer;  
int userGuess = 0;  
int digitIndex = 0;
```

```
void setup() {  
    pinMode(buzzerPin, OUTPUT);  
  
    for (int i = 0; i < 10; i++) {  
        pinMode(buttonPins[i], INPUT_PULLUP);  
    }  
  
    pinMode(confirmButtonPin, INPUT_PULLUP);  
  
    lcd.begin(16, 2);
```

```
    lcd.print("Adivina el numero");
    delay(2000);
    lcd.clear();

    randomSeed(analogRead(0)); // Inicializa la semilla para la
generación de números aleatorios
    generateRandomNumber();
}

void loop() {
    displayGuess();
    getUserInput();

    if (digitalRead(confirmButtonPin) == LOW) {
        checkGuess();
    }
}

void generateRandomNumber() {
    answer = random(1000); // Genera un número aleatorio entre 0 y
999
}

void displayGuess() {
    lcd.clear();
    lcd.print("Tu adivinanza:");
    lcd.setCursor(0, 1);
    lcd.print(String(userGuess));
}
```

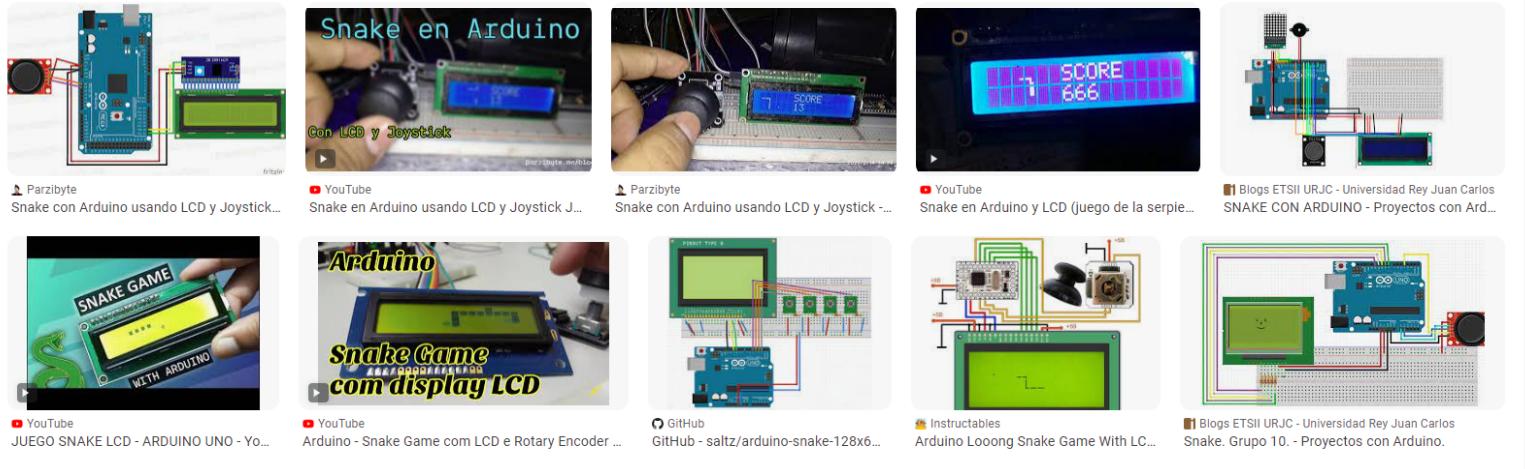
```
void getUserInput() {  
    for (int i = 0; i < 10; i++) {  
        if (digitalRead(buttonPins[i]) == LOW) {  
            delay(150); // Debouncing  
            userGuess = userGuess * 10 + i;  
            tone(buzzerPin, 1000, 50);  
            while (digitalRead(buttonPins[i]) == LOW) {  
                // Espera a que se libere el botón  
            }  
            delay(150); // Debouncing  
        }  
    }  
}
```

```
void checkGuess() {  
    noTone(buzzerPin);  
    if (userGuess == answer) {  
        lcd.clear();  
        lcd.print("¡Correcto!");  
        tone(buzzerPin, 1000, 500);  
        delay(2000);  
        lcd.clear();  
        generateRandomNumber();  
        userGuess = 0;  
    } else {  
        lcd.clear();  
        lcd.print("Incorrecto. Intentalo");  
        tone(buzzerPin, 500, 200);  
    }  
}
```

```
    delay(1000);
    lcd.clear();
    userGuess = 0;
}
}
```

6/10/23

yo: Snake



Lógica del Juego: Comprende la lógica del juego de Snake. Esto incluye cómo la serpiente se mueve, cómo se genera la comida, cómo crece la serpiente y cómo se detectan las colisiones.

Manejo de Entrada: Aprende a manejar la entrada del usuario. En el caso de Snake, esto implica detectar la dirección en la que el usuario quiere que la serpiente se mueva.

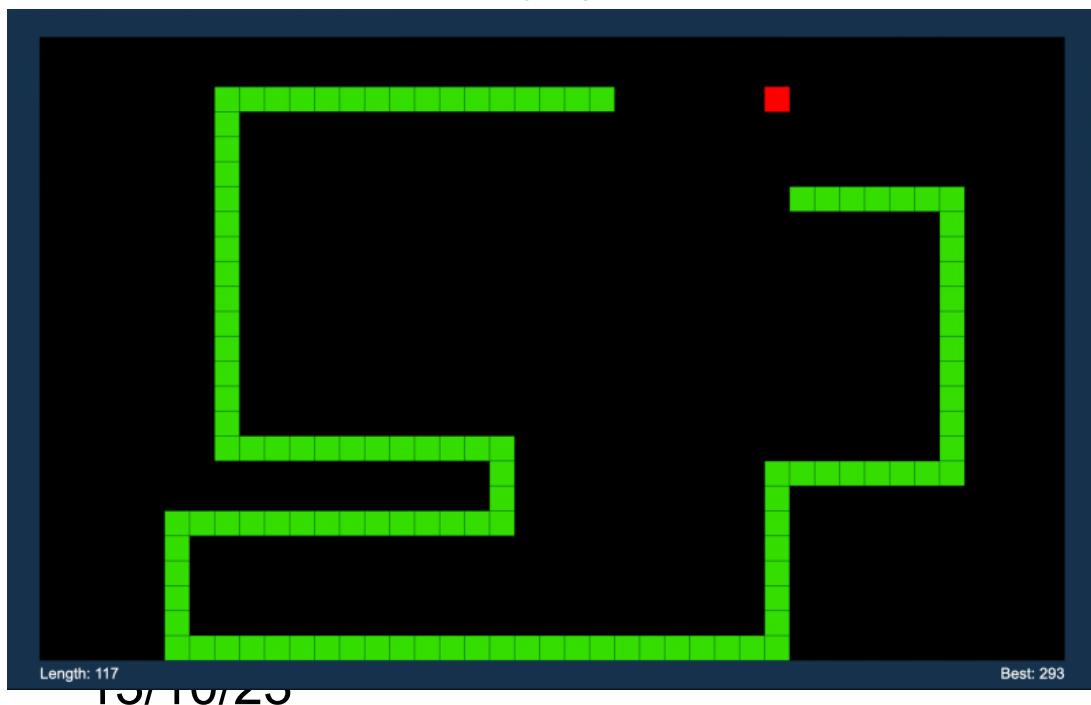
Actualización de la Pantalla: Comprende cómo actualizar la pantalla en cada ciclo del juego para reflejar los cambios en la posición de la serpiente y la ubicación de la comida.

Control de Tiempo: Utiliza el control de tiempo para regular la velocidad del juego y para decidir cuándo mover la serpiente y cuándo generar nueva comida.

Desarrollo Iterativo: Iteración del Desarrollo: Comienza con un conjunto mínimo de características y ve agregando funcionalidades paso a paso. Por ejemplo, comienza con una serpiente que puede moverse y crecer al comer, y luego agrega puntuación.

Depuración: Aprende a utilizar la salida de depuración y la monitorización seria en Arduino IDE para encontrar errores y problemas en tu código.

Recursos Adicionales: Tutoriales y Ejemplos: Buscare algunos tutoriales y ejemplos de juegos de Snake para Arduino en línea. podria encontrar un videos que me proporcione un código de referencia y ayudarte a entender cómo se estructura un juego simple.



debido a que fue feriado no pude reunirme con mis compañeros y pude investigar algunos juegos que le podríamos agregar al BMO.

<https://www.xataka.com/makers/13-proyectos-asombrosos-con-arduino-para-ponerte-a-prueba-y-pasar-un-gran-rato>



Esto no me ayudó, ya que no era lo que estaba buscando.

Los juegos que vi son muy avanzados y no son los componentes que nosotros usamos, así que sería muy difícil adaptarlos o modificarlos para la situación

27/10/23

Después del día de modelar la carcasa del BMO empezamos a colocar las partes, es decir, el display y unir cables y botones, pero

primero nos fijamos de que el display funciona así que buscando un código de Internet de ejemplo para ver si funcionaba.

Cosa que al parecer no funcionó, el profe midió todas las conexiones a ver si pasaba corriente pero no.

Para ello uso el tester.

2/11/2

El
pantalla



con la

Información de este LCD 128x64

<https://forum.arduino.cc/t/informacion-de-este-lcd-128x64/282153>

tutoriales

<https://arduino-tutorials.net/tutorial/control-graphic-lcd-display-spi-st7920-128x64-with-arduino?authuser=0>

Interfaz de una pantalla LCD

<https://electropeak.com/learn/interfacing-128x64-graphical-lcd-display-with-arduino/?authuser=0>

Uso de una pantalla LCD

<https://www.makerguides.com/es/interfacing-128-x-64-graphical-lcd-with-arduino/?authuser=0>

Graficos en la pantalla LCD

<https://www.mauroalfieri.it/elettronica/lcd-grafico-128x64-arduino.html?authuser=0>

configuración de la LCD

<https://github.com/olikraus/u8g2/wiki/u8g2setupcpp?authuser=0>

Graficar en arduino

<https://www.youtube.com/watch?v=fOYwBObRBkl>

Biblioteca de gráficos LCD

<https://playground.arduino.cc/Code/LCD12864/?authuser=0>

Un ejemplo en pantalla LCD

<https://forum.arduino.cc/t/pantalla-lcd-128x64-imprime-datos-basura-con-el-serial-print/129720/6?authuser=0>

Como crear IMAGENES y LOGOS

https://www.youtube.com/watch?v=NuYw2y9xm_Y

3/11/23

Yo me ausente este día, pero puede investigar más sobre qué juegos podíamos usar. Por lo que me dijeron es que la pantalla si funciona, así que me podría tranquilizar con eso.

<https://proyectosinteresantes.com/10-proyectos-de-arduino-para-crear-juegos-con-leds/>



2.- Videojuego en pantalla Led

Utiliza una pantalla Led sencilla, para recrear un juego en el que tendrás que sortear obstáculos para aumentar tu posición. Genial ¿No?



Link del proyecto | <http://bit.ly/2WIAcDj>

3.- Juego de la serpiente

Recrea el clásico y original juego de la serpiente con Arduino, gracias a una matriz de leds de 8 x 8.



Lastimosamente no me servían ya que se usaban los leds .

8/11/23

En este día charlé un poco con mis compañeros de como iban con la carpeta de campo.



Al escuchar sus opiniones decidí modificar y actualizar la carpeta de campo.

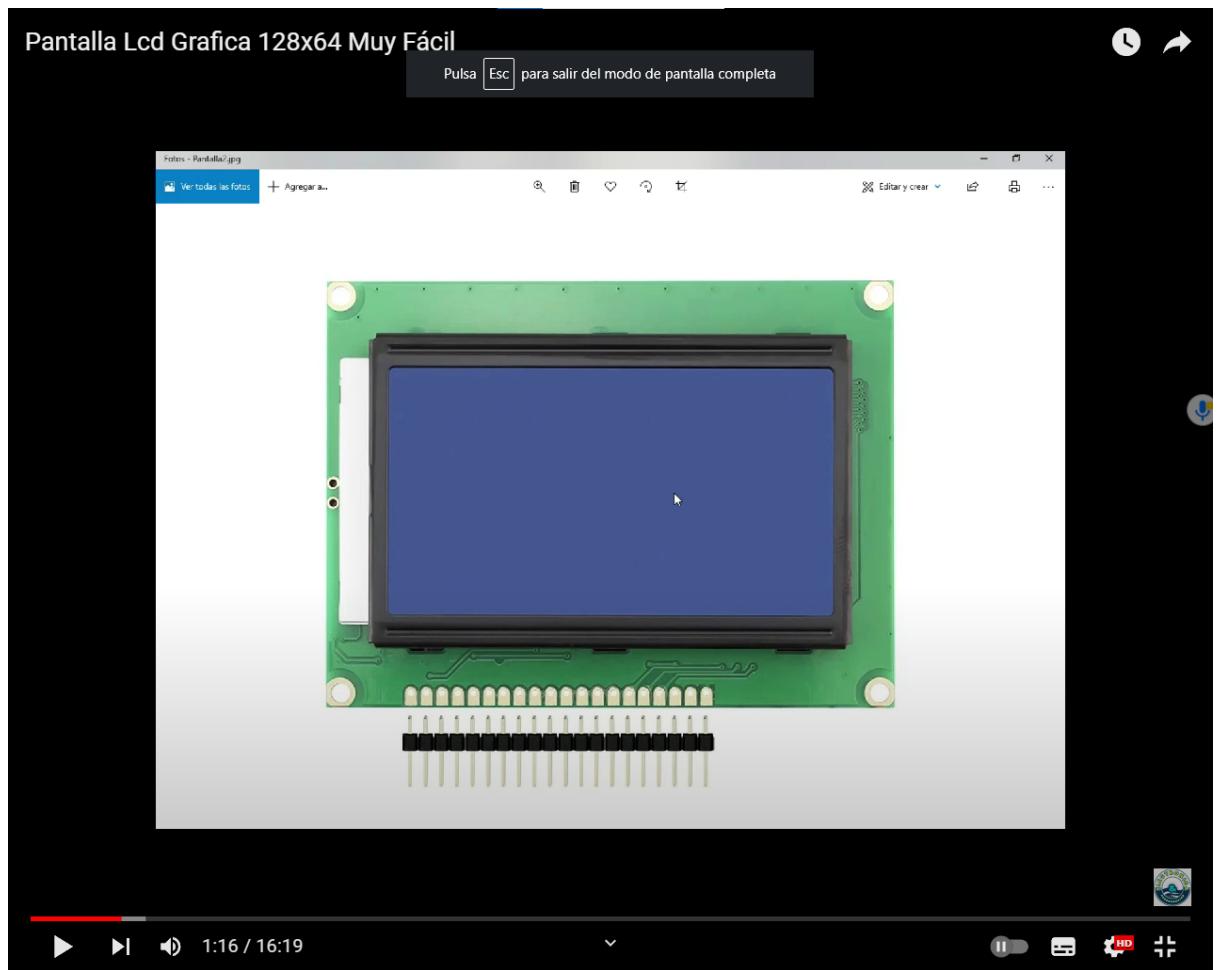
Su actualización fue simplemente imágenes, por decir, agregar links de las búsquedas.

Se agregó todo lo que hacía referente al proyecto una de ellas fueron búsquedas y charlas.

10/11/23

No vino el profe, así que no hubo clases y faltamos, yo por mi cuenta busque videos videos en casa de como conectar los cables en la pantalla lcd con el arduino.

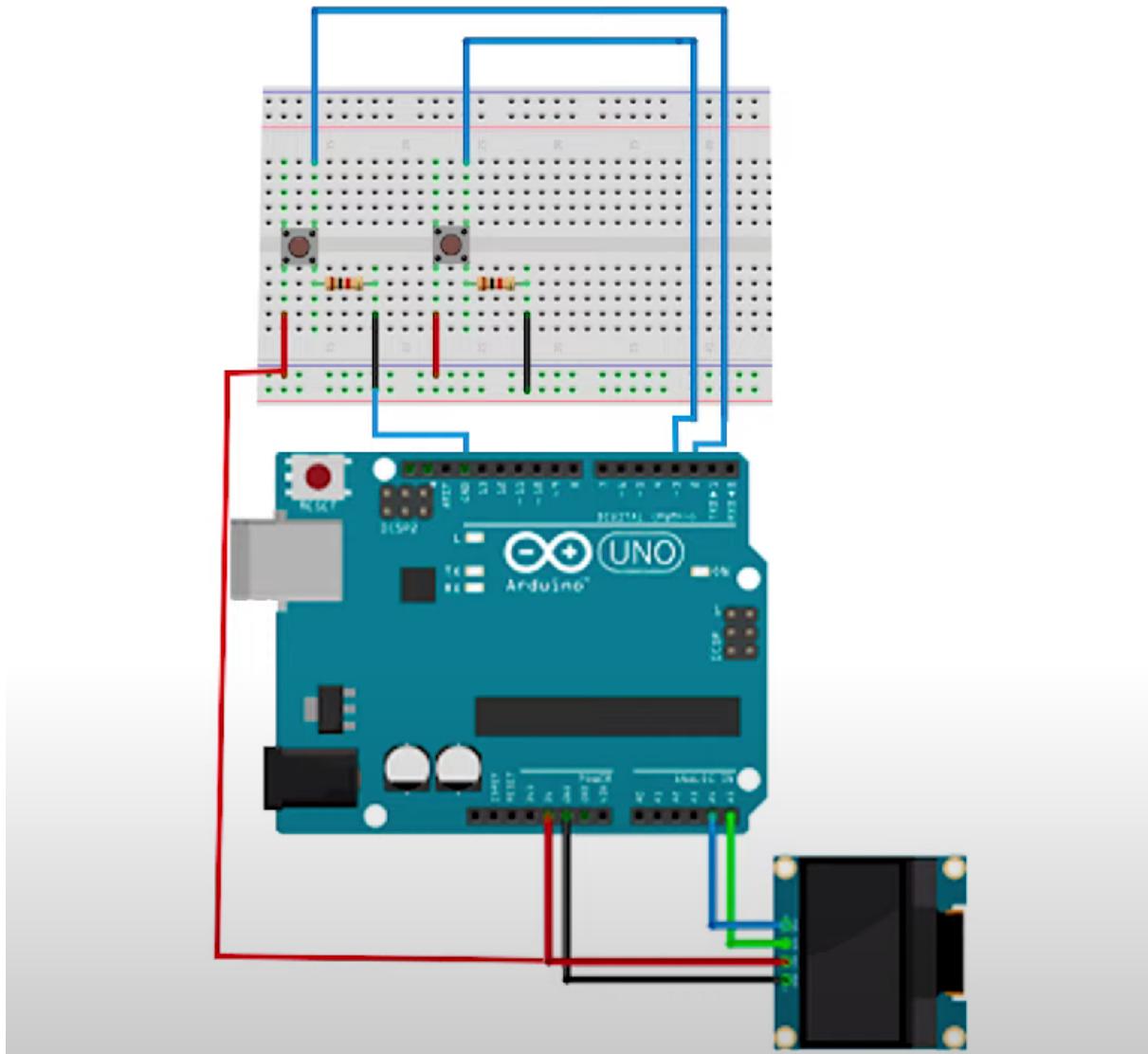
<https://www.youtube.com/watch?v=AVTL4mhbevM>



17/11/23

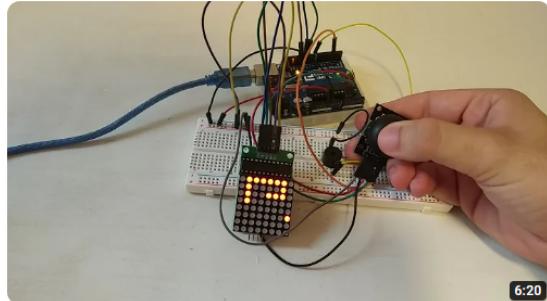
Sofía se encargó de buscar el pong en YouTube de como lo han hecho otras personas y como programarlo.

<https://www.youtube.com/watch?v=lqkQoR620b4&t=98s>



Por otra parte, voy a investigar cómo ejecutar y hacer las conexiones que necesita mi juego de la serpiente(snake).

Cosa que no puede encontrar ya que usan otros tipos de pantallas.



✓ SNAKE GAME con MATRIZ LED 8x8 || MINIJUEGO con ARDUINO || [Explicado paso a paso] FÁCIL
30 k vistas • hace 3 años
Robot UNO
→COMENTA que te ha parecido y dale a →SUSCRIBETE para no perderte ningún vídeo.



✓ Cómo hacer tu propio juego ↗ de Snake con Arduino y una matriz de LED
970 vistas • hace 8 meses
Innovabots
aprende como hacer clásico juego de Snake con un joystick y Arduino sígueme en mis redes sociales Facebook: innovabots ...



Arduino Juego de la Serpiente con Arduino y Display LCD
2.4 K vistas • hace 3 años
Héctor Pérez
Demostración de Arduino: Juego de la Serpiente (Snake) con Arduino y una pantalla LCD Link de los proyectos ...

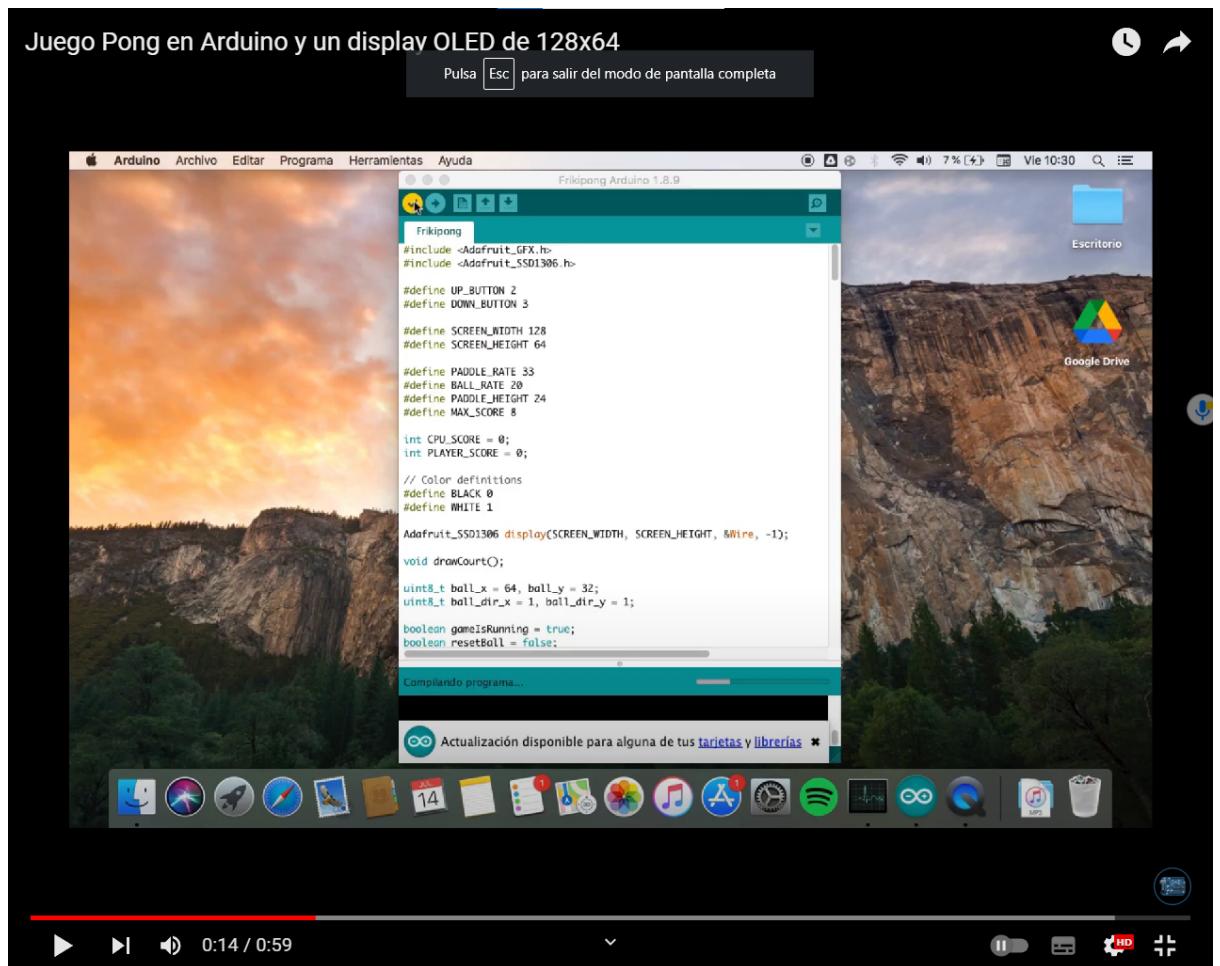
gente usa otro tipo de pantalla y entonces es difícil saber como hacer las conexiones.

21/11/23

Hoy no fue Sofia pero en hora de organizacion computacional ya que habia presentado todo, me puse a pensar como podriamos entregar el avance que hemos obtenido.

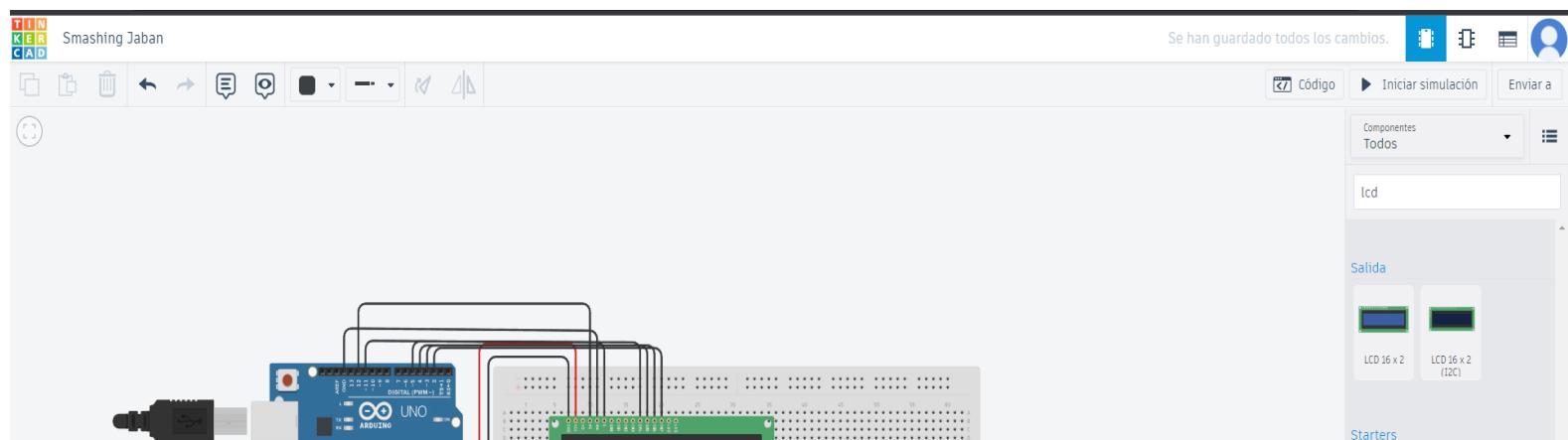
Un dia muy tranquilo pero a la final empezo a llover, me puso a ver videos en youtube del pong, ya que podria ayudar a Sofia con eso.

https://www.youtube.com/watch?v=xk_qRnc4oM8



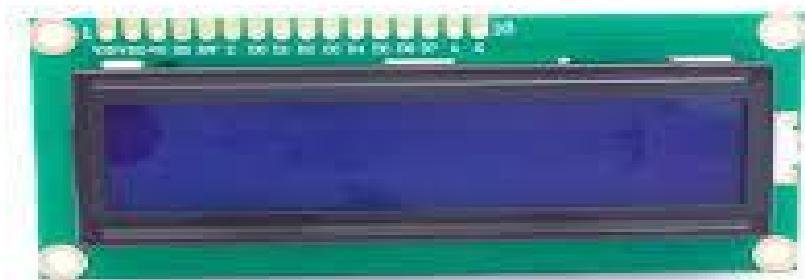
24/11/23

Hoy en la hora de Programación me puse a adelantar y programar el código, pero me di cuenta que no está la pantalla que estamos utilizando la pantalla 128x64



Cosa que me pareció muy rara, ya que si estaban esas pantallas
lcd 12x64 deberían de estar las de 128x64.

Investigue sobre eso pero no me decía nada, y si salía la pantalla
lcd 128x64 era físicamente y no por tinkercad
Pantalla LCD 16X2



PANTALLA 128X64



En la hora del proyecto me decidí de actualizar la carpeta de campo, cosa que hice hasta que me di cuenta que no se encontraba Sofía en el aula.

Después de esperar un rato revise en lab donde se encontraba con el Mono y otros, imprimiendo partes de sus proyectos en 3D.

En nuestro caso se complicó un poco, ya que pasó algo con la carcasa de BMO, así que tardaremos en resolver eso, un poco.

Sofia me dijo que ya me pasaría los bocetos después que terminemos con el problema de la carcasa



Le pregunté al profe para aclarar mi duda sobre porque no sale la pantalla de 128x64, bueno él me dijo que solo se trabaja en físico y no sale en tinkercad.

No se como podriamos ejecutar en tinkercad, sin quemar un arduino ya que para eso es arduino, para hacer un prototipo sin dañar nada, trataremos de hacerlo con el arduino y la pantalla lcd en físico y que sea lo que dios quiera, cuidando los equipos y herramientas.

Me puse a investigar a ver si había simuladores parecidos al tinkercad y de lcd pero no dio un buen resultado, ya que no se encontraba lo que queria.

26/11/23

Hoy me levante y empeze actualizar la carpeta.

Estaba viendo en classroom como hacer la entrega de mi proyecto, para ello debia imprimir este documento en pdf.

Las imagenes las tenia sofia a si que se las pedi y solo espere a que ella me las pasara para subirla en el git hub.

Cree el repositorio