

# DEBIDO AL TAMAÑO DEL VIDEO LO HE SUBIDO A YOUTUBE:

- En el video voy explicando esta presentación y hago una demo del juego.
- Te dejo el link por aquí:

- <https://youtu.be/OA1QPDT8k2s>

# JUEGO PONG

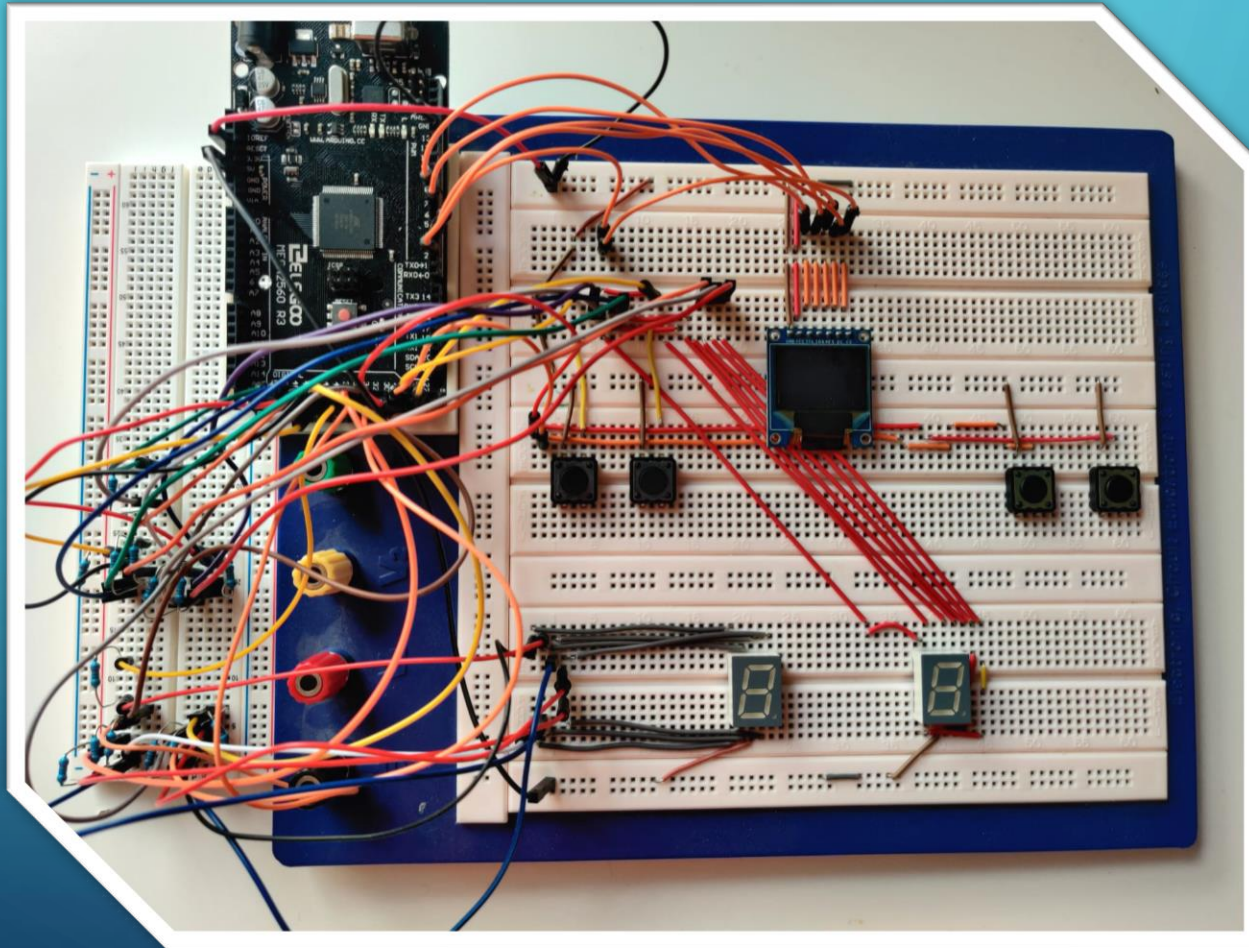
JORGE LÓPEZ ALONSO DEL HOYO (52439)

# INTRODUCCIÓN

- Para el trabajo de este año decidí centrarme más en la programación que en el aspecto electrónico y por ello he desarrollado una versión del juego clásico Pong.
- El modo de juego es 1v1, para ello controlas el movimiento de la raqueta con dos botones.
- Está todo desarrollado en el entorno de Arduino y controlado por una placa Arduino Mega.



# COMPONENTES

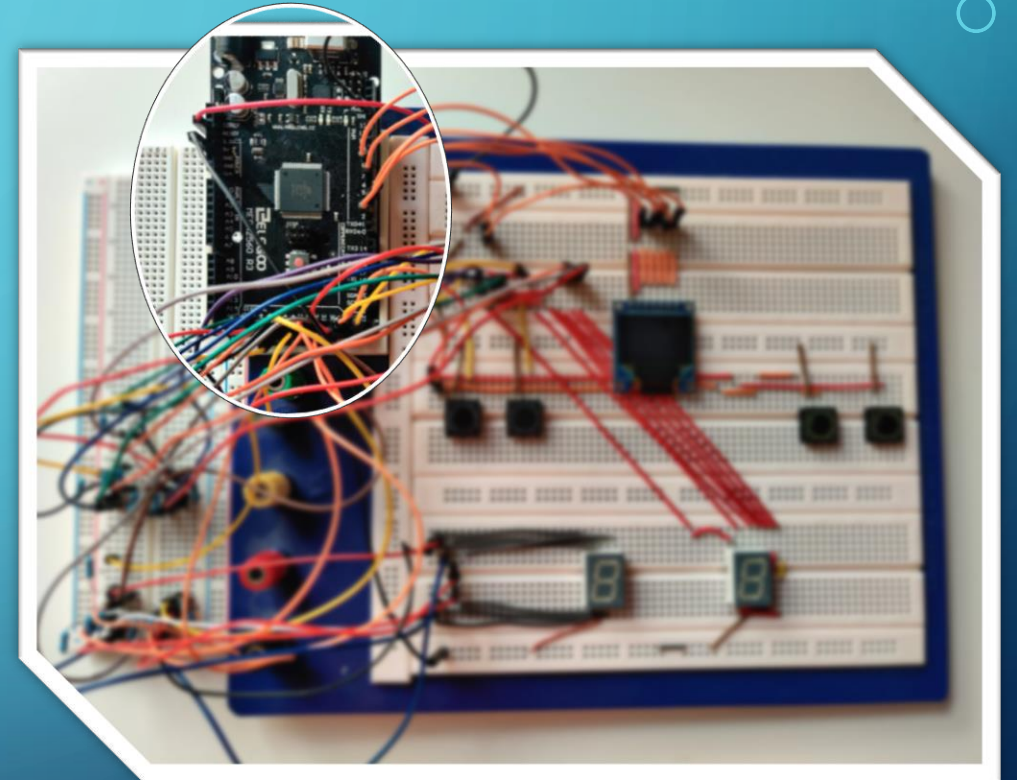


# PLACA ARDUINO MEGA

Este es el microcontrolador que nos permite programar y controlar el resto de componentes.

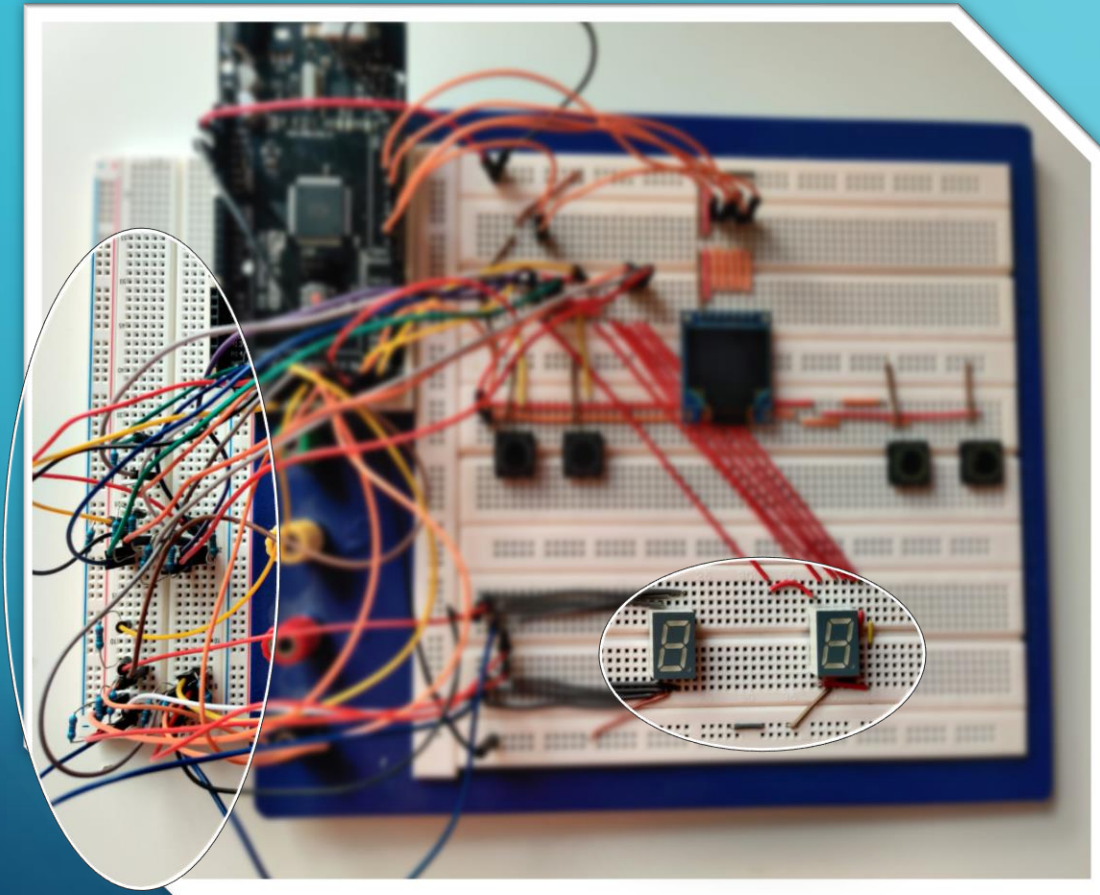
En la parte de programación comentaré algún aspecto más sobre el entorno.

En un principio use una placa de Arduino Uno, pero pronto me di cuenta que me iban a faltar pines de conexión.





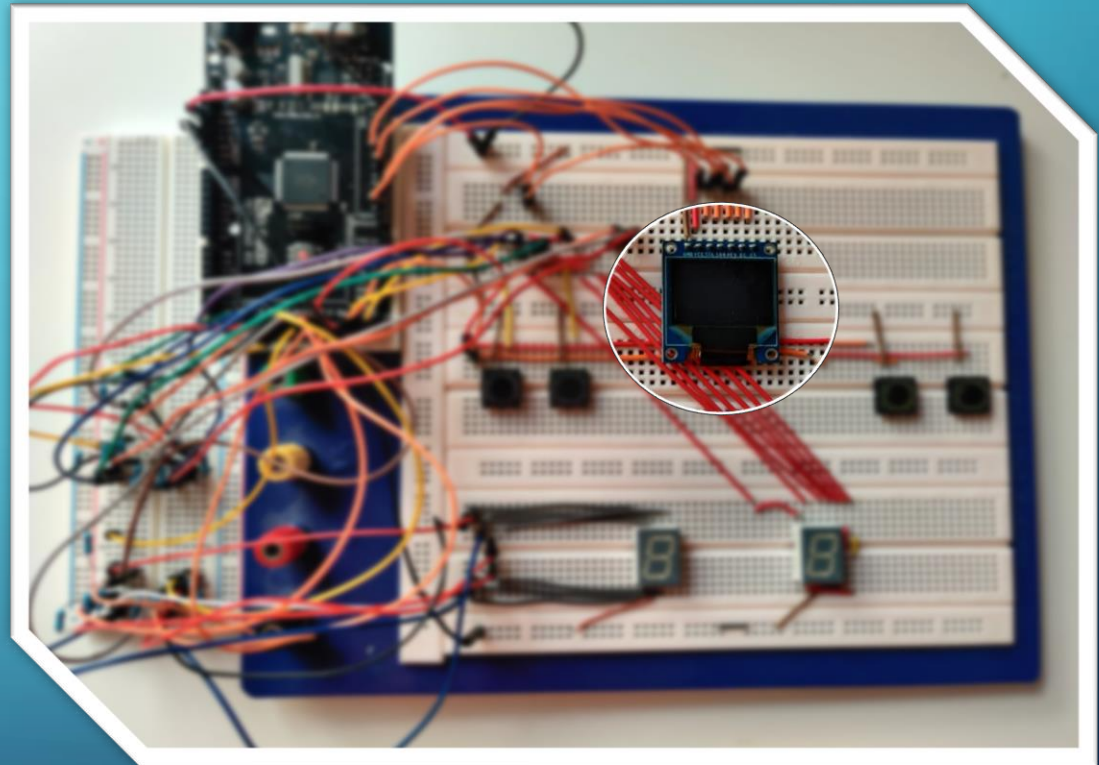
# DISPLAY DE 7 SEGMENTOS



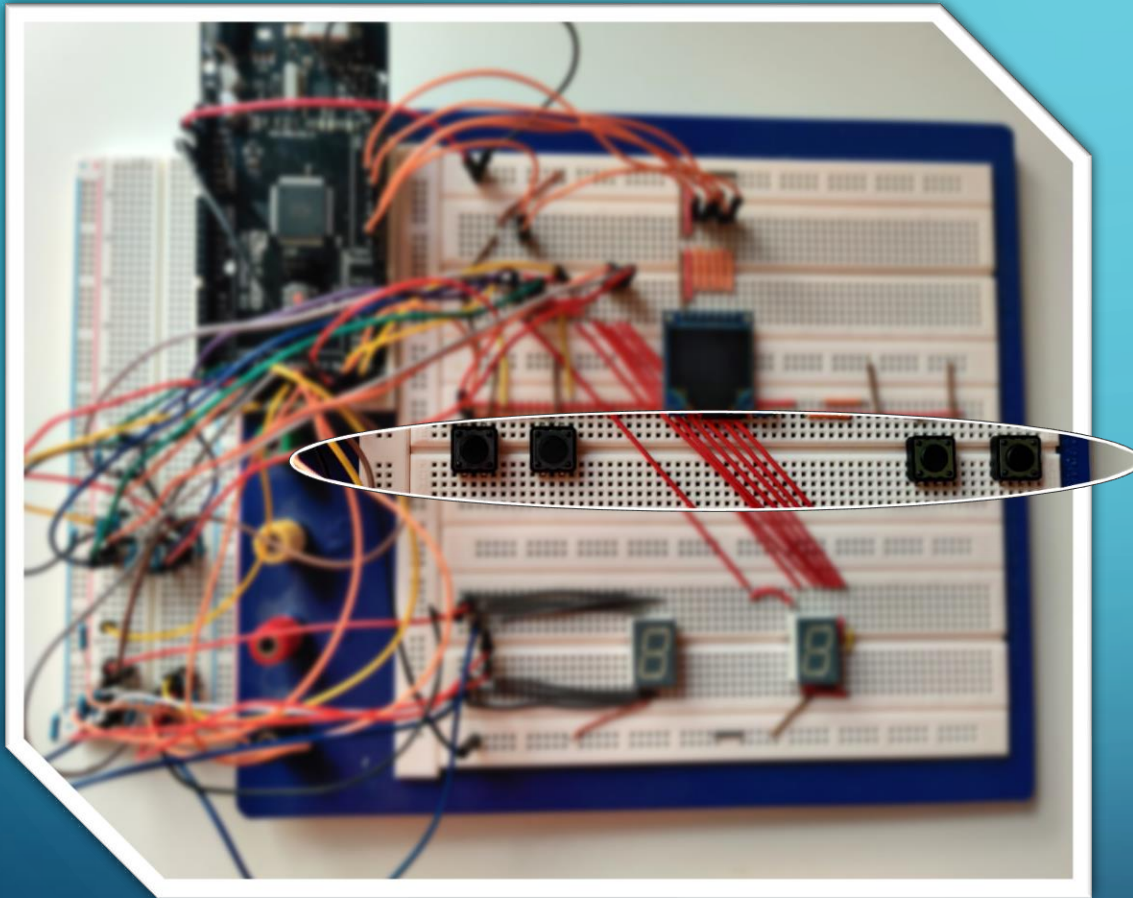
- Ofrecen una funcionalidad extra que es la de mostrar el marcador en tiempo real.
- Son los culpables de la mayoría de cables que hay en el circuito, ya que aunque aparentemente sencillos, requieren bastante atención.
- En el apartado de programación entraré en más detalle.

# OLED DISPLAY

- Esta pequeña pantalla de 94x94 pixeles nos permite mostrar el juego así como el marcador.
- La podemos controlar con gran facilidad gracias a la librería Adafruit.h
- Su uso es bastante sencillo y permite hacer infinidad de cosas



# BOTONES



- Nos permiten mover las raquetas de cada jugador.
- Los botones de la izquierda harán que la pala suba y los de la derecha que bajen. Cada par de botones corresponde a un jugador.



# CÓDIGO

## STRUCTS

## FUNCIONES

```
///Definimos las funciones que vamos a usar///
void choqueParedHorizontal ();
void choqueParedVertical ();
void choquePala ();

void Reset ();
void resetPelota ();
void resetJugadores();
void setJugadores ();

void dibujaPelota();
void dibujarPista();

void finJuego();
void marcador();
```

```
void mostrarPuntuacion();
```

```
void mostrarJugadores();
```

```
///Usaremos struct para facilitar el código///
/*Vector2D nos permite usar unas
//coordenadas x,y para la posición de la pelota y el jugador*/
struct Vector2D {
    uint8_t x;
    uint8_t y;
};

/*Jugador tiene 3 variables, la posición, la puntuación
y el refresco, que equivale a la velocidad que lleva.*/
struct jugador
{
    struct Vector2D pos;
    int punt;
    long refre;
};

//Añadimos la variable dirección a la pelota
struct pelota
{
    struct Vector2D pos;
    struct Vector2D dir;
    long refre;
};

//Inicializamos las estructuras
//Creamos dos pelotas;
pelota pel_ini; //{{64,32},{0,0},0}; //Se crea al ppio y cada vez que hacemos un punto
pelota pel; // {{0,0},{1,1},0}; //Pelota que usamos mientras estamos jugando

//Creamos dos jugadores
jugador jug1 {{12,16},0,0}; //Jugador de la izda
jugador jug2 {{85,16},0,0}; //dcha
```

```
void mostrarPuntuacion();
```

```
void mostrarJugadores();
```

```
void mostrarJugadores();
```

## DISPLAY DE 7 SEGMENTOS

```
///Display 7 Segmentos Catodo Común///  
int contador[10]= {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x67};  
byte a=32;  
byte b=33;  
byte c=34;  
byte d=35;  
byte e=36;  
byte f=37;  
byte g=38;  
  
byte h=42;  
byte i=43;  
byte j=44;  
byte k=45;  
byte l=46;  
byte m=47;  
byte n=48;
```

```
//////////FUNCIONES PARA EL MARCADOR//////////  
/*Función que coloca en el puerto de salida los bits comenzando  
desde el pin ini hasta el pin fin*/  
void puerto(int bits,int ini,int fin)  
{  
    for(int i=ini;i<=fin;i++)  
    {  
        digitalWrite(i,bitRead(bits,i-ini));  
    }  
}  
  
//Dibujamos tanto el marcador en la pantalla como en el display de 7seg  
void marcador()  
{  
    //Llamamos a la función puerto de arriba  
    puerto(contador[jug1.punt],a,g);  
    puerto(contador[jug2.punt],h,n);
```

# ELECTRÓNICA Y DIFICULTADES

