

# CARPETA DE CAMPO

**nombre:** juan

**apellido:** Luna

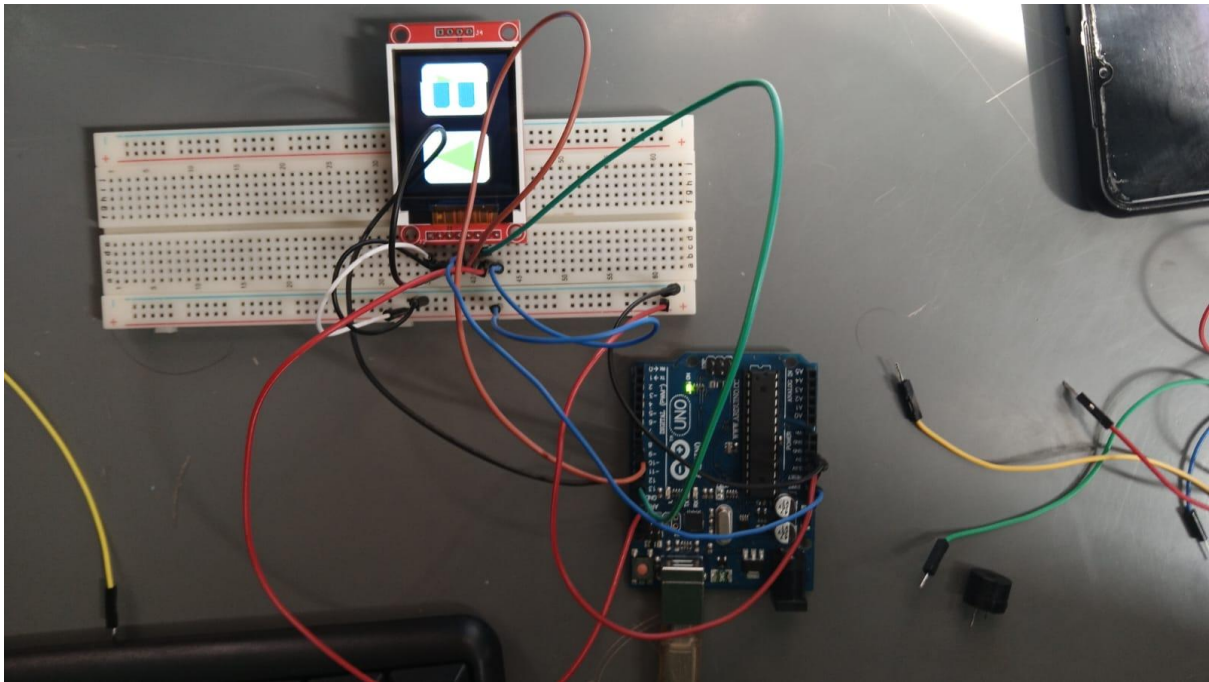
**año y curso:** 4to 2da

con mi grupo al final cambiamos de juego arcade que es el “pong” que estamos usando una pantalla TFT de 1.4 a 3.2 dos joysticks y un buzzer junto a un diseño 3D para que quedara una “maquina arcade”. La idea es que con este juego se pueda jugar con dos personas haci pueden jugar uno contra el otro

en el juego del pong ya tengo un aidea de como va a ser y voy a hacer como una “binvenida” y que comience el juego el juego va a ser hasta 3 o mas por si lo cambio. despues de los 3 puntos queria hacer una despedida haci queda mejor

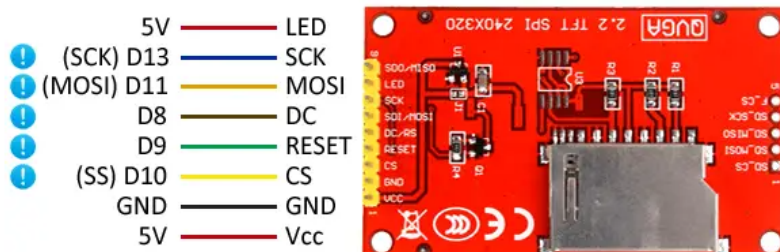
otra de las ideas ya que vamos a hacerlo con los joystisck tenia pensado que con los analogicos la paleta tire con mas fuerza y haci llendo mas rapido la pelota como para ponerle algo que lo haga mas divertido

En el primer dia nos pusimos a conectar la pantalla y probar varios ejemplos para ver como es y para guiarnos.



**conexiones:**

**joysticks:** A0 y A1 para el jugador 1 A2 y A3 para el jugador2 gnd y 5v  
**pantalla :**



Ya con todo conectado empezamos con el código y probamos con uno que yo lleve la clase siguiente de referencia lo saque de google porque no tenía ni idea de cómo se dibuja en la pantalla, así que tuvimos el error de que la pantalla se dibujaba mal y no hacía nada de lo que queríamos y además agregué librerías que no eran para la pantalla

```
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>
#include <Adafruit_STMPE610.h>
```

```
#define TFT_CS 10
#define TFT_DC 9
```

```
#define JOYSTICK1_X A0
#define JOYSTICK1_Y A1
#define JOYSTICK2_X A2
#define JOYSTICK2_Y A3
```

```
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);
Adafruit_STMPE610 ts = Adafruit_STMPE610();
```

```
int ballX, ballY, ballSpeedX, ballSpeedY;
int paddle1Y, paddle2Y;
int paddleHeight = 40;
int paddleWidth = 5;
int ballSize = 5;
int screenWidth, screenHeight;
```

```
void setup() {
  tft.begin();
  ts.begin();
  tft.setRotation(3);
  screenWidth = tft.width();
  screenHeight = tft.height();
```

```
  ballX = screenWidth / 2;
  ballY = screenHeight / 2;
  ballSpeedX = 2;
  ballSpeedY = 2;
```

```
  paddle1Y = screenHeight / 2 - paddleHeight / 2;
  paddle2Y = screenHeight / 2 - paddleHeight / 2;
}
```

```
void loop() {
  // Leer la posición de los joysticks
  int joystick1X = analogRead(JOYSTICK1_X);
  int joystick1Y = analogRead(JOYSTICK1_Y);
  int joystick2X = analogRead(JOYSTICK2_X);
  int joystick2Y = analogRead(JOYSTICK2_Y);

  // Actualizar la posición de las paletas
  paddle1Y = map(joystick1Y, 0, 1023, 0, screenHeight - paddleHeight);
```

```

paddle2Y = map(joystick2Y, 0, 1023, 0, screenHeight - paddleHeight);

// Actualizar la posición de la pelota
ballX += ballSpeedX;
ballY += ballSpeedY;

// Detección de colisiones con paletas
if (ballX < paddleWidth && ballY > paddle1Y && ballY < paddle1Y +
paddleHeight) {
    ballSpeedX = -ballSpeedX;
}
if (ballX > screenWidth - paddleWidth && ballY > paddle2Y && ballY <
paddle2Y + paddleHeight) {
    ballSpeedX = -ballSpeedX;
}

// Detección de colisiones con bordes
if (ballY < 0 || ballY > screenHeight - ballSize) {
    ballSpeedY = -ballSpeedY;
}

// Detección de puntos
if (ballX < 0 || ballX > screenWidth) {
    // Punto para uno de los jugadores, reiniciar la pelota
    ballX = screenWidth / 2;
    ballY = screenHeight / 2;
    ballSpeedX = -ballSpeedX;
}

// Limpiar pantalla
tft.fillScreen(ILI9341_BLACK);

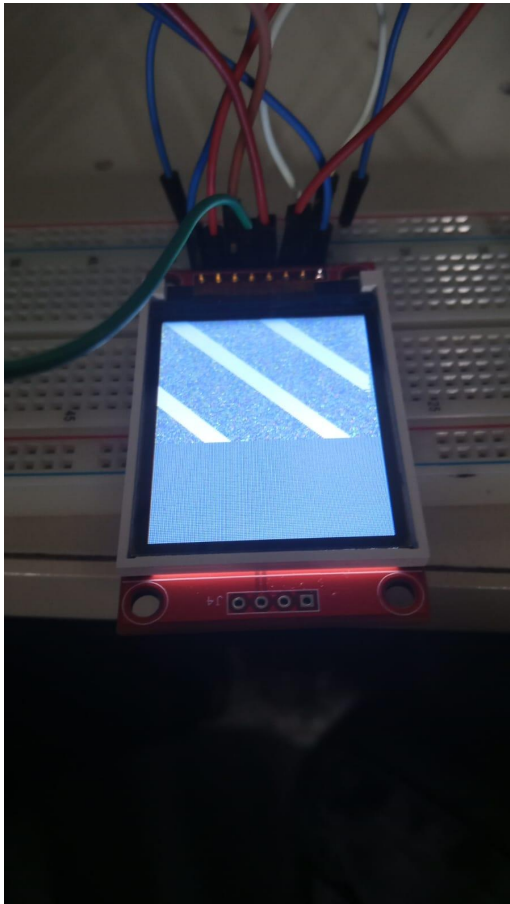
// Dibujar paletas y pelota
tft.fillRect(0, paddle1Y, paddleWidth, paddleHeight, ILI9341_WHITE);
tft.fillRect(screenWidth - paddleWidth, paddle2Y, paddleWidth, paddleHeight,
ILI9341_WHITE);
tft.fillCircle(ballX, ballY, ballSize, ILI9341_WHITE);

// Esperar un breve tiempo
delay(10);

```

}

y nos quedaba de la siguiente manera (nada que ver con un pong)



entonces nos pusimos a hacer otro código nuevo porque el problema eran las librerías porque agarre unas de otra pantalla ya que la librerías de la pantalla en la guía de la pantalla que nos mando consorti no eran las mismas que yo usaba en el código entonces me puse a ver videos para saber como usar la pantalla y poder arreglar el problema que tuvimos con la pantalla y ya poder hacer que de alguna imagen y bien.

### **CODIGOS DE LA PANTALLA:**

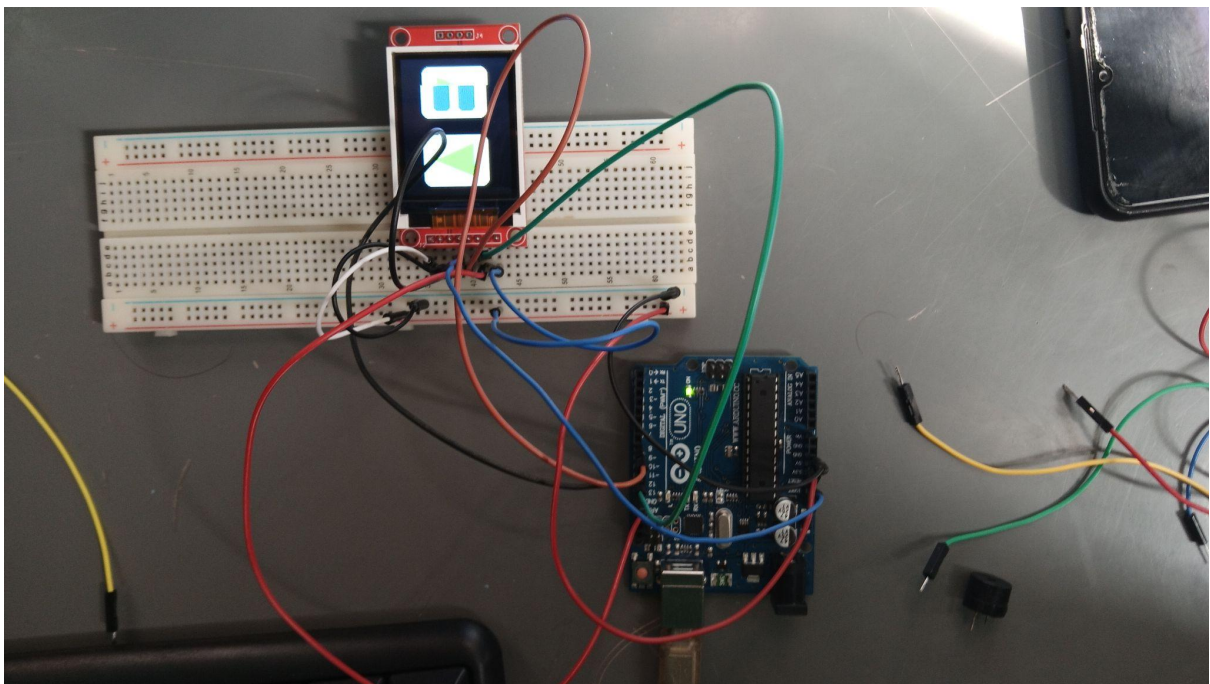
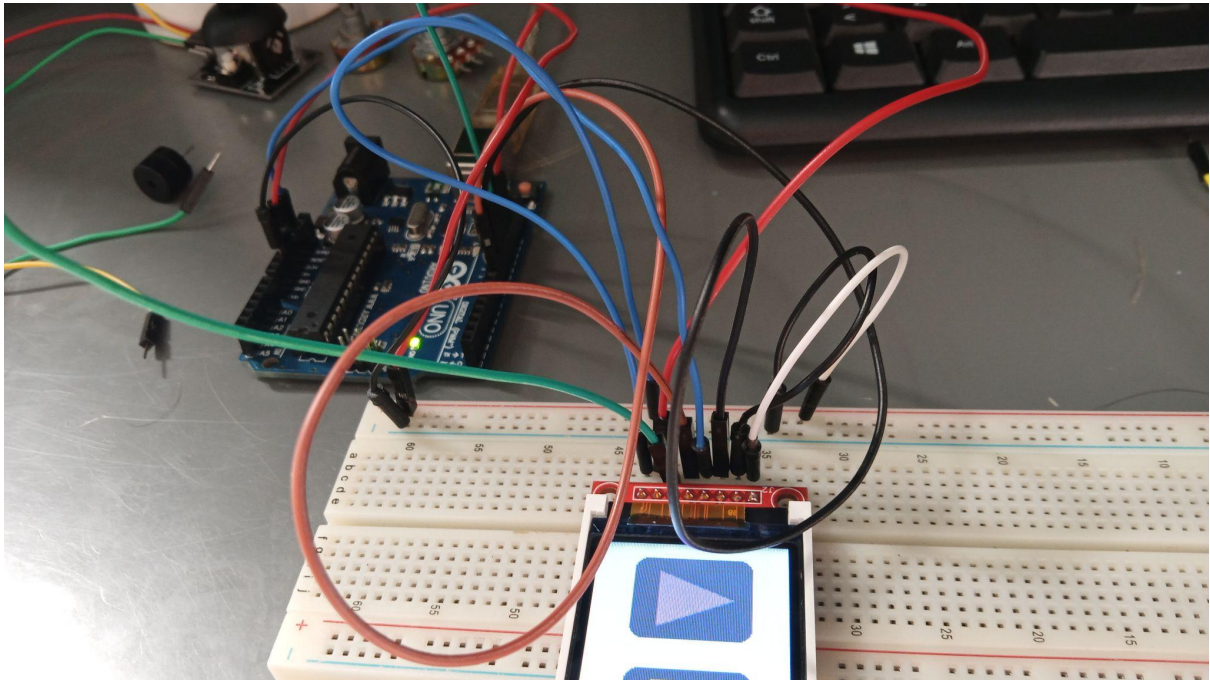
**tft.setTextSize:** La función `tft.setTextSize` ajusta el tamaño del texto que se imprimirá en la pantalla TFT

**tft.setCursor:** La función `tft.setCursor` establece la posición de inicio para imprimir texto en la pantalla TFT.

**tft.setTextColor(tft.color565:** La función `tft.setTextColor(tft.color565(...))` establece el color del texto que se imprimirá en la pantalla TFT utilizando el formato de color RGB565. Específicamente, `tft.color565` toma valores para los componentes rojo, verde y azul y los combina en un solo valor de 16 bits que representa el color.

**tft.print:** La función `tft.print` se utiliza para emplea para mostrar mensajes, números u otras cadenas de texto

con los ejemplos de que teníamos con las librerías pudimos saber como funciona cada cosa para poder hacer dibujar la pantalla



```
#include <Adafruit_GFX.h>  
#include <Adafruit_ILI9341.h>
```



```

#define TFT_CS 10
#define TFT_DC 9
#define TFT_RST -1 // RST pin es opcional, si no se utiliza, establecer en -1
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC, TFT_RST);

// Variables para la pelota
float ballX, ballY, ballSpeedX, ballSpeedY, ballRadius;

// Variables para las paletas
float paddle1Y, paddle2Y, paddleWidth, paddleHeight, paddleSpeed;

// Variables para el tamaño de la pantalla
int screenWidth, screenHeight;

void setup() {
  Serial.begin(9600);

  tft.begin();
  tft.setRotation(3); // Ajusta la orientación de la pantalla si es necesario

  screenWidth = tft.width();
  screenHeight = tft.height();

  ballX = screenWidth / 2;
  ballY = screenHeight / 2;
  ballSpeedX = 2.0;
  ballSpeedY = 1.5;
  ballRadius = 5.0;

  paddleWidth = 10;
  paddleHeight = 40;
  paddleSpeed = 2.0;

  paddle1Y = screenHeight / 2 - paddleHeight / 2;
  paddle2Y = screenHeight / 2 - paddleHeight / 2;
}

void loop() {
  // Actualizar la posición de la pelota y las paletas
  updateBall();
}

```

```

updatePaddles();

// Limpiar la pantalla
tft.fillScreen(ILI9341_BLACK);

// Dibujar la pelota y las paletas
tft.fillCircle(ballX, ballY, ballRadius, ILI9341_WHITE);
tft.fillRect(0, paddle1Y, paddleWidth, paddleHeight, ILI9341_WHITE);
tft.fillRect(screenWidth - paddleWidth, paddle2Y, paddleWidth, paddleHeight,
ILI9341_WHITE);

// Esperar un breve periodo de tiempo
delay(10);
}

void updateBall() {
    ballX += ballSpeedX;
    ballY += ballSpeedY;

    // Rebote en los bordes superior e inferior
    if (ballY - ballRadius < 0 || ballY + ballRadius > screenHeight) {
        ballSpeedY = -ballSpeedY;
    }

    // Rebote en las paletas
    if (ballX - ballRadius < paddleWidth && ballY > paddle1Y && ballY <
paddle1Y + paddleHeight) {
        ballSpeedX = -ballSpeedX;
    }

    if (ballX + ballRadius > screenWidth - paddleWidth && ballY > paddle2Y &&
ballY < paddle2Y + paddleHeight) {
        ballSpeedX = -ballSpeedX;
    }

    // Verificar si se anotó un punto (pelota pasó las paletas)
    if (ballX - ballRadius < 0 || ballX + ballRadius > screenWidth) {
        // Reiniciar la posición de la pelota
        ballX = screenWidth / 2;
        ballY = screenHeight / 2;
    }
}

```



```
}  
}
```

```
void updatePaddles() {  
    // Controlar las paletas con botones (por ejemplo, usando analogRead)  
    int paddle1Input = analogRead(A0);  
    int paddle2Input = analogRead(A1);  
  
    // Mapear los valores de entrada a la posición de las paletas  
    paddle1Y = map(paddle1Input, 0, 1023, 0, screenHeight - paddleHeight);  
    paddle2Y = map(paddle2Input, 0, 1023, 0, screenHeight - paddleHeight);  
}
```

con este código ya estuve usando las librerías de la pantalla

**LIBRERÍAS PARA LA PANTALLA:** [Adafruit ILI9341](#), [Adafruit ST7735](#) y [Adafruit HX8357](#)

21/11

para esta clase trajimos un código nuevo para el pong pero el problema que teníamos es que las paletas y la pelota no se movía

aca dejo el código y explico que hace cada cosa y después dejo el código nuevo y explico que se agregó y porque

```
#include <Adafruit_GFX.h>  
#include <Adafruit_ST7735.h>  
#include <SPI.h>
```

```
#define TFT_CS 10  
#define TFT_RST 9  
#define TFT_DC 8
```

//aca se incluye las librerías para la pantalla y defino los Aquí se definen pines para el control del display: TFT\_CS, TFT\_RST para el reset, y TFT\_DC para el modo de datos/mando.

```
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);
```

//Se crea una instancia de Adafruit\_ST7735 utilizando los pines definidos anteriormente.

```
int paddleWidth = 30;  
int paddleHeight = 5;  
int ballSize = 6;
```

//aca se definen las dimensiones de las paletas del jugador y computadora y la pelota

```
int playerPaddleX = 60;  
int computerPaddleX = 60;  
int playerPaddleY = 120;  
int computerPaddleY = 10;
```

//Se establecen las posiciones iniciales de las paletas del jugador y la computadora.

```
int ballX = 50;  
int ballY = 50;  
int ballSpeedX = 5;  
int ballSpeedY = 5;
```

//Se definen las coordenadas iniciales y las velocidades de la pelota.

```
void setup() {  
  tft.initR(INITR_BLACKTAB);  
  tft.setRotation(3);  
  tft.fillScreen(tft.color565(0, 0, 0)); // Fondo negro  
  
  // Inicializa la posición de las paletas y la pelota  
  // ...  
  
  // Dibuja las paletas y la pelota  
  drawPaddle(playerPaddleX, playerPaddleY);  
  drawPaddle(computerPaddleX, computerPaddleY);  
  drawBall(ballX, ballY);  
}
```

//en el setup se inicia la pantalla en negro y luego dibuja la pantalla y las paletas en su posicion inicial

```

void loop() {
  // Lógica del juego aquí
  moveBall(); // Mueve la pelota
  // Otras acciones del juego como detección de colisiones y actualización de
  puntajes

  // Borra las posiciones anteriores de la pelota y las paletas
  clear(playerPaddleX, playerPaddleY, paddleWidth, paddleHeight);
  clear(computerPaddleX, computerPaddleY, paddleWidth, paddleHeight);
  clear(ballX - ballSpeedX, ballY - ballSpeedY, ballSize, ballSize);

  // Dibuja las paletas y la pelota en las nuevas posiciones
  drawPaddle(playerPaddleX, playerPaddleY);
  drawPaddle(computerPaddleX, computerPaddleY);
  drawBall(ballX, ballY);

  delay(100); // Ajusta el retraso según sea necesario
}

```

//en el void loop se supone que tiene que borrar las posiciones anteriores y dibujar las nuevas pero tenemos problemas con eso ya que el clear no nos lo toma y queda dibujado. y le agregamos un delay de 100 para que sea rapido la limpieza y el dibujado en la pantalla

```

void drawPaddle(int x, int y) {
  tft.fillRect(x, y, paddleWidth, paddleHeight, tft.color565(255, 255, 255));
}

```

//dibuja la paleta en las coordenadas especificas de la pantalla

```

void drawBall(int x, int y) {
  tft.fillCircle(x, y, ballSize / 2, tft.color565(255, 255, 255));
}

```

//lo mismo pero con la pelota

```

void moveBall() {
  // Actualiza las coordenadas de la pelota
  ballX += ballSpeedX;
  ballY += ballSpeedY;
}

```

```

// Lógica para invertir la dirección si la pelota alcanza los bordes
if (ballX <= 0 || ballX >= tft.width()) {
    ballSpeedX = -ballSpeedX;
}
if (ballY <= 0 || ballY >= tft.height()) {
    ballSpeedY = -ballSpeedY;
}
}

```

//en la funcion moveball actualiza las coordenadas de la pelota y verifica si llega a los bordes de la pantalla haciendo que cuando choque rebote

```

void clear(int x, int y, int width, int height) {
    tft.fillRect(x, y, width, height, tft.color565(0, 0, 0));
}

```

//La función clear borra un rectángulo en las coordenadas especificadas.

**entonces me puse a trabajar para arreglar el problema y poder hacer que se mueva la pelota pero tambien tenia que crear una funcion para que la pelota cuando se mueva se valla borrando por donde pasa haci no deja el rastro lo mismo las paletas. y ya trabajar con los puntos, colisión y el final del juego para que cuando uno haga un punto que borre toda la pantalla por completo y empiece todo de nuevo**

**codigo corregido:**

```

#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <SPI.h>

```

```

#define TFT_CS 10
#define TFT_RST 9
#define TFT_DC 8

```

```

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

```

```

int paddleWidth = 30;
int paddleHeight = 5;

```

```

int playerPaddleX = 60;
int computerPaddleX = 60;
int playerPaddleY = 120;
int computerPaddleY = 0;

int ballSize = 6;
int ballX = 40;
int ballY = 40;
int ballSpeedX = 2;
int ballSpeedY = 2;

void setup() {
  Serial.begin(9600);
  tft.initR(INITR_GREENTAB);
  tft.setRotation(3);
  tft.fillScreen(tft.color565(0, 0, 0)); // Fondo negro

  // Inicializa la posición de las paletas y la pelota
  // ...

  // Dibuja las paletas y la pelota
  drawPaddle(playerPaddleX, playerPaddleY);
  drawPaddle(computerPaddleX, computerPaddleY);
  drawBall(ballX, ballY);
}

void loop() {

  // Lógica del juego aquí

  //clear(ballX, ballY, ballSize, ballSize);

  delay(20);
  clear(ballX, ballY );
  moveBall(); // Mueve la pelota

  ////////////
  //esto se cambia por el movimiento de joytick
  if( playerPaddleX <= 130 )
  {
    playerPaddleX = playerPaddleX + 1;
  }
}

```

```
if( computerPaddleX >= 0 )
{
    computerPaddleX = computerPaddleX - 1;
}
```

```
//////////
```

```
drawPaddle(playerPaddleX, playerPaddleY);
drawPaddle(computerPaddleX, computerPaddleY);
drawBall(ballX, ballY);
```

```
// delOtras acciones juego como detección de colisiones y actualización de puntajes
```

```
// Borra las posiciones anteriores de la pelota y las paletas
//clear(playerPaddleX, playerPaddleY, paddleWidth, paddleHeight);
//clear(computerPaddleX, computerPaddleY, paddleWidth, paddleHeight);
// clear(ballX - ballSpeedX, ballY - ballSpeedY, ballSize, ballSize);
```

```
// Dibuja las paletas y la pelota en las nuevas posiciones
//drawPaddle(playerPaddleX, playerPaddleY);
//drawPaddle(computerPaddleX, computerPaddleY);
```

```
//clearScreen();
//delay(2); // Ajusta el retraso según sea necesario
}
```

```
void drawPaddle(int x, int y) {
    tft.fillRect(x, y, paddleWidth, paddleHeight, tft.color565(255, 255, 255));
}
```

```
void drawBall(int x, int y) {
    tft.fillCircle(x, y, ballSize / 2, tft.color565(255, 255, 255));
}
```

```
void moveBall() {
    // Actualiza las coordenadas de la pelota
    ballX += ballSpeedX;
    ballY += ballSpeedY;
```

```
// Lógica para invertir la dirección si la pelota alcanza los bordes
if (ballX <= 0 || ballX >= tft.width()) {
    ballSpeedX = -ballSpeedX;
```

```

    }
    if (ballY <= 0 || ballY >= tft.height()) {
        ballSpeedY = -ballSpeedY;
    }

    if( ballY <= 0 )
    {
        Serial.println("Gana Player");
    }
    if( ballY >= 128 )
    {
        Serial.println("Gana PC");
    }
}
/*
void clear(int x, int y, int width, int height) {
    tft.fillRect(x, y, width, height, tft.color565(0, 0, 0));
}

void clear(int x, int y, int width, int height) {
    tft.fillRect(x, y, width, height, tft.color565(0, 0, 0));
}

void clear(int x, int y, int width, int height) {
    int centerX = x + width / 2;
    int centerY = y + height / 2;
    tft.fillCircle(centerX, centerY, width / 2, tft.color565(0, 0, 0));
}
*/

void clear(int x, int y) {
    int centerX = x + ballSize;
    int centerY = y + ballSize;
    tft.fillCircle(x, y, 15, tft.color565(0, 0, 0));

    tft.fillRect(0, 0, 300, 10, tft.color565(0, 0, 0));
    tft.fillRect(0, 120, 300, 10, tft.color565(0, 0, 0));
}

void clearScreen() {
    //tft.fillScreen(tft.color565(0, 0, 0));

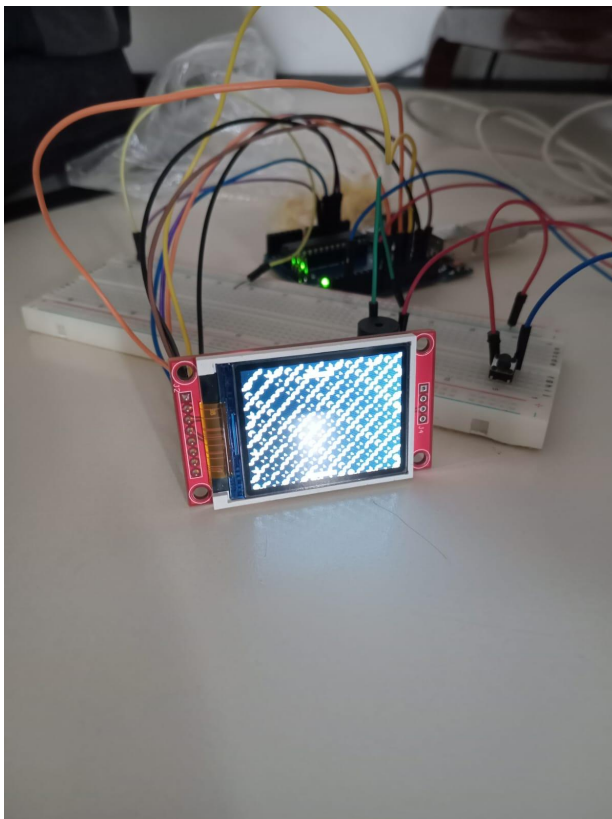
```



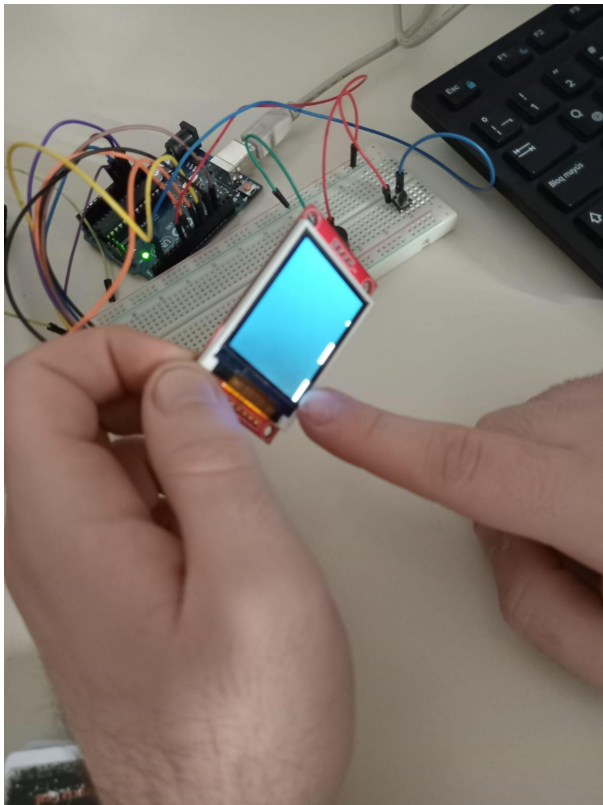
```
tft.fillRect(0, 0, 300, 300, tft.color565(0, 0, 0));  
delayMicroseconds(200);  
}
```

con este codigo agregue lo que viene a ser los clear de la pelota y pantalla, el tope de las paletas, en el codigo agregamos un printLN para que en la consola aparezca quien hizo el punto para corroborar que si funcionaba y haci proseguir con el clearscreen para que se reinicie cuando uno haga un punto y a mediados que hago esto tengo ideas como agregarle una trampa o un mini juego en medio del pong para que sea mas divertido e dinamico como que la pelota se haga el triple de grande o mas rapida o que se duplique o triplique cuando colisione o agregar dos paletas para cada uno o que la paleta del otro cambie el sentido en decir que si quiere ir a la derecha valla a la izquierda o se le congele la paleta al otro. pero bueno eso vere a mediados que valla haciendo el codigo. si se me hace muy dificil ya se me ocurrira otra cosa

agrego imagenes de lo que nos pasaba con el codigo en pantalla:



este es unos de los problemas que teniamos y es que la funcion que hicimos “clear” para que la pelota sea borrada de donde paso no se borraba o en algunos casos se borraba la mitad y la otra quedaba dibujada o demas.



despues de solucionar lo del clear con el profe queriamos saber cual era la paleta de palyer y la computadora o player2 y despues de solucionar ese tema colocamos las paletas en su lugar arreglamos tambien un problema del clear en las paletas y que tuvieran un tope en los costados de la pantalla.(no tengo imagenes de eso ya que me colgue)

**VIERNES 24/11**

hoy me puse a trabajar con el tema de las colisiones con las paletas ya que la pelota pasaba y traspasaba la paleta hacique hize una nueva funcion que hace que guarde las coordenadas de las paletas para que cuando la pelota este en las mismas coordenadas que la paleta colisione e rebote

```
void checkCollision() {  
    // Colisión con la paleta del jugador  
    if (ballY + ballSize / 2 >= playerPaddleY && ballY - ballSize / 2 <= playerPaddleY + paddleHeight) {  
        if (ballX + ballSize / 2 >= playerPaddleX && ballX - ballSize / 2 <= playerPaddleX + paddleWidth) {  
            // Colisión con la paleta del jugador  
            ballSpeedY = -ballSpeedY;  
        }  
    }  
}
```

```

    }

    // Colisión con la paleta de la computadora
    if (ballY + ballSize / 2 >= computerPaddleY && ballY - ballSize / 2
<= computerPaddleY + paddleHeight) {
        if (ballX + ballSize / 2 >= computerPaddleX && ballX - ballSize / 2
<= computerPaddleX + paddleWidth) {
            // Colisión con la paleta de la computadora
            ballSpeedY = -ballSpeedY;
        }
    }
}

```

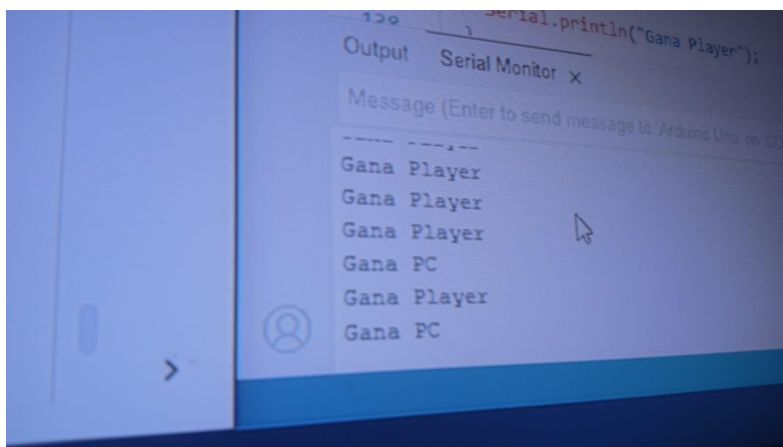
hize un if para las dos paletas como para la computadora como la del player, y despues la deje en el void loop para que sea llamada y realice la colision. y gracias a la consola y los siguientes codigos pude ver si realmente colisionaba y para quien iba el punto

```

    if (ballX <= 0 ||
ballX >= tft.width()) {
        ballSpeedX = -ballSpeedX;
    }
    if (ballY <= 0 || ballY >= tft.height()) {
        ballSpeedY = -ballSpeedY;
    }

    if( ballY <= 0 )
    {
        Serial.println("Gana Player");
    }
    if( ballY >= 128 )
    {
        Serial.println("Gana PC");
    }
}

```



**ahora trabaje en la funcion de “final del juego” para que cuando el player 1 o 2 haga un punto se limpie todo y se reinicie a todo a su lugar de inicio y haci mostrando el puntaje de cada uno**

```
void finDelJuego() {
    // Limpia la pantalla
    clearScreen();

    // Muestra el puntaje (puedes personalizar esta parte según tu lógica de puntaje)
    int puntajePlayer = 0;
    int puntajeComputer = 0;

    // Dibuja el puntaje en la pantalla
    tft.setTextSize(2);
    tft.setCursor(40, 40);
    tft.setTextColor(tft.color565(255, 255, 255));
    tft.print("Player: ");
    tft.print(puntajePlayer);

    tft.setCursor(40, 80);
    tft.print("Computer: ");
    tft.print(puntajeComputer);

    delay(2000); // Espera unos segundos para que el jugador pueda ver el puntaje

    // Limpia la pantalla nuevamente antes de reiniciar el juego
    clearScreen();

    // Reinicia las posiciones iniciales de las paletas y la pelota
    playerPaddleX = 60;
    computerPaddleX = 60;
    playerPaddleY = 120;
    computerPaddleY = 0;
    ballX = 40;
    ballY = 40;
    ballSpeedX = 2;
    ballSpeedY = 2;
}
```

**con este codigo hice el “final del juego” al principio tuve problemas con el dibujado ya que no se borraba pero lo solucione con un clear, le deje un delay como para que ambos pudieran ver el puntaje, deje la pelota y paleta en su lugar inicial, use el clear screen como para limpiar toda la**

**pantalla y a la funcion de colision le agregue in IF para que verifique que llegue la pelota al extremo inferior o superior para que haci sea llamada la funcion fin del juego y limpiar la pantalla y empezar otra vez**

```
if (ballY <= 0 || ballY >= tft.height()) {  
    finDelJuego();  
}
```



**y luego de hacer esto hice una funcion nueva “inicio de juego” que basicamente es unos segundos donde muestra en pantalla”pong del grupo 3” y una cuenta regresiva de 3 segundos**

```
void inicioDelJuego() {  
    // Limpia la pantalla  
    clearScreen();  
  
    // Muestra el mensaje de inicio en tres líneas  
    tft.setTextSize(2);  
    tft.setCursor(30, 20);  
    tft.setTextColor(tft.color565(255, 255, 255));  
    tft.print("Pong");  
  
    tft.setCursor(30, 60);  
    tft.print("del");  
  
    tft.setCursor(30, 100);  
    tft.print("Grupo 3");  
  
    // Espera 1 segundo antes de comenzar la cuenta regresiva  
    delay(3000);
```

```

// Cuenta regresiva
for (int i = 3; i > 0; i--) {
    // Limpia la pantalla antes de mostrar el siguiente número
    clearScreen();

    // Muestra el número de la cuenta regresiva
    tft.setCursor(150, 40);
    tft.print(i);

    // Espera 1 segundo antes de mostrar el siguiente número
    delay(1000);
}

// Limpia la pantalla antes de comenzar el juego
clearScreen();
}

```



**ahora cambie la funcion fin del juego porque me queda incomodo para hacer la funcion de los puntos porque los guardaba en “fin del juego”, entonces hice como un orden iniciodejuego, puntaje y fin del juego para que despues de los 3 puntos se reinicie termine y muestre en pantalla “!gracias por jugar!”**

```

void finDelJuego() {
    // Limpia la pantalla
    clearScreen();

    // Muestra el mensaje de agradecimiento
    tft.setTextSize(2);
    tft.setCursor(3, 40);
    tft.setTextColor(tft.color565(255, 255, 255));
    tft.print("!Gracias por");
    tft.setTextSize(2);
    tft.setCursor(3, 80);
    tft.setTextColor(tft.color565(255, 255, 255));
}

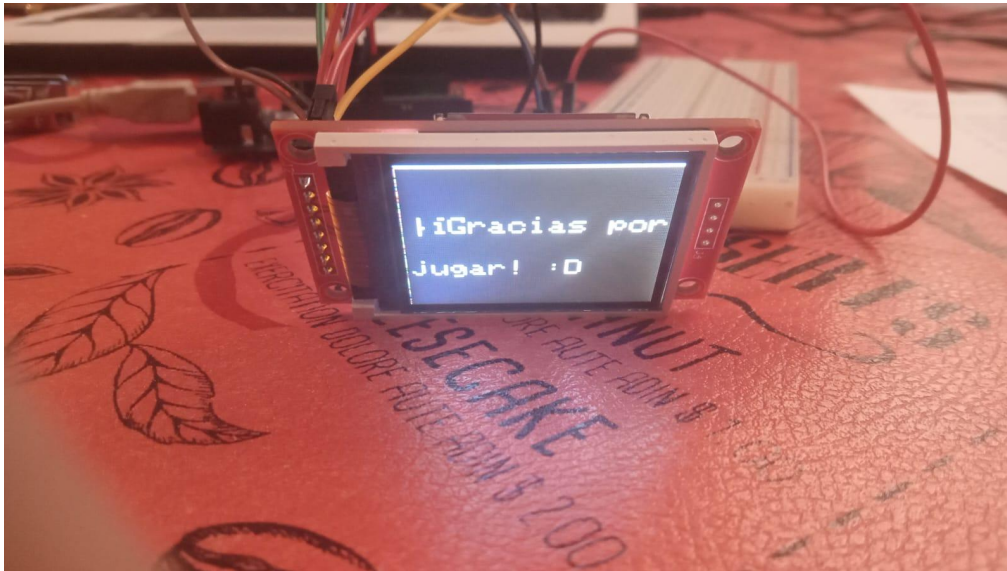
```



```
tft.print("jugar! :D");
```

```
delay(3000); // Espera unos segundos para que el jugador pueda ver el mensaje
```

```
// Limpia la pantalla nuevamente antes de reiniciar el juego  
clearScreen();
```



```
playerPaddleX = 60;  
computerPaddleX = 60;  
playerPaddleY = 120;  
computerPaddleY = 0;  
ballX = 40;  
ballY = 40;  
ballSpeedX = 2;  
ballSpeedY = 2;
```

coordenadas de las paletas y la pelota

**en el codigo agregue dos funciones mas, una me reinicia las posiciones de las paletas y la pelota porque se me hacia quilombo al ponerlo en “findeljuego” y la otra marca el gol y los cuenta hasta 3.**

```
void mostrarGol() {  
  clearScreen();  
  |  
  tft.setTextSize(2);  
  tft.setCursor(30, 40);  
  tft.setTextColor(tft.color565(255, 255, 255));  
  tft.print("¡Gol!");  
}
```



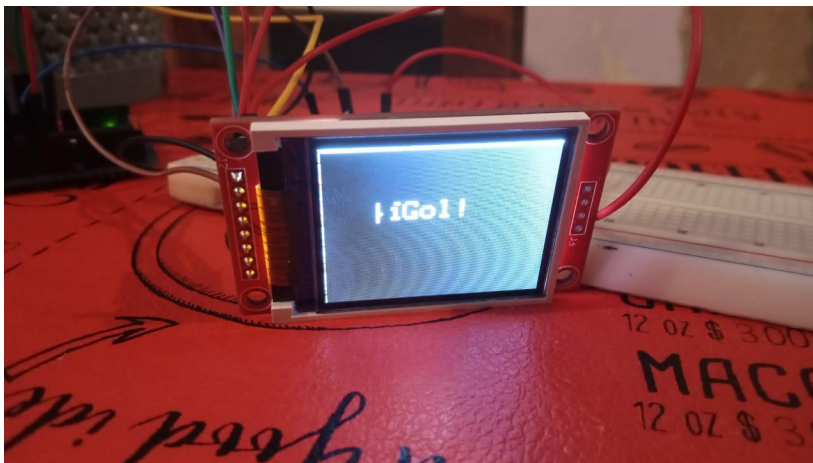
```

    delay(2000);
    clearScreen();
    posiciones();
}

void posiciones() {

    playerPaddleX = 60;
    computerPaddleX = 60;
    playerPaddleY = 120;
    computerPaddleY = 0;
    ballX = 40;
    ballY = 40;
    ballSpeedX = 2;
    ballSpeedY = 2;
}

```



con la funcion fin del juego la cambie porque mostraba el puntaje de cada uno de los jugadores y para hacer todo el tema de los puntos y que haga un print en cada linea para los numeros me quedaba mas comodo a mi gusto de hacer las funciones por separado y hacer que la funcion fin del juego nada mas funcione cuando los dos jugadores hallan alcanzado los puntos suficientes y haci mostrar en pantalla el “gracias por jugar”

POSIBLE CODIGO PARA LOS JOYSTICKS:

```
#define JOYSTICK_PLAYER A0
```

```
#define JOYSTICK_COMPUTER A1
```

```
// Resto del código...
```

```
void loop() {  
  // Lógica del juego aquí  
  delay(20);  
  clear(ballX, ballY);  
  moveBall(); // Mueve la pelota  
  checkCollision();  
  
  // Lee la entrada de los joysticks y ajusta las posiciones de las paletas  
  int playerJoystickValue = analogRead(JOYSTICK_PLAYER);  
  int computerJoystickValue = analogRead(JOYSTICK_COMPUTER);  
  
  // Ajusta la posición de la paleta del jugador  
  playerPaddleX = map(playerJoystickValue, 0, 1023, 0, tft.width() -  
paddleWidth);  
  
  // Ajusta la posición de la paleta de la computadora  
  computerPaddleX = map(computerJoystickValue, 0, 1023, 0, tft.width() -  
paddleWidth);
```

## **POSIBLE CODIGO PARA USAR LOS POTENCIOMETROS**

```
#define POT_PLAYER A0  
#define POT_COMPUTER A1
```

```
// Resto del código...
```

```
void loop() {  
  // Lógica del juego aquí  
  delay(20);  
  clear(ballX, ballY);  
  moveBall(); // Mueve la pelota  
  checkCollision();  
  
  // Lee la entrada de los potenciómetros y ajusta las posiciones de las paletas  
  int playerPotValue = analogRead(POT_PLAYER);  
  int computerPotValue = analogRead(POT_COMPUTER);  
  
  // Ajusta la posición de la paleta del jugador  
  playerPaddleX = map(playerPotValue, 0, 1023, 0, tft.width() - paddleWidth);
```

```
// Ajusta la posición de la paleta de la computadora
computerPaddleX = map(computerPotValue, 0, 1023, 0, tft.width() - paddleWidth);

// Dibuja las paletas y la pelota en las nuevas posiciones
drawPaddle(playerPaddleX, playerPaddleY);
drawPaddle(computerPaddleX, computerPaddleY);
drawBall(ballX, ballY);

// Resto del código...
}
```

**a la funcion fin del juego le agregue dos lineas de codigo que son los puntos haci cada vez que reinicia vuelven a tener los dos jugadores 0 puntos y haci que sigan jugando los dos**