

1.- Instrucción para modificar el estado de los LED

a) MFS.writeLeds();

b) MFS.blinkLeds();

c) MFS.write();

2.- Encender el LED 1, esperar 1 segundo y apagarlo.

```
a) void loop() {  
    MFS.writeLeds(LED_1, ON); delay(1000);  
    MFS.writeLeds(LED_1, OFF); delay(1000);  
}
```

```
b) void loop() {  
    MFS.writeLeds(LED_2, ON); delay(1000);  
    MFS.writeLeds(LED_2, OFF); delay(1000);  
}
```

```
c) void loop() {  
    MFS.writeLeds(LED_3, ON); delay(1000);  
    MFS.writeLeds(LED_3, OFF); delay(1000);  
}
```

3.- Encender el LED 2, esperar 1 segundo y apagarlo.

```
a) void loop() {  
    MFS.writeLeds(LED_2, ON); delay(1000);
```

```
MFS.writeLeds(LED_2, OFF); delay(1000);  
}
```

b) void loop() {
 MFS.writeLeds(LED_1, ON); delay(1000);
 MFS.writeLeds(LED_1, OFF); delay(1000);
}

c) void loop() {
 MFS.writeLeds(LED_3, ON); delay(1000);
 MFS.writeLeds(LED_3, OFF); delay(1000);
}

4.- Encender el LED 3, esperar 1 segundo y apagarlo.

a) void loop() {
 MFS.writeLeds(LED_3, ON); delay(1000);
 MFS.writeLeds(LED_3, OFF); delay(1000);
}

b) void loop() {
 MFS.writeLeds(LED_1, ON); delay(1000);
 MFS.writeLeds(LED_1, OFF); delay(1000);
}

c) void loop() {
 MFS.writeLeds(LED_4, ON); delay(1000);
 MFS.writeLeds(LED_4, OFF); delay(1000);
}

5.- Encender el LED 4, esperar 1 segundo y apagarlo.

```
a) void loop() {  
    MFS.writeLeds(LED_4, ON); delay(1000);  
    MFS.writeLeds(LED_4, OFF); delay(1000);  
}
```

```
b) void loop() {  
    MFS.writeLeds(LED_2, ON); delay(1000);  
    MFS.writeLeds(LED_2, OFF); delay(1000);  
}
```

```
c) void loop() {  
    MFS.writeLeds(LED_3, ON); delay(1000);  
    MFS.writeLeds(LED_3, OFF); delay(1000);  
}
```

6.- Encender el LED 1, esperar 1 segundo, apagarlo, esperar 1 segundo
Encender el LED 2, esperar 1 segundo, apagarlo, esperar 1 segundo
e iniciar de nuevo.

```
a) void loop() {  
    MFS.writeLeds(LED_1, ON); delay(1000);  
    MFS.writeLeds(LED_1, OFF); delay(1000);  
    MFS.writeLeds(LED_2, ON); delay(1000);  
    MFS.writeLeds(LED_2, OFF); delay(1000);  
}
```

b) void loop() {

```
MFS.writeLeds(LED_1, ON); delay(1000);  
MFS.writeLeds(LED_2, OFF); delay(1000);  
MFS.writeLeds(LED_1, ON); delay(1000);  
MFS.writeLeds(LED_2, OFF); delay(1000);  
}
```

c) void loop() {

```
MFS.writeLeds(LED_2, ON); delay(1000);  
MFS.writeLeds(LED_1, OFF); delay(1000);  
MFS.writeLeds(LED_1, ON); delay(1000);  
MFS.writeLeds(LED_2, OFF); delay(1000);  
}
```

7.- Encender el LED 2, esperar 1 segundo, apagarlo, esperar 1 segundo
Encender el LED 3, esperar 1 segundo, apagarlo, esperar 1 segundo
e iniciar de nuevo.

a) void loop() {

```
MFS.writeLeds(LED_2, ON); delay(1000);  
MFS.writeLeds(LED_2, OFF); delay(1000);  
MFS.writeLeds(LED_3, ON); delay(1000);  
MFS.writeLeds(LED_3, OFF); delay(1000);  
}
```

b) void loop() {

```
MFS.writeLeds(LED_1, ON); delay(1000);  
MFS.writeLeds(LED_3, OFF); delay(1000);  
MFS.writeLeds(LED_1, ON); delay(1000);
```

```
MFS.writeLeds(LED_3, OFF); delay(1000);  
}
```

```
c) void loop() {  
    MFS.writeLeds(LED_4, ON); delay(1000);  
    MFS.writeLeds(LED_2, OFF); delay(1000);  
    MFS.writeLeds(LED_1, ON); delay(1000);  
    MFS.writeLeds(LED_2, OFF); delay(1000);  
}
```

8.- Encender el LED 3, esperar 1 segundo, apagarlo, esperar 1 segundo
Encender el LED 4, esperar 1 segundo, apagarlo, esperar 1 segundo
e iniciar de nuevo.

```
a) void loop() {  
    MFS.writeLeds(LED_3, ON); delay(1000);  
    MFS.writeLeds(LED_3, OFF); delay(1000);  
    MFS.writeLeds(LED_4, ON); delay(1000);  
    MFS.writeLeds(LED_4, OFF); delay(1000);  
}
```

```
b) void loop() {  
    MFS.writeLeds(LED_1, ON); delay(1000);  
    MFS.writeLeds(LED_3, OFF); delay(1000);  
    MFS.writeLeds(LED_1, ON); delay(1000);  
    MFS.writeLeds(LED_3, OFF); delay(1000);  
}
```

```
c) void loop() {
```

```
MFS.writeLeds(LED_4, ON); delay(1000);  
MFS.writeLeds(LED_3, OFF); delay(1000);  
MFS.writeLeds(LED_3, ON); delay(1000);  
MFS.writeLeds(LED_4, OFF); delay(1000);  
}
```

9.- Instrucción para que los LED parpadeen

a) `MFS.writeLeds();`

b) `MFS.blinkLeds();`

c) `MFS.write();`

10.- Código para que el LED 1 parpadee durante dos segundos

a) `void loop() {`
 `MFS.writeLeds(LED_1, ON);`
 `MFS.blinkLeds(LED_1, ON); delay(2000);`
 `MFS.blinkLeds(LED_1, OFF); delay(2000);`
 `MFS.writeLeds(LED_1, OFF);`
 `}`

b) `void loop() {`
 `MFS.writeLeds(LED_2, ON);`
 `MFS.blinkLeds(LED_2, ON); delay(2000);`
 `MFS.blinkLeds(LED_2, OFF); delay(2000);`
 `MFS.writeLeds(LED_2, OFF);`
 `}`

```
c) void loop() {  
    MFS.writeLeds(LED_1, ON);  
    MFS.blinkLeds(LED_2, ON); delay(2000);  
    MFS.blinkLeds(LED_2, OFF); delay(2000);  
    MFS.writeLeds(LED_1, OFF);  
}
```

11.- Código para que el LED 2 parpadee durante dos segundos

```
a) void loop() {  
    MFS.writeLeds(LED_2, ON);  
    MFS.blinkLeds(LED_2, ON); delay(2000);  
    MFS.blinkLeds(LED_2, OFF); delay(2000);  
    MFS.writeLeds(LED_2, OFF);  
}
```

```
b) void loop() {  
    MFS.writeLeds(LED_3, ON);  
    MFS.blinkLeds(LED_3, ON); delay(2000);  
    MFS.blinkLeds(LED_3, OFF); delay(2000);  
    MFS.writeLeds(LED_3, OFF);  
}
```

```
c) void loop() {  
    MFS.writeLeds(LED_1, ON);  
    MFS.blinkLeds(LED_3, ON); delay(2000);  
    MFS.blinkLeds(LED_3, OFF); delay(2000);  
    MFS.writeLeds(LED_1, OFF);  
}
```

```
}
```

12.- Código para que el LED 3 parpadee durante dos segundos

a) void loop() {

```
    MFS.writeLeds(LED_3, ON);  
    MFS.blinkLeds(LED_3, ON); delay(2000);  
    MFS.blinkLeds(LED_3, OFF); delay(2000);  
    MFS.writeLeds(LED_3, OFF);  
}
```

b) void loop() {

```
    MFS.writeLeds(LED_1, ON);  
    MFS.blinkLeds(LED_1, ON); delay(2000);  
    MFS.blinkLeds(LED_1, OFF); delay(2000);  
    MFS.writeLeds(LED_1, OFF);  
}
```

c) void loop() {

```
    MFS.writeLeds(LED_4, ON);  
    MFS.blinkLeds(LED_4, ON); delay(2000);  
    MFS.blinkLeds(LED_2, OFF); delay(2000);  
    MFS.writeLeds(LED_2, OFF);  
}
```

13.- Código para que el LED 4 parpadee durante dos segundos

a) void loop() {

```
    MFS.writeLeds(LED_4, ON);
```



```
MFS.blinkLeds(LED_4, ON); delay(2000);  
MFS.blinkLeds(LED_4, OFF); delay(2000);  
MFS.writeLeds(LED_4, OFF);  
}
```

b) void loop() {

```
MFS.writeLeds(LED_2, ON);  
MFS.blinkLeds(LED_2, ON); delay(2000);  
MFS.blinkLeds(LED_2, OFF); delay(2000);  
MFS.writeLeds(LED_2, OFF);  
}
```

c) void loop() {

```
MFS.writeLeds(LED_3, ON);  
MFS.blinkLeds(LED_2, ON); delay(2000);  
MFS.blinkLeds(LED_3, OFF); delay(2000);  
MFS.writeLeds(LED_2, OFF);  
}
```

14.- Código para que los 4 LED parpadeen durante dos segundos

a) void loop() {

```
MFS.writeLeds(LED_ALL, ON);  
MFS.blinkLeds(LED_ALL, ON); delay(2000);  
MFS.blinkLeds(LED_ALL, OFF); delay(2000);  
MFS.writeLeds(LED_ALL, OFF);  
}
```

b) void loop() {

```
MFS.writeLeds(ALL_LED, ON);  
MFS.blinkLeds(ALL_LED, ON); delay(2000);  
MFS.blinkLeds(ALL_LED, OFF); delay(2000);  
MFS.writeLeds(ALL_LED, OFF);  
}
```

```
c) void loop() {  
    MFS.writeLeds(LEDALL, ON);  
    MFS.blinkLeds(LEDALL, ON); delay(2000);  
    MFS.blinkLeds(LEDALL, OFF); delay(2000);  
    MFS.writeLeds(LEDALL, OFF);  
}
```

15.- Código para que los 4 LED parpadeen durante dos segundos

```
a) void loop() {  
    MFS.writeLeds(LED_ALL, ON); delay(2000);  
    MFS.writeLeds(LED_ALL, OFF); delay(2000);  
}
```

```
b) void loop() {  
    MFS.writeLeds(LED_1, ON); delay(2000);  
    MFS.writeLeds(LED_4, OFF); delay(2000);  
}
```

```
c) void loop() {  
    MFS.writeLeds(ALL_LED, ON); delay(2000);  
    MFS.writeLeds(ALL_LED, OFF); delay(2000);  
}
```

16.- Código para que haga una secuencia encendido/apagado desde el LED 1 hasta el LED 4

a) void loop() {

```
MFS.writeLeds(LED_1, ON); delay(500);  
MFS.writeLeds(LED_1, OFF); delay(500);  
MFS.writeLeds(LED_2, ON); delay(500);  
MFS.writeLeds(LED_2, OFF); delay(500);  
MFS.writeLeds(LED_3, ON); delay(500);  
MFS.writeLeds(LED_3, OFF); delay(500);  
MFS.writeLeds(LED_4, ON); delay(500);  
MFS.writeLeds(LED_4, OFF); delay(500);  
}
```

b) void loop() {

```
MFS.writeLeds(LED_1, ON); delay(500);  
MFS.writeLeds(LED_2, OFF); delay(500);  
MFS.writeLeds(LED_3, ON); delay(500);  
MFS.writeLeds(LED_4, OFF); delay(500);  
MFS.writeLeds(LED_1, ON); delay(500);  
MFS.writeLeds(LED_2, OFF); delay(500);  
MFS.writeLeds(LED_3, ON); delay(500);  
MFS.writeLeds(LED_4, OFF); delay(500);  
}
```

c) void loop() {

```
MFS.writeLeds(LED_4, ON); delay(500);  
MFS.writeLeds(LED_3, OFF); delay(500);  
MFS.writeLeds(LED_2, ON); delay(500);
```

```
MFS.writeLeds(LED_1, OFF); delay(500);  
MFS.writeLeds(LED_1, ON); delay(500);  
MFS.writeLeds(LED_2, OFF); delay(500);  
MFS.writeLeds(LED_3, ON); delay(500);  
MFS.writeLeds(LED_4, OFF); delay(500);  
}
```

17.- Código para que haga una secuencia encendido/apagado desde el LED 4 hasta el LED 1

a) void loop() {

```
MFS.writeLeds(LED_4, ON); delay(500);  
MFS.writeLeds(LED_4, OFF); delay(500);  
MFS.writeLeds(LED_3, ON); delay(500);  
MFS.writeLeds(LED_3, OFF); delay(500);  
MFS.writeLeds(LED_2, ON); delay(500);  
MFS.writeLeds(LED_2, OFF); delay(500);  
MFS.writeLeds(LED_1, ON); delay(500);  
MFS.writeLeds(LED_1, OFF); delay(500);  
}
```

b) void loop() {

```
MFS.writeLeds(LED_4, ON); delay(500);  
MFS.writeLeds(LED_3, OFF); delay(500);  
MFS.writeLeds(LED_2, ON); delay(500);  
MFS.writeLeds(LED_1, OFF); delay(500);  
MFS.writeLeds(LED_1, ON); delay(500);  
MFS.writeLeds(LED_2, OFF); delay(500);  
MFS.writeLeds(LED_3, ON); delay(500);  
MFS.writeLeds(LED_4, OFF); delay(500);
```

```
}
```

c) void loop() {

```
    MFS.writeLeds(LED_1, ON); delay(500);  
    MFS.writeLeds(LED_1, OFF); delay(500);  
    MFS.writeLeds(LED_2, ON); delay(500);  
    MFS.writeLeds(LED_2, OFF); delay(500);  
    MFS.writeLeds(LED_3, ON); delay(500);  
    MFS.writeLeds(LED_4, OFF); delay(500);  
    MFS.writeLeds(LED_3, ON); delay(500);  
    MFS.writeLeds(LED_4, OFF); delay(500);  
}
```

18.- Código para que decremente una barra generada en los LED

a) void loop() {

```
    MFS.writeLeds(LED_ALL, ON); delay(500);  
    MFS.writeLeds(LED_1, OFF); delay(500);  
    MFS.writeLeds(LED_2, OFF); delay(500);  
    MFS.writeLeds(LED_3, OFF); delay(500);  
    MFS.writeLeds(LED_4, OFF); delay(500);  
}
```

b) void loop() {

```
    MFS.writeLeds(LED_ALL, ON); delay(500);  
    MFS.writeLeds(LED_4, OFF); delay(500);  
    MFS.writeLeds(LED_3, OFF); delay(500);  
    MFS.writeLeds(LED_2, OFF); delay(500);  
    MFS.writeLeds(LED_1, OFF); delay(500);
```

```
}
```

c) void loop() {

```
    MFS.writeLeds(LED_ALL, ON); delay(500);
```

```
    MFS.writeLeds(LED_1, OFF); delay(500);
```

```
    MFS.writeLeds(LED_2, OFF); delay(500);
```

```
    MFS.writeLeds(LED_2, OFF); delay(500);
```

```
    MFS.writeLeds(LED_1, OFF); delay(500);
```

```
}
```

19.- Código para que incremente una barra generada en los LED

a) void loop() {

```
    MFS.writeLeds(LED_ALL, OFF); delay(500);
```

```
    MFS.writeLeds(LED_4, ON); delay(500);
```

```
    MFS.writeLeds(LED_3, ON); delay(500);
```

```
    MFS.writeLeds(LED_2, ON); delay(500);
```

```
    MFS.writeLeds(LED_1, ON); delay(500);
```

```
}
```

a) void loop() {

```
    MFS.writeLeds(LED_ALL, OFF); delay(500);
```

```
    MFS.writeLeds(LED_1, ON); delay(500);
```

```
    MFS.writeLeds(LED_2, ON); delay(500);
```

```
    MFS.writeLeds(LED_3, ON); delay(500);
```

```
    MFS.writeLeds(LED_4, ON); delay(500);
```

```
}
```

a) void loop() {

```

MFS.writeLeds(LED_ALL, OFF); delay(500);

MFS.writeLeds(LED_4, ON); delay(500);

MFS.writeLeds(LED_2, ON); delay(500);

MFS.writeLeds(LED_2, ON); delay(500);

MFS.writeLeds(LED_4, ON); delay(500);

}

```

20.- Código que muestre una barra que incremente en los LED de acuerdo al valor actual del potenciómetro

```

a) void loop() {

    if( analogRead(POT_PIN) >= 0 && analogRead(POT_PIN) <= 255 ){

        MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_4, ON);

    }

    else if( analogRead(POT_PIN) >= 256 && analogRead(POT_PIN) <= 511 ){

        MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_4, ON); MFS.writeLeds(LED_3, ON);

    }

    else if( analogRead(POT_PIN) >= 512 && analogRead(POT_PIN) <= 768 ){

        MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_4, ON); MFS.writeLeds(LED_3, ON);
MFS.writeLeds(LED_2, ON);

    }

    else{

        MFS.writeLeds(LED_ALL, ON)

    }

}

```

```

b) void loop() {

    if( analogRead(POT_PIN) >= 0 && analogRead(POT_PIN) <= 255 ){

        MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, ON);

    }

}

```

```

else if( analogRead(POT_PIN) >= 256 && analogRead(POT_PIN) <= 511 ){
    MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, ON); MFS.writeLeds(LED_3, ON);
}
else if( analogRead(POT_PIN) >= 512 && analogRead(POT_PIN) <= 768 ){
    MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, ON); MFS.writeLeds(LED_3, ON);
MFS.writeLeds(LED_2, ON);
}
else{
    MFS.writeLeds(LED_ALL, ON)
}
}

```

```

c) void loop() {
    if( analogRead(POT_PIN) >= 0 && analogRead(POT_PIN) <= 255 ){
        MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, OFF);
    }
    else if( analogRead(POT_PIN) >= 256 && analogRead(POT_PIN) <= 511 ){
        MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, OFF); MFS.writeLeds(LED_3, OFF);
    }
    else if( analogRead(POT_PIN) >= 512 && analogRead(POT_PIN) <= 768 ){
        MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, OFF); MFS.writeLeds(LED_3, OFF);
MFS.writeLeds(LED_2, OFF);
    }
    else{
        MFS.writeLeds(LED_ALL, OFF)
    }
}
}

```

21.- Instrucción para escribir algún texto en los display

a) MFS.writeLeds();

b) Serial.write();

c) MFS.write();

22.- Código que muestre cuenta en aumento en los display

```
a) void loop() {  
    if(Contador <= 9999){  
        MFS.write( Contador );  
        Contador++;  
    }  
    delay(50);  
}
```

```
b) void loop() {  
    if(Contador <= 0){  
        MFS.write( Contador );  
        Contador--  
    }  
    delay(50);  
}
```

```
c) void loop() {  
    if(Contador = 9999){  
        MFS.write( Contador );  
        Contador++;  
    }  
}
```

```
    delay(50);  
}
```

23.- Código que muestre cuenta en decremento en los display

a) void loop() {
 if(Contador <= 0){
 MFS.write(Contador);
 Contador--;
 }
 delay(50);
}

b) void loop() {
 if(Contador <= 9999){
 MFS.write(Contador);
 Contador--;
 }
 delay(50);
}

c) void loop() {
 if(Contador = 0){
 MFS.write(Contador);
 Contador--;
 }
 delay(50);
}

24.- Código para que muestre el valor actual del potenciómetro en los display

```
a) void loop() {  
    MFS.write(analogRead(POTENTIOMETER));  
}
```

```
b) void loop() {  
    MFS.write(analogRead(POT_PIN));  
}
```

```
c) void loop() {  
    MFS.write(POT);  
}
```

25.- Código para que muestre el valor actual del potenciómetro en los display y además muestre una barra que incremente en los LED de acuerdo al valor actual del potenciómetro.

```
a) void loop() {  
    if( analogRead(POT_PIN) >= 0 && analogRead(POT_PIN) <= 255 ){  
        MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_4, ON);  
    }  
    else if( analogRead(POT_PIN) >= 256 && analogRead(POT_PIN) <= 511 ){  
        MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_4, ON); MFS.writeLeds(LED_3, ON);  
    }  
    else if( analogRead(POT_PIN) >= 512 && analogRead(POT_PIN) <= 768 ){  
        MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_4, ON); MFS.writeLeds(LED_3, ON);  
        MFS.writeLeds(LED_2, ON);  
    }  
    else{
```

```

    MFS.writeLeds(LED_ALL, ON)
}
MFS.write(analogRead(POT_PIN));
}

```

```

b) void loop() {
    if( analogRead(POT_PIN) >= 0 && analogRead(POT_PIN) <= 255 ){
        MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, ON);
    }
    else if( analogRead(POT_PIN) >= 256 && analogRead(POT_PIN) <= 511 ){
        MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, ON); MFS.writeLeds(LED_3, ON);
    }
    else if( analogRead(POT_PIN) >= 512 && analogRead(POT_PIN) <= 768 ){
        MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, ON); MFS.writeLeds(LED_3, ON);
MFS.writeLeds(LED_2, ON);
    }
    else{
        MFS.writeLeds(LED_ALL, ON)
    }
    MFS.write(analogRead(POTENTIOMETER));
}

```

```

c) void loop() {
    if( analogRead(POT_PIN) >= 0 && analogRead(POT_PIN) <= 255 ){
        MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, OFF);
    }
    else if( analogRead(POT_PIN) >= 256 && analogRead(POT_PIN) <= 511 ){
        MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, OFF); MFS.writeLeds(LED_3, OFF);
    }
}

```

```

else if( analogRead(POT_PIN) >= 512 && analogRead(POT_PIN) <= 768 ){
    MFS.writeLeds(LED_ALL, ON); MFS.writeLeds(LED_4, OFF); MFS.writeLeds(LED_3, OFF);
MFS.writeLeds(LED_2, OFF);
}
else{
    MFS.writeLeds(LED_ALL, OFF)
}
MFS.write(POT);
}

```

26.- Código para que muestre una barra que incremente en los display de acuerdo al valor actual del potenciómetro.

```

a) void loop() {
    if( analogRead(POT_PIN) >= 0 && analogRead(POT_PIN) <= 255 ){
        MFS.write("0 ");
    }
    else if( analogRead(POT_PIN) >= 256 && analogRead(POT_PIN) <= 511 ){
        MFS.write("00 ");
    }
    else if( analogRead(POT_PIN) >= 512 && analogRead(POT_PIN) <= 768 ){
        MFS.write("000 ");
    }
    else{
        MFS.write("0000");
    }
    MFS.write(analogRead(POT_PIN));
}

```

```

b) void loop() {

```

```

if( analogRead(POT_PIN) >= 0 && analogRead(POT_PIN) <= 255 ){
    MFS.write("0");
}
else if( analogRead(POT_PIN) >= 256 && analogRead(POT_PIN) <= 511 ){
    MFS.write("00");
}
else if( analogRead(POT_PIN) >= 512 && analogRead(POT_PIN) <= 768 ){
    MFS.write("000");
}
else{
    MFS.write("0000");
}
MFS.write(POT);
}

```

```

c) void loop() {
    if( analogRead(POT_PIN) >= 0 && analogRead(POT_PIN) <= 255 ){
        MFS.write("0 0");
    }
    else if( analogRead(POT_PIN) >= 256 && analogRead(POT_PIN) <= 511 ){
        MFS.write("0 0 ");
    }
    else if( analogRead(POT_PIN) >= 512 && analogRead(POT_PIN) <= 768 ){
        MFS.write("00 0");
    }
    else{
        MFS.write("0 ");
    }
    MFS.write(analogRead(POTENCIOMETRO));
}

```

```
}
```

27.- Código que hace posible detectar una pulsación corta en el botón 1

a) void loop() {

```
    if( buttonAction == BUTTON_PRESSED_IND && buttonNumber == 1){
```

```
        // Código
```

```
    }
```

```
}
```

b) void loop() {

```
    if( buttonAction == BUTTON_SHORT_PRESS_IND && buttonNumber == 1){
```

```
        // Código
```

```
    }
```

```
}
```

c) void loop() {

```
    if( buttonAction == BUTTON_PRESS_IND && buttonNumber == 1){
```

```
        // Código
```

```
    }
```

```
}
```

28.- Código que hace posible detectar una pulsación corta en el botón 2

a) void loop() {

```
    if( buttonAction == BUTTON_PRESSED_IND && buttonNumber == 2){
```

```
        // Código
```

```
    }
```

```
}
```

b) void loop() {

```
    if( buttonAction == BUTTON_SHORT_PRESS_IND && buttonNumber == 2){
```

```
        // Código
```

```
    }
```

```
}
```

c) void loop() {

```
    if( buttonAction == BUTTON_PRESS_IND && buttonNumber == 2){
```

```
        // Código
```

```
    }
```

```
}
```

29.- Código que hace posible detectar una pulsación corta en el botón 3

a) void loop() {

```
    if( buttonAction == BUTTON_PRESSED_IND && buttonNumber == 3){
```

```
        // Código
```

```
    }
```

```
}
```

b) void loop() {

```
    if( buttonAction == BUTTON_SHORT_PRESS_IND && buttonNumber == 3){
```

```
        // Código
```

```
    }
```

```
}
```

c) void loop() {

```
    if( buttonAction == BUTTON_PRESS_IND && buttonNumber == 3){
```



```
    // Código
  }
}
```

30.- Código que hace posible detectar una pulsación larga en el botón 1

```
a) void loop() {
    if( buttonAction == BUTTON_LONG_PRESSED_IND && buttonNumber == 1){
        // Código
    }
}
```

```
b) void loop() {
    if( buttonAction == BUTTON_LONG_PRESSED && buttonNumber == 1){
        // Código
    }
}
```

```
c) void loop() {
    if( buttonAction == BUTTON_LONG_PRESS_IND && buttonNumber == 1){
        // Código
    }
}
```

31.- Código que hace posible detectar una pulsación larga en el botón 2

```
a) void loop() {
    if( buttonAction == BUTTON_LONG_PRESSED_IND && buttonNumber == 2){
        // Código
    }
}
```

```
}  
}
```

```
b) void loop() {  
    if( buttonAction == BUTTON_LONG_PRESSED && buttonNumber == 2){  
        // Código  
    }  
}
```

```
c) void loop() {  
    if( buttonAction == BUTTON_LONG_PRESS_IND && buttonNumber == 2){  
        // Código  
    }  
}
```

32.- Código que hace posible detectar una pulsación larga en el botón 3

```
a) void loop() {  
    if( buttonAction == BUTTON_LONG_PRESSED_IND && buttonNumber == 3){  
        // Código  
    }  
}
```

```
b) void loop() {  
    if( buttonAction == BUTTON_LONG_PRESSED && buttonNumber == 3){  
        // Código  
    }  
}
```

```
c) void loop() {  
    if( buttonAction == BUTTON_LONG_PRESS_IND && buttonNumber == 3){  
        // Código  
    }  
}
```

33.- Código que hace posible detectar una liberación corta en el botón 1

```
a) void loop() {  
    if( buttonAction == BUTTON_SHORT_RELEASE_IND && buttonNumber == 1){  
        // Código  
    }  
}
```

```
b) void loop() {  
    if( buttonAction == BUTTON_SHORT_RELEASE && buttonNumber == 1){  
        // Código  
    }  
}
```

```
c) void loop() {  
    if( buttonAction == BUTTON_RELEASE_IND && buttonNumber == 1){  
        // Código  
    }  
}
```

34.- Código que hace posible detectar una liberación corta en el botón 2

```
a) void loop() {
```

```
if( buttonAction == BUTTON_SHORT_RELEASE_IND && buttonNumber == 2){  
    // Código  
}  
}
```

b) void loop() {
 if(buttonAction == BUTTON_SHORT_RELEASE && buttonNumber == 2){
 // Código
 }
}

c) void loop() {
 if(buttonAction == BUTTON_RELEASE_IND && buttonNumber == 2){
 // Código
 }
}

35.- Código que hace posible detectar una liberación corta en el botón 3

a) void loop() {
 if(buttonAction == BUTTON_SHORT_RELEASE_IND && buttonNumber == 3){
 // Código
 }
}

b) void loop() {
 if(buttonAction == BUTTON_SHORT_RELEASE && buttonNumber == 3){
 // Código
 }
}

```
}
```

```
c) void loop() {
```

```
    if( buttonAction == BUTTON_RELEASE_IND && buttonNumber == 3){
```

```
        // Código
```

```
    }
```

```
}
```

36.- Código que hace posible detectar una liberación larga en el botón 1

```
a) void loop() {
```

```
    if( buttonAction == BUTTON_LONG_RELEASE_IND && buttonNumber == 1){
```

```
        // Código
```

```
    }
```

```
}
```

```
b) void loop() {
```

```
    if( buttonAction == BUTTON_LONG_RELEASE && buttonNumber == 1){
```

```
        // Código
```

```
    }
```

```
}
```

```
c) void loop() {
```

```
    if( buttonAction == BUTTON_LONG_IND && buttonNumber == 1){
```

```
        // Código
```

```
    }
```

```
}
```

37.- Código que hace posible detectar una liberación larga en el botón 2

a) void loop() {

```
    if( buttonAction == BUTTON_LONG_RELEASE_IND && buttonNumber == 2){
```

```
        // Código
```

```
    }
```

```
}
```

b) void loop() {

```
    if( buttonAction == BUTTON_LONG_RELEASE && buttonNumber == 2){
```

```
        // Código
```

```
    }
```

```
}
```

c) void loop() {

```
    if( buttonAction == BUTTON_LONG_IND && buttonNumber == 2){
```

```
        // Código
```

```
    }
```

```
}
```

38.- Código que hace posible detectar una liberación larga en el botón 3

a) void loop() {

```
    if( buttonAction == BUTTON_LONG_RELEASE_IND && buttonNumber == 3){
```

```
        // Código
```

```
    }
```

```
}
```

b) void loop() {

```
    if( buttonAction == BUTTON_LONG_RELEASE && buttonNumber == 3){
```

```

    // Código
}
}

```

```

c) void loop() {
    if( buttonAction == BUTTON_LONG_IND && buttonNumber == 3){
        // Código
    }
}

```

39.- Código que cambia de posición (1 a 4) un dígito en los display de acuerdo a un objeto de tipo Contador

```

a) Contador PosicionDisplay(1, 1, 4);

void loop() {
    ++PosicionDisplay;
    switch( PosicionDisplay.GetN() ){
        case 1: MFS.write("0 "); break;
        case 2: MFS.write(" 0 "); break;
        case 3: MFS.write(" 0 "); break;
        case 4: MFS.write(" 0"); break;
    }
    delay(1000);
}

```

```

b) Contador PosicionDisplay(1, 1, 4);

void loop() {
    ++PosicionDisplay;
    switch( PosicionDisplay.GetN() ){

```

```

    case 1: MFS.write("0 "); break;
    case 2: MFS.write(" "); break;
    case 3: MFS.write("0000"); break;
    case 4: MFS.write(" 0"); break;
}
delay(1000);
}

```

c) Contador PosicionDisplay(1, 1, 2);

```

void loop() {
    ++PosicionDisplay;
    switch( PosicionDisplay.GetN() ){
        case 1: MFS.write("0 "); break;
        case 2: MFS.write(" 0 "); break;
        case 3: MFS.write(" 0 0 "); break;
        case 4: MFS.write(" 0 0 "); break;
    }
    delay(1000);
}

```

40.- Código que cambia de posición (1 a 4) un LED de acuerdo a un objeto de tipo Contador

a) Contador PosicionLED(1, 1, 4);

```

void loop() {
    ++PosicionLED;
    switch( PosicionLED.GetN() ){
        case 1: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_1, ON); break;
        case 2: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_2, ON); break;
        case 3: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_3, ON); break;
    }
}

```



```

        case 4: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_4, ON); break;
    }
    delay(1000);
}

```

a) Contador PosicionLED(0, 0, 0);

```

void loop() {
    ++PosicionLED;
    switch( PosicionLED.GetN() ){
        case 1: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_1, ON); break;
        case 2: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_2, ON); break;
        case 3: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_3, ON); break;
        case 4: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_4, ON); break;
    }
    delay(1000);
}

```

a) Contador PosicionLED(1, 1, 1);

```

void loop() {
    ++PosicionLED;
    switch( PosicionLED.GetN() ){
        case 1: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_1, ON); break;
        case 2: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_2, ON); break;
        case 3: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_3, ON); break;
        case 4: MFS.writeLeds(LED_ALL, OFF); MFS.writeLeds(LED_4, ON); break;
    }
    delay(1000);
}

```

41.- Instrucción que hace funcionar el buzzer

a) `MFS.writeLeds();`

b) `MFS.beep();`

c) `MFS.write();`