



**DEPARTAMENTO DE CIENCIAS EXACTAS-DECE**

**FUNDAMENTOS DE PROGRAMACIÓN**

**PROYECTO “BARRERA CON CONTRASEÑA  
ARDUINO”**

**DOCENTE: ING. JORGE RAURA**

**INTEGRANTES:**

**JEAN MERA**

**SARAHÍ CALAHORRANO**

**MELANY LARA**

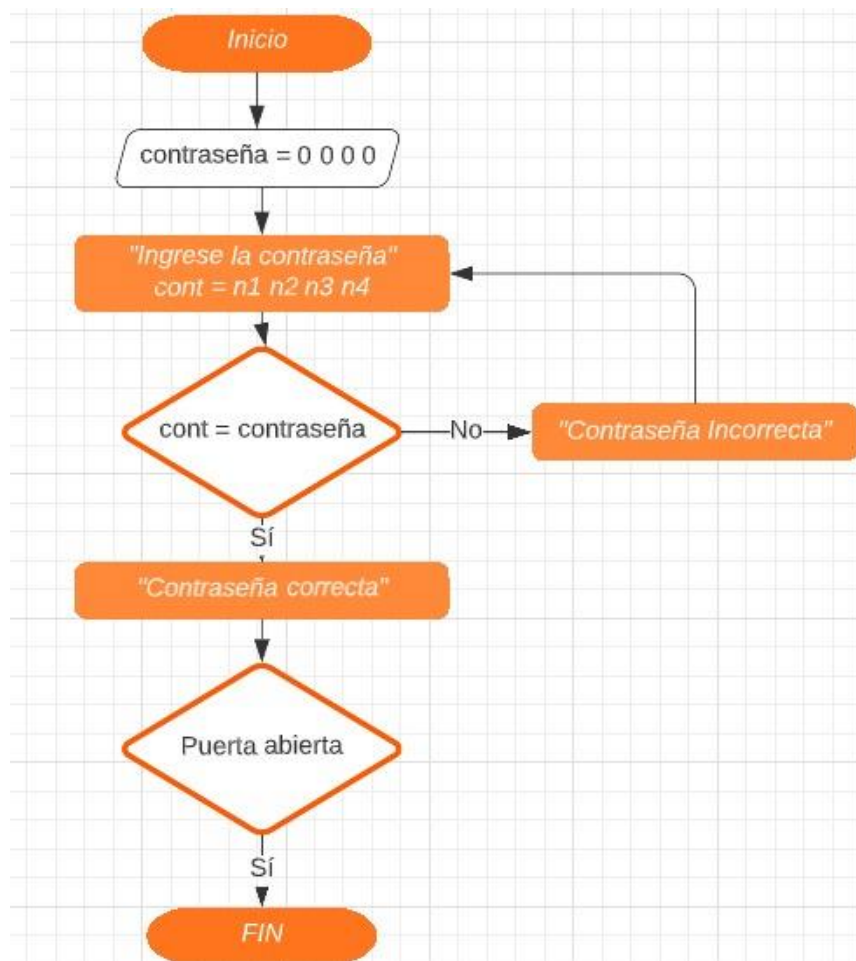
**JUAN SIMBAÑA**

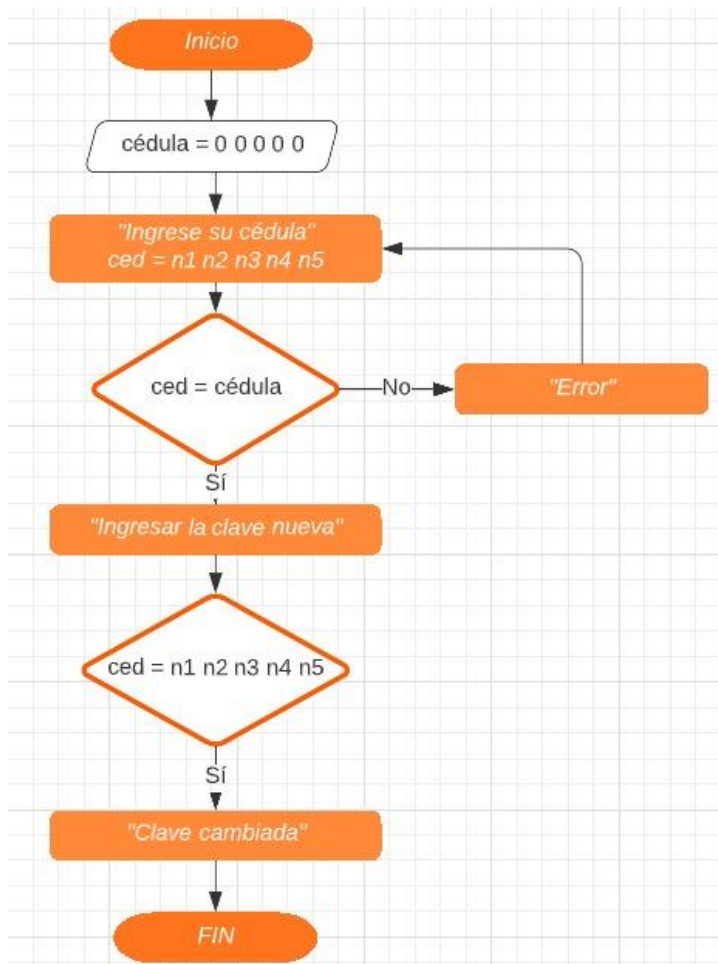
**DANIEL VILLARREAL**

## OBJETIVO:

El objetivo principal de este parcial es formar una conexión y comunicación entre usuario y cerradura por medio de un código programado para que la persona pueda ingresar con clave pero también la pueda cambiar.

## FLUJOGRAMA:



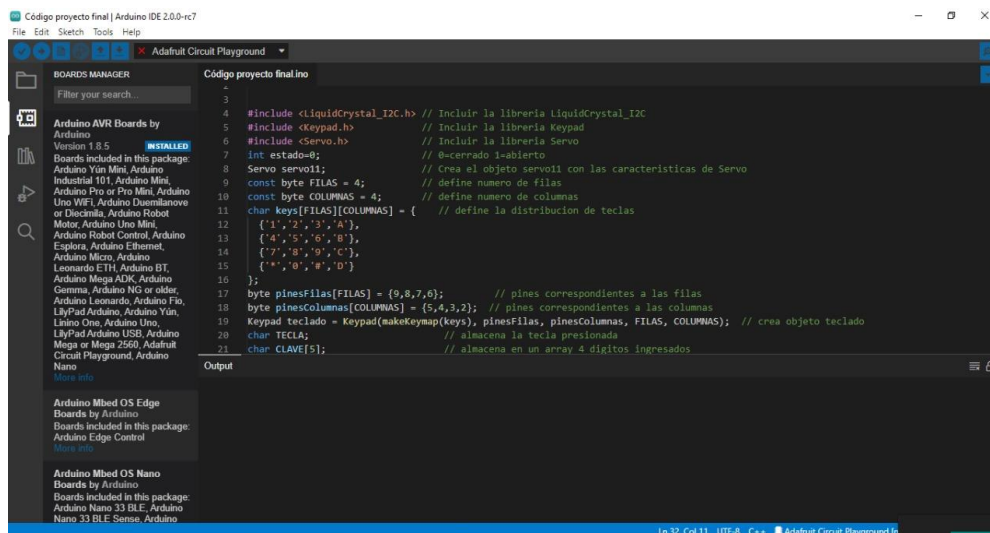


### ESQUEMA DEL ALGORITMO:

1. Incluir librerías LiquidCrystal\_I2C, Keypad, Servo
2. Definir variables 0=cerrado 1=abierto
3. Crear el objeto servo11 con las características de Servo
4. Definir el numero de filas y de columnas del pad numérico a utilizar
5. Seguir definiendo variables
6. Almacenar teclas guardando en la variable char
7. Fijar una contraseña inicial, que viene por default
8. Comandos para llamar al pantalla lcd, dependiendo del fabricante del I2C el código 0x27 puede cambiar
9. Función inicial (void setup) para establecer ciertas instrucciones para iniciar con el programa
10. Inicia con el programa (void loop) define el ingreso de 4 teclas que digite el usuario, si la contraseña coincide con la inicial, se abre la puerta; Si no coincide la pantalla lcd imprime “Clave Incorrecta”
11. Código (void abierto) si se coloca bien la contraseña, este paso imprime en la pantalla lcd “Abierto”, por lo tanto el servo gira 30 grados para abrir la puerta; Para cerrar se vuelve a digitar la clave, imprimiendo en la pantalla lcd “Cerrado” y el servo girara 150 grados.

12. Función nueva clave (void nueva clave), tocara definir nuevamente variables para 5 espacios, en los cuales constara los 5 últimos dígitos de la cédula, por lo tanto en la pantalla lcd se imprime “Ingrese su cedula”. Si los números ingresado son los mismos a los que se encuentra guardados en el programa, el programa va a limpiar y se abre la opción de ingresar la nueva clave, la cual se va a almacenar, y la pantalla lcd imprimirá “Clave cambiada”. En el caso de no haber digitado bien los 5 dígitos de la cedula, se reflejará en la pantalla lcd “Error”, y el usuario deberá volver a ingresar los 5 números.
13. Función (void limpia), limpiará todo lo que exista en la pantalla y volverá a imprimir “Clave”

## CÓDIGO Fuente:



```
1 // Código proyecto final | Arduino IDE 2.0.0-rc7
2
3 #include <LiquidCrystal_I2C.h> // Incluir la libreria LiquidCrystal_I2C
4 #include <Keypad.h> // Incluir la libreria Keypad
5 #include <Servo.h> // Incluir la libreria Servo
6
7 int estado=0; // 0=cerrado 1=abierto
8 Servo servo11; // Crea el objeto servo11 con las características de Servo
9 const byte FILAS = 4; // define numero de filas
10 const byte COLUMNAS = 4; // define numero de columnas
11 char keys[FILAS][COLUMNAS] = { // define la distribucion de teclas
12   {'1','2','3','A'},
13   {'4','5','6','B'},
14   {'7','8','9','C'},
15   {'*','0','#','D'}
16 };
17 byte pinesFilas[FILAS] = {9,8,7,6}; // pines correspondientes a las filas
18 byte pinesColumnas[COLUMNAS] = {5,4,3,2}; // pines correspondientes a las columnas
19 Keypad teclado = Keypad(makeKeymap(keys), pinesFilas, pinesColumnas, FILAS, COLUMNAS); // crea objeto teclado
20 char TECLA; // almacena la tecla presionada
21 char CLAVE[5]; // almacena en un array 4 digitos ingresados
22
23 Output
```

// Apertura con Clave y opcion de cambio de Clave

#include <LiquidCrystal\_I2C.h> // Incluir la libreria LiquidCrystal\_I2C

#include <Keypad.h> // Incluir la libreria Keypad

#include <Servo.h> // Incluir la libreria Servo

int estado=0; // 0=cerrado 1=abierto

Servo servo11; // Crea el objeto servo11 con las características de Servo

const byte FILAS = 4; // define numero de filas

const byte COLUMNAS = 4; // define numero de columnas

char keys[FILAS][COLUMNAS] = { // define la distribucion de teclas

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{' ','0','#','D'}

};

```

byte pinesFilas[FILAS] = {9,8,7,6};    // pines correspondientes a las filas

byte pinesColumnas[COLUMNAS] = {5,4,3,2}; // pines correspondientes a las columnas

Keypad teclado = Keypad(makeKeymap(keys), pinesFilas, pinesColumnas, FILAS,
COLUMNAS); // crea objeto teclado

char TECLA;           // almacena la tecla presionada

char CLAVE[5];         // almacena en un array 4 digitos ingresados

char CLAVE_MAESTRA[5] = "0000"; // almacena en un array la contraseña inicial

byte INDICE = 0;       // indice del array

LiquidCrystal_I2C lcd(0x27,16,2); // dependiendo del fabricante del I2C el codigo 0x27
cambiar a

                                // 0x3F , 0x20 , 0x38 ,

void setup()
{
    lcd.init();           // inicializa el LCD

    lcd.backlight();

    servo11.attach(11,660,1400); // Asocia el servo1 al pin 11, define el min y max del ancho
del pulso

    servo11.write(150);    // Gira el servo a 150 grados Cierra la puerta

    limpia();
}

void loop(){
    TECLA = teclado.getKey(); // obtiene tecla presionada y asigna a variable

    if (TECLA)               // comprueba que se haya presionado una tecla
    {
        CLAVE[INDICE] = TECLA; // almacena en array la tecla presionada

        INDICE++;             // incrementa indice en uno

        lcd.print(TECLA);     // envia al LCD la tecla presionada
    }

    if(INDICE == 4)          // si ya se almacenaron los 4 digitos
    {
        if(!strcmp(CLAVE, CLAVE_MAESTRA)) // compara clave ingresada con clave maestra

            abierto();

        else

```

```

        error();

    INDICE = 0;
}

if(estado==1 && (analogRead(A3)==0))    // si esta abierta y se pulsa boton de Nueva Clave
    nueva_clave();
}

////////// Error //////////

void error(){
    lcd.setCursor(0,1);
    lcd.print("ERROR DE CLAVE  ");
    limpia();
}

//////////abierto o cerrado //////////

void abierto(){
    if(estado==0){
        estado=1;
        lcd.setCursor(0,1);
        lcd.print("Abierta      "); // imprime en el LCD que esta abierta
        servo11.write(30);          // Gira el servo a 30 grados abre la puerta
    }
    else{
        estado=0;
        lcd.setCursor(0,1);
        lcd.print("Cerrada      "); // imprime en el LCD que esta cerrada
        servo11.write(150);         // Gira el servo a 150 grados cierra la puerta
    }
    limpia();
}

//////////Nueva_Clave //////////

void nueva_clave(){
    const char num[5]="81942";

```

```

char codigo[5];

lcd.setCursor(0,0);

lcd.print("INGRESE SU CEDULA      ");

lcd.setCursor(1,0);

INDICE=0;

codigo[5]=teclado.getKey();

if
(codigo[0]==num[0]&&codigo[1]==num[1]&&codigo[2]==num[2]&&codigo[3]==num[3]&&
codigo[4]==num[4]){ //condición, para cambiar contraseña primero se pregunta los ultimos 5
dígitos de la cedula

    limpia();

    lcd.print("NUEVA CLAVE:      ");

    while (INDICE<=3) {

        TECLA = teclado.getKey(); // obtiene tecla presionada y asigna a variable TECLA

        if (TECLA)                // comprueba que se haya presionado una tecla

        {

            CLAVE_MAESTRA[INDICE] = TECLA; // almacena en array la tecla presionada

            CLAVE[INDICE] = TECLA;

            INDICE++;                // incrementa indice en uno

            lcd.print(TECLA);        // envia a monitor serial la tecla presionada

        }

    }

    estado=0;

    lcd.setCursor(0,1);

    lcd.print("CLAVE CAMBIADA");

    delay(2000);

    limpia();

}

else {

    lcd.print("ERROR");

    limpia();

}

}

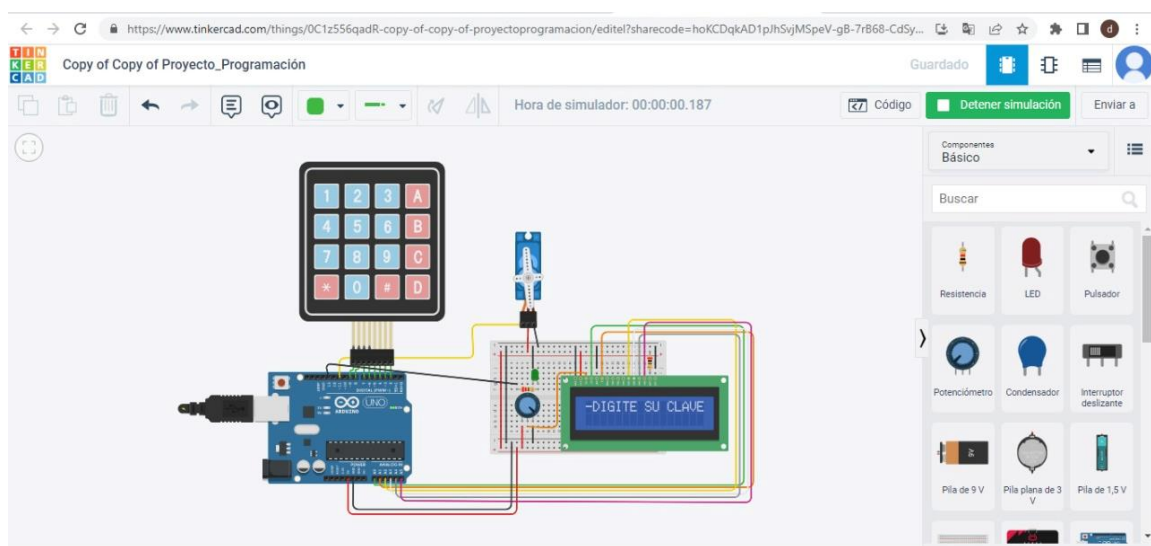
```

```

//////////////////// limpia //////////////////////
void limpia(){
  lcd.setCursor(0,0);
  lcd.print("CLAVE :      ");
  lcd.setCursor(9,0);
}

```

## SIMULACIÓN EN TINKERCAD:



## CONCLUSIONES:

- El diseño electrónico cuenta con una interacción dinámica y familiar al utilizarse ya que su método es igual que un celular al ingresar la clave y poder cambiarla.
- Al ser un diseño totalmente electrónico el consumo de energía es muy bajo, no afectando drásticamente servicios energéticos.

## BIBLIOGRAFÍA:

AMarcos. (2018). La importancia de la Domótica en la vivienda. Obtenido de <https://www.tecnocosas.es/la-importancia-de-la-domotica-en-la-vivienda/> Arduino.cl. (s.f).

¿QUÉ ES ARDUINO? Obtenido de <https://arduino.cl/que-es-arduino> / Artero, O. T. (2014).

Arduino Curso practico de formacion. Del valle: Alfaomega. Bntz, E. (13 de 06 de 2015). La robotica educativa. Obtenido de <http://robotc.blogspot.com/2015/06/morfologia-de-un-robot.html>

Herramientas, L. d. (2018). LAS 8 MEJORES CERRADURAS INTELIGENTES DE 2018. Obtenido de <https://herramientas10.top/mejores-cerraduras-inteligentes/>