



UNIVERSIDAD MAYOR DE SAN SIMON
FACULTAD DE CIENCIAS Y TECNOLOGIA
INGENIERIA ELECTRONICA



TALLER DE PROGRAMACIÓN EN BAJO NIVEL

ESTUDIANTES:

ENCALADA CAMARGO JOSE CRISTIAN

MONTAÑO TORO LAURA HELEN

DOCENTE:

ING. ABRAHAM MAMANI VELIZ

FECHA:

18/02/2017

CANDADO ELECTRÓNICO

DESCRIPCIÓN DEL PROYECTO

El presente proyecto esta destinado a la seguridad, mediante la verificación del usuario pidiéndole su nombre de usuario y su contraseña, posteriormente ingresa a la ventana principal que permite abrir la puerta al ingresar la clave correcta que es “11111” – presionando los botones de ON OFF de la ventana -, esto se verifica con el encendido de un led verde, si la clave ingresada es correcta; caso contrario, el led permanece apagado.

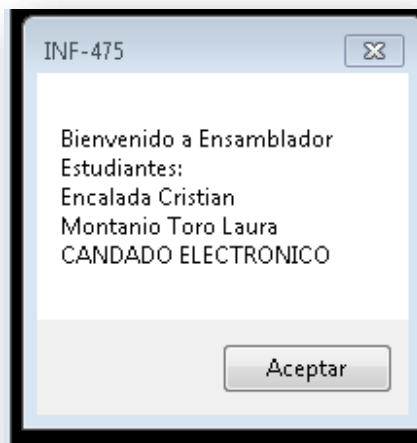
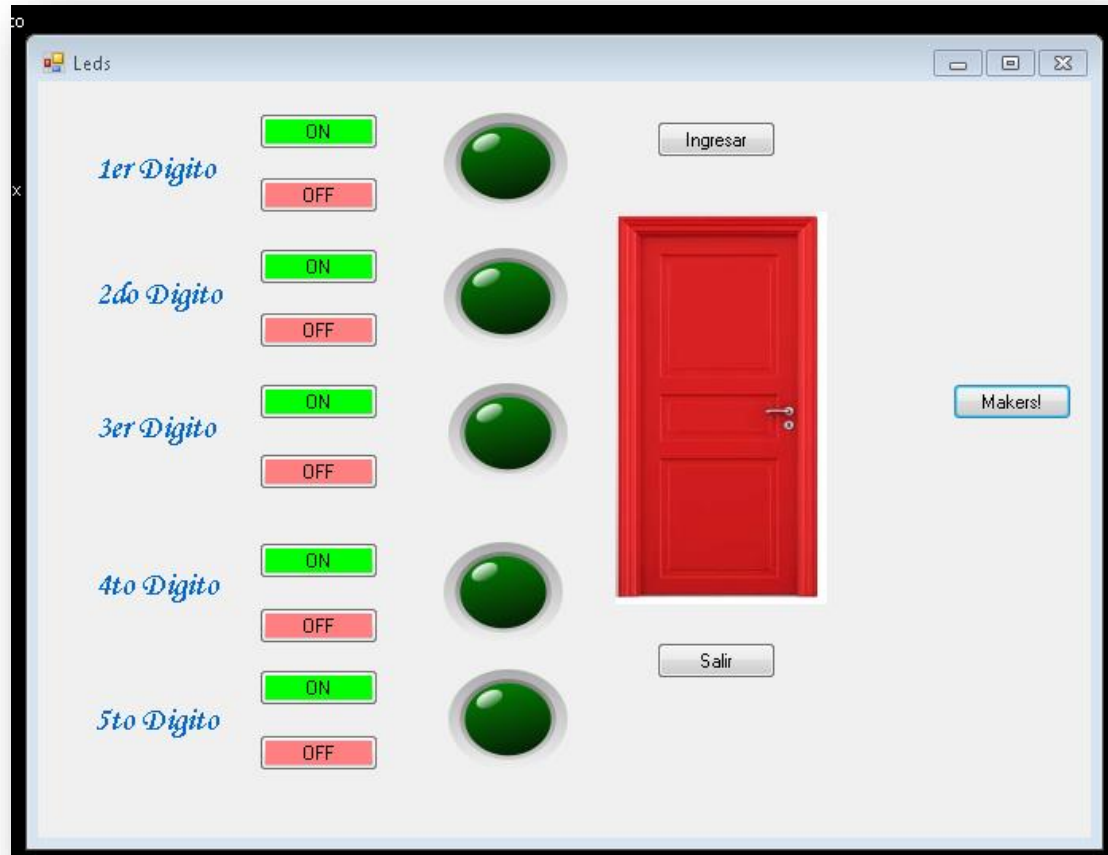
OBJETIVOS

- Implementar el candado electrónico enlazando Visual Studio 2013, Arduino y librerías dinámicas (dll) en ensamblador.

DESARROLLO DEL PROYECTO

PROGRAMA PRINCIPAL

1.1. 1er Requerimiento: Mensaje Makers(assembler)



Código Assembler:

1.1.dll1.def

LIBRARY midll
EXPORTS holaMundo

1.2. dll11.asm

.386
.model flat,stdcall
include \masm32\include\windows.inc
include \masm32\include\user32.inc

includelib \masm32\lib\user32.lib

.data
Text db "Bienvenido a Ensamblador",10,13,"Estudiantes:",10,13,"Encalada
Cristian",10,13,"Montanio Toro Laura",10,13,"CANDADO
ELECTRONICO",10,13,0
Titulo db "INF-475",0

.code
entrada proc hInstance:HINSTANCE,entrda:dword,entrda2:dword
 mov eax,TRUE
 ret
entrada endp
holaMundo proc
 invoke MessageBox,NULL,addr Text,addr Titulo,MB_OK
 ret
holaMundo endp
end entrada

1.2.2do Requerimiento: Ventana Login(assembler)

Código Assembler:

2.1.dll2.def

LIBRARY comp
EXPORTS mensaje
EXPORTS comparar
EXPORTS tamaño

2.2.dll2.asm

.386

.model flat,stdcall

option casemap:none

include \masm32\include\windows.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

tamano proto:dword

comparar proto :dword,:dword,:dword

.code

entry proc hInstance:dword,var1:dword,var2:dword

mov eax,1

ret

entry endp

mensaje proc text:dword,pass:dword

invoke tamano,text

mov ecx,edx

invoke tamano,pass

cmp ecx,edx

jne error

invoke comparar,text,pass,ecx

cmp eax,1

jne error

mov eax,1

ret

error:

xor eax,eax

ret

mensaje endp;devuelve un 0 si hay error --1 si es correcto

comparar proc uses esi edi ecx texto:dword,password:dword,tam:dword

mov esi,texto

mov edi,password

mov ecx,tam

cld

repz cmpsb

jcxz iguales

```

        xor eax,eax
        ret
iguales:
        mov eax,1
        ret
comparar endp

tamanio proc uses ebx eax ecx cadena:dword
        xor edx,edx
        mov ebx,cadena
ciclo:
        mov al,[ebx]
        cmp al,0
        je salir
        inc edx
        inc ebx
        jmp ciclo
salir:
        ret
tamanio endp
end entry

```

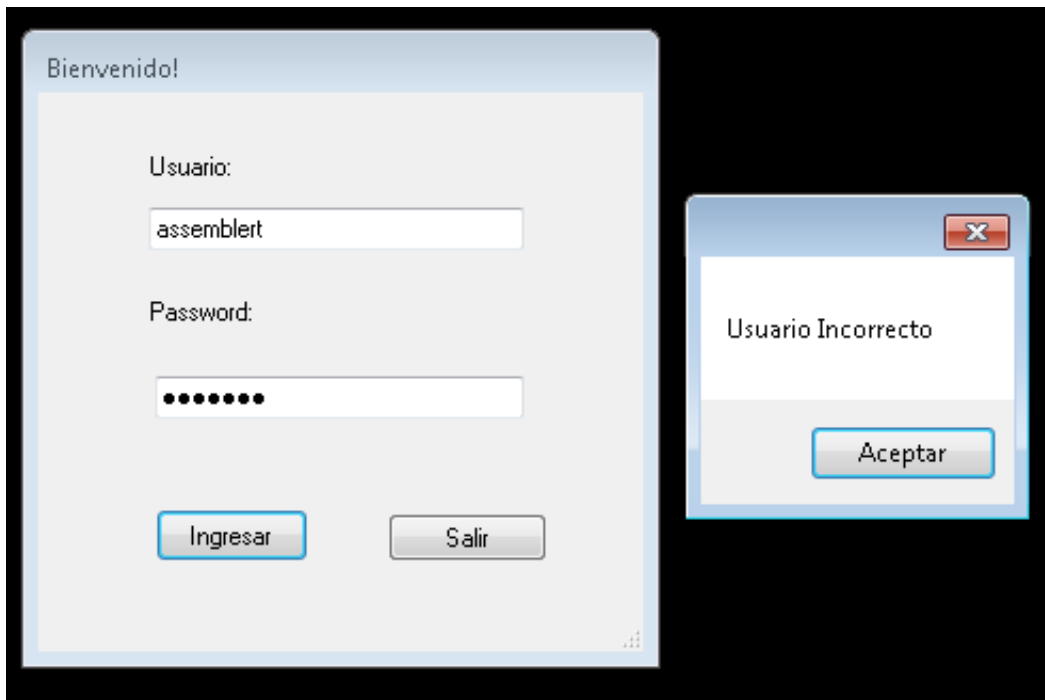
PRUEBAS:

Ventana de Login (Inicio)

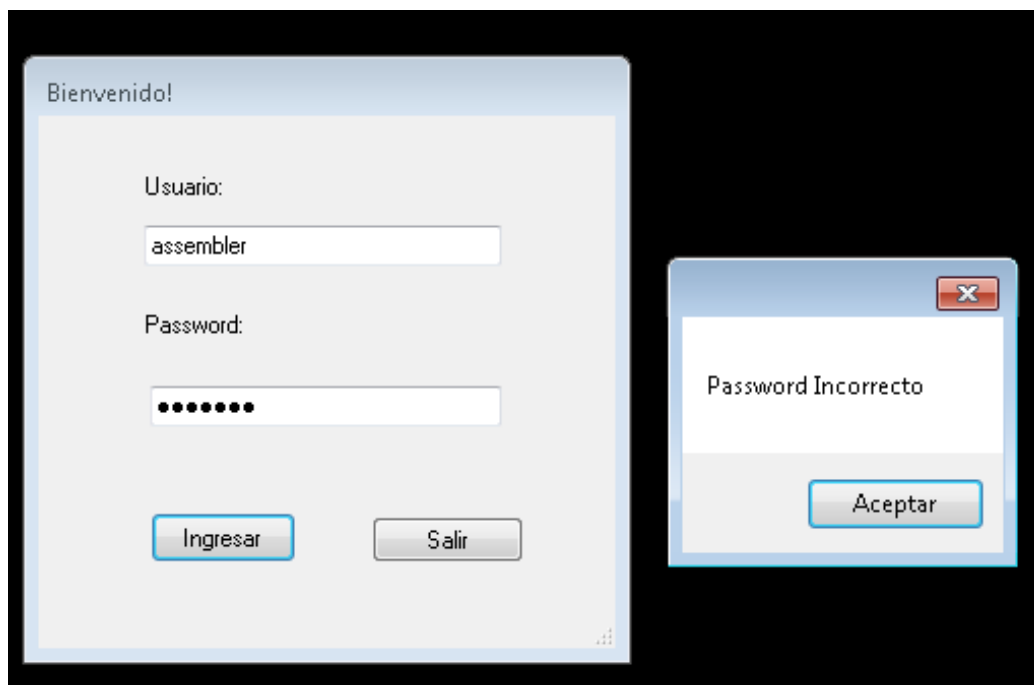


The image shows a Windows-style login dialog box. The title bar is blue and contains the text 'Bienvenido!'. The main area has a light gray background. There are two text labels: 'Usuario:' and 'Password:'. Below each label is a white rectangular input field. At the bottom of the dialog, there are two buttons: 'Ingresar' (Login) and 'Salir' (Exit). The 'Ingresar' button is slightly larger and has a gray border, while the 'Salir' button is smaller and has a blue border. The dialog box has a black border and a shadow effect.

Usuario Incorrecto (Login)



Password Incorrecto (Login)



Datos Correctos! (Bienvenido)



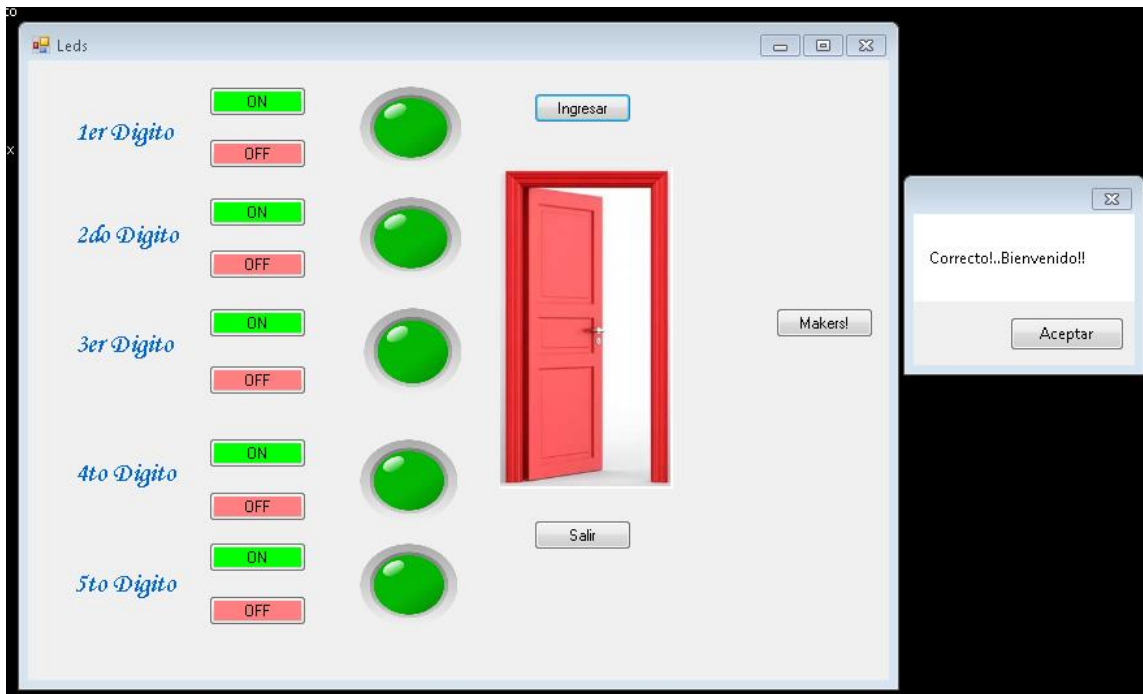
Ventana Principal



Codigo Incorrecto (Ventana Principal)



Codigo Correcto (Ventana Principal)



Codigo Boton Ingresar (Login)

1 referencia

```
private void btnIngresar_Click(object sender, EventArgs e)
{
    string usuario = txtUser.Text;
    string password = txtPass.Text;
    string user = "assembler";
    string pass = "2017";

    int n = mensaje(usuario, user);
    // MessageBox.Show(n.ToString());
    if (n == 1)
    {
        int m = mensaje(password, pass);
        if (m==1)
        {
            MessageBox.Show("Bienvenido!");
            this.Close();
        }
        else
        {
            MessageBox.Show("Password Incorrecto");
        }
    }
    else
    {
        MessageBox.Show("Usuario Incorrecto");
    }
}
```

Codigo Boton Ingresar y Salir (Principal)

```

1 referencia
private void btnIngresar_Click(object sender, EventArgs e)
{
    serialPort1.WriteLine("X");
    if (a == 1 && b == 1 && c == 1 && d == 1 && f == 1)
    {
        pictureBox12.Visible = true;
        pictureBox11.Visible = false;
        MessageBox.Show("Correcto!..Bienvenido!!");
    }
    else
    {
        pictureBox12.Visible = false;
        pictureBox11.Visible = true;
        MessageBox.Show("Incorrecto!!!");
    }
}

1 referencia
private void btnSalir_Click(object sender, EventArgs e)
{
    //Close();
    salir();
}

```

Codigo programa principal (Visual Studio – C#)

Leds_Arduino.Form1

▼ btnIngr

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Runtime.InteropServices;

namespace Leds_Arduino
{
    3 referencias
    public partial class Form1 : Form
    {
        [DllImport("midll.dll", CharSet = CharSet.Auto)]
        1 referencia
        public static extern void holaMundo();

        [DllImport("dll2.dll", CharSet = CharSet.Auto)]
        1 referencia
        public static extern void salir();
        1 referencia
        public Form1()
        {
            InitializeComponent();
            serialPort1.Open();
        }
        public int a, b, c, d, f;
    }
}
```

%

Codigo Botones ON/OFF Programa principal (Visual Studio – C#)

1 referencia

```
public Form1()
{
    InitializeComponent();
    serialPort1.Open();
}
public int a, b, c, d, f;
```

1 referencia

```
private void btnON_Click(object sender, EventArgs e)
{
    serialPort1.WriteLine("1");
    pictureBox2.Visible = true;
    pictureBox1.Visible = false;
    a = 1;
}
}
```

1 referencia

```
private void btnOFF_Click(object sender, EventArgs e)
{
    serialPort1.WriteLine("A");
    pictureBox1.Visible = true;
    pictureBox2.Visible = false;
    a = 0;
}
}
```

COMPLEMENTOS UTILIZADOS

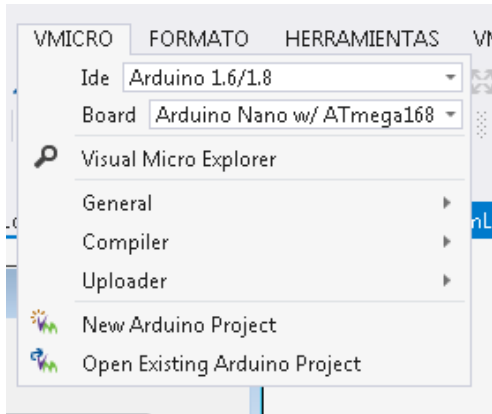
Software

-Visual Studio 2013

-“Visual.Micro.Arduino.Studio.vsix”

Pagina de descarga: <http://www.visualmicro.com/>

-IDE de Arduino



Codigo de Arduino

```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(12,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int a,b,c,d,f;
  if(Serial.available())
  {
    int datol=Serial.read();
    if(datol=='1'){
      digitalWrite(2,HIGH);

      a=1;
      Serial.println(a);
      Serial.println("1 ESTA ENCENDIDO");
    }
    if(datol=='A'){
      digitalWrite(2,LOW);
      a=0;
      Serial.println(a);
      Serial.println("1 ESTA APAGADO");
    }
  }
}
```

```

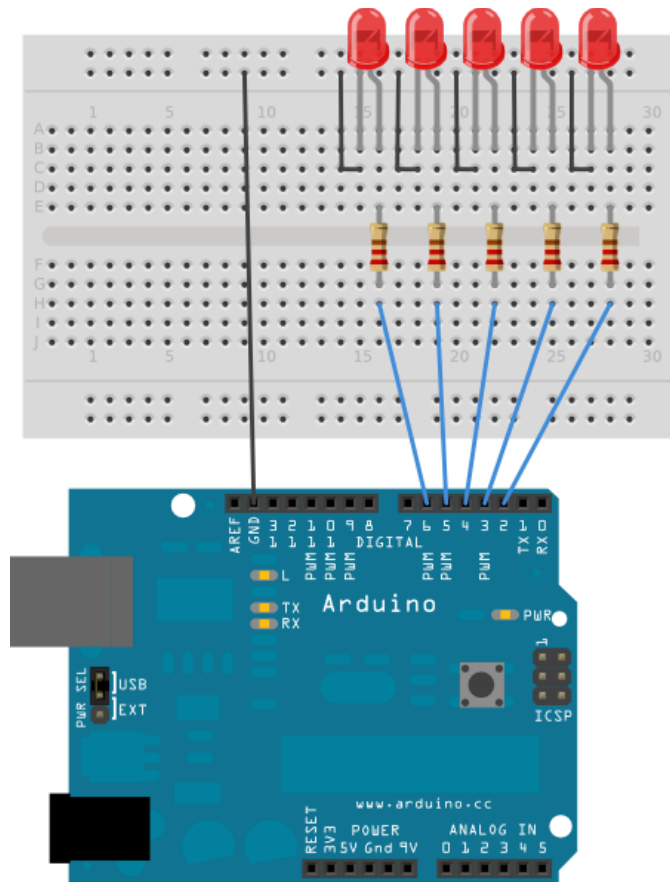
    }
    if(datol=='2'){
digitalWrite(3,HIGH);
b=1;
    }
    if(datol=='B'){
digitalWrite(3,LOW);
b=0;
    }
    if(datol=='3'){
digitalWrite(4,HIGH);
c=1;
    }
    if(datol=='C'){
digitalWrite(4,LOW);
c=0;
    }
    if(datol=='4'){
digitalWrite(5,HIGH);
d=1;
    }
    if(datol=='D'){
digitalWrite(5,LOW);
d=0;
    }
    if(datol=='5'){
digitalWrite(6,HIGH);
    }
    if(datol=='E'){
digitalWrite(6,LOW);
f=0;
    }

    if(a=1 && b=1 && c=1 && d=1 && f=1 && datol=='X')
    {
        Serial.println("Correcto!");
        digitalWrite(12,HIGH);
        
        else
        {
            // Serial.println("Incorrecto!");
            digitalWrite(12,LOW);
        }
    }
}

```

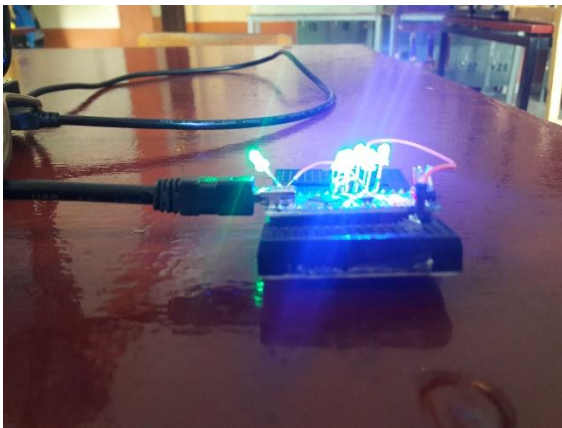
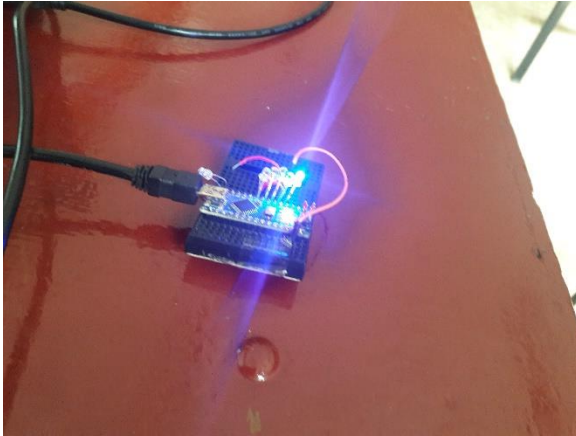

HARDWARE

DIAGRAMA DE CONEXIONADO DE ARDUINO



FUNCIONAMIENTO





CONSTRUCCIÓN DE LA PUERTA

MATERIALES

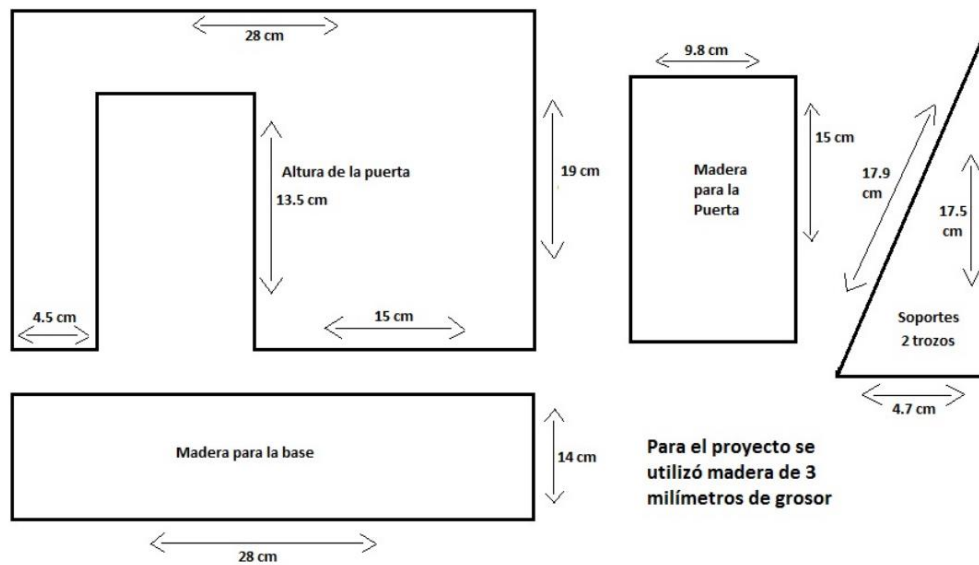
- Trozos de madera u otro material
- Bombilla
- Lectora en desuso de PC
- Alambre de 30 cm de largo, 3 mm de grosor
- Cables
- Silicona caliente y pegamento
- Leds

OTROS

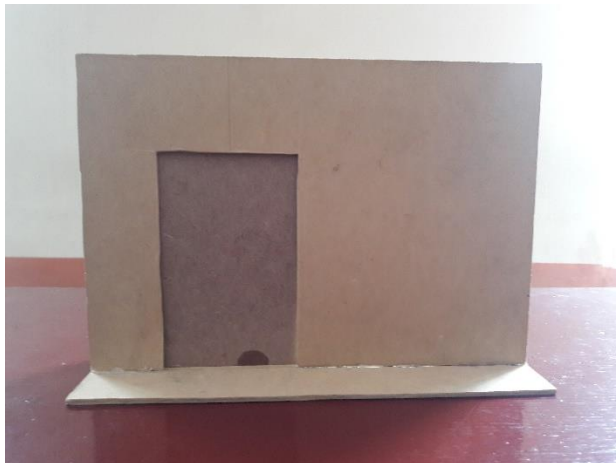
- Regla
- Plumon
- Lapicero
- Sierra de mano
- Estaño
- Cautin

PROCEDIMIENTO

1. Procedemos acortar la madera de acuerdo a las siguiente medidas:



2. Procedemos a ramar la puerta y a conectar el motor y la riel que sacamos de la lectora usada
3. Puerta terminada





CONCLUSIONES

Se logró implementar el candado electrónico enlazando Visual Studio 2013, Arduino y librerías dinámicas (dll) en ensamblador; teniendo las siguientes dificultades:

- Para enlazar Visual Studio con Arduino se utilizó el complemento Visual Micro para Arduino, con lo quedó subsanado este problema
- Para enlazar Visual Studio con Masm32 se utilizaron librerías dinámicas.
- Para la creación de botones en ensamblador en Visual Studio se utilizaron las funciones de Win32 implementadas en librerías dinámicas.
- En cuanto al funcionamiento del programa principal: se tuvo problemas comparando valores en el IDE de Arduino, por lo que se optó hacer la comparación en Visual Studio.

Y Homemade 3D printer