**INSTITUCIÓN UNIVERSITARIA CESMAG**

**FACULTAD DE INGENIERÍA - INGENIERÍA ELECTRÓNICA**

**ELECTIVA DE PROGRAMACIÓN 2A**

**PRESENTACIÓN PROYECTO:**

VIVIENDAS INTELIGENTES

|  |
| --- |
| **Nombre del proyecto:** |
| PARADISE DOMOTIC (Laura Ramos, Harold Chamorro) |
| **Objetivo general:** |
| Crear una casa la cual sea manipulada a través de tecnología |
| **Objetivos específicos:** |
| **•** Entender cómo implementar un código en una casa  • Dar una idea con el fin de controlar una casa con tecnología  • Saber si los códigos impuestos están realizando sus funciones |
| **1. FASE DE ANÁLISIS**  **\*** Se deben detallar claramente las funciones y procesos que realizará y  se ejecutarán en la viviente inteligente |
| Se desarrollara un sistema en Python y arduino en los cuales hay funciones tales como la definición de los mismos en Python es (def funcion( ):), y en arduino soy void loop( ) { } y void set up ( ) { } |
| **2. FASE DE DISEÑO – MAQUETACIÓN Y PROTOTIPADO**  **\*** Se deben relacionar las imágenes correspondientes a la maqueta (arquitectura) de la vivienda,  y el diseño (Mockups) de todas las interfaces desde las cuales se controlará la vivienda, y que luego serán programadas |
| **2.1. Maqueta o prototipo de la vivienda:** |
|  |
| **2.2. Mockups Desktop App:** |
|  |
| **2.2. Arquitectura del sistema de control: \*** Debe ser representada de manera gráfica y lógica |
|  |
| **2.4. Componentes electrónicos:**   |  |  |  | | --- | --- | --- | | **Elemento/componente** | **Referencia/característica/Descripción** | **Función** | | **Led** | **Led:** amarillos, rojos, verdes, azul, blanco  **Led:** es un tipo especial de diodo que trabaja como un diodo común, pero que al ser atravesado por la corriente eléctrica, emite luz.  **Referencia:** BL-B5141, RE-WW03A3SWQ4-N2 | Iluminación de la casa y indicador de un semáforo | | **Lm35** | **Referencia:** LM35DZ/LFT1  **Lm35:** registrar datos de temperatura | Controla la temperatura ambiente de la casa | | **Jumper** | **Jumper:** conductor de corriente eléctrica | Permite la conducción de energía a los componentes en el circuito | | **servomotor** | **Referencia:** SG90  **Servomotor:** Servomotores de rotación continúa: se caracterizan por ser capaces de girar 0 a 180 grados, es decir, una rotación completa. | Permite el acceso a la vivienda | | **Ventilador 5v** | **Referencia:** GDT4020B5V2P2.54 | Realiza la refrigeración de la casa | | **protoboard** | **Referencia:** WB-102  **Protoboard:** Una placa de pruebas o placa de inserción es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipo de circuitos electrónicos y sistemas similares | Facilita la conexión eléctrica de todo el circuito | | **Arduino** | **Referencia:** MEGA 2560 | Leer código fuente | |
| **3. FASE DE PROGRAMACIÓN - CÓDIGOS FUENTE**  **\*** El código fuente debe estar debidamente comentado,  explicando la función de las líneas principales |
| **3.1. Código fuente Arduino (.ino):** |
| //declarar variable  char numero;  //incluir bibliotecas externas  #include <Servo.h>  //incluye lcd  #include <LiquidCrystal.h>  LiquidCrystal lcd(7,6,5,4,3,2);  //declarar servos  Servo ServoD\_P;  Servo ServoD\_G;  //declarar servos y pines  const int Luz\_C = 22;  int EstadoLC = 0;  const int Luz\_S = 24;  int EstadoLS = 0;  const int Luz\_Ba = 32;  int EstadoLB = 0;  const int Luz\_H = 28;  int EstadoLH = 0;  const int Luz\_L = 34;  int EstadoLL = 0;  const int Luz\_G = 50;  int EstadoLG = 0;  const int LedR = 51;  const int LedV = 49;  const int LedA = 47;  #define FogonA 31  #define FogonB 33  #define FogonC 35  #define FogonD 37  int EstadoF1 = 0;  int EstadoF2 = 0;  int EstadoF3 = 0;  int EstadoF4 = 0;  const int Ext = 52;  int EstadoEx = 0;  const int Ven = 53;  int EstadoVe = 0;  float TG;  int SensorTG = 0;  float TC;  int SensorTC = 0;  int S\_PP = 0;  int a\_PPC = 0;  int a\_PPA = 130;  int S\_PG = 0;  int a\_PGC = 0;  int a\_PGA = 145;  void setup() {  // put your setup code here, to run once:  //se coloca a los pines como salidas  Serial.begin(9600);  pinMode(Luz\_C,OUTPUT);  pinMode(FogonA,OUTPUT);  pinMode(FogonB,OUTPUT);  pinMode(FogonC,OUTPUT);  pinMode(FogonD,OUTPUT);  pinMode(Luz\_S,OUTPUT);  pinMode(Luz\_Ba,OUTPUT);  pinMode(Luz\_H,OUTPUT);  pinMode(Luz\_L,OUTPUT);  pinMode(Luz\_G,OUTPUT);  pinMode(Ext,OUTPUT);  pinMode(Ven,OUTPUT);  pinMode(LedR,OUTPUT);  pinMode(LedA,OUTPUT);  pinMode(LedV,OUTPUT);  //se define los pines de los servos  ServoD\_P.attach(38);  ServoD\_G.attach(53);  //se define las filas y columnas del lcd  lcd.begin (12,2);  }  void loop() {  // put your main code here, to run repeatedly:  //definir variables para el nivel de luz de los fogones  int iF\_1 = 0;  int iF\_2 = 0;  int iF\_3 = 0;  int iF\_4 = 0;    TG = analogRead(SensorTG);  TC = analogRead(SensorTC);    //Probar si el lcd funciona  lcd.setCursor(0,1);  lcd.print (“funcionando”);    //lectura desde Python  if(Serial.available()>0){  numero=Serial.read();  Serial.print(numero);  //Luz sala########################  if (numero == 'a' and EstadoLS == 0){  digitalWrite(Luz\_S,HIGH);  lcd.setCursor(0,0);  lcd.print("Luz de sala");  lcd.setCursor(0,1);  lcd.print("Esta encendida");  EstadoLS = 1;  }else{  if (numero == 'a' and EstadoLS == 1){  digitalWrite(Luz\_S,LOW);  lcd.setCursor(0,0);  lcd.print("Luz de sala");  lcd.setCursor(0,1);  lcd.print("Esta apagada");  EstadoLS = 0;  }  }  //Luz cocina##############  if (numero == 'b' and EstadoLC == 0){  digitalWrite(Luz\_C,HIGH);  lcd.setCursor(0,0);  lcd.print("Luz de cocina");  lcd.setCursor(0,1);  lcd.print("Esta encendida");  EstadoLC=1;  }else{  if (numero == 'b' and EstadoLC == 1){  digitalWrite(Luz\_C,LOW);  lcd.setCursor(0,0);  lcd.print("Luz de cocina");  lcd.setCursor(0,1);  lcd.print("Esta apagada");  EstadoLC=0;  }  }  //Fogon Nº1##############  if (numero == 'c' and EstadoF1 == 0){  digitalWrite(FogonA,HIGH);  lcd.setCursor(0,0);  lcd.print("Fogon Nº1");  lcd.setCursor(0,1);  lcd.print("Esta encendido");  EstadoF1=1;  }else{  if (numero == 'c' and EstadoF1 == 1){  digitalWrite(FogonA,LOW);  lcd.setCursor(0,0);  lcd.print("Fogon Nº1");  lcd.setCursor(0,1);  lcd.print("Esta apagado");  EstadoF1=0;  }  }  //Fogon Nº2##############  if (numero == 'd' and EstadoF2 == 0){  digitalWrite(FogonB,HIGH);  lcd.setCursor(0,0);  lcd.print("Fogon Nº2");  lcd.setCursor(0,1);  lcd.print("Esta encendido");  EstadoF2=1;  }else{  if (numero == 'd' and EstadoF2 == 1){  digitalWrite(FogonB,LOW);  lcd.setCursor(0,0);  lcd.print("Fogon Nº2");  lcd.setCursor(0,1);  lcd.print("Esta apagado");  EstadoF2=0;  }  }  //Fogon Nº3##############  if (numero == 'e' and EstadoF3 == 0){  digitalWrite(FogonC,HIGH);  lcd.setCursor(0,0);  lcd.print("Fogon Nº3");  lcd.setCursor(0,1);  lcd.print("Esta encendido");  EstadoF3=1;  }else{  if (numero== 'e' and EstadoF3 == 1){  digitalWrite(FogonC,LOW);  lcd.setCursor(0,0);  lcd.print("Fogon Nº3");  lcd.setCursor(0,1);  lcd.print("Esta apagado");  EstadoF3=0;  }  }  //Fogon Nº4##############  if (numero == 'f' and EstadoF4 == 0){  digitalWrite(FogonD,HIGH);  lcd.setCursor(0,0);  lcd.print("Fogon Nº4");  lcd.setCursor(0,1);  lcd.print("Esta encendido");  EstadoF4=1;  }else{  if (numero == 'f' and EstadoF4 == 1){  digitalWrite(FogonD,LOW);  lcd.setCursor(0,0);  lcd.print("Fogon Nº4");  lcd.setCursor(0,1);  lcd.print("Esta apagado");  EstadoF4=0;  }  }  //Luz Garaje##############  if (numero == 'g' and EstadoLG == 0){  digitalWrite(Luz\_G,HIGH);  lcd.setCursor(0,0);  lcd.print("Luz de Garaje");  lcd.setCursor(0,1);  lcd.print("Esta encendida");  EstadoLG=1;  }else{  if (numero == 'g' and EstadoLG == 1){  digitalWrite(Luz\_G,LOW);  lcd.setCursor(0,0);  lcd.print("Luz de Garaje");  lcd.setCursor(0,1);  lcd.print("Esta apagada");  EstadoLG=0;  }  }  //Luz Baño##############  if (numero == 'h' and EstadoLB == 0){  digitalWrite(Luz\_Ba,HIGH);  lcd.setCursor(0,0);  lcd.print("Luz de Baño");  lcd.setCursor(0,1);  lcd.print("Esta encendida");  EstadoLB=1;  }else{  if (numero == 'h' and EstadoLB == 1){  digitalWrite(Luz\_Ba,LOW);  lcd.setCursor(0,0);  lcd.print("Luz de Baño");  lcd.setCursor(0,1);  lcd.print("Esta apagada");  EstadoLB=0;  }  }  //Luz Habitacion##############  if (numero == 'i' and EstadoLH == 0){  digitalWrite(Luz\_H,HIGH);  lcd.setCursor(0,0);  lcd.print("Luz de Habitacion");  lcd.setCursor(0,1);  lcd.print("Esta encendida");  EstadoLH=1;  }else{  if (numero == 'i' and EstadoLH == 1){  digitalWrite(Luz\_H,LOW);  lcd.setCursor(0,0);  lcd.print("Luz de Habitacion");  lcd.setCursor(0,1);  lcd.print("Esta apagada");  EstadoLH=0;  }  }  //Luz Lavadero##############  if (numero=='j' and EstadoLL == 0){  digitalWrite(Luz\_L,HIGH);  lcd.setCursor(0,0);  lcd.print("Luz de Lavadero");  lcd.setCursor(0,1);  lcd.print("Esta encendida");  EstadoLL=1;  }else{  if (numero=='j' and EstadoLL == 1){  digitalWrite(Luz\_L,LOW);  lcd.setCursor(0,0);  lcd.print("Luz de Lavadero");  lcd.setCursor(0,1);  lcd.print("Esta apagada");  EstadoLL=0;  }  }  if (numero == 'k'){  TG = analogRead(SensorTG);  TG = (5.0 \* TG \* 100)/1024.0;  lcd.setCursor(0,0);  lcd.print("Temperatura Garaje");  lcd.setCursor(0,1);  lcd.print(TG);  }  if (numero == 'l'){  TC = analogRead(SensorTC);  TC = (5.0 \* TC \* 100)/1024.0;  lcd.setCursor(0,0);  lcd.print("Temperatura Cocina");  lcd.setCursor(0,1);  lcd.print(TC);  }  //semáforo########  if (numero == 'l'){  digitalWrite (LedR,HIGH);  delay(4000);  digitalWrite (LedR,LOW);  digitalWrite (LedA,HIGH);  delay(2000);  digitalWrite (LedA,LOW);  digitalWrite (LedV,HIGH);  delay (4000);  digitalWrite (LedV,LOW);  digitalWrite (LedA,HIGH);  delay(2000);  digitalWrite (LedA,LOW);  digitalWrite (LedR,HIGH);  }  //extractor##########  if (numero == 'm'){  }if (TC>=27){  digitalWrite(Ext,HIGH);  }else{  if(TC <27){  digitalWrite(Ext,LOW);  }  }  //ventilador#####  if(numero == 'n'){  if(TG >= 27){  digitalWrite(Ven,HIGH);  }else{  if(TG <27){  digitalWrite(Ven,LOW);  }  }  }  //puerta principal############  if(numero == 'o' and S\_PP == 0 ){  ServoD\_P.write(a\_PPA);  delay(15);  S\_PP=1;}  else{  if(numero == 'o' and S\_PP == 1 ){  ServoD\_P.write(a\_PPC);  delay(15);  S\_PP=0;  }  }  //puerta garaje###########  if (numero == 'p' and S\_PG == 0 ){  if(S\_PG==0){  ServoD\_G.write(a\_PGA);}  S\_PG=1;  }else{  if(numero == 'p' and S\_PP == 1 ){  ServoD\_G.write(a\_PGC);  S\_PG=0;  }  }  }  } |
| **3.2. Código fuente Python (.py):** |
| #IMPORTAR LIBRERIAS  from tkinter import \*  import tkinter  import serial  import os  #arduinoData =serial.Serial("COM10",9600)  #############################  #CREAR VENTANA  v\_l =tkinter.Tk()  v\_l.title("LOGIN")  v\_l.geometry("500x500")  v\_l.resizable(FALSE,FALSE)  v\_l.configure(background='#46E0EB')#color del fondo de la 1ra ventana  color\_b=("gray77")  #############################  #VENTANAS  def On\_Off (numero):  os.system('cls')  numero = str(numero)  arduinoData.write(str.encode(numero))  print("ON/OFF")  def menu():  global v\_2  v\_2 = tkinter.Tk()  v\_2.title("Menu principal")  v\_2.geometry("400x400")  v\_2.resizable(FALSE,FALSE)  b\_m\_Sala = Button (v\_2, text = "Sala-Cocina",width=15,height=2,command=Sala\_Cocina).place(x=30,y=30)  b\_m\_ga = Button (v\_2, text = "Garaje",width=15,height=2,command=Garaje).place(x=30,y=100)  b\_hab = Button (v\_2, text = "Habitacion",width=15,height=2,command=Habitacion).place(x=150,y=30)  b\_baño = Button (v\_2, text = "Baño",width=15,height=2,command=Baño).place(x=270,y=100)  b\_lav = Button (v\_2, text = "Lavadero",width=15,height=2,command=Lavadero).place(x=150,y=100)  b\_sem = Button (v\_2, text = "semaforo",width=15,height=2,command=lambda:On\_Off(l)).place(x=270,y=30)  def Sala\_Cocina():  global v\_3  v\_3 = tkinter.Tk()  v\_3.title("Sala-Cocina")  v\_3.geometry("400x400")  v\_3.resizable(FALSE,FALSE)  v\_3.configure(background="#000000")  b\_Luz\_s = Button(v\_3,text="on/off Luz",width=10,height=2,command=lambda:On\_Off(a)).place(x=30,y=30)  b\_door\_p = Button(v\_3,text="puerta principal",width=15,height = 2, command = lambda:On\_Off(o)).place(x=120,y=30)  b\_luz\_cocina =Button(v\_3,text="Luz de la cocina",width=15,height = 2, command = lambda:On\_Off(b)).place(x=240,y=30)  b\_estufa =Button(v\_3,text="Estufa",width=5,height = 2, command = Estufa).place(x=30,y=80)  def Garaje():  global v\_5  v\_5 = tkinter.Tk()  v\_5.title("VENTILADORES")  v\_5.geometry("400x400")  v\_5.resizable(FALSE,FALSE)  b\_Luz = Button(v\_5, text = "Luz", width = 15 , height = 2, command = lambda:On\_Off(g)).place(x=31.5,y=15)  b\_door = Button(v\_5, text = "Puerta Garaje", width = 15 , height = 2, command = lambda:On\_Off(p)).place(x=31.5,y=60)  b\_t\_g = Button(v\_5, text = "Temp", width = 15 , height = 2, command = lambda:On\_Off(n)).place(x=31.5,y=105)    def Estufa():  global v\_6  v\_6 = tkinter.Tk()  v\_6.title("Menu cocina")  v\_6.geometry("400x400")  v\_6.resizable(FALSE,FALSE)  v\_6.resizable(FALSE,FALSE)  b\_extractor = Button(v\_6, text = "Temp", width = 15 , height = 2, command = lambda:On\_Off(m)).place(x=31.5,y=15)  b\_Fogon1 = Button(v\_6, text = "Fogon 1", width = 15 , height = 2, command = lambda:On\_Off(c)).place(x=31.5,y=55)  b\_Fogon2 = Button(v\_6, text = "Fogon 2", width = 15 , height = 2, command = lambda:On\_Off(d)).place(x=31.5,y=95)  b\_Fogon3 = Button(v\_6, text = "Fogon 3", width = 15 , height = 2, command = lambda:On\_Off(e)).place(x=31.5,y=135)  b\_Fogon4 = Button(v\_6, text = "Fogon 4", width = 15 , height = 2, command = lambda:On\_Off(f)).place(x=31.5,y=175)  def Habitacion():  global v\_7  v\_7 = tkinter.Tk()  v\_7.title("Menu de Control")  v\_7.geometry("175x130")  v\_7.resizable(FALSE,FALSE)  b\_Luz = Button(v\_7, text = "Luz", width = 15 , height = 2, command = lambda:On\_Off(i)).place(x=31.5,y=15)  def Baño():  global v\_8  v\_8 = tkinter.Tk()  v\_8.title("Menu de Control")  v\_8.geometry("175x130")  v\_8.resizable(FALSE,FALSE)  b\_Luz = Button(v\_8, text = "Luz", width = 15 , height = 2, command = lambda:On\_Off(h)).place(x=31.5,y=15)  def Lavadero():  global v\_9  v\_9 = tkinter.Tk()  v\_9.title("Menu de Control")  v\_9.geometry("175x130")  v\_9.resizable(FALSE,FALSE)  b\_Luz = Button(v\_9, text = "Luz", width = 15 , height = 2, command = lambda:On\_Off(j)).place(x=31.5,y=15)  #############################  input\_text\_u = StringVar()  Screen\_text = ""  input\_text\_k = StringVar()  Screen\_text = ""  D\_u = Entry(v\_l,font=('Comic Sans MS',20,'bold'),width=25,bd=5,insertwidth=3,bg="#BCBBBF",justify="center",textvariable=input\_text\_u).place(x=40,y=140)  D\_k = Entry(v\_l,font=('Comic Sans MS',20,'bold'),width=25,bd=5,insertwidth=3,bg="#BCBBBF",justify="center",textvariable=input\_text\_k).place(x=40,y=260)  ############################  B\_Acceso = Button(v\_l,text="ingresar",width=30,height=3,command=menu).place(x=140,y=360)  B\_Help = Button(v\_l,text="?",width=3,height=1,command=menu).place(x=237,y=440)  v\_l.mainloop() |
| **4. GUI (INTERFAZ GRÁFICA DE USUARIO)**  **\*** Se deben relacionar las interfaces gráficas de usuario que ya han sido desarrolladas.  Tenga en cuenta que cada interfaz debe coincidir con los Mockups creados en la fase de diseño |
| **C:\Users\Laurita R\Downloads\WhatsApp Image 2018-12-06 at 6.19.37 PM (3).jpeg** |
| **OBSERVACIONES GENERALES:** |
| En la fase de análisis debe relacionar en detalle las funciones y operaciones que serán controladas de manera automática y a través de interfaz gráfica:   * **Luces:**   + Diodo led * **Ventanas y puertas:**   + Uso de servomotores * **Ventiladores:**   + Climatización   + Sensores de temperatura * Aportes libres de nuevo conocimiento y/o autoaprendizaje del estudiante. * En esta experiencia se aprendió como realizar en físico lo que se hace en códigos dando una experiencia retroalimentaría de como se hace una maqueta y lo que se conecta coincida con lo físico. |

****