



**PROGRAMACIÓN PARA
ARTE MULTIMEDIA**
PROYECTO FINAL

JOSE LUIS CUENCA MONCADA
MÁSTER EN ARTES VISUALES Y MULTIMEDIA 2018 - 2019
jocuel@epsg.upv.es

INDICE

| | |
|--|----|
| TÍTULO DEL TRABAJO..... | 2 |
| ABSTRACT | 2 |
| ELEMENTOS DE LA INSTALACIÓN..... | 6 |
| ESQUEMA DE INTERACCIÓN..... | 7 |
| ESQUEMA DE PROCESAMIENTO | 8 |
| PREVISUALIZACIÓN..... | 8 |
| REFERENTES..... | 9 |
| REGISTRO | 10 |
| BIBLIOGRAFÍA | 10 |
| ANEXO | 11 |
| ARDUINO MEGA (solo recepción de mensaje para encender luces)..... | 11 |
| ARDUINO UNO (envió mensaje de datos) | 13 |
| PROCESSING (recepción, almacenamiento, procesamiento y envío de respuesta) | 14 |
| PURE DATA (recepción, especialización del sonido)..... | 17 |

TÍTULO DEL TRABAJO

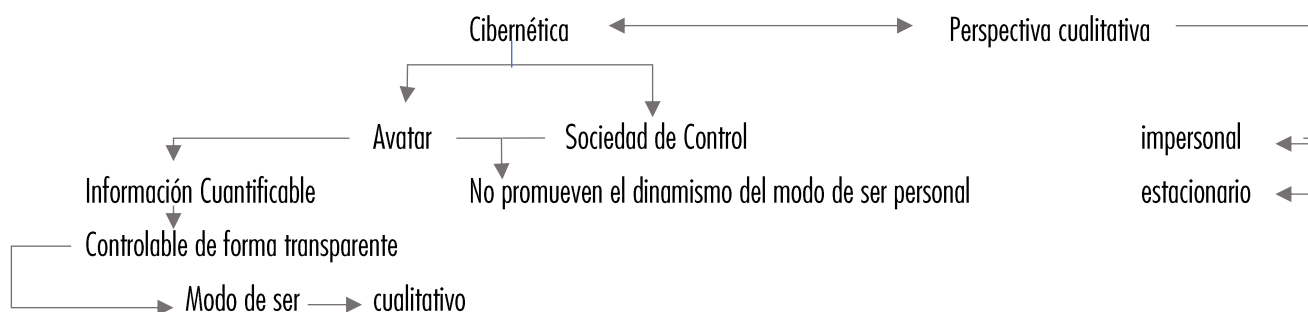
Sin TITULO 2.0

ABSTRACT

Sin TITULO 2.0 es una instalación interactiva que plantea un espacio virtual-analógico, una perspectiva del espacio virtual del internet, en donde el usuario al ingresar esta para a transformarse de manera metafórica en información. Esta información no tiene un rasgos característicos sino que difiere en el mundo real con los personas, para la cibernética solo somos información cuantificables (ontología de la factualidad) que vista desde esta perspectiva(N. Wiener, 1958) esta es impersonal, esta no permite adquirir el dinamismo vocacional del ser, en Jaramillo-Gaviria (2018): *“consideramos que la ontología de la factualidad, con la cual opera la cibernética no da espacio para comprender al hombre como persona, pues lo personal hace referencia a una realidad cualitativa que difiere ontológicamente de la realidad cuantitativa”*. En la instalación la persona una vez “transformada” en datos se pasea por el mundo virtual, este será identificado por medio de sensores ultrasónicos ubicados en lugares determinados en el espacio que permitirán identificar la posición de la persona, en el centro de procesamiento en tiempo real, y posteriormente esta enviara un mensaje de respuesta a los periféricos de salida (tanto al módulo de relés y los altavoces) para que cumplan una determinada acción. El uso del procesamiento en tiempo real nos permite utilizar el concepto de cibernética de segundo orden o “la retroalimentación o feedback”, desarrollada por Wiener, en la que enmarca que la maquina al recibir una información esta pasa a almacenarse, procesarse y obtener una respuesta actualizada considerando los mensajes recibidos y la nueva, llevando este concepto a la instalación la información obtenida de los sensores de distancia, ya procesada se podría estimar cual podría ser los siguientes posibles pasos del actuante (teniendo en cuenta en la programación las limitaciones y posibles salidas que este pueda realizar). Con el mensaje enviado en el caso que se utiliza en los relés este determinara las luces que se encenderán alrededor del usuario conforme realice su movimiento, así mismo con el mensaje que enviara a Pure Data, en el cual activara y desactivara una determinada salida de audio en búsqueda de la especialización del sonido (ver figura 2).

En las sociedades contemporáneas por distinta que esta sea, existe tipos de interfaces con las que el individuo interactúan, llámese este dispositivo movil, máquinas expendedoras, cajeros automáticos, todas tienen en común que observan e interactúan con el usuario como una fuente de datos cuantificables(mensajes), en la cibernética se estudia el mensaje que se envían entre estos objetos (humano-maquina, maquina-humano o máquina-maquina). En Jaramillo-Gaviria (2018) encontramos una visión más en profundidad:

[...] implica un modo de subjetivación del hombre como avatar y un concepción de la sociedad de cibernética como una sociedad de control. El hombre Avatar ¹se caracteriza, especialmente, por desarrollar un cerebro capaz de procesar información, la cual es traducible al lenguaje de la matemática probabilística e intercambiable con las maquinas que proveen las nuevas tecnologías de la información y la comunicación. La sociedad de control, por su parte se caracteriza por ser una sociedad transparente en la que la relaciones entre procesadores cognitivos, ya sean hombres o maquinas, están condicionadas por el control, la actualidad, la velocidad y a comunicación de los flujos de información. En la sociedad de control, que opera con una ontología de la factualidad, el hombre avatar no funge como un ser singular e interior, sino como un ser lógico y exterior, homologable a un procesador de información.”



PALABRAS CLAVE

Cibernética, Cuerpo, Tecnología, Información, Realidad

¹ Modo de subjetivación que opera la cibernética

ESQUEMA

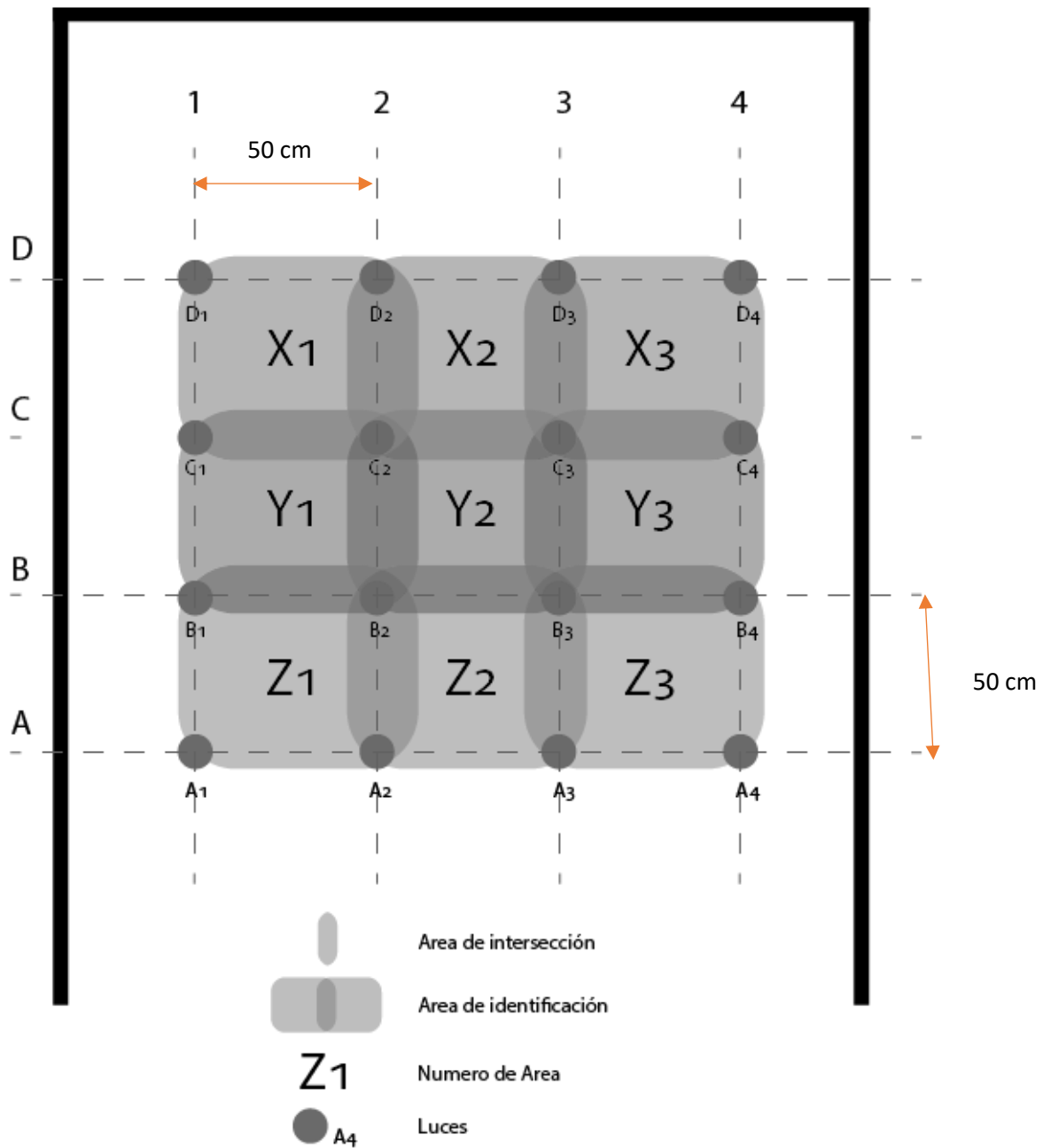
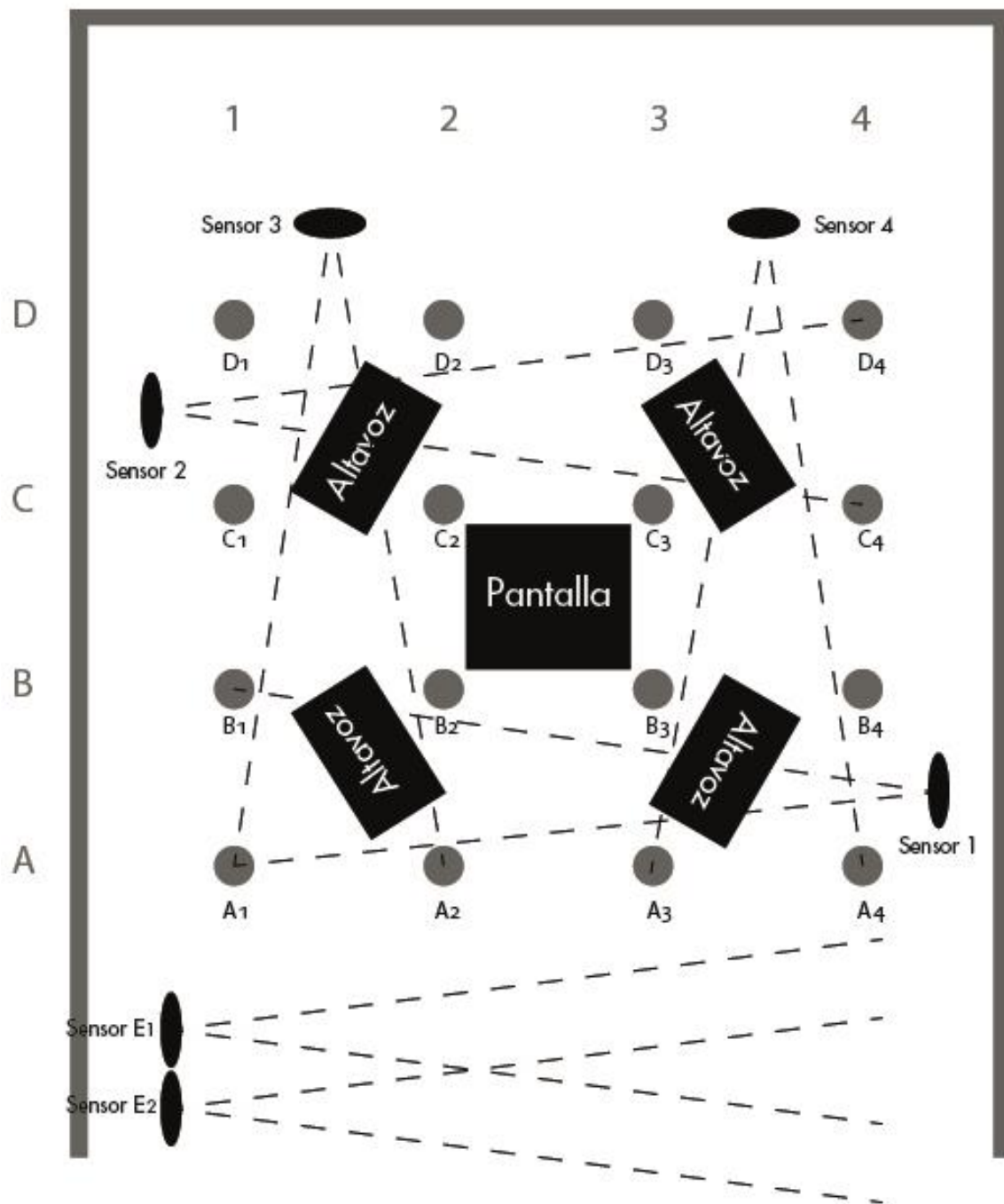
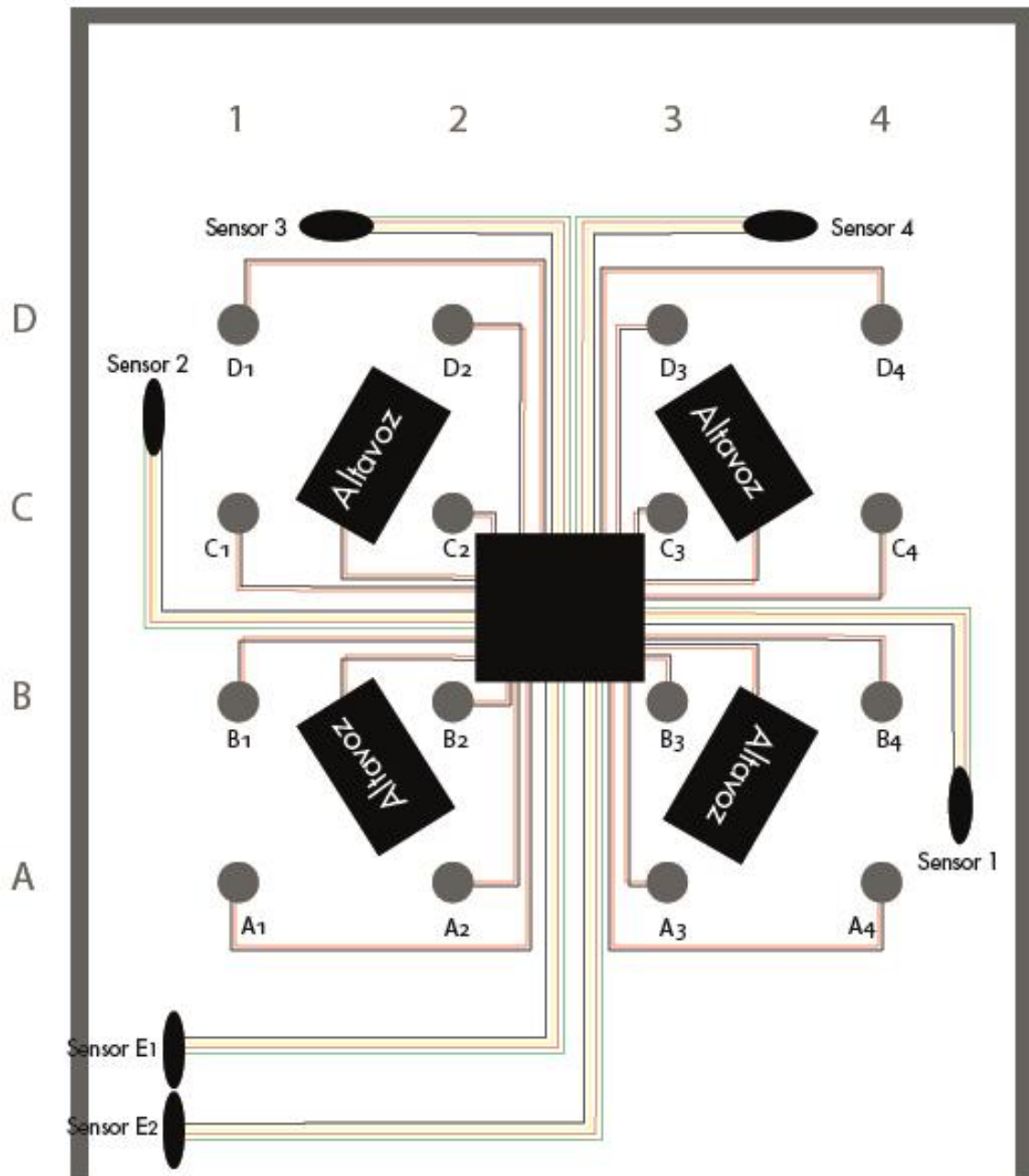


Figura 1.- Vista Superior de la Instalación



Sensor E1, E2.- corta fuego instalado en el espacio que condiciona a la actividad de la instalación.
Sensor1,2,3,4.- sensores que nos daran las posiciones X y Y de la persona en el espacio.
Pantalla.- en este dispositivo se mostrara la ubicación de la persona en el espacio.
X1,X2,X3,Y1,Y3,Z1,Z2,Z3.- valores de ubicación en cuadrantes del espacio.
A1,A2,A3,A4,B1,B2,B3,B4,C1,C2,C3,C4, D1,D2,D3,D4.- ubicación de luces circundantes a la posición de la persona.

Figura 2.- Vista Superior con la ubicación de los elementos en la Instalación



Sensor E1, E2.- corta fuego instalado en el espacio que condiciona a la actividad de la instalación.
Sensor1,2,3,4.- sensores que nos daran las posiciones X y Y de la persona en el espacio.
Cable Rojo.- energia, Cable Negro.- tierra, Cable Amarillo.- información, Cable Verde.- información de retorno.
A1 ,A2,A3,A4,B1 ,B2,B3,B4,C1 ,C2,C3,C4, D1 ,D2,D3,D4.- ubicación de luces circundantes a la posición de la persona.

Figura 3.- Esquema electrónico de la instalación

ELEMENTOS DE LA INSTALACIÓN

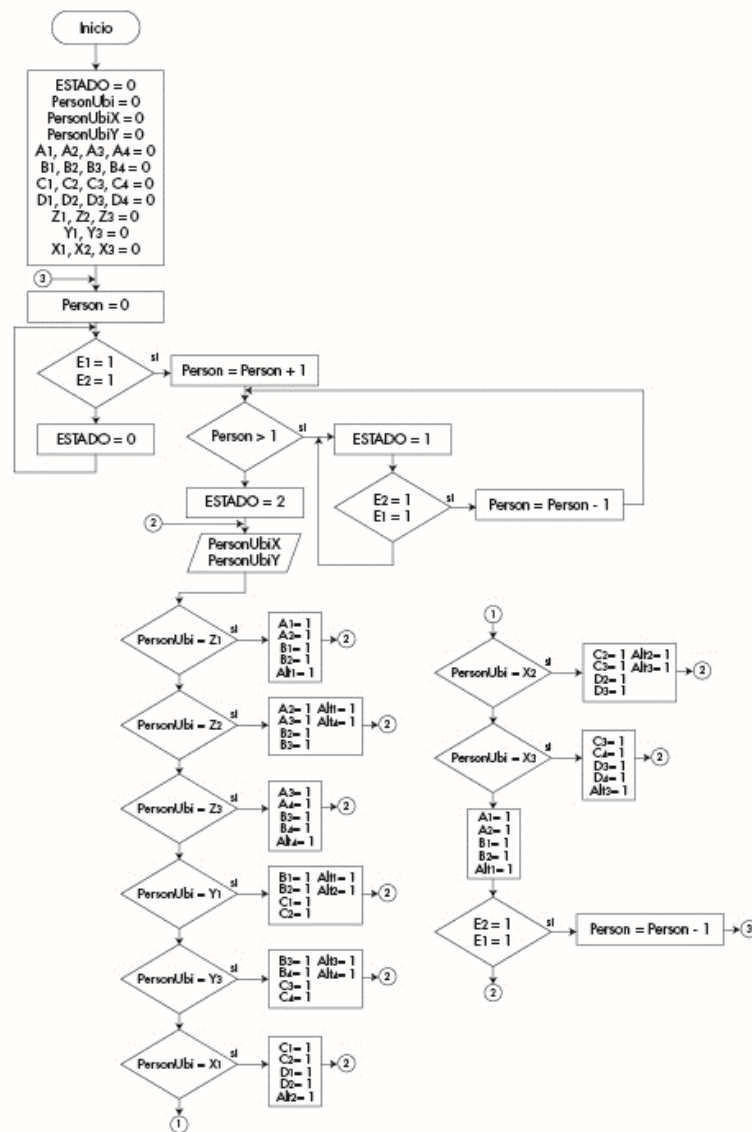
| | | |
|--|-----|---------|
| - Arduino Mega2560 R3, tarjeta placa con microcontrolador con USB. | 1 | € 10,77 |
| - Modulo de expansión sensor Shiel V 2.0 para Arduino Mega | 1 | € 2,79 |
| - Arduino Uno Rev 3 | 1 | € 20,00 |
| - Modulo Relé 12 V de 16 Canales AVR | 1 | € 16,56 |
| - Sensores ultrasónico HC-SR04 | 8 | € 11,66 |
| - tiras de luces LED (blancos) | 16 | € 56,45 |
| - Alimentador 12VDC con conector de 5.5 mm x 2.1 mm. | 1 | € 3,90 |
| - Conector Jack 2.1 x 5.5 mm. Alimentación Hembra DC. | 1 | € 0,80 |
| - Placa Protoboard 830 contactos | 1 | € 4,54 |
| - Cable Pin Hembra a Hembra 20 cm Jumper | 20 | € 2,39 |
| - Cable Pin Macho a Hembra 20 cm Jumper | 20 | € 2,39 |
| - Cable Pin Macho a Macho 20 cm Jumper | 20 | € 2,39 |
| - Baterías AAA 1.5v | 12 | € 7.35 |
| - Cable 2.5 mm | 200 | € 29,78 |



Figura 4.- Elementos de la instalación

ESQUEMA DE INTERACCIÓN

El procesado de la información es muy sencillo (ver figura 5). La instalación empieza con un contador de persona (person) arranca en cero, antes de ingresar al espacio donde se encuentra la instalación, estarán 2 sensores ultrasónicos (E1 yE2) que realizan la función de corta fuego y determinaran el estado que se encuentra la instalación, el estado que arranca el sistema es cero(inactivo) en donde todo el sistema se encuentra en reposo, mas solo se encuentran en lectura E1 y E2, que al ingresar un usuario el contador person suma 1 con lo que el sistema cambia a estado uno(activado), en este estado se encienden las luces y los sensores 1,2,3,4,5 y 6 empiezan a realizar la lectura para determinar la posición de las persona, realizando cada cierto tiempo una lectura y una retroalimentación con los datos (mensajes) que recibe el cerebro (de la instalación), estos llegan a Processing que a continuación procederá a almacenarlo, procesarlo y enviara las respuestas tanto a la placa de Arduino (que controla las luces) así como a Pure Data (que controla el audio). Como están en constante lectura los sensores del corta fuego al presenciar el ingreso de otra persona, el estado cambiara a dos (modo error) debido a que no podrá determinar la posición de dos figurantes en el espacio y el sistema esta diseñado de manera unipersonal, y no saldrá de este estado hasta que las personas hayas salido del sistema y este posteriormente se "reinicie" y vuelva al estado cero (inactivo). Así mismo al estar dentro solo una persona los sensores del cortafuego sentirán o determinarán cuando este abandone el espacio y pase a estado cero (inactivo).

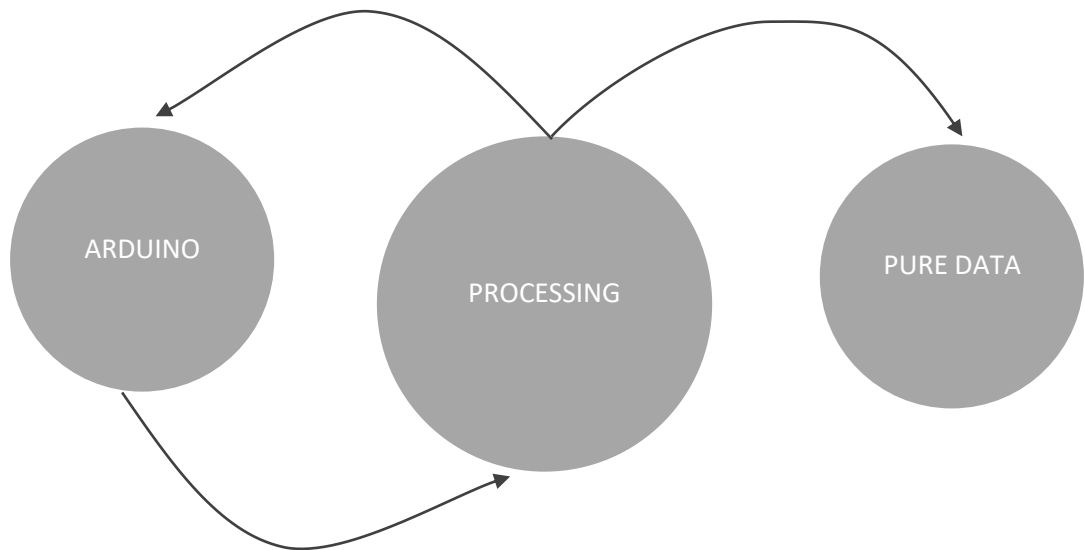


Person.- valor numero asignado a la persona que participa en la instalación. (contador)
 ESTADO.- actividad que realizara la instalación.
 E1, E2.- corta fuego instalado en el espacio que condiciona a la actividad de la instalación.
 PersonUbi.- resultado del calculo proveniente de los valores en los sensores, PersonUbiX y PersonUbiY.
 X1,X2,X3,Y1,Y3,Z1,Z2,Z3.- valores de ubicación en cuadrantes del espacio.
 A1 ,A2,A3,A4,B1,B2,B3,B4,C1,C2,C3,C4, D1,D2,D3,D4.- ubicación de luces circundantes a la posición de la persona.

Figura 5.- Diagrama de Flujo del programa

ESQUEMA DE PROCESAMIENTO

La instalación realiza un procesamiento de dos formas tanto unidireccional como bidireccional, teniendo en cuenta esto nos da la siguiente representación:



PREVISUALIZACIÓN

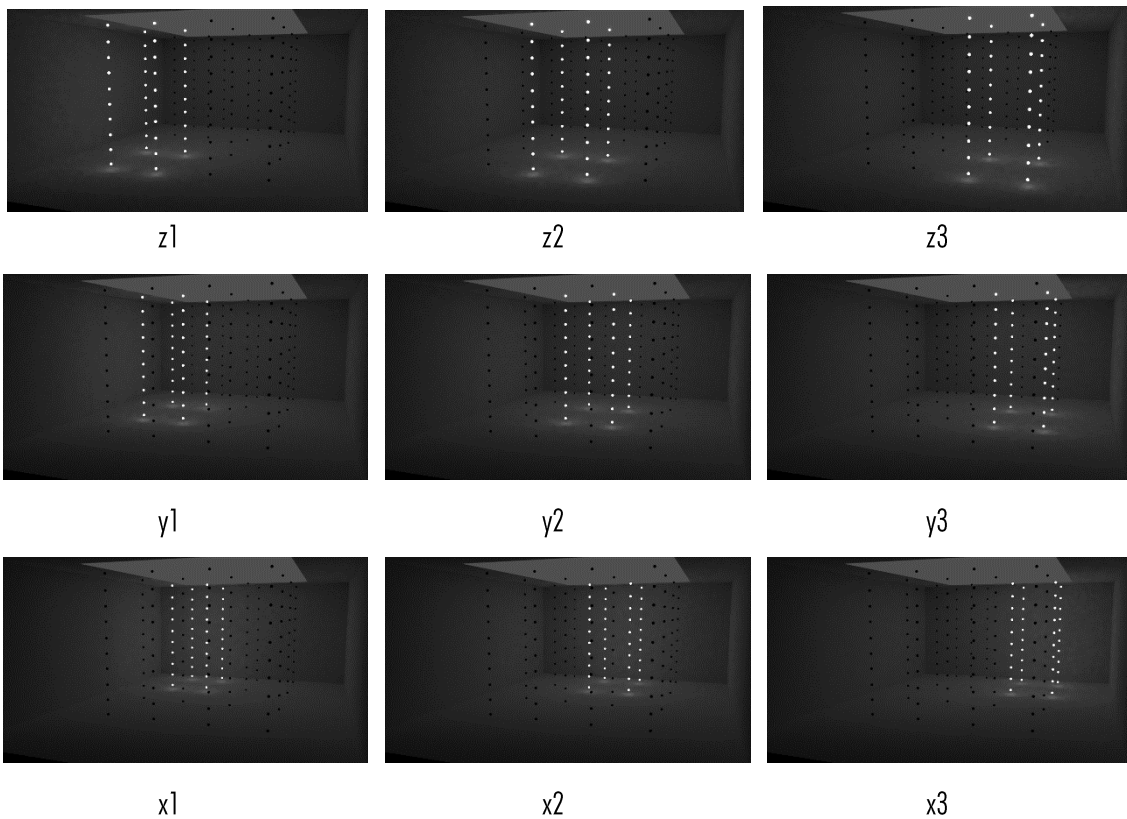
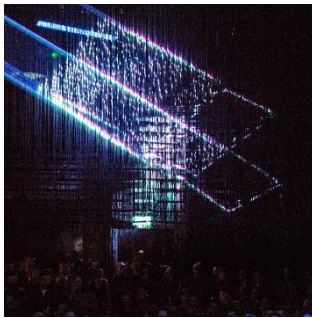


Figura 6.- Vista en Perspectiva de la Instalación

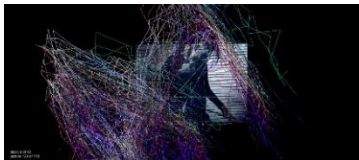
REFERENTES



Lozano-Hemmer, R. (2010). Exposición de corazonadas [Imagen]
Recuperado de: http://www.lozano-hemmer.com/rafael_lozano-hemmer_pulse_show.php



Cooper, M. (2018). Aether [Imagen]
Recuperado de: <https://maxcooper.net/aether>



Stain (2016). Picchain-net [Imagen]
Recuperado de: <https://www.stain.ws/PICCHAIN-NET>



Stain (2018). Pattern [Captura de Pantalla]
Recuperado de: <https://stain.ws/PATTERN>



Stanza (2010). Data, Data, Data [Captura de Pantalla]
Recuperado de: <http://www.stanza.co.uk/data/index.html>



Anai, Y. (2018). In the Rain [Captura de Pantalla]
Recuperado de: <http://www.stanza.co.uk/data/index.html>

REGISTRO

<https://vimeo.com/341902728>

BIBLIOGRAFÍA

- Aguilar, T. (2008). *Ontología Cyborg: el cuerpo en la nuevas sociedad tecnológica*. Editorial Gedisa.
- Chavarría, G. (2013). *El posthumanismo y el transhumanismo: transformaciones del concepto de ser humano en la era tecnológica*. Editorial Universidad de Costa Rica. pp. 1-55.
- Cooney, B. (2003). *Post humanity. Thinking philosophically about the Future*. Maryland: The Rowman and Littlefield.
- Gros, B. (2001). *De la cibernética clásica a la cibercultura: herramientas conceptuales desde mirar el mundo cambiante*. FUENTE: https://campus.usal.es/~teoriaeducacion/rev_numero_02/n2_art_gros.htm
- Gutiérrez, F. (2016). Nihilismo, bioética y biopolítica. *Revista FAIA*. 2016, Vol. 1, no 4.
- Izagirre, J. (1997). *Anotaciones en los márgenes de un arte cibernético*. Editorial Virus.
- Jaramillo-Gaviria, C., et al. (2018). El avatar: un modo de ser cibernético cualitativamente estacionario. *Folios*. Vol. 48, pp. 193-206.
- Mejía, I. (2005). *El cuerpo posthumano en el arte y la cultura contemporánea*. Editorial Universidad Nacional Autónoma de México.
- Stelarc (1997). *Das estrategias psicológicas ás ciberestrategias: a protética, a robótica e a existencia remota*, En Domingues, D. (ed.). *A arte no século XXI. A humanização das tecnologias*, UNESP, pp. 54-59.
- Stelarc (2008). *EXCESS AND INDIFFERENCE: Alternate Body Architectures*.
- Stelarc (2009). *THE CADAVER, THE COMATOSE & THE CHIMERA: Alternate Anatomical Architectures*.
- Tijerina, A. B. N. (2017). *Imaginaris urbanos y transhumanismo: tensiones entre la ficción tecno urbana y la humanidad salvaje*. *Sociología y Tecnociencia*. Vol. 7(1), pp. 5-36.
- Tiqun, C. (2013). *La hipótesis cibernética*.
- Zurburgg, N. (1999). *Virilio, Stelarc and 'Terminal' Technoculture*. *Theory, Cultura & Society*. Vol. 16 (5-6, pp. 177-199.

ANEXO

ARDUINO MEGA (solo recepción de mensaje para encender luces)

```
#include <AP_Sync.h>

AP_Sync streamer(Serial);

void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(14, OUTPUT);
  pinMode(15, OUTPUT);

  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(9, HIGH);
  digitalWrite(11, HIGH);
  digitalWrite(13, HIGH);
  digitalWrite(14, HIGH);
  digitalWrite(15, HIGH);
}

void loop() {
  if(Serial.available()){
    String command = Serial.readString();
    if(command == "z"){
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(5, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(9, HIGH);
      digitalWrite(13, HIGH);
      digitalWrite(14, HIGH);
      digitalWrite(15, HIGH);
      digitalWrite(11, HIGH);
    }else if (command == "p") { //— ESPACIO 1
      digitalWrite(2, LOW);
      digitalWrite(3, HIGH);
      digitalWrite(5, LOW);
      digitalWrite(7, HIGH);
      digitalWrite(9, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(13, LOW);
      digitalWrite(14, HIGH);
      digitalWrite(15, LOW);
    }else if (command == "q") { //— ESPACIO 2
      digitalWrite(2, LOW);
      digitalWrite(3, LOW);
      digitalWrite(5, HIGH);
      digitalWrite(7, LOW);
      digitalWrite(9, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(13, HIGH);
      digitalWrite(14, LOW);
      digitalWrite(15, LOW);
    }else if (command == "r") { //— ESPACIO 3
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(5, LOW);
      digitalWrite(7, HIGH);
      digitalWrite(9, LOW);
      digitalWrite(11, HIGH);
      digitalWrite(13, HIGH);
      digitalWrite(14, LOW);
      digitalWrite(15, LOW);
    }else if (command == "s") { //— ESPACIO 4
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(5, HIGH);
      digitalWrite(7, LOW);
      digitalWrite(9, HIGH);
      digitalWrite(11, LOW);
      digitalWrite(13, HIGH);
      digitalWrite(14, LOW);
      digitalWrite(15, LOW);
    }else if (command == "t") { //— LUZ 1
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(5, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(9, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(13, LOW);
      digitalWrite(14, HIGH);
      digitalWrite(15, HIGH);
    }else if (command == "u") { //— LUZ 2
      digitalWrite(2, LOW);
      digitalWrite(3, HIGH);
      digitalWrite(5, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(9, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(13, HIGH);
      digitalWrite(14, HIGH);
      digitalWrite(15, HIGH);
    }else if (command == "v") { //— LUZ 3
      digitalWrite(2, HIGH);
      digitalWrite(3, LOW);
      digitalWrite(5, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(9, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(13, HIGH);
      digitalWrite(14, HIGH);
      digitalWrite(15, HIGH);
    }else if (command == "w") { //— LUZ 4
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(5, LOW);
      digitalWrite(7, HIGH);
      digitalWrite(9, HIGH);
    }
  }
}
```

```

digitalWrite(11, HIGH);
digitalWrite(13, HIGH);
digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
}else if (command == "x") {    ///— LUZ 5
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(5, HIGH);
digitalWrite(7, HIGH);
digitalWrite(9, HIGH);
digitalWrite(11, HIGH);
digitalWrite(13, HIGH);
digitalWrite(14, LOW);
digitalWrite(15, HIGH);
}else if (command == "y") {    ///— LUZ 6
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(5, HIGH);
digitalWrite(7, LOW);
digitalWrite(9, HIGH);
digitalWrite(11, HIGH);
digitalWrite(13, HIGH);
digitalWrite(14, HIGH);
digitalWrite(15, LOW);
}else if (command == "d") {    ///— LUZ 7
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(5, HIGH);
digitalWrite(7, HIGH);
digitalWrite(9, LOW);
digitalWrite(11, HIGH);
digitalWrite(13, HIGH);
digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
}else if (command == "e") {    ///— LUZ 8
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(5, HIGH);
digitalWrite(7, HIGH);
digitalWrite(9, HIGH);
digitalWrite(11, HIGH);
digitalWrite(13, HIGH);
digitalWrite(14, LOW);
digitalWrite(15, HIGH);
}else if (command == "f") {    ///— LUZ 9
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(5, HIGH);
digitalWrite(7, HIGH);
digitalWrite(9, HIGH);
digitalWrite(11, LOW);
digitalWrite(13, HIGH);
digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
}else if (command == "h") {    /// — SECUENCIA DE ERROR
digitalWrite(13,LOW);
digitalWrite(9,LOW);
digitalWrite(7,LOW);
digitalWrite(5,LOW);
delay(200);
digitalWrite(13,HIGH);
digitalWrite(9,HIGH);
digitalWrite(7,HIGH);
digitalWrite(5,HIGH);
delay(250);
digitalWrite(9,LOW);
digitalWrite(15,LOW);
digitalWrite(14,LOW);
digitalWrite(3,LOW);
delay(400);
digitalWrite(9,HIGH);
digitalWrite(15,HIGH);
digitalWrite(14,HIGH);
digitalWrite(3,HIGH);
delay(250);
digitalWrite(9,LOW);
digitalWrite(15,LOW);
digitalWrite(3,LOW);
digitalWrite(2,LOW);
delay(200);
digitalWrite(9,HIGH);
digitalWrite(15,HIGH);
digitalWrite(3,HIGH);
digitalWrite(2,HIGH);
delay(250);
digitalWrite(15,LOW);
digitalWrite(5,LOW);
digitalWrite(7,LOW);
digitalWrite(2,LOW);
digitalWrite(14,LOW);
delay(400);
digitalWrite(15,HIGH);
digitalWrite(5,HIGH);
digitalWrite(7,HIGH);
digitalWrite(2,HIGH);
digitalWrite(14,HIGH);
delay(250);
}
}
}

```

/* el presente código está basada con la presentación, las variaciones del código original con 16 es mínima solo de adicionar las respectivas luces y espacios, pero siguen la misma lógica*/

ARDUINO UNO (envió mensaje de datos)

```

#include <Ultrasonic.h>

Ultrasonic ultrasonic1(12, 13);
Ultrasonic ultrasonic2(10, 11);
Ultrasonic ultrasonic3(8, 9);
Ultrasonic ultrasonic4(6, 7);
Ultrasonic ultrasonic5(4, 5);
Ultrasonic ultrasonic6(2, 3);

int firstSensor = 0; // first sensor
int secondSensor = 0; // second sensor
int thirdSensor = 0; // third sensor
int fourthSensor = 0; // fourth sensor
int fifthSensor = 0; // fifth sensor
int sixthSensor = 0; // digital sensorsen1,sen2,sen3,sen4,sen5,sen6;
int timer = 100;
int val=0;
int inByte=0;

void setup() {
  Serial.begin(9600);
  while (!Serial) {}
  establishContact();
}

void loop() {
  if (Serial.available() > 0) {
    inByte = Serial.read();
    firstSensor = ultrasonic1.read(CM);
    secondSensor = ultrasonic2.read(CM);
    thirdSensor = ultrasonic3.read(CM);
    fourthSensor = ultrasonic4.read(CM);
    fifthSensor = ultrasonic5.read(CM);
    sixthSensor = ultrasonic6.read(CM);
    //DATOS A ENVIAR
    Serial.write(firstSensor);
    Serial.write(secondSensor);
    Serial.write(thirdSensor);
    Serial.write(fourthSensor);
    Serial.write(fifthSensor);
    Serial.write(sixthSensor);
  }
  delay(100);
}

void establishContact() {
  while (Serial.available() <= 0) {
    Serial.print('A'); // send a capital A
    delay(300);
  }
}

```

PROCESSING (recepción, almacenamiento, procesamiento y envío de respuesta)

```
import processing.serial.*;
import oscP5.*;
import netP5.*;

Serial myPort;
OscP5 oscP5;
NetAddress myRemoteLocation;

int[] serialInArray = new int[5]; // Where we'll put what we receive
int serialCount = 0; // A count of how many bytes we receive
boolean firstContact = false; // Whether we've heard from the microcontroller
int persona;
int firstSensor = 0; // Los valores A se obtienen del sensor 1
int secondSensor = 0; // Los valores B se obtienen del sensor 2
int thirdSensor = 0; // Los valores C se obtienen del sensor 3
int fourthSensor = 0; // Los valores D se obtienen del sensor 4
int fifthSensor = 0; // Los valores E se obtienen del sensor 5
int sixthSensor = 0;
int distGen;
int A,B,C,D,E,F,A1,A2,A3,B1,B2,B3,C1,C2,C3,D1,D2,D3,E1,E2,E3,F1,F2,F3;
//los espacios están distribuidos desde la izquierda a derecha, donde A esta
situado entre las luces A1,A2,B1,B2.
int persoUbi;
char val;

void setup(){
  frameRate(15);
  size(200, 500);
  persona=0;
  val=0;
  printArray(Serial.list());

  String portName = Serial.list()[0];
  myPort = new Serial(this, portName, 9600);

  oscP5 = new OscP5(this,12001);
  myRemoteLocation = new NetAddress("localhost",12000);
}

void draw(){
  background(0);
  frameRate(6);
  //if ( myPort.available() > 0) { // If data is available,
  // val = myPort.read(); // read it and store it in val
  // println(val);

  OscMessage myMessage = new OscMessage("n");

  myMessage.add(firstSensor);
  myMessage.add(secondSensor);
  myMessage.add(thirdSensor);
  oscP5.send(myMessage, myRemoteLocation);

  distGen=50; // el valor de distGen es la medida en cm2 de los espacios
  A=firstSensor; // Los valores A se obtienen del sensor 1
```

```
B=secondSensor; // Los valores B se obtienen del sensor 2
C=thirdSensor; // Los valores C se obtienen del sensor 3
D=fourthSensor; // Los valores D se obtienen del sensor 4
E=fifthSensor; // Los valores E se obtienen del sensor 5
F=sixthSensor; // Los valores F se obtienen del sensor 6
A1=distGen+distGen; // el valor de A1 son los limitantes del espacio 1
A2=A1+distGen; // el valor de A1 son los limitantes del espacio 2
A3=A2+distGen; // el valor de A1 son los limitantes del espacio 3
B1=distGen+distGen; // el valor de A1 son los limitantes del espacio 4
B2=B1+distGen; // el valor de A1 son los limitantes del espacio 5
B3=B2+distGen; // el valor de A1 son los limitantes del espacio 6
C1=distGen+distGen; // el valor de A1 son los limitantes del espacio 7
C2=C1+distGen; // el valor de A1 son los limitantes del espacio 8
C3=C2+distGen; // el valor de A1 son los limitantes del espacio 9
D1=distGen+distGen; // el valor de A1 son los limitantes del espacio 7
D2=D1+distGen; // el valor de A1 son los limitantes del espacio 4
D3=D2+distGen; // el valor de A1 son los limitantes del espacio 1
E1=distGen+distGen; // el valor de A1 son los limitantes del espacio 8
E2=E1+distGen; // el valor de A1 son los limitantes del espacio 5
E3=E2+distGen; // el valor de A1 son los limitantes del espacio 2
F1=distGen+distGen; // el valor de A1 son los limitantes del espacio 9
F2=F1+distGen; // el valor de A1 son los limitantes del espacio 6
F3=F2+distGen; // el valor de A1 son los limitantes del espacio 3
//Por lo que se obtienen coordenadas (orden de lectura de arriba izquierda
hacia abajo derecha):
//
// (A3,F3) (A2,E2) (A1,D3) | Espacio 9 Espacio 8 Espacio7 |
// (B3,F2) (B2,E2) (B1,D2) | Espacio 6 Espacio 5 Espacio4 |
// (A3,F3) (A2,E3) (A1,D3) | Espacio 3 Espacio 2 Espacio1 |
//
if (persoUbi==0){
  if(((A > distGen) && (A < A1)) && ((D > D2) && (D < D3))){
    persoUbi=1;
    myPort.write('p');
    persona= persona + 1;
  } else if(((A > A1) && (A < A2)) && ((E > E2) && (E < E3))){
    persoUbi=2;
    myPort.write('q');
    persona=persona + 1;
  } else if(((A > A2) && (A < A3)) && ((F > F2) && (F < F3))){
    persoUbi=3;
    myPort.write('r');
    persona=persona + 1;
  }
} else if(persoUbi==1){
  if(((A > A1) && (A < A2)) && ((E > E2) && (E < E3))){
    persoUbi=2;
    myPort.write('q');
  } else if((B > distGen) && (B < B1)) && ((D > D1) && (D < D2))){
    persoUbi=4;
    myPort.write('s');
  } else if((D > D3)){
    persoUbi=0;
    myPort.write('z');
    persona=persona - 1;
  }
} else if(persoUbi==2){
  if(((A > distGen) && (A < A1)) && ((D > D2) && (D < D3))){
    persoUbi=1;
    myPort.write('p');
```

```

}else if((A > A2) && (A < A3) && ((F > F2) && (F < F3))){
  persoUbi=3;
  myPort.write('r');
  persona=persona + 1;
}else if((B > B1) && (B < B2) && ((E > E1) && (E < E2))){
  persoUbi=5;
  myPort.write('t');
}else if((E > E2)){
  persoUbi=0;
  myPort.write('z');
  persona=persona - 1;
}
}else if(persoUbi==3){
  if((A > A1) && (A < A2) && ((E > E2) && (E < E3))){
    persoUbi=2;
    myPort.write('q');
    persona=persona + 1;
  }else if((B > B2) && (B < B3) && ((F > F1) && (F < F2))){
    persoUbi=6;
    myPort.write('u');
  }else if((F > F3)){
    persoUbi=0;
    myPort.write('z');
    persona=persona - 1;
  }
}
}else if(persoUbi==4){
  if((A > distGen) && (A < A1) && ((D > D2) && (D < D3))){
    persoUbi=1;
    myPort.write('p');
  }else if((B > B1) && (B < B2) && ((E > E1) && (E < E2))){
    persoUbi=5;
    myPort.write('t');
  }else if(((C > distGen) && (C < C1) && ((D > distGen) && (D < D1))){
    persoUbi=7;
    myPort.write('v');
  }
}
}else if(persoUbi==5){
  if((A > A1) && (A < A2) && ((E > E2) && (E < E3))){
    persoUbi=2;
    myPort.write('q');
  }else if((B > distGen) && (B < B1) && ((D > D1) && (D < D2))){
    persoUbi=4;
    myPort.write('S');
  }else if(((C > C1) && (C < C2) && ((E > distGen) && (E < E1))){
    persoUbi=8;
    myPort.write('w');
  }else if((B > B2) && (B < B3) && ((F > F1) && (F < F2))){
//-----
    persoUbi=6;
    myPort.write('u');
  }
}
}else if(persoUbi==6){
  if((A > A2) && (A < A3) && ((F > F2) && (F < F3))){
    persoUbi=3;
    myPort.write('r');
  }else if((B > B1) && (B < B2) && ((E > E1) && (E < E2))){
    persoUbi=5;
    myPort.write('t');
  }else if(((C > C2) && (C < C3) && ((F > distGen) && (F < F1))){
    persoUbi=9;
    myPort.write('x');
  }
}

```

```

}
}else if(persoUbi==7){
  if(((B > distGen) && (B < B1) && ((D > D1) && (D < D2))){
    persoUbi=4;
    myPort.write('t');
  }else if((B > B1) && (B < B2) && ((E > E1) && (E < E2))){
    persoUbi=8;
    myPort.write('w');
  }
}
}else if(persoUbi==8){
  if(((B > B1) && (B < B2) && ((E > E1) && (E < E2))){
    persoUbi=5;
    myPort.write('t');
  }else if(((C > distGen) && (C < C1) && ((D > distGen) && (D < D1))){
    persoUbi=7;
    myPort.write('v');
  }else if(((C > C2) && (C < C3) && ((F > distGen) && (F < F1))){
    persoUbi=9;
    myPort.write('x');
  }
}
}else if(persoUbi==9){
  if((B > B2) && (B < B3) && ((F > F1) && (F < F2))){
    persoUbi=6;
    myPort.write('u');
  }else if(((C > C1) && (C < C2) && ((E > distGen) && (E < E1))){
    persoUbi=8;
    myPort.write('w');
  }
}
}
//-----
/*
// FILA A
  if((A > distGen) && (A < A1) && ((D > D2) && (D < D3))){
    persoUbi=1;
  }else if((A > A1) && (A < A2) && ((E > E2) && (E < E3))){
    persoUbi=2;
  }else if((A > A2) && (A < A3) && ((F > F2) && (F < F3))){
    persoUbi=3;
  }else{
    persoUbi=0;
  }
//FILA B
  if((B > distGen) && (B < B1) && ((D > D1) && (D < D2))){
    persoUbi=4;
  }else if((B > B1) && (B < B2) && ((E > E1) && (E < E2))){
    persoUbi=5;
  }else if((B > B2) && (B < B3) && ((F > F1) && (F < F2))){
    persoUbi=6;
  }else{
    persoUbi=0;
  }
//FILA C
  if(((C > distGen) && (C < C1) && ((D > distGen) && (D < D1))){
    persoUbi=7;
  }else if(((C > C1) && (C < C2) && ((E > distGen) && (E < E1))){
    persoUbi=8;
  }else if(((C > C2) && (C < C3) && ((F > distGen) && (F < F1))){
    persoUbi=9;
  }else{
    persoUbi=0;
  }
}
}

```



```

  */
  //-----
  /*
  // RESPUESTA las respuesta se obtiene de adelantes hacia atras
  if(persoUbi==1){
    println("ESTAS EN ESPACIO 1");
    myPort.write(1);
  }
  if(persoUbi==2){
    println("ESTAS EN ESPACIO 2");
    myPort.write(2);
  }
  if (persoUbi==3) {
    println("ESTAS EN ESPACIO 3");
    myPort.write(3);
  }
  if (persoUbi==4){
    println("ESTAS EN ESPACIO 4");
    myPort.write(4);
  }
  if (persoUbi==5){
    println("ESTAS EN ESPACIO 5");
    myPort.write(5);
  }
  if (persoUbi==6){
    println("ESTAS EN ESPACIO 6");
    myPort.write(6);
  }
  if (persoUbi==7){
    println("ESTAS EN ESPACIO 7");
    myPort.write(7);
  }
  if (persoUbi==8){
    println("ESTAS EN ESPACIO 8");
    myPort.write(8);
  }
  */
  println(val);
}

void serialEvent(Serial myPort) {
  // read a byte from the serial port:
  int inByte = myPort.read();
  // if this is the first byte received, and it's an A, clear the serial
  // Otherwise, add the incoming byte to the array:
  if (firstContact == false) {
    if (inByte == 'G') {
      myPort.clear(); // clear the serial port buffer
      firstContact = true; // you've had first contact from the microcontroller
      myPort.write('G'); // ask for more
    }
  }
  else {
    // Add the latest byte from the serial port to array:
    serialInArray[serialCount] = inByte;
    serialCount++;

    if (serialCount > 5 ) {
      firstSensor = serialInArray[0];
      secondSensor = serialInArray[1];
      thirdSensor = serialInArray[2];
      fourthSensor = serialInArray[3];
      fifthSensor = serialInArray[4];
      sixthSensor = serialInArray[5];

      // Send a capital A to request new sensor readings:
      myPort.write('G');
      // Reset serialCount:
      serialCount = 0;
    }
  }
  println(persoUbi);
}

```

PURE DATA (recepción, especialización del sonido)

//no terminado

