

# PAC 4

Borja Javier Lanza López

Anexo, investigación realizada para el proyecto de tesis

# Anexo

---

## Elaboración de Personas y Experimentos

Para las primeras fases del prototipo y poder ir preparando tanto las necesidades y características de la instalación como los requisitos de validación en los experimentos.

**Ana Martín:** 75 años, viuda, vive sola.

A su edad ya empieza a tener dificultades en la vista y en el oído, lo que hace que tenga problemas para identificar objetos o escuchar el timbre de la puerta. No tiene una alfabetización digital.

**Paco Soria:** 69 años, vive con su mujer María de 65.

Paco desgraciadamente empieza a sufrir los primeros estados de Alzheimer por lo que empieza a tener problemas de memoria, al identificar objetos y personas o recordar acciones a realizar. Dependen de su hijo para cualquier cuestión tecnológica.

**Mario López:** 6 años, vive con sus padres pero es algo introvertido, a veces por cuestiones laborales se tiene que quedar en casa y sus padres están preocupados por qué pueda pasar algún accidente doméstico y no sea capaz de contactar con ellos, al no tener vecinos. A pesar de ser un nativo digital, Mario aún no puede hacer acciones complejas por su dificultad y peligro. Esto se complica porque la madre de Mario, Susana tiene ataques epilépticos, que la incapacitan siendo aún Mario demasiado pequeño para hacer algo.

Con estos perfiles creados podemos probar las reacciones ante tres experimentos que permiten comprobar las capacidades del sistema:

La Realidad Aumentada, mostrando información adicional y de manera multimodal a un usuario con deficiencias sensoriales. Cambiando la iluminación, generando alertas audiovisuales en la habitación donde se encuentre el usuario. Incluso transmitiéndolas a un wearable que lleve el usuario. En este caso por ejemplo se puede adoptar a los gustos de quien lo lleva. Para doña Ana Martín es factible crear una mantilla que gracias al cambio del color de los hilos y una pequeña vibración avise de que llaman a la puerta o se ha hecho la comida.

Los objetos inteligentes pueden ayudar a potenciar la memoria y asociar conceptos o personas a dichos elementos. A la vez que ayudamos a los ejercicios mentales que son fundamentales para combatir el progreso de las enfermedades neurodegenerativas.

Un objeto con la foto del hijo de Paco y María, puede ayudar a Paco a realizar acciones que por sus problemas con el Alzheimer le serían más difíciles y frustrantes.

Si acerca el objeto a un lector NFC/RFID este puede proyectar fotos de Paco con su hijo y ayudarle a recordar quien es. Podemos identificar que elementos como canciones o fotografías despiertan su memoria y tienen mejor asociación. Quizás Paco no se acuerde del aspecto de su hijo en el tiempo actual, pero una foto de su comunión esté aun fuertemente arraigada y le ayude acceder a esas memorias veladas. Lo mismo puede pasar con una canción que haya sonado el día de su boda para identificar a María.

Si acercamos el mismo objeto a un teléfono NFC puede llamar al hijo, sin tener que recordar el número, o establecer una videoconferencia.

Los objetos inteligentes también pueden ser usados como localizadores, ayudando a Paco a saber donde están ciertos objetos. Podemos integrar esta característica con un wearable, que ya sea mediante un dispositivo GPS o un sensor de radiofrecuencia, vaya guiando mediante vibraciones o leds en la dirección correcta.

La inteligencia ambiental también puede ayudar a Paco con el mismo propósito, encendiendo y apagando las luces de la habitación donde se encuentre el objeto.

En el caso de Mario podemos experimentar con la realización de acciones complejas, por encima de las capacidades del usuario. En este caso, debido a una emergencia, ya sea un accidente doméstico o un ataque epiléptico de su madre, ya sea por el propio análisis del sistema que detecte un temperatura elevada, humo o un problema con un sensor que tenga la madre, o mediante la invocación del protocolo por el uso de un objeto inteligente por parte de Mario. El sistema AAL es capaz de realizar acciones que van más allá de las capaces por Mario por su corta edad. Llamando a emergencias, contactando con el progenitor disponible, señalando la salida, desconectando la electricidad, desbloqueando una puerta para que se acceda a la vivienda.

## Esquema de una instalación AAL

En estos pasos previos de investigación, hay que elaborar esquemas de la instalación que permitan avanzar que elementos se necesitan. Incluyendo las funciones y eventos que la aplicación tenga que ejecutar y analizar. Los esquemas ayudan a prever las necesidades y posibles problemas a solucionar.

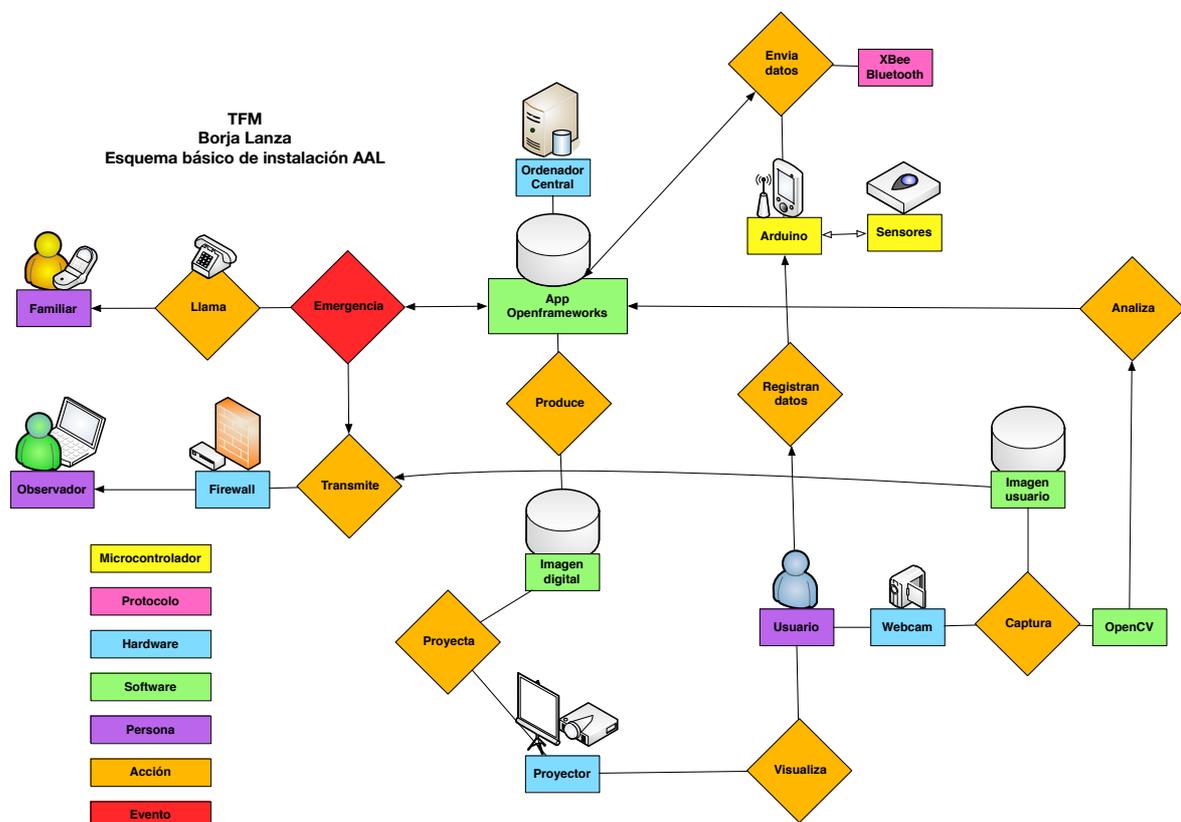


Ilustración 1 - Esquema inicial de una instalación AAL

En el esquema expuesto, tenemos un pequeño flujo, donde la aplicación del sistema AAL, en este caso programada en C++ con las librerías Openframeworks, realiza una serie de acciones.

Desde seguir con openCV las acciones del usuario mediante una webcam, proyectar la imagen sintética que compone la realidad aumentada y hacer un seguimiento de los valores que envían los sensores de un arduino en forma de wearable.

En el caso de que se produzca una emergencia, el sistema llama a un familiar, y posibilita el acceso remoto de un observador de la imagen captada por la webcam.

## **Esquemas elementos electrónicos**

Con la herramienta gratuita Fritzing es posible diseñar los esquemas de conexión de los posibles elementos de la instalación AAL estudiada. Desde wearables hasta objetos inteligentes pasando por bases de sensores. Fritzing muestra no solo la visión con iconos, sino un esquema electrónico y el listado de componentes necesarios.

### **Wearable con Arduino Lilypad**

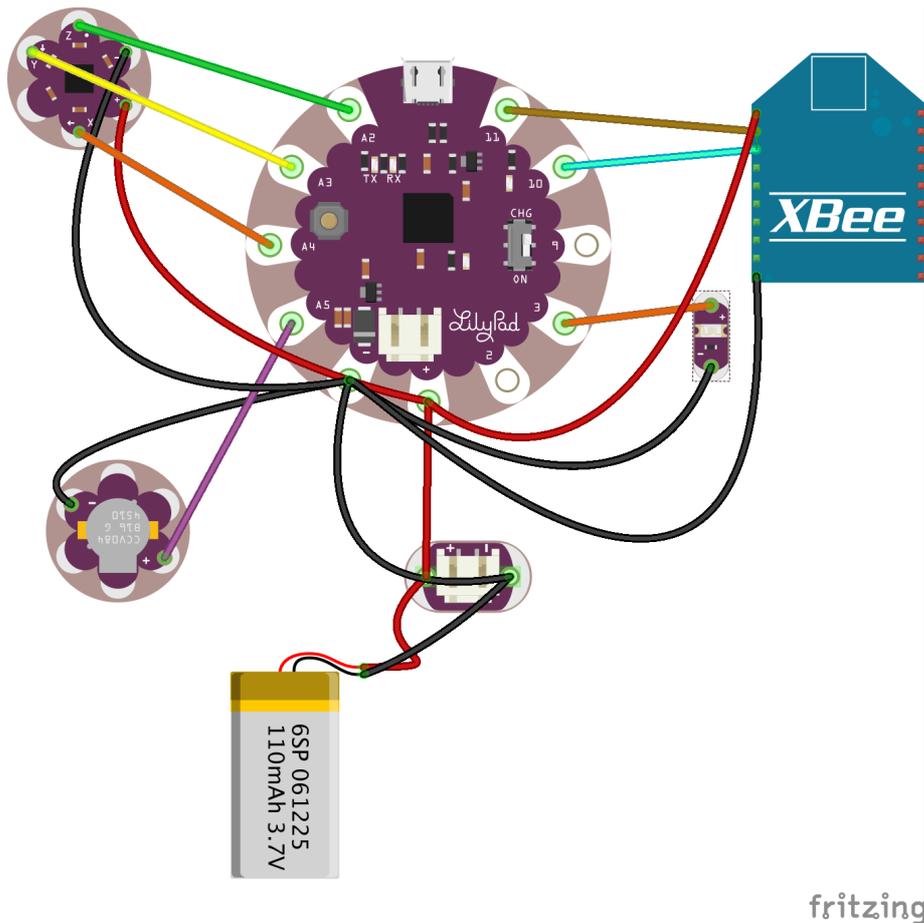


Ilustración 2 - Esquema de un wearable

Para diseñar un wearable, necesitamos un microcontrolador, necesario para las operaciones de computo y controlar las entradas y salidas de datos, para luego ser transmitidas al ordenador central.

En este esquema podemos apreciar una placa Arduino, especialmente diseñada para ser usada en aplicaciones de computadoras vestibles y que es denominada LilyPad. Al ser portable necesita una batería, en este caso aunque ponga 110mAh necesitaríamos una más grande de unos 800-1000 mAh de capacidad.

Complementa el dispositivo, un acelerómetro para conocer la posición y si se experimentan caídas. Para poder interactuar con el usuario el wearable incluye un vibrador (buzzer) y un led que se ilumina parpadeando ante los eventos del sistema.

Para comunicarse inalámbricamente disponemos de un XBee que es quien envía los datos serial por radiofrecuencia.

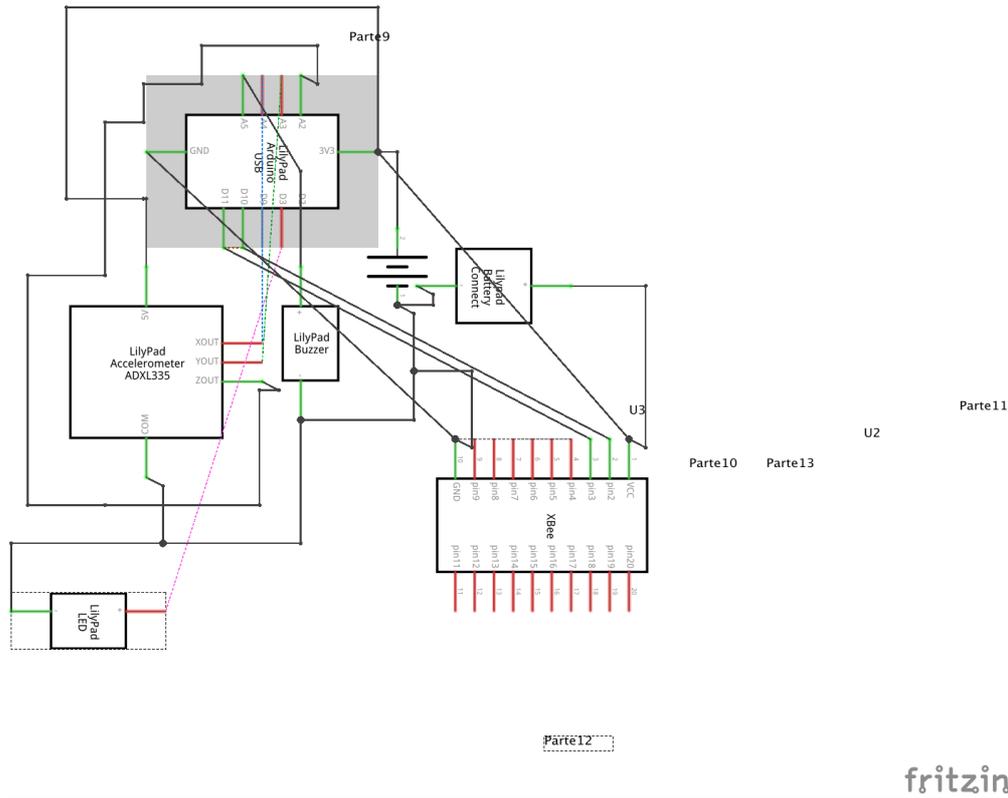


Ilustración 3 - Esquema electrónico de un wearable

fritzing

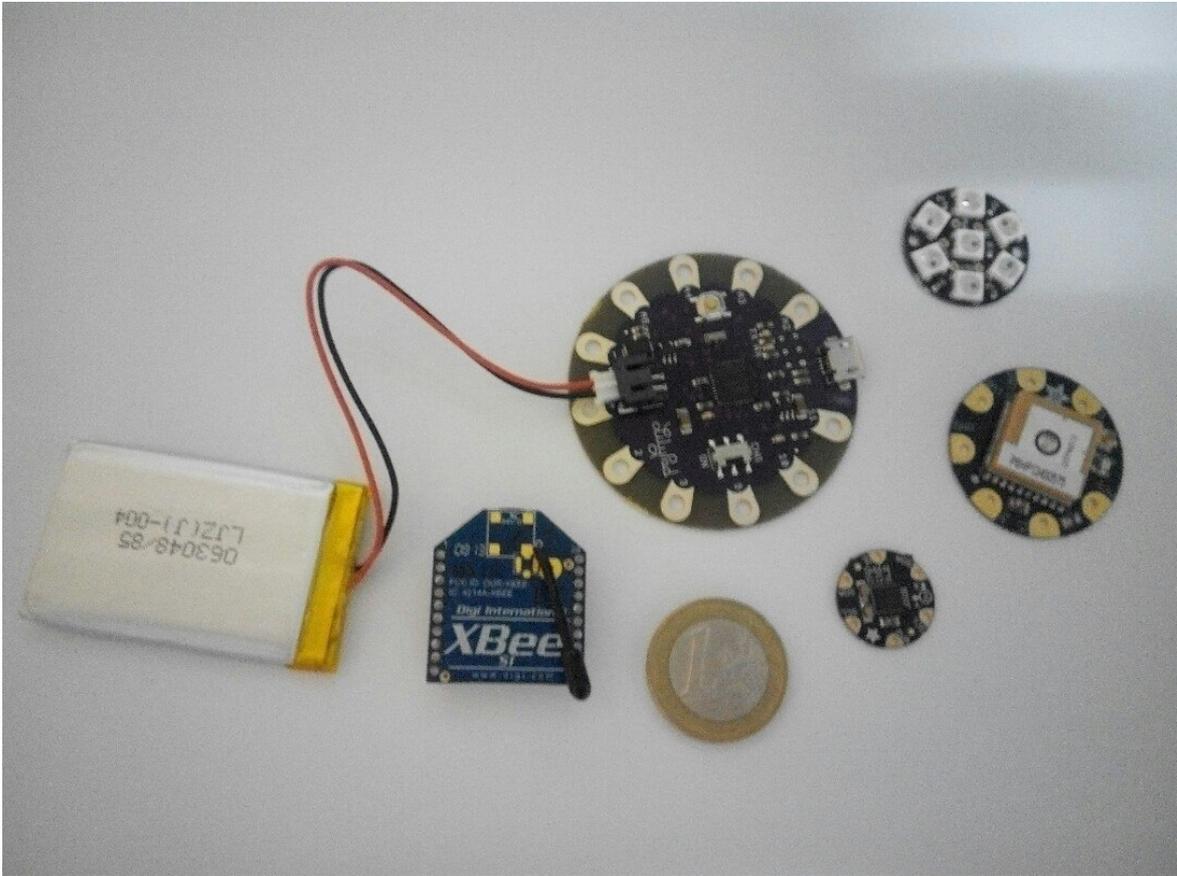
Otra de las ventajas de Fritzing es que genera un listado de las piezas necesarias.

### Assembly List

Label	Part Type	Properties
<b>Parte9</b>	LilyPad Arduino USB	Tipo Lilypad Arduino USB
<b>Parte10</b>	LilyPad Accelerometer ADXL335	interface Analog; Tipo Accelerometer Breakout; range +/- 3g
<b>Parte11</b>	LilyPad Battery Connect	connector JST; Tipo Power Connector
<b>Parte12</b>	LilyPad LED	colors 1; Tipo LED
<b>Parte13</b>	LilyPad Buzzer	Tipo Buzzer
<b>U2</b>	LIPO-110mAh	variant 110mAh; paquete lipo-110
<b>U3</b>	XBee	model XBee ZNet 2.5; pins 20

En la siguiente foto podemos observar una distribución similar, de los componentes con los que se ha estado trabajando. En este caso se añade un GPS, y el sistema de LEDs al ser RGB de 7 leds direccionales poder realizar muchas más acciones.

El XBee sigue presente para la comunicación inalámbrica.

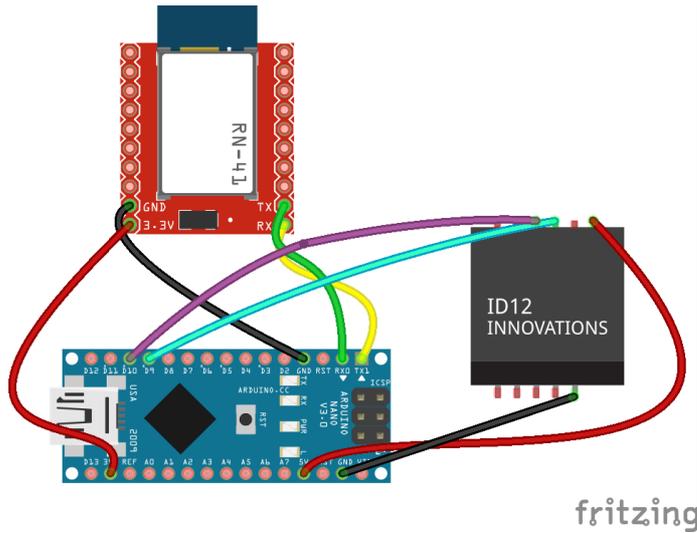


**Ilustración 4 - Piezas necesarias para un Wearable, moneda de un euro para escala**

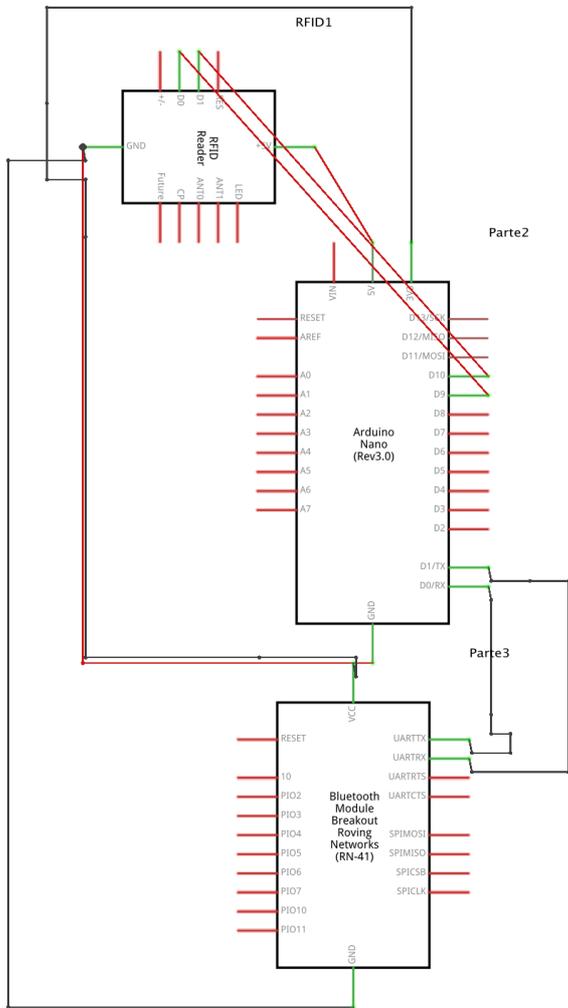
### **Objeto inteligente – IoT (Internet of Things)**

Para crear un objeto inteligente conectado y que en este caso lea etiquetas RFID podemos hacerlo de una manera muy sencilla.

Con un arduino nano, un lector rfid y un módulo bluetooth que permite conectarse a un móvil o un ordenador, ya es posible empezar a procesar objetos marcados con etiquetas NFC/RFID

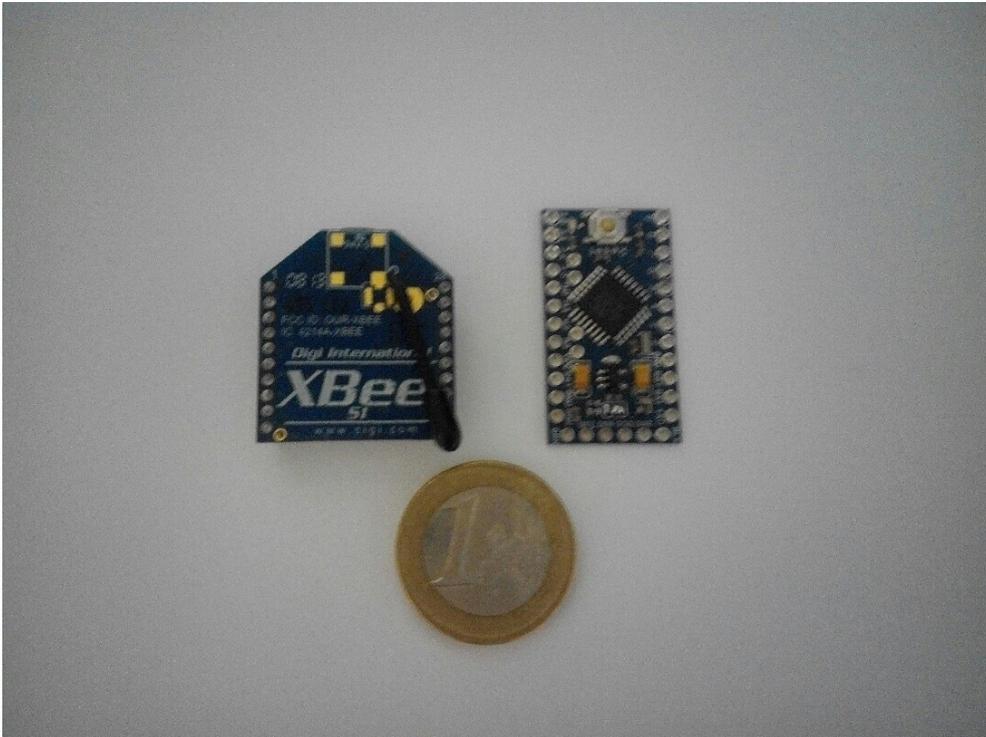


Il·lustració 5 - Esquema de element IoT



Il·lustració 6 - Components de un dispositiu IoT

En las dos siguientes fotografías se observa que con un tamaño reducido es posible dotar a un elemento de capacidades de computo y transmisión, conectándolo al Internet de las Cosas.



**Ilustración 7 - Taamaño de un módulo Xbee y un Arduino Nano**

Las etiquetas NFC/RFID vienen en diferentes tamaños y formatos, desde pequeños encapsulados como los chips de identificación subdermica animal. Hasta las etiquetas pasivas con antena de cobre, antena que también tienen los lectores de rfid para generar un campo electromagnético que sea capaz de leer la información disponible en las etiquetas.

En la foto se aprecia los códigos hexadecimales, que hacen única a cada etiqueta y que podemos capturar y generar eventos que correspondan a cada una de ellas.



Il·lustració 8 - Arduino Nano, lector RFID y diferentes tipos de etiquetas NFC/RFID

En el caso de instalaciones con mayor demanda de características y que además sean fijas, se opta por usar de arduinos clásicos con mayor tamaño pero mejores prestaciones. Facilitando la incorporación de otros sensores y elementos gracias al uso de shields. En la foto siguiente se puede observar un Arduino, en este caso de la compañía Sparkfun, con un shield Adafruit NFC.



Il·lustració 9 - Arduino con shield NFC

## Estaciones de sensores

Para dotar de inteligencia ambiental al sistema AAL, es necesario tener diferentes nodos de sensores que suministran información para su proceso tanto por el código interno del micro controlador como del sistema central.

Un ejemplo perfectamente válido es el mostrado a continuación. Donde un arduino ya dotado de capacidades de transmisión bluetooth, tiene conectados dos sensores multifunción. Uno se encarga de controlar la humedad y temperatura, algunos modelos incluso registran también la presión atmosférica. El segundo es un sensor que analiza las combinaciones de varios gases que pueden ser peligrosos para las personas, como el dióxido de carbono.

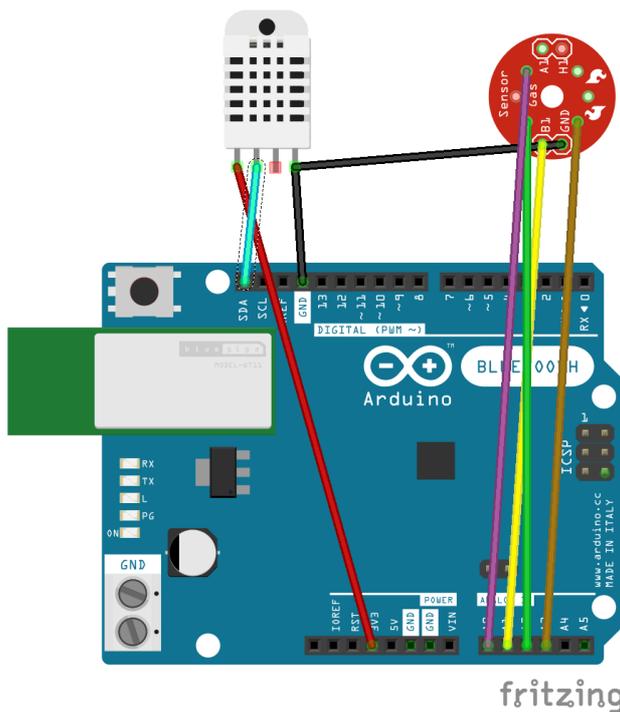


Ilustración 10 - Arduino BT con sensores ambientales

## Realidad Aumentada

En estas primeras etapas de investigación, se ha decidido que la realidad aumentada se haga mediante el programa Video Projection Tool (<https://hcgilje.wordpress.com/vpt/>). Programa gratuito que es un aplicación generada en Max/MSP y que acepta comandos mediante OSC.

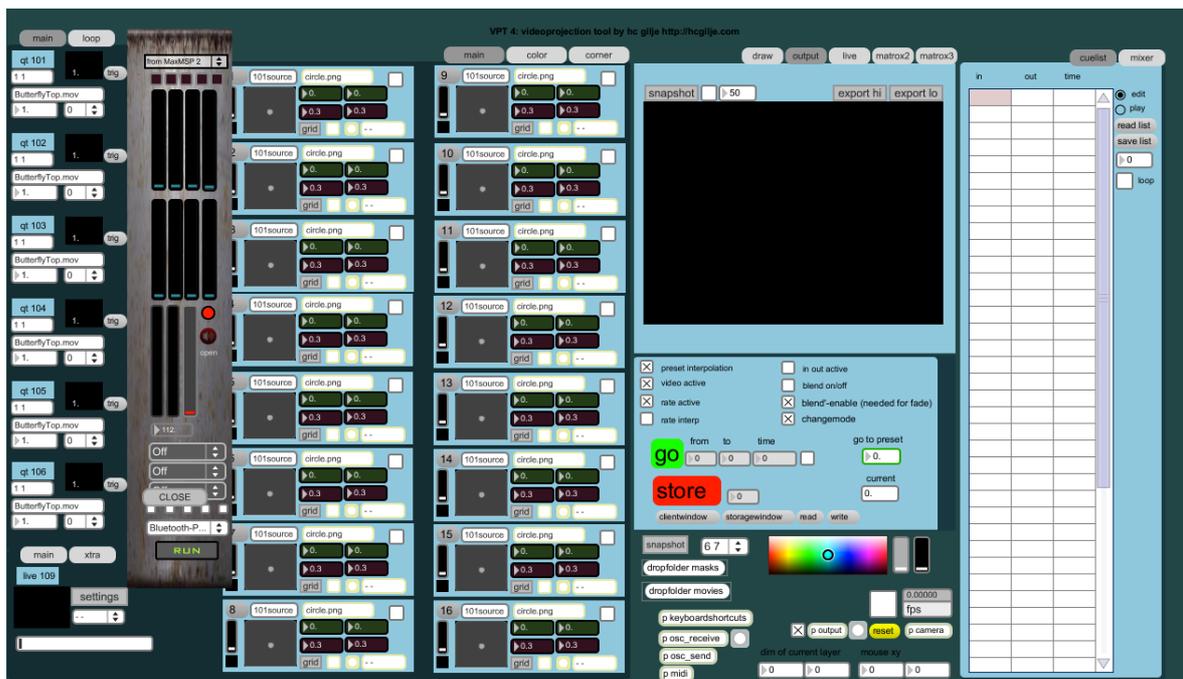


Ilustración 11 - Aplicación VPT

La aplicación tiene diferentes capas, que se pueden mapear para que la proyección de la imagen digital se ajuste al entorno físico. Es capaz de mostrar tanto videos como imágenes, junto a listas de reproducción de dicho contenido.

Para proyectar, es factible que por su tamaño y potencia sea suficiente con pico-proyectores, creando pequeñas superficies de realidad aumentada.



**Ilustración 12 - Pico proyector**

Otra solución sería un proyector clásico que cubra toda una pared, o quizás para algunas aplicaciones láseres con el protocolo ILDA, que permiten crear imágenes vectoriales.

## **Código de comunicación entre aplicaciones (Serial – OSC)**

Ya que no se ha programado ninguna aplicación los fragmentos de código son escasos. Sin embargo es factible ver como se transmiten los datos entre aplicaciones. Ya sea mediante el protocolo serial, que es mediante el cual los microcontroladores envían los datos de los sensores, estos son transmitidos mediante radiofrecuencia o bluetooth pero siempre mediante serial.

## Código arduino

```
void loop()
{
  // if we get a valid byte, read analog ins:
  if (Serial.available() > 0) {
    // get incoming byte:
    inByte = Serial.read();
    // read first analog input:
    firstSensor = analogRead(A0);
    // read second analog input:
    secondSensor = analogRead(A1);
    // read switch, map it to 0 or 255L
    thirdSensor = map(digitalRead(2), 0, 1, 0, 255);
    // send sensor values:
    Serial.print(firstSensor);
    Serial.print(",");
    Serial.print(secondSensor);
    Serial.print(",");
    Serial.println(thirdSensor);
  }
}
```

Esté código de ejemplo, es el que envía tras recibir un comando, los valores de sus sensores con la función Serial.print();

Estos datos pueden ser recibidos en OpenFrameworks

```
// Read all bytes from the device;
uint8_t buffer[16];

while (device.available() > 0)
{
  std::size_t sz = device.readBytes(buffer, 16);

  for (int i = 0; i < sz; ++i)
  {
    std::cout << buffer[i];
    mensaje[i] =buffer[i];

    /*if (buffer[i]=='\n') {
      std::cout <<"Fin De línea";
    }*/
  }
  //std::cout << mensaje[1];

  if (mensaje[0]=='F') {
    std::cout <<" Se ha declarado un Fuego";
  }
  else if (mensaje[0]=='C') {
    std::cout <<" La cena esta lista";
  }
  else if (mensaje[0]=='R') {
    std::cout <<" El sistema esta listo";
  }
  device.flush();
}
```

El código esquemáticamente espera a que haya datos en el buffer del protocolo serie. Según el primer digito del mensaje que acaba cuando detecta el carácter de

nueva línea, realiza una acción dependiendo de la valoración mediante un bucle if.

Según el bucle if podemos emitir otros mensajes por el mismo protocolo serie, que serán escuchados por otros dispositivos, por ejemplo conectados a un Xbee. Con una función en C++ que acepte los siguientes parámetros podemos tanto enviar datos a otros dispositivos como cambiar las imágenes de la aplicación VTP que es la que se encarga de la realidad aumentada

```
void testApp::setEvent(string event, Byte evento_serial, int ard_pin)
```

```
if (evento_serial == 'F') {  
    mensa[0] = 0x41;  
    serial.writeBytes(mensa, 8);  
    serial_Xbee.writeByte('B');  
}  
else if (evento_serial == 'C') {  
    mensa[0] = 0x5a;  
    serial.writeBytes(mensa, 8);  
    serial_Xbee.writeByte('Z');  
}  
}
```

El lenguaje OSC establece rutas, que son las que admiten como comandos los programas. En el siguiente código de la misma función anterior, según el evento detectado, enviamos un comando a otro ordenador en este caso un iMac, que realizará otras acciones, y a dos ordenadores que en este caso estarían en dos salas diferentes, que cambien la fuente de su imagen para mostrar la correspondiente al evento detectado.

```
/Enviar Mensajes OSC
```

```
//Eventos iMac  
ofxOscMessage msg;  
msg.setAddress("/evento/e"+event);  
osciMac.sendMessage(msg);  
  
//VideoMapping VTP  
ofxOscMessage m_VTP;  
m_VTP.setAddress("/sources/1video/clip");  
m_VTP.addStringArg("portrait_"+event+".jpg");  
oscSender.sendMessage(m_VTP);  
  
ofxOscMessage m_VTP2;  
m_VTP2.setAddress("/sources/7video/clip");  
m_VTP2.addStringArg("avatar_"+event+".png");  
oscSender.sendMessage(m_VTP2);
```

Can parte de la aplicación será gestionar los eventos producidos y ya sea mediante bucles condicionales if o case ir llamando a las diferentes funciones.

## **Código RFID – NFC**

```
uint8_t success;
uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 };
uint8_t data[] = { 0, 0, 0, 0, 0, 0 }; // Buffer to store the returned UID
uint8_t uidLength; // Length of the UID (4 or 7 bytes depending on ISO14443A card type)

// Wait for an ISO14443A type cards (Mifare, etc.). When one is found
// 'uid' will be populated with the UID, and uidLength will indicate
// if the uid is 4 bytes (Mifare Classic) or 7 bytes (Mifare Ultralight)
success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uidLength);

if (success) {
  // Display some basic information about the card
  //Serial.println("Found an ISO14443A card");
  //Serial.print(" UID Length: ");Serial.print(uidLength, DEC);Serial.println(" bytes");
  //Serial.print(" UID Value: ");
  //nfc.PrintHex(uid, uidLength);
  //Serial.println("");

  data[0] = uid[0];
  //Serial.println("Data serial O");
  //Serial.println(data[0],DEC);

  if (data[0] ==18){
    //Serial.println("Apagar Luces");
    Serial.write("Ze\n");
  }
  else if (data[0] ==130){
    //Serial.println("Llamar Juan");
    Serial.write("Rs\n");
  }
}
```

En este código destinado a un arduino con un shield NFC como el mostrado de la compañía Adafruit podemos enviar datos serial según la tarjeta detectada. En este ejemplo ya que el código del primer dígito hexadecimal es diferente a cada una de ellas no hace falta comparar más dígitos.