Ra Ximhai
ISSN: 1665-0441
raximhai@uaim.edu.mx
Universidad Autónoma Indígena de México
México

Chan-Canche, Abraham Obed; Díaz-Rodríguez, Miriam
SISTEMA DISTRIBUIDO PARA LA INTERACCIÓN HOMBRE MÁQUINA EN AMBIENTES VIRTUALES
Ra Ximhai, vol. 13, núm. 3, julio-diciembre, 2017, pp. 107-122
Universidad Autónoma Indígena de México
El Fuerte, México

Disponible en: http://www.redalyc.org/articulo.oa?id=46154070007

RA XIMHAI ISSN 1665-0441
Volumen 13 Número 3 Edición Especial
Julio-Diciembre 2017
107-122

SISTEMA DISTRIBUIDO PARA LA INTERACCIÓN HOMBRE MÁQUINA EN AMBIENTES VIRTUALES

DISTRIBUTED SYSTEM FOR HUMAN MACHINE INTERACTION IN VIRTUAL ENVIRONMENTS

Abraham Obed **Chan-Canche**[1] y Miriam **Díaz-Rodríguez**[2]
[1]Egresado de Maestría en Sistemas Computacionales. Instituto Superior de Jalisco José Mario Molina Pasquel y Henríquez Unidad Académica Zapopan. [2]Profesora de tiempo completo Titular A. Departamento de Investigación y Posgrados. Instituto Superior de Jalisco José Mario Molina Pasquel y Henríquez Unidad Académica Zapopan.

RESUMEN
La existencia de redes de comunicación conformadas por múltiples dispositivos y sensores se está volviendo cada vez más frecuente. Estas redes de dispositivos interconectados entre sí, permiten desarrollar una interacción hombre máquina la cual procura mejorar el desempeño de la persona generando un entorno adaptativo en respuesta a la información que el mismo entorno emite. La problemática de este trabajo es la integración ágil de una red de dispositivos que permita desarrollar un entorno inmersivo flexible a distintos casos de uso.
Palabras claves: red de dispositivos, gestor de comandos, sockets de red, disparadores.

SUMMARY
The communication networks built by multiple devices and sensors are becoming more frequent. These device networks allow human-machine interaction development which aims to improve the human performance generating an adaptive environment in response to the information provided by it. The problem of this work is the quick integration of a device network that allows the development of a flexible immersive environment for different uses.
Keywords: network devices, manager command, network sockets, triggers.

INTRODUCCIÓN

La interconexión de múltiples dispositivos es algo común hoy en día; la existencia de redes de comunicación conformadas por múltiples sistemas y protocolos se están volviendo cada vez más frecuentes. El Internet crece de manera exponencial y la cantidad de dispositivos personales interconectados también (Zhang *et al.* 2008); tal es el ejemplo de los teléfonos inteligentes y las tabletas, los cuales han presentado tal crecimiento que desde 2010 el número de dispositivos conectados es superior a los habitantes del planeta (Evans, 2011).

Dado a que muchos dispositivos cuentan con la capacidad de interconectarse entre sí, se puede establecer un sistema de comunicación entre ellos, a tal grado de que el conjunto de dispositivos en una misma red definen un concepto actual llamado Internet de las cosas (en inglés, Internet of things, IoT). El IoT aprovecha cada una de las características de comunicación de los distintos dispositivos, lo cual permite la interacción entre ellos, alcanzando distintos objetivos en base a la información que comparten entre sí.

Gracias al conjunto de información que se comparte en una red se puede desarrollar un entorno en donde brinde una interacción entre el usuario y los dispositivos por medio de la sincronización de las acciones de los dispositivos que se encuentran conectados, este resultado nos lleva al concepto de entornos inteligentes (smart environments).

Recibido: 26 de febrero de 2017. Aceptado: 15 de mayo de 2017.
Publicado como ARTÍCULO CIENTÍFICO en Ra Ximhai 13(3): 107-122.

Para aprovechar toda la estructura de un entorno inteligente, se integra un proceso de minería de datos en donde, por medio de cálculos estadísticos y proyección futura, se pueda entender el ambiente que rodea al usuario y predecir ciertas situaciones y acciones en base al estado actual del entorno leído por los sensores y la proyección del trabajo de minería.

Este trabajo tiene como objetivo el integrar una metodología de implementación que permita la construcción rápida y funcional de un entorno inmersivo por medio de la capacidad de comunicación que tienen ciertos dispositivos y la gestión de la información que estos pueden compartir dentro de la red.

MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN

Para la integración del trabajo, se investigaron varias referencias de trabajos relacionados, entre los cuales destaca, en el área de IoT y entornos inteligentes, el MavHome (Cook D. J., *et al.*, 2004), que es un proyecto que desarrolla un agente inteligente y lo integra a la estructura arquitectónica de una casa. Este trabajo busca maximizar la comunidad y la productividad de los habitantes de la casa disminuyendo el costo de operación.

Otros trabajos relacionados, dentro del conjunto de redes de sensores, se encuentran los proyectos de agricultura de precisión, en donde se implementan nodos sensores del ambiente y un nodo central de gestión y almacenamiento, permitiendo la toma de decisiones en base a la información recabada y analizada (López J. A., *et al.*, 2011).

Tomando en cuenta el desarrollo y las problemáticas de estos trabajos relacionados, se definieron los elementos de hardware y de software necesarios para el desarrollo de este proyecto; algunos de estos elementos son los siguientes:

- *Dispositivo Intel® Galileo*
  El dispositivo Intel® Galileo es una placa de desarrollo, pertenecientes a la familia de Arduino, que cuentan con el conjunto de puertos GPIO (General Purpose Input/Output [Entrada/Salida de Propósito General]) compatibles con Arduino, con un procesador Intel R QuarkTM SoC X1000 y una tarjeta de red integrada, entre otras cosas. Este dispositivo es usado como un nodo sensor y actuador al mismo tiempo, dado a la capacidad de integrar sensores por medio de los puertos análogos y emitir salidas digitales por los puertos correspondientes.
- *Entorno de desarrollo Unity*
  Unity se define a sí mismo como una solución ya preparada cuyo uso también es intuitivo y altamente personalizable. Con el poder de renderizado, y shading de base física altamente optimizado, se pueden crear juegos con suma rapidez. Este entorno de desarrollo de videojuegos permite realizar una integración rápida de un entorno virtual entregando así la creación rápida de una aplicación simple pero con una presentación de calidad la cual se integra en el proyecto.
- *Socket de red*
  En TCP/IP, antes de poder realizar cualquier tipo de comunicación es necesario habilitar un socket y después realizar las demás tareas de comunicación. Cuando se crea un socket se debe especificar primero una familia de protocolos, así como el tipo de servicio requerido, si se trata de un servicio orientado a conexión o no. Finalmente se debe especificar qué protocolo se utilizará para dar este servicio. En TCP/IP para servicios orientados a conexión se utiliza el TCP y para servicios no orientados a conexión se utiliza el UDP. El desarrollo de los sockets de red dentro del proyecto establece la parte

del trabajo de comunicación con los dispositivos, para generar la capacidad de envío de datos entre todos.

Tomando en cuenta los trabajos relacionados y aclarada información de ciertos elementos ocupados para la construcción de una red de dispositivos y la integración en un ambiente virtual, se procede a detallar la integración realizada de los dispositivos para conformar una red de sensores acorde al objetivo del proyecto.

Para lograr la integración de la red, se seleccionó un conjunto de dispositivos capaces de comunicarse entre sí, a los cuales se les configuró un lenguaje de instrucciones simples y un sistema de mensajes de respuesta a dichas instrucciones. Además de esto, tales dispositivos fueron asignados en dos tipos de actividades diferentes, dependiendo de su configuración interna

- *Los dispositivos actuadores (nodo esclavo),* que cuentan con un conjunto de instrucciones de entrada, un conjunto de acciones en base a las instrucciones y ciertos mensajes de respuesta.
- *El dispositivo gestor de comandos (nodo central)* el cual se encarga de administrar las instrucciones del usuario e invocar los comandos de los dispositivos, almacenar la información de lectura y entregar un estado de la red de dispositivos al usuario.

Integración del dispositivo actuador (nodo esclavo)

Como se ha mencionado antes, es necesario que el nodo esclavo cuente con un conjunto de instrucciones y tenga la capacidad de emitir una respuesta a dichas instrucciones, ya sea una actividad de lectura o una actividad de salida, por lo que dentro del nodo esclavo se integra un intérprete de instrucciones en base a la definición de ciertos comando básicos que se establecieron por parte del desarrollador del nodo. En la *Figura 1* podemos observar un diagrama simple de un nodo esclavo.

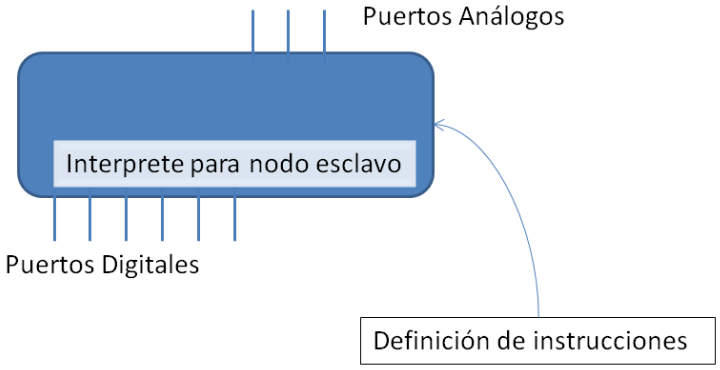

Figura 1. Nodo esclavo.

Para el desarrollo e integración del nodo esclavo, se seleccionó la tarjeta de desarrollo Intel® Galileo ya que cuenta con una interfaz de red que puede establecer comunicación por medio de los protocolos de red comunes, como TCP/IP. Así también debido al conjunto de puertos GPIO compatible con Arduino se puede integrar un desarrollo de dispositivos de manera rápida, bajo las herramientas que Intel® y Arduino proveen.

Configuración de Interfaz Arduino

En este trabajo se optó por el uso de la interfaz Arduino de la tarjeta, por lo que para la integración y uso del dispositivo Intel® Galileo como una interfaz Arduino se realizó una actualización de firmware. Dicha actualización se llevó a cabo con la herramienta de desarrollo de Sketchs Arduino que Intel® provee. Después de esta actualización, la tarjeta verifica si cuenta con una unidad de almacenamiento (tarjeta micro SD) y habilita todas las funciones que provee la interfaz Arduino.

Configuración de interfaz de red

Una vez configurada la interfaz Arduino, es necesario establecer las configuraciones de conectividad en el dispositivo. Una de las ventajas que nos proporciona el dispositivo Intel® Galileo es que no requiere la interfaz SPI (Serial Pherical Interface) como el común de los Shields de Arduino, esto nos permite realizar una configuración directa de la tarjeta de red, ligando la dirección física del dispositivo con la dirección IP de la red. En el *Cuadro 1* se describen los parámetros de uno de los dispositivos que integran a la red.

Cuadro 1. Parámetros de configuración de red

Propiedad	Valor
Dirección Física	0x98 0x4F 0xEE 0x00 0xE5 0x1D
Dirección IP	192.168.1.30
Máscara de Red	255.255.255.0
Puerta de Enlace	192.168.1.1

Para poder habilitar la configuración de red en el dispositivo Intel® Galileo se utiliza la biblioteca <Ethernet.h> la cual permite establecer la definición de un servidor de red por medio de la asignación de un puerto de entrada. En este caso se define el servidor de red con puerto 1056 para la entrada de datos, dado a que este puerto se encuentra libre. Esta configuración es necesaria para que, en base al estado del servidor en la red, se habilite un cliente de red en cada dispositivo permitiendo recibir las entradas y enviar las salidas dentro de la red.

Configuración de los GPIO con interfaz Arduino

Puesto que previamente de configuro el dispositivo Intel® Galileo como una interfaz Arduino, los puertos GPIO toman las mismas características de Arduino, por lo se cuenta con un conjunto de 14 pines de entrada y salida digital y otros 6 más de entrada análoga. Como se observa en el *Cuadro 2*, se definieron los pines para la salida digital y otros para la lectura análoga.

Cuadro 2. Configuración de puertos del dispositivo Intel® Galileo

Puerto	Tipo	Configuración
9	Digital	Output
10	Digital	Output
1	Análogo	Input

Definición de conjunto de instrucciones

En base a la configuración del comportamiento de los pines se estableció un algoritmo de interpretación de instrucciones para la habilitación de la salida del pin digital o para la lectura del pin análogo en base a instrucciones simples. El *Cuadro 3* contiene la definición de dos comandos, uno de lectura del pin análogo y otro de salida alta o baja del pin digital. En el *Cuadro 4* se pueden apreciar algunos ejemplos de los comandos completamente definidos y la salida esperada del dispositivo Intel® Galileo.

Cuadro 3. Comandos simples

Tipo	Palabra Clave	Parámetros	Salida
Lectura	LEC		Valor de Lectura Análoga | Mensaje: Error de Lectura
Salida	PIN	[H|L],N	Salida digital del pin N, en base al parámetro H|L

Cuadro 4. Ejemplo de instrucciones básicas

Comando	**Salida**
LEC	Lectura análoga del pin 1
PINH13	Salida de encendido del pin 13
PINH9	Salida de encendido del pin 9
PINL12	Salida de apagado del pin 12
PINL13	Salida de apagado del pin 13

Dicho algoritmo de interpretación establece un ciclo de lectura en base a la disponibilidad del cliente en red y la secuencia de los paquetes entrantes para que cuando la entrada contenga una instrucción bien definida, se interprete y ejecute la acción de dicha instrucción. El Algoritmo 1 muestra el proceso de interpretación en donde se observa el ciclo de lectura de paquetes, la verificación del comando válido y la ejecución de la salida en base a los parámetros.

Cuadro 5. Algoritmo 1

Algoritmo 1: Ciclo de lectura

```
Entrada: Cadena = "", C = cliente de red
Salida:
1:      Mientras C está conectado hacer
2:          Si C está disponible entonces
3:              S = Lee cadena del cliente
4:              CAD += S
5:              Si CAD inicia con "PIN" entonces
6:                  ACCIÓN = cuarto carácter de CAD
7:                  Si ACCIÓN = "H" entonces
8:                      Si longitud de CAD >= 4 entonces
9:                          LED = Subcadena a partir del 4 símbolo de CAD
10:                         Salida digital Alta en LED
11:                     Fin Si
12:                 Fin Si
13:                 Si ACCIÓN = "L" entonces
14:                     Si longitud de CAD >= 4 entonces
15:                         LED = Subcadena a partir del 4 símbolo de CAD
```

```
16:                                                    Salida digital Baja en LED
17:                                                Fin Si
18:                                            Fin Si
19:                                        Fin Si
20:                                        Si CAD inicia con “LEC” entonces
21:                                            V = 25 * lectura análoga del pin 1 / 1023
22:                                            Si V > 2.5 O V < 0.25 entonces
23:                                                Mensaje al servidor “Error”
24:                                            Sí no
25:                                                Mensaje al servidor V
26:                                            Fin Si
27:                                        Fin Si
28:                                Fin Si
29:                        Fin Mientras
```

Con la definición e interpretación de estas dos instrucciones simples en el nodo esclavo, se realizó una replicación de varios dispositivos Intel Galileo, conformando así la red de sensores, sin embargo dichos sensores son gestionados por un nodo central el cual configuramos de la siguiente manera.

Integración del dispositivo gestor de comandos (nodo central)
El nodo central funciona como gestor de comunicación entre los comandos del usuario y los dispositivos en la red, además de que se encarga de leer y almacenar los datos recolectados por los sensores y depositarlos en un repositorio a fin de aprovecharlos para cálculos estadísticos y predicción de estados futuros del entorno. Este nodo cuenta con un intérprete de instrucciones enfocadas a entregárselas al usuario a fin de que por medio de ellas pueda hacer uso de los nodos esclavos que conforman la red. En la *Figura 2* podemos observar la correlación entre el usuario y los dispositivos a través del nodo central.

El nodo central cuenta con dos procesos que se ejecutan de manera simultánea:

- *El proceso de comunicación* con el usuario y los dispositivos el cual toma los parámetros de conexión del dispositivo y habilita un puerto de entrada para la recepción de los datos que envía tanto el usuario como el dispositivo sensor.
- *El proceso de lecturas y almacenamiento* de los datos censados. El cual proporciona modularidad a la red, la cual de manera autónoma se encuentra almacenando una imagen del entorno de implementación y entrega de manera automática una base de datos completamente consumible para aplicaciones externas.

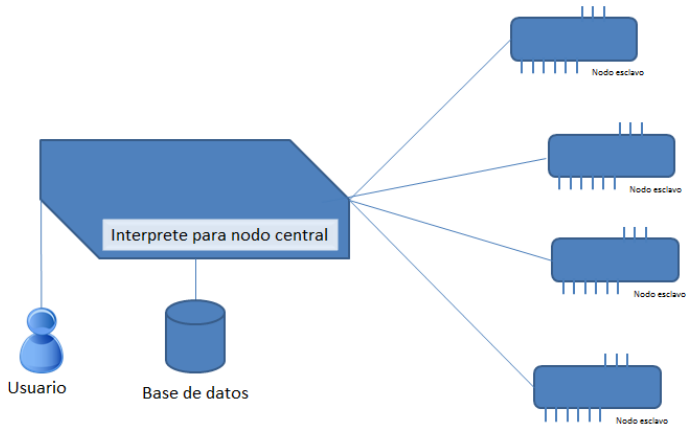

Figura 2. Nodo central en la red de dispositivos.

Proceso de comunicación

Al igual que el nodo esclavo, en el nodo central se estableció un algoritmo de interpretación de instrucciones simples enfocadas al usuario, esto es para facilitar la comunicación entre el usuario y los dispositivos sin que el usuario tenga la necesidad de reaprender las instrucciones de cada uno de los dispositivos esclavos. Estas instrucciones simples se aprecia en el *Cuadro 6*, en donde se observan, tres comandos básicos: la obtención de la lista de dispositivos que se encuentran en la red, el envío de comando a un solo dispositivo y el envío de comando a todos los dispositivos. Cada instrucción se define por una palabra de control y un conjunto de parámetros.

Cuadro 6. Instrucciones para el usuario

Nombre	Comando	Parámetros
Discover devices	DD	
Send instruction	SI	-d IP del dispositivo -p pin a utilizar -o [H|L] salida del pin para Alto voltaje: H o para bajo voltaje L
Send All	SA	-p pin a utilizar -o [H|L] salida del pin para Alto voltaje: H o para bajo voltaje L

Como se muestra en el Algoritmo 2, el proceso de interpretación es una capa intermedia entre que gestiona el envío de comando del usuario y lo transforma al comando del dispositivo.

Cuadro7. Algoritmo 2

Algoritmo 2: Proceso de comunicación

Entrada: ENTRADA, SALIDA, CONTADOR
Salida:

```
Crea el conjunto de COMANDOS
Mientras Entrada != Nulo hacer
CONTADOR++
ENTRADA = ENTRADA reemplaza espacio por carácter vació
COMARR = ENTRADA transforma a arreglo por separador "-"
Si COMARR[0] = "SI" entonces
                    Si Tamaño de COMARR = 4 entonces
                    Si COMMARR[1] = "d" Y COMMARR[2] = "p" Y COMMARR[3] = "o" entonces
          COMANDOS.SI(COMMARR[1], "PIN"+COMMARR[2]+COMMARR[3])
          Si no
                    SALIDA imprime "Error en estructura"
          Fin Si
Si no
                    SALIDA imprime "Parámetros insuficientes"
Fin Si
Si no Si COMARR[0] = "DD" entonces
                    DISPOSTIVOS = COMANDOS.DD
          Para i hasta  DISPOSTIVOS Tamaño Siguiente
SALIDA imprime "Dispositivo disponible:" + DISPOSITIVOS[i]
```

```
20:     Fin Para
21:     Si no Si COMARR[0] = "SA" entonces
22:                     Si tamaño de COMARR = 3 entonces
23:                     Si COMMARR[1] = "p" Y COMMARR[2] = "o"  entonces
24:     Para i hasta  Tamaño de DISPOSTIVOS Siguiente
25:                             COMANDOS.SI(DISPOSITIVOS[i],"PIN"+ COMMARR[1]+COMMARR[2])
26:                             SALIDA imprime  "Envío global"
27:                     Si no
28:                             SALIDA imprime "Error en estructura"
29:                             Fin Si
30:             Si no
31:                             SALIDA imprime "parámetros insuficientes"
32:                     Fin Si
33:     Fin Si
34:     Si ENTRADA = "exit" entonces
35:             SALIDA = true
36:     Forzar Salida de Mientras
37:     Fin Si
38:     Fin Mientras
```

Los Algoritmos 3 y 4 definen la estructura del conjunto de comandos utilizados en COMANDO del Algoritmo 3

Cuadro 8. Algoritmo 3

Algoritmo 3: Comando SI

Entrada: IP, ACCIÓN

Salida: MENSAJE

```
1:      MENSAJE = ""
2:      Intenta
3:      CONEXIÓN = IP, 1056
4:      SALIDA = Controlador de salida de CONEXIÓN
5:      ENTRADA = Controlador de lectura de entrada de CONEXIÓN
6:              SALIDA envía ACCION
7:              MENSAJE = ENTRADA realiza lectura
8:      Atrapa Excepción de anfitrión remoto desconocido
9:      MENSAJE = "No sabemos acerca del anfitrión" + IP
10:     Atrapa Excepción de entrada y salida
11:     MENSAJE = "No podemos establecer conexión de entrada y salida de " + IP
12:     Fin Intenta
13:     Devuelve MENSAJE
```

Cuadro 9. Algoritmo 4

Algoritmo 4: Comando DD

```
Entrada:
Salida: DISPOSITIVOS
1:    DISPOSITIVOS = Lista vacía de cadenas
2:    Para i de 1 hasta 254  siguiente
3:    ANFITRION = "192.168.1." + i
4:    DIRECCION
5:        Intenta
6:            DIRECCION = Obtiene dirección por nombre (ANFITRION)
7:            Si DIRECCION es accesible entonces
8:                Agrega dirección a DISPOSITIVOS
9:            Fin Si
10:       Atrapa Excepción de anfitrión remoto desconocido
11:       Imprime trazado de pila
12:       Atrapa Excepción de entrada y salida
13:       Imprime trazado de pila
14:       Fin Intenta
16:   Fin Para
17:   Devuelve DISPOSITIVOS
```

Proceso de Lectura y almacenamiento

Junto con el proceso de comunicación se ejecuta el proceso de lectura y almacenamiento el cual establece la comunicación con los dispositivos por medio del puerto de comunicación configurado previamente.

Los algoritmos 5, 6 y 7 muestran, en conjunto, la forma en la que funciona el proceso de lectura y almacenamiento. Cabe mencionar que en esta implementación se define una base de datos MYSQL la cual cuenta con una tabla que identifica el dispositivo, la fecha y hora de lectura y el valor leído.

Cuadro 10. Algoritmo 5

Algoritmo 5: Proceso de Lectura y Almacenamiento

```
Entrada: IP, ACCION, BANDERA
Salida:
1:    Mientras BANDERA hacer
2:    Intenta
3:        Lectura de sensor (IP, ACCION)
4:        Espera 500 milisegundos
5:    Atrapa Excepción de entrada y salida
6:        Imprime pila de instrucciones
7:    Fin Intenta
```

El Algoritmo 5 es la definición del hilo el cual realiza una lectura cada medio segundo, el proceso de lectura (Algoritmo 6) genera la comunicación con el dispositivo e invoca al proceso de almacenamiento (Algoritmo 7).

Cuadro 11. Algoritmo 6

Algoritmo 6: Lectura de sensor	
Entrada: IP; ACCION	
Salida:	
1:	Lanza excepción en base a Error en formato de Número
2:	Crea el conjunto de COMANDOS
3:	DATA = COMANDOS.SI(IP,ACCIÓN)
4:	Almacenamiento(DATA)

Cuadro 12. Algoritmo 7

Algoritmo 7: Almacenamiento	
Entrada: DATA	
Salida:	
1:	Lanza excepción en base a Error en formato de Número
2:	Crea el controlador de base de datos DB
3:	Conecta la base de datos
5:	Registra lectura (IP, DATA, ”Voltaje”)

Existe un evento que detiene la ejecución del proceso de almacenamiento, el cual solo modifica el valor de bandera a falso del Algoritmo 5. En esta sección se habló de la interconexión de dispositivos y este sistema distribuido en un entorno de realidad virtual.

Implementación de la red de sensores en un entorno virtual

Definida la red de sensores y habilitada para recibir comandos por parte del usuario se procedió a integrarla en el caso de uso de un ambiente virtual. Como podemos ver en la *Figura 3* se define la conexión por medio de unos disparadores del entorno virtual hacia el nodo central.

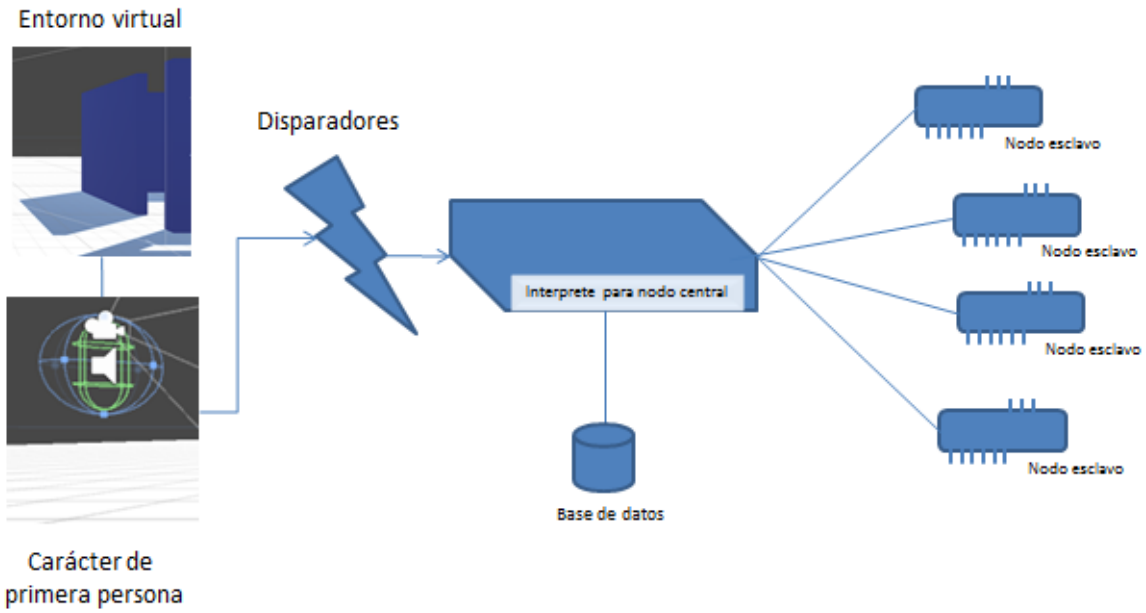

Figura 3. Integración en entorno virtual.

En este caso de uso se dio paso a la representación virtual de un planteamiento de desarrollo de un entorno inteligente en una casa habitación, en donde un conjunto de disparadores activan acciones de los dispositivos por medio de la invocación del gestor de comandos. Este entorno virtual ejerce con un disparador la comunicación con el nodo central lo que provoca que se ejecute una acción de los nodos esclavos en el mundo real dando como resultado la integración de una realidad mixta y demostrando el funcionamiento de la red de sensores.

Para realizar esta implementación se trabajó con 3 elementos en el ambiente virtual:

- El desarrollo del entorno virtual
- El carácter de primera persona
- La configuración de los disparadores

Desarrollo del Entorno Virtual

Para el desarrollo de entorno virtual se definió una escena en Unity en donde se depositó un plano de una habitación simple (*Figura 4*). Este plano fue diseñado desde una aplicación de modelado y luego exportado a un objeto .FBX el cual se implementó en la escena Unity dándole propiedades sólidas para la buena interacción con los demás objetos que se encuentran dentro del mismo entorno virtual.

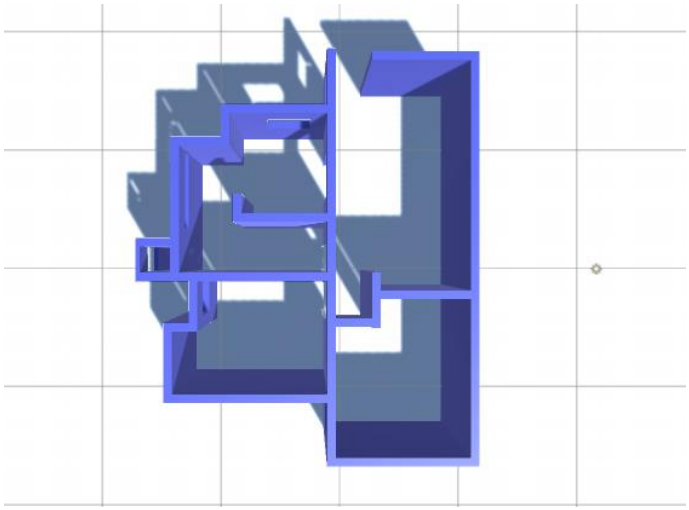

Figura 4. Plano de habitación simple.

Carácter de primera persona

Una vez establecida la escena, se estableció un personaje que interactúe en ella, para esto Unity por defecto provee de ciertos recursos los cuales permiten una integración rápida de objetos complejos pero comunes en un entorno cualquiera, dentro de estos objetos se encuentra los recursos de caracteres entre los cuales está el carácter de la primera persona. La *Figura 5* muestra la integración del carácter de primera persona en la escena de Unity. El carácter en primera persona es un objeto que integra ciertas características físicas y de control, así como efectos de sonido y una visión de cámara que simula la inmersión del usuario. Este carácter se integró para proporcionar control al usuario y de esta forma pueda moverse dentro del entorno virtual a fin de activar los disparadores definidos dentro del entorno.

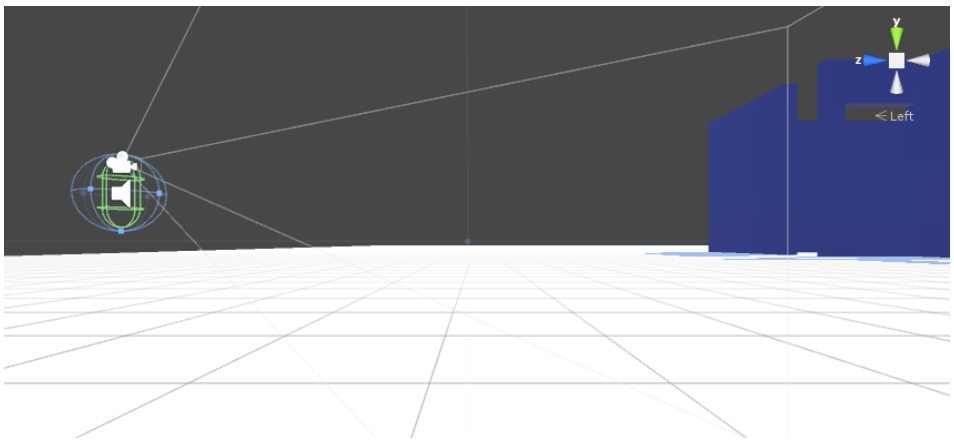

Figura 5. Carácter de primera persona.

Configuración de los disparadores

El carácter en primera persona es aquel que activa los disparadores en la escena, para establecer dichos disparadores se define un objeto no sólido e invisible al usuario el cual por medio de la asignación isTrigger invoca los eventos del entorno de desarrollo. Como podemos observar en la *Figura 6* se establecen varios

elementos como disparadores dentro de la escena para posteriormente configurar los eventos al gestor de comandos del sistema distribuido.

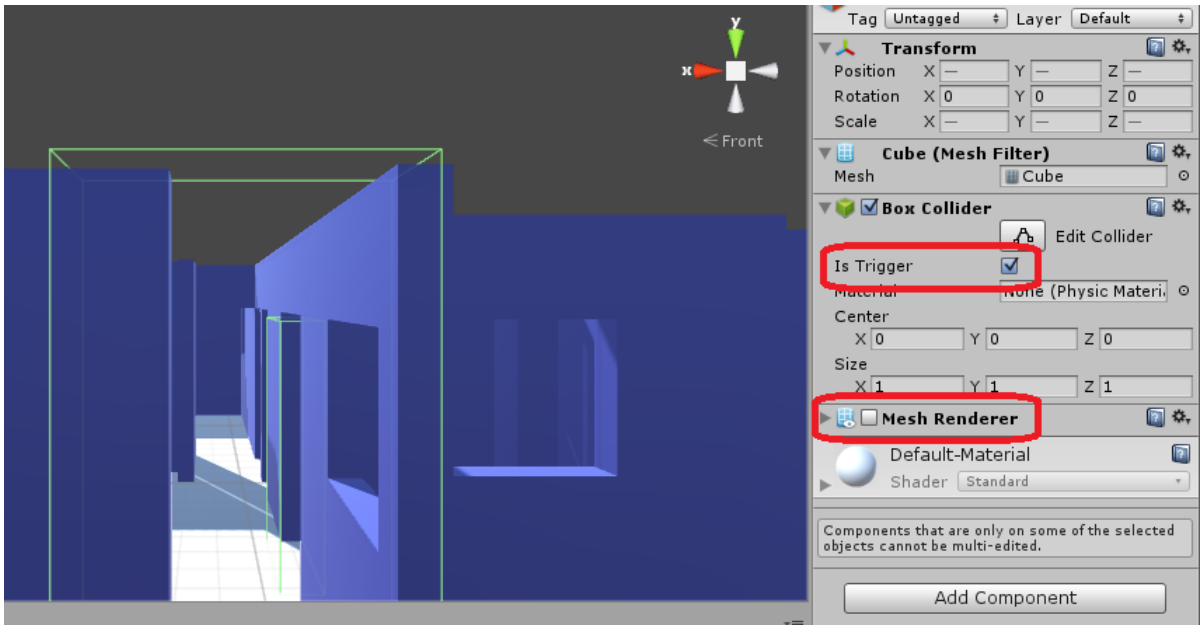

Figura 6. Configuración de disparadores.

Script de respuesta de los disparadores

Teniendo en el entorno los disparadores listos, es necesario asignarles una acción cuando éstos se activen, en este caso es el envío de una instrucción por medio de la red al nodo central. Por defecto las bibliotecas que permiten la comunicación por conexiones de red a nivel de transferencia de paquetes no se encuentran integradas en el conjunto de recursos que proporciona Unity, sin embargo, se puede integrar la conexión de red por medio de la construcción de una biblioteca de que defina un cliente de red y la integración a los recursos de la escena Unity.

- *Biblioteca para el cliente de red*

La construcción de la biblioteca la comunicación crea el cliente de red a partir de una clase la cual integra dos bibliotecas del sistema: la biblioteca *System.Net* y la biblioteca *System.Net.Sockets*. Estas dos entregan los métodos suficientes para la conexión con el nodo central. Esta clase llamada Cliente integra la construcción de un Socket y un EndPoint apuntando a una IP y un Puerto. Al crear el socket se definen el tiempo de familia de direcciones, el tipo de socket y el tipo de protocolo.

El método de envío verifica que exista conexión con el dispositivo gestor y, en caso de ser verdadero, establece un controlador de envío de paquetes, codifica el mensaje y lo envía por el socket. Este método también recupera la respuesta enviada por el dispositivo y lo imprime en la consola de ejecución.

- *Script*

Después de integrar la biblioteca del cliente de red se procede a realizar la configuración de los disparadores. Estos disparadores funcionan con la definición de dos métodos que proporciona el entorno Unity los cuales son

- OnTriggerEnter()

  Construye una instancia de la biblioteca cliente, conecta con el gestor de comandos y envía un comando de encendido apuntando a un dispositivo en específico.

- OnTriggerExit()

  Construye una instancia de la biblioteca Cliente, conecta con el gestor de comandos y envía un comando de apagado a un dispositivo en específico.

Ambos comandos utilizan el conjunto de instrucciones definido en el Gestor de comandos y es él quien se encarga de la comunicación con los dispositivos, reduciendo la cantidad de código necesario para la integración y agilizando el uso de las instrucciones que entregan los dispositivos.

RESULTADOS Y DISCUSIÓN

Dentro del conjunto de resultados obtenidos en el trabajo, podemos observar en la Figura 7 la red de dispositivos integradas con 5 tarjetas Intel® Galileo conectando un led en el puerto 12, una computadora (izquierda) que ejecuta el sistema interprete del nodo central y otra (derecha) que ejerce la función de usuario la cual se comunica con el nodo central. La *Figura 7* se tomó al momento de realizar la prueba de funcionamiento y comunicación entre el nodo central y los nodos esclavos; y entre el usuario y el nodo central, a manera de que al momento del envío de un comando al nodo central este la reinterprete de forma adecuada y envíe la instrucción específica a los nodos esclavos.

Figura 7. Red de dispositivos.

Después de haberse realizado las pruebas del funcionamiento de la red, se integró el entorno Unity a la maquina cliente. En este entorno los disparadores configurados dentro de Unity funcionan como el cliente que envía comandos al nodo central. En la Figura 8 se puede observar el resultado de la prueba del entorno virtual funcionando en conjunto con la red, en donde al momento de activar uno de los disparadores se encienda el led conectado en el nodo esclavo por medio de la comunicación con el nodo cliente.

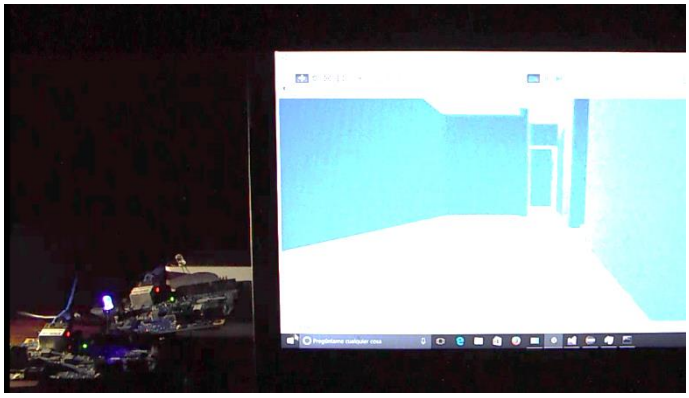

Figura 8. Integración del entorno virtual.

Una de las actividades a futuro es poder hacer reaccionar el entorno virtual en base a la lectura de los sensores de los dispositivos, para poder entregarle al usuario una mejor inmersión. Otro trabajo a futuro es la Integración de repositorio de los datos almacenados en la nube, para permitir el análisis de datos de varias redes y poder compartir dicha información por medio del internet. Este trabajo se integrará en el caso de uso en domo de inmersión de realidad virtual, en donde múltiples dispositivos ejecutan sus eventos en base a una proyección audiovisual. Para hacer esta integración también se desarrollará un lenguaje de alto nivel el cual gestionará los comandos del Gestor y así sincronizar los dispositivos.

CONCLUSIONES

En este trabajo se demostró la flexibilidad y dinamismo de las tarjetas Intel® Galileo permitiendo utilizar sus recursos acordes a los requerimientos de la problemática. Cabe mencionar que se busca integrar otros tipos de sistemas de IoT para verificar la eficiencia de la interfaz Arduino en contra de otros sistemas IoT que la tarjeta Intel® Galileo permite integrar. Sin embargo con esta implementación demostró el funcionamiento de los dispositivos al trabajar en conjunto con el nodo central, con el uso un cliente simple como usuario, donde se comprobó el funcionamiento de la red y se verificó la respuesta esperada a las instrucciones establecidas demostrado en un primer caso de uso de comunicación sin interfaz.

Al tener un entorno distribuido, con las instrucciones se pudo integrar un entorno virtual a la red de dispositivos, corroborando de esta forma la flexibilidad de la red, además de probar el tiempo de integración en un caso de uso de comunicación con una interfaz (entorno Unity), donde se observa que, con la red definida, el desarrollo del proceso de comunicación es sencillo y ágil.

En base a los elementos resultantes y las pruebas realizadas se demostró que la red de dispositivos cumple con el objetivo buscado, el cual es permitir la construcción rápida y funcional de un entorno inmersivo, con esto se demostró dicha funcionalidad en un entorno virtual aunque cabe mencionar que puede ser aplicado en diversos casos de uso, dependiendo de los requerimientos del entorno en donde se implemente, por ejemplo una casa inteligente, un hospital, una sala de inmersión audiovisual entre otros. En la delimitación de este trabajo se validó uno de los diversos escenarios en donde puede integrarse el trabajo.

LITERATURA CITADA

Arduino community, (2016, Agosto), Arduino - Sketch, disponible en https://www.arduino.cc/en/Tutorial/Sketch [26 de Agosto de 2016].

Cook, D., Das, S. (2003) Karthik Gopalratnam, and Abhishek Roy, Health Monitoring in an Agent-Based Smart Home, to appear in Proceedings of the International Conference on Aging, Disability and Independence Advancing Technology and Services to Promote Quality of Life.

Cook, D., Das, S. (2004). MavHome: Work in progress, IEEE Pervasive Computing.

Evans, D. (2011). Internet de las cosas. Cómo la próxima evolución de Internet lo cambia todo.

Intel®, (2016, Agosto), Intel® Galileo Gen 2 Development Board—Empower Your Prototype, disponible en http://www.intel.la/content/www/xl/es/do-it-yourself/galileo-makerquark-board.html [28 de Agosto de 2016].

López, J. A., García, A. J., Soto, F., Iborra, A., García F., Iborra A. (2011). Design and validation of a wireless sensor network architecture for precision horticulture applications, PRECISION AGRICULTURE (vol. 12, pp. 280-295).

Molina J. M., Navarro, P. J., Jiménez, M., Soto, F., Ruiz, A. García, D. (2012). VIPMET: a new real-time data filtering based automatic, JOURNAL OF IRRIGATION AND DRAINAGE ENGINEERING (vol. 138, pp. 823-829).

Quintana, M. A. (1996), Introducción a la programación en red: Sockets y Windows Sockets, disponible en http://hdl.handle.net/10553/601 [4 de Septiembre de 2016]

Unity - Game engine, tools and multiplatform, disponible en https://unity3d.com/es/unity [02 de septiembre de 2016]

Zhang, Guo-Qiang., Yang, Qing-Feng., Cheng, Su-Qi., Zhou, Tao. (2008), Evolution of the Internet and its cores, disponible en http://iopscience.iop.org/article/10.1088/1367-2630/10/12/123027/meta [11 de octubre de 2016]

SÍNTESIS CURRICULAR

Abraham Obed Chan Canche

Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Cancún, es gerente de sistemas en la empresa SYE decodificaciones en donde implementa sistemas de automatización por código de barras. Actualmente es estudiante de maestría en el Instituto Tecnológico Mario Molina Pasquel y Henríquez, dentro de sus áreas de interés se encuentra el desarrollo de sistemas de control y minería de datos. Correo electrónico: achan.msc@itszapopan.edu.mx

Miriam Díaz Rodríguez

Ingeniera en computación por el Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) en 2007, obtuvo su grado de maestra en Ciencias en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV) Unidad Guadalajara en 2010, donde realiza actualmente su doctorado. Es profesora titular en el Instituto Tecnológico José Mario Molina Pasquel y Henríquez, entre sus áreas de interés se encuentran sistemas de eventos discretos y teoría computacional. Correo electrónico: mdiaz@itszapopan.edu.mx