

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación



**ESTUDIO, DISEÑO E IMPLEMENTACIÓN DE
SISTEMAS DE INTERNET DE LAS COSAS
APLICADOS A ESCENARIOS DE REALIDAD
AUMENTADA EN MUSEÍSTICA**

TRABAJO FIN DE MÁSTER

Germán Paolo Jácome Zambrano

2017

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos**

TRABAJO FIN DE MÁSTER

**ESTUDIO, DISEÑO E IMPLEMENTACIÓN DE
SISTEMAS DE INTERNET DE LAS COSAS
APLICADOS A ESCENARIOS DE REALIDAD
AUMENTADA EN MUSEÍSTICA**

Autor

Germán Paolo Jácome Zambrano

Director

Borja Bordel Sánchez

Ponente

Diego Martín de Andrés

Departamento de Ingeniería de Sistemas Telemáticos

2017

Resumen

La realidad aumentada combina los entornos reales con los elementos virtuales, presentando así una realidad “mixta”, en donde sus usuarios perciben una nueva experiencia, en la que se les presenta una realidad explicada o simplificada mediante la continua interacción de los diversos elementos virtuales con los reales. Las aplicaciones de realidad aumentada se han extendido al ámbito de la museística, donde las nuevas tecnologías están sustituyendo a las tradicionales audio guías.

Por otro lado, la Internet de las Cosas se ha convertido en poco tiempo en una herramienta que aporta inteligencia y control sobre un variado y diverso conjunto de elementos tanto de la industria como de la vida cotidiana. Los Módulos de Internet de las Cosas (IoT) son una nueva tendencia tecnológica, que presenta grandes capacidades de recopilación y tratamiento de datos a través de dispositivos de tamaño reducido.

Actualmente una de las corrientes en el mundo de las redes es cubrir ámbitos más reducidos en donde se aprovechen y desarrollen mejor las capacidades de las tecnologías de red existentes. El presente trabajo evidencia las potencialidades de las tecnologías inalámbricas aplicadas al ámbito de Redes de Área Personal (PAN), como lo son: IEEE 802.11, Bluetooth, ZigBee y Radio Frequency Identification (RFID); con dicho análisis y la investigación que se realiza de museos en espacios abiertos, entorno sobre el cual se aplicará la implementación, es así que se escoge a Bluetooth como la opción que mejor se adapta al escenario establecido.

En este proyecto se diseña e implementa, con tecnología Bluetooth, un sistema de Internet de las Cosas para aplicaciones de realidad aumentada en entornos de museística. La implementación involucra fundamentalmente los módulos IoT: ‘receptor’ que lo portarán los visitantes al museo y ‘nodos’ que se incorporan a las piezas de la colección museística; mediante la interacción de estos dispositivos se facilitara la difusión de contenidos inherentes a los recursos del museo. La implementación del sistema incluye las etapas de montaje y programación de los módulos receptor y nodos; el uso de herramientas de código libre, las cuales pueden ser modificadas y utilizadas a conveniencia, ha permitido el despliegue de la programación en el presente trabajo, es así que para esta etapa se ha empleado principalmente, el script de la Bash y los lenguajes de programación C y C++.

Para complementar el diseño e implementación, se presenta una primera revisión tecnológica con RFID, con el fin de disponer de una alternativa a la implementación desarrollada que sirva como base para posibles trabajos y aplicaciones futuras.

Finalmente se realiza un conjunto de pruebas que acercan al sistema desarrollado a un a un entorno real, con lo que se valida el funcionamiento de los diferentes módulos. Se describen también, algunas líneas de acción sobre las cuales se puedan seguir desarrollando proyectos con características similares.

Abstract

The augmented reality combines the real environments with the virtual elements, presenting a "mixed" reality, where it's users perceive a new experience, in which they are presented an explained or simplified reality through the continuous interaction of the various virtual elements with real elements. Applications of augmented reality have been extended to the field of museum studies, where new technologies are replacing traditional audio guides.

On the other hand, the Internet of Things has become in a short time a tool that provides intelligence and control over a varied and diverse set of elements of both industry and everyday life. The Internet of Things (IoT) Modules are a new technological trend, which presents great capacities of gathering and processing of data through devices of reduced size.

Currently one of the tendencies in the network world is to cover smaller areas where the capabilities of existing network technologies are better exploited and developed. The present work shows the potential of wireless technologies applied to the environment of Personal Area Networks (PAN), such as: IEEE 802.11, Bluetooth, ZigBee and Radio Frequency Identification (RFID); With this analysis and the research that is carried out of museums in open spaces, the environment on which the implementation will be applied, is to choose Bluetooth as the option that best suits the established scenario.

In this project is designed and implemented, with Bluetooth technology, an Internet system of Things for augmented reality applications in museum studies environments. The implementation basically involves the IoT modules: 'receiver' that will be carried by the visitors to the museum and 'nodes' that are incorporated into the pieces of the museum collection; Through the interaction of these devices facilitate the dissemination of contents inherent to the museum resources. The implementation of the system includes the stages of assembly and programming of the receiver modules and nodes; The use of free code tools, which can be modified and used at convenience, has allowed the deployment of programming in the present work, so for this stage has been used mainly, the Bash script and programming languages C and C ++.

To complement, the design and implementation presents a first technological revision with RFID, with the goal to have an alternative to the implementation developed that serves as a basis for possible future work and applications.

Finally, a set of tests is performed that bring the developed system closer to a real environment, thereby validating the operation of the different modules. It also describes

some lines of action that allow further development of projects with similar characteristics.

Índice general

Resumen	i
Abstract.....	iii
Índice general.....	v
Índice de figuras	ix
Índice de tablas.....	xii
Siglas	xiii
1 Introducción.....	1
1.1 Objetivos	2
1.2 Estructura del Documento	3
2 Redes Inalámbricas	5
2.1 WLAN (Wireless Local Area Network)	6
2.2 WPAN (Wireless Personal Area Network).....	6
2.2.1 Grupos de Trabajo WPAN de la IEEE (Institute of Electrical and Electronics Engineers) 802.15: [1]	6
2.3 Red Inalámbrica AD HOC	9
2.3.1 Tipos.....	9
2.4 IEEE 802.11-2012.....	10
2.4.1 Grupos de estandarización 802.11	10
2.4.2 Componentes	11
2.4.3 Topologías de Red.....	12
2.4.4 Stack de Protocolos	15
2.4.5 IEEE 802.11ac-2013	19
2.4.6 Formato de la PDU.....	19
2.5 IEEE 802.15.4	23
2.5.1 Componentes de IEEE 802.15.4	23

2.5.2	Topologías de Red.....	24
2.5.3	Arquitectura de Protocolos	24
2.5.4	Formato de la PDU.....	27
2.5.5	Inicialización y mantenimiento de la PAN.....	31
2.6	Bluetooth.....	32
2.6.1	Topologías Bluetooth.....	32
2.6.2	Stack de Protocolos	32
2.6.3	Arquitectura de Protocolos	34
2.6.4	Formato de la PDU.....	36
2.6.5	Bluetooth Core Specification Version 3.0 High Speed	38
2.7	ZigBee	39
2.7.1	Tipos de Dispositivos.....	39
2.7.2	Topologías de Red.....	40
2.7.3	Stack de Protocolos	41
2.7.4	Formato de la PDU.....	45
2.8	RFID (Radio Frequency IDentification).....	49
2.8.1	Componentes del Sistema	49
2.8.2	Radios de operación.....	52
2.8.3	Estándares	52
2.8.4	ISO/IEC 18000-7 [8]	52
2.8.5	Formato de la PDU.....	54
2.9	Plataformas Hardware.....	57
2.9.1	Samsung ARTIK™ 530 [9].....	57
2.9.2	Arduino.....	58
2.9.3	XBee Radio	61
2.9.4	Kits de Desarrollo RFID [15].....	63
3	Entornos Museísticos y Tecnologías Inalámbricas	67
3.1	Análisis de Entornos Museísticos	67
3.1.1	Museografía.....	67
3.1.2	Proyecto Museográfico	68
3.1.3	Distribución de Piezas Museísticas (Definición del Área de Trabajo) ..	68

3.2	Elección de la Tecnología	69
3.2.1	Factores de Evaluación	69
	Características tecnológicas:	69
3.2.2	Evaluación de Wi-Fi (IEEE 802.11a/b/g/n).....	70
3.2.3	Evaluación de Bluetooth.....	70
3.2.4	Evaluación de ZigBee	71
3.2.5	Evaluación de RFID	72
3.2.6	Elección de la Tecnología a Implementarse.....	73
4	Diseño e Implementación del Sistema.....	76
4.1	Análisis de Requerimientos	76
4.1.1	Modelo Básico de la Red	76
4.1.2	Dimensionamiento de Tráfico	76
4.1.3	Consumo de Potencia (Batería)	77
4.1.4	Distancia entre Dispositivos	77
4.2	Diseño de la Red de Área Personal.....	77
4.2.1	Distribución de Dispositivos	77
4.2.2	Planteamiento de Requerimientos por Etapas para el Desarrollo del Software	78
4.3	Montaje y Programación del Módulo Receptor (Dispositivo Móvil).....	79
4.3.1	Componente de Hardware	79
4.3.2	Programación del Componente.....	79
4.4	Montaje y Programación de los Nodos (Dispositivos Móviles)	92
4.4.1	Componentes de Hardware.....	93
4.4.2	Programación de Componentes	94
4.5	Revisión Tecnológica con RFID.....	97
4.5.1	Interconexión del Hardware.....	97
4.5.2	Configuración para la Interconexión.....	98
5	Experimentación y Resultados	101
5.1	Despliegue y Experimentación.....	101
5.1.1	Instalación del Prototipo	101
5.1.2	Escenario de Pruebas	101

5.1.3	Pruebas de Campo	102
5.2	Resultados Obtenidos.....	104
6	Conclusiones y Líneas Futuras de Investigación.....	105
6.1	Conclusiones	105
6.2	Líneas Futuras de Investigación.....	107
	Bibliografía	109

Índice de figuras

Figura 1. Clasificación de las Redes Inalámbricas por su Ámbito de Cobertura	5
Figura 2. Topología BSS.....	13
Figura 3. Topología IBSS	13
Figura 4. Topología ESS.....	14
Figura 5. Topología MBSS.....	14
Figura 6. Stack de Protocolos Base IEEE 802.11	15
Figura 7. Distribución esquemática de la Trama Física VHT.....	19
Figura 8. Formato del Sub-campo VHT-SIG-A1.....	20
Figura 9. Formato del Sub-campo VHT-SIG-A2.....	20
Figura 10. Formato del Campo VHT-SIG-B	20
Figura 11. Formato del campo Carga Útil	21
Figura 12. Formato del Sub-campo Servicio.....	21
Figura 13. Distribución Esquemática de la Trama General MAC	21
Figura 14. Formato del Campo Control de Trama	21
Figura 15. Formato del Campo Control QoS.....	22
Figura 16. Formato del Campo Control HT	22
Figura 17. Formato PPDU	27
Figura 18. Formato Campo SHR.....	27
Figura 19. Formato del Campo Control de Trama	28
Figura 20. Vista esquemática de la Trama Beacon.....	28
Figura 21. Formato del campo Especificación de Supertrama.....	29
Figura 22. Estructura del campo GTS.....	29
Figura 23. Estructura del sub-campo Especificación GTS.....	29
Figura 24. Formato del sub-campo Direccionamiento Pendiente GTS.....	29
Figura 25. Estructura sub-campo Lista GTS.....	29
Figura 26. Formato del campo Direccionamiento Pendiente.....	29
Figura 27. Formato del sub-campo Especificación de direccionamiento pendiente .	30
Figura 28. Vista esquemática de la trama de datos	30
Figura 29. Vista esquemática de la trama ACK	30
Figura 30. Vista esquemática de la trama de comandos MAC	31
Figura 31. Stack de Protocolos Bluetooth.....	33
Figura 32. Formato general del paquete Bluetooth.	36
Figura 33. Formato del campo Código de Acceso	36
Figura 34. Formato de la Encabezado de Bluetooth.....	37

Figura 35. Formato del campo Carga útil	37
Figura 36. Sub-campo Encabezado de carga útil - paquetes de ranura simple.....	38
Figura 37. Sub-campo Encabezado de carga útil - paquetes de ranura múltiple.....	38
Figura 38. Topología Estrella.....	40
Figura 39. Topología Malla	40
Figura 40. Topología Árbol	40
Figura 41. Stack de Protocolos ZigBee e IEEE 802.15.4.....	41
Figura 42. Vista esquemática de la NPDU.....	45
Figura 43. Formato del campo Control de Trama.	45
Figura 44. Formato del campo Control Multicast.....	46
Figura 45. Formato del campo Ruta de Origen de Sub-trama	47
Figura 46. Vista esquemática de la APS.....	47
Figura 47. Formato del campo Control de Trama	48
Figura 48. Componentes sistema RFID.....	50
Figura 49. Vista esquemática del Mensaje Lectora -Etiqueta.....	54
Figura 50. Formato del campo Tipo de Comando.....	54
Figura 51. Formato Mensaje Respuesta de Broadcast.....	56
Figura 52. Formato Campo Estado de Etiqueta	56
Figura 53. Formato Mensaje de Respuesta Unicast.....	56
Figura 54. ARTIK™ 530 [10]	58
Figura 55. Arduino Nano [12]	60
Figura 56. Distribución de Piezas Museísticas en Entornos Abiertos [17]	69
Figura 57. Comparativa entre Tecnologías (Dispositivos Fijos)	73
Figura 58. Comparativa entre Tecnologías (Dispositivos Móviles)	74
Figura 59. Elementos Básicos del Sistema.....	78
Figura 60. Dispositivos Escaneados, mediante ‘scan on’	80
Figura 61. Código en C para Escanear Dispositivos Bluetooth Cercanos [21]	82
Figura 62. Dispositivo escaneado mediante, programa en C con BlueZ.....	83
Figura 63. Estructura ‘inquiry_info’ de la librería ‘hci.h’ [22].....	83
Figura 64. Estructura ‘inquiry_info_with_rssi’ de la librería ‘hci.h’ [22]	84
Figura 65. Estructura ‘if’ con la función ‘hci_read_rssi’, para obtener el ‘rssi’	84
Figura 66. Errores desplegados al ejecutar el código modificado.	84
Figura 67. Función ‘hci_inquiry’ de la librería ‘hci_lib.h’	85
Figura 68. Función ‘hci_read_rssi’ de la librería ‘hci_lib.h’	85
Figura 69. Librerías SDK incluidas en el Programa con Paquetes SDK	86
Figura 70. Código en C correspondiente a ‘main.c’	86
Figura 71. Fragmento de Código en C relacionado con ‘hardLibrary.c’	87
Figura 72. Fragmento de Código en C correspondiente a ‘hardLibrary.h’	87

Figura 73. Fragmento de Código en C relacionado con 'lista_dispositivos_cercanos_Bluetooth.c'	87
Figura 74. Fragmento de Código en C correspondiente a 'lista_dispositivos_emparejados_Bluetooth.c'	88
Figura 75. Fragmento de Código en C relacionado con 'lista_dispositivos_todos_Bluetooth.c'	88
Figura 76. Fragmento de Código en C correspondiente a 'middleware.c'	89
Figura 77. Fragmento de Código en C relacionado con 'middleware.h'	89
Figura 78. Compilación de 'main.c'	90
Figura 79. Compilación de 'main.c', modificado con parámetro 0	90
Figura 80. Compilación de 'main.c', modificado con parámetro 1	90
Figura 81. Modificación de 'middleware.h'	90
Figura 82. Modificación de 'middleware.c'	91
Figura 83. Compilación de 'main.c', cambiando 'middleware.h' y 'middleware.c' ..	91
Figura 84. Compilación de 'main.c' utilizando 'pkg-config'	91
Figura 85. Ejecución y resultados, comando 'hcitool scan'	92
Figura 86. Script de Bash.....	92
Figura 87. Ejecución y resultados, script 'scan5.sh'	93
Figura 88. Componentes de un Nodo	94
Figura 89. Montaje Nodo.....	94
Figura 90. Fragmento de Código, configuración Módulo Bluetooth HC-06 con Arduino Nano.....	95
Figura 91. Fragmento de Código, configuración Acelerómetro mediante Arduino Nano.....	96
Figura 92. Fragmento de Código, configuración Zumbador con Arduino Nano	97
Figura 93. Cableado para la Conexión del Lector Serial L-RX202 [25]	98
Figura 94. Interconexión entre DB9 del Lector Serial L-RX202 y ARTIK™ 530	98
Figura 95. Fragmento de Código en C, comunicación Serial Loopback [26]	100
Figura 96. Nodo Instalado en un Objeto Real.	101
Figura 97. Escenario de Pruebas, ETSI Telecomunicación – Bloque B.....	102
Figura 98. Prueba de Campo 1	103
Figura 99. Prueba de Campo 2	103

Índice de tablas

Tabla 1. Características de los Principales Estándares IEEE 802.11 [3].....	11
Tabla 2. Bandas de Frecuencia y Velocidad de Datos para IEEE 802.15.4	25
Tabla 3. Frecuencias de Operación y Distancias de Lectura se Sistemas RFID [7]	52
Tabla 4. Distribución de Pines Arduino Nano	60
Tabla 5. Distribución de Pines XBee [13]	62
Tabla 6. Pines de Conexión entre XBee y Arduino Board Adapter Hack [14]	63
Tabla 7. Comparación de los Estándares Bluetooth 3.0, ZigBee 3.0, IEEE 802.11ac y RFID	66
Tabla 8. Características de Evaluación, IEEE 802.11a/b/g/n.....	70
Tabla 9. Características de Evaluación, Bluetooth 3.0	71
Tabla 10. Características de Evaluación, ZigBee.....	71
Tabla 11. Características de Evaluación, a 433 MHz	72

Siglas

ACL	Asynchronous Connectionless
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AIB	Application Support Sub-layer Information BASE
AIFS	Arbitration Interframe Space
AM_ADDR	Active Member Address
AMP	Alternate Medium Access Control / Physical
A-MPDU	Aggregated Medium Access Control Protocol Data Unit
A-MSDU	Aggregated Medium Access Control Service Data Unit
AODV	Ad hoc On-Demand Distance Vector
APDU	Application Protocol Data Unit
API	Application Programming Interface
APS	Application Support Sub-layer
APSDE	APS Data Entity
APSME	APS Management Entity
ARQ	Automatic Repeat Request
ARQN	Automatic Repeat reQuest Number
ASK	Amplitude-Shift Keying
BD_ADDR	Bluetooth Device Address
BPSK	Binary Phase Shift Keying
BSA	Basic Service Area
BSS	Basic Service Sets
BSSID	BSS Identifier
BTR	Broadcast Transaction Record
BTT	Broadcast Transaction Table
CAC	Channel Access Code
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCA	Clear Channel Assessment
CCITT	Consultative Committee for International Telegraphy and Telephony
CCK	Complementary Code Keying
CCMP	Counter Mode Cipher Block Chaining Message Authentication Code Protocol
CoAP	Constrained Application Protocol
CSMA	Carrier Sense Multiple Access
CSMA/CA	CSMA/Collision Avoidance
CTS	Clear-to-Send
DAC	Device Access Code
DCF	Distributed Coordination Function
DQPSK	Differential Quadrature Phase Shift Keying
DS	Distribution System
DSDV	Destination-Sequenced Distance Vector Routing
DSM	Distribution System Medium
DSSS	Direct-Sequence Spread Spectrum

EDCA	Enhanced Distributed Channel Access
EDCAF	EDCA Function
EIA	Electronic Industries Alliance
EIRP	Equivalent Isotropically Radiated Power
EPCglobal	Electronic Product Code global
ERP	Effective Radiated Power
ESS	Extended Service Set
ESSID	ESS Identifier
ETSI	European Telecommunications Standards Institute
EWMA	Exponentially Weighted Moving Average Algorithms
FCC	Federal Communications Commission
FCS	Frame Check Sequence
FEC	Frame Check Sequence
FFD	Full-Function Device
FHSS	Frequency-Hopping Spread Spectrum PHY
FH-TDD-TDMA	Frequency Hopping- Time Division Duplex - Time Division Multiple Access
FTDI	Future Technology Devices International
GCC	GNU Compiler Collection
GFSK	Gaussian Frequency Shift Keying
GNU	GNU's Not Unix
GPU	Graphics Processing Unit
GTS	Guaranteed Time Slot
HC	Hybrid Coordinator (IEEE 802.11)
HCCA	HC Function Controlled Channel Access
HCF	HC Function
HDMI	High-Definition Multimedia Interface
HEC	Header Error Control
HF	High Frequency
HR-DSSS	High-Rate DSSS
HS	High Speed
HSIC	HS Inter-Chip
HT	High Throughput
HWMP	Hybrid Wireless Mesh Protocol
I²C	Inter-Integrated Circuit
IAC	Inquiry access code
IBSS	Independent BSS
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IFS	Interframe Space
IoT	Internet of Things
IP	Internet Protocol
IR	Infrared
ISM	Industrial, Scientific and Medical
ISO	International Organization for Standardization
ISR	Integrated Services Routers
KPM	Key Management Protocol
L_CH	Logical Channel
L2CAP	Logical Link Control and Adaptation Protocol

L2R	Layer 2 Routing
LAP	Lower Address Part
LC	Link Controller (Bluetooth)
LCD	Liquid Cristal Display
LF	Low Frequency
LLC	Logical Link Control
L-LTF	Non-HT Long Training Field
LMP	Link Manager Protocol
LQI	Link Quality Indicator
L-SIG	Non-HT Signal
L-STF	Non-HT Short Training Field
MAC	Medium Access Control
MANETS	Mobile Ad-Hoc Networks
MBSS	Mesh BSS
MCPS-SAP	MAC Common Part Sublayer Service Access Point
MCS	Modulation and Coding Set
MFR	MAC Footer
MHR	MAC Header
MLME	MAC Sublayer Management Entity
MP	Mesh Point
MPDU	MAC Protocol Data Unit
MPP	Mesh Point Portal
MQTT	Message Queue Telemetry Transport
MSDU	MAC Service Data Unit
NAK	Negative Acknowledgment
NIB	Network Information Base
NLDE	Network Layer Data Entity
NLME	Network Layer Management Entity
NPC	Network Coprocessor
NPDU	Network Protocol Data Unit
OBEX	Object Exchange Protocol
OFDM	Orthogonal Frequency-Division Multiplexing
OLED	Organic Light-Emitting Diode
OQPSK	Offset Quadrature Phase-Shift Keying
OSI	Open System Interconnection
OTG	On-The-Go
PAC	Peer Aware Communications
PCB	Printed Circuit Board
PCF	Point Coordination Function
PDU	Protocol Data Unit
PHR	Physical Header
PLCP	Physical Layer Convergence Procedure
PMD	Physical Medium Dependent
POS	Personal Operating Space
PPDU	Physical PDU (IEEE 802.15.4)
PPDU	PLCP PDU (IEEE 802.11)
PPP	Point to Point Protocol
PSDU	PLCP Service Data Unit
PSMP	Power Save Multi-Poll
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying

REST	Representational State Transfer
RFCOMM	Radio Frequency Communication
RFD	Reduced-Function Device
RPSMA	Reverse Polarity Sub-Miniature Version A
RSSI	Received Signal Strength Indication
RTS/CTS	Request-to-Send/Clear-to-Send
RTT	Round-Trip Time
SCO	Synchronous Connection Oriented
SDF	Start-of-Frame Delimiter
SDK	Software Development Kit
SDP	Service Discovery Protocol
SEQN	Sequential Numbering scheme
SHR	Synchronization Header
SIFS	Short IFS
SMPS	Spatial Multiplexing Power Save
SPDU	Service-Specific Convergence Sublayer PDU
SPI	Serial Peripheral Interface
SSCS	Service-Specific Convergence Sublayer
SSID	Service Set Identifier
SSP	Service Security Provider
TCP	Transmission Control Protocol
TCS-BIN	Telephony Control Protocol – Binary
TG	Task Group
TKIP	Image Temporal Key Integrity Protocol
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol
UHF	Ultra High Frequency
UIT-R	Radiocommunication Sector of the International Telecommunication Union
USB	Universal Serial Bus
UTP	Unshielded Twisted Pair
VHT	Very High Throughput
VHT-LTF	VHT Long Training Field
VHT-SIG	VHT Signal
VHT-STF	VHT Short Training Field
WAE/WAP	Wireless Application Environment/Wireless Application Protocol
WBAN	Wireless Body Area Network
WDS	Wireless Distribution System
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
X-CTU	XBee Configuration and Test Utility
ZDO	ZigBee Device Object
ZDP	ZigBee Device Profile

1 Introducción

En un ambiente museográfico los dispositivos tecnológicos incorporados, a su cotidianidad artística, facilitan la difusión de contenidos inherentes al museo y sus recursos a un conjunto amplio de visitantes, brindando beneficios que inciden directamente sobre las partes involucradas en este tipo de entornos.

Si bien las audioguías han sido elementos han aportado considerablemente a los entornos de museística, su uso en la actualidad ha decaído, pues a comparación de otros sistemas, los cambios incorporados han sido muy limitados, a pesar de su implementación en dispositivos móviles actualizados.

A lo mencionado se suma que solo los museos con gran presupuesto tienen acceso a sistemas actualizados y sofisticados de audioguías, los que generalmente son sistemas propietarios.

En este contexto el presente trabajo pone a consideración el uso de tecnologías del Internet de las Cosas (IoT), que promete revolucionar muchos ámbitos de la vida cotidiana, en escenarios de realidad aumentada, específicamente en ambientes de museística, donde las nuevas tecnologías están sustituyendo a las tradicionales audio guías.

La realidad aumentada es la combinación de los entornos reales con los elementos virtuales, logrando de esa manera una realidad “mixta”. La arquitectura básica consiste en un conjunto de dispositivos que añaden información de origen virtual, a la información física reciba por los usuarios de forma natural.

Los usuarios de realidad amentada perciben una nueva experiencia, que es el resultado de capturar automáticamente las situaciones de la realidad mediante diferentes métodos; de este modo se presenta una realidad simplificada o explicada mediante la continua interacción de diversos elementos virtuales con reales.

La realidad aumentada muestra al usuario información añadida del entorno, mejorando su percepción e interacción con situaciones reales; para cumplir con dicho propósito, los elementos virtuales deben realizar un seguimiento preciso de los acontecimientos en tiempo real, propiciando la coincidencia de los entornos. A lo mencionado debe añadirse la movilidad de los dispositivos (desvinculando de esta forma la realidad aumentada de un ambiente de propósito especial) así, éstos deberán ser capaces (idealmente) de trabajar en cualquier entorno mediante el uso de tecnologías como comunicaciones inalámbricas y computación portátil.

Las nuevas tecnologías se han ido incorporando al acontecer diario de casi todas las instituciones, son entonces los elementos virtuales de bajo coste los que permitirán a museos, poco conocidos y con un limitado presupuesto, a involucrarse de una forma más amplia al mundo tecnológico, brindándoles a estas instituciones una nueva forma de promoción cultural haciendo más atractiva la experiencia museística.

1.1 Objetivos

El objetivo general de este proyecto es implementar un sistema de Internet de las Cosas para aplicaciones de realidad aumentada en entornos de museística. El foco es el estudio de un conjunto de tecnologías inalámbricas en las que se pueda basar un sistema capaz de presentar a los usuarios información enriquecida sobre la experiencia que viven, en tiempo real, con las muestras e instalaciones.

Para cumplir con dicho objetivo se plantean varios ámbitos sobre los cuales se irá desarrollando el proyecto:

- Analizar las diferentes tecnologías inalámbricas de área local de amplio uso como IEEE 802.11 y área personal, centrándose en las redes Ad Hoc como Bluetooth 3.0, ZigBee y RFID; este análisis toma en cuenta las características de cada una de las tecnologías y cómo éstas pueden o no adaptarse a un entorno de museística.
- Analizar especialmente la forma de comunicación P2P (peer to peer), de las tecnologías seleccionadas, y la posibilidad de admitir además de comunicaciones unicast; esquemas multicast, broadcast o anycast.
- En concordancia con las tecnologías inalámbricas, que son sujeto de estudio, se analizan módulos (hardware) de comunicación, sus características, versiones y demás elementos y aspectos que permitan su operación en una red de comunicaciones.
- Se implementarán en dos módulos IoT para realidad aumentada, una de las tecnologías de comunicación que son sujeto de análisis (Bluetooth 3.0); en donde uno de los módulos (fijo) identificará y se ubicará en la pieza museística; y el otro módulo (móvil) lo portará el visitante para la difusión de contenido multimedia (audio).
- Conseguir que los módulos IoT establezcan comunicación entre peers, esto implica que la emisión continua de señal del dispositivo fijo sea identificada, a una distancia adecuada por la escucha continua de señales del dispositivo móvil con lo cual éste empezará a difundir información al usuario referente a la pieza museística a que se ha acercado.
- Se implementará los peers de acuerdo a la tecnología de comunicación inalámbrica que se escoja; es así que en cuanto a los nodos se plantea las siguientes opciones: para el caso de Wi-Fi Intel Edison, en Bluetooth el módulo

HC-06, con ZigBee a través de XBee y RFID mediante tecnologías activas a 433 MHz.

- Se analizará cuál de las cuatro tecnologías se adapta mejor al entorno de museos y sus piezas de exhibición; para finalmente recomendar justificadamente la tecnología que mejor se ajuste al ambiente de museística.

1.2 Estructura del Documento

El presente trabajo presenta el análisis, desarrollo e implementación de una aplicación de Realidad Aumentada empleando tecnologías de redes inalámbricas basados en módulos de IoT, considerando que la integración de las mencionadas tecnologías se aplicará en entornos de museística; por lo que se divide al presente documento en seis capítulos que abarcan la temática descrita a continuación:

- **Capítulo 1 - Introducción:** Se realiza un breve análisis de la situación actual de las tecnologías usadas en entornos de museística, los motivos que han provocado su decaimiento. Además, se plantea la implementación de un sistema de Internet de las Cosas en escenarios de realidad aumentada en entornos de museística como una alternativa económica, viable y moderna, para reemplazar a las tradicionales audioguías.
- **Capítulo 2 - Estado del Arte Redes Inalámbricas para Sistemas IoT de corto Alcance:** Se analiza desde la perspectiva del estándar correspondiente los aspectos conceptuales y técnicos que rigen las tecnologías de comunicación inalámbrica IEEE 802.11, Bluetooth, ZigBee y RFID, con el objetivo de conocer su funcionamiento y como se pueden adaptar a entornos y piezas de museística. Así también, se presenta un conjunto de módulos IoT relacionados con las tecnologías mencionadas, de los cuales se empleará algunos en la implementación.
- **Capítulo 3 - Análisis de Entornos Museísticos y Adopción de Tecnología:** Se elabora un estudio de los ambientes de museos, su estructura, distribución de piezas y demás aspectos a tomar en cuenta; para luego en base a un análisis comparativo de las tecnologías investigadas se pueda escoger aquella que mejor se adapte a estos entornos.
- **Capítulo 4 - Diseño e Implementación del Sistema:** Se plantea los requerimientos para la implementación del sistema, con lo que se procede a desarrollar el diseño de la Red de Área Personal en base a la tecnología seleccionada. El diseño comprende el montaje y programación tanto del módulo receptor (dispositivo móvil) como de varios nodos (dispositivos fijos).
- **Capítulo 5 - Experimentación y Resultados:** Se muestran los resultados obtenidos de la implementación de los módulos del sistema en un ambiente real. Se realiza una implementación tipo prototipo en las instalaciones del

edificio Bloque B de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid

- **Capítulo 6 - Conclusiones y Líneas Futuras:** Se presenta las conclusiones derivadas del desarrollo del presente trabajo. Para finalmente, analizar y proponer posibles líneas de trabajo futuras, que evidencie el potencial de desarrollo de las tecnologías IoT.
- **Bibliografía**

2 Redes Inalámbricas

Uno de los factores que incentivó el desarrollo de esta tecnología es la necesidad de los usuarios de desplazamiento. Si bien la adopción de esquemas inalámbricos por parte del sector de la telefonía constituye uno de los más importantes espacios de aplicación y despliegue cada vez son más los ambientes en los que la tecnología inalámbrica se ha incorporado.

Algunas de las ventajas de las redes inalámbricas, a parte de la movilidad, son la facilidad de despliegue ya que únicamente se necesita la instalación de algunos nodos en poco tiempo para que la red empiece a dar servicio; otra ventaja es la facilidad de crecimiento ya que la estructura de la red es la misma para dar conectividad a uno o varios terminales, así también el incorporar nuevos usuarios implica una autorización.

Si bien la velocidad de transmisión de los datos y la seguridad son han sido aspectos limitantes para las redes inalámbricas, con el despliegue de investigaciones esto se ha ido superando paulatinamente.

Las redes inalámbricas son un conjunto de dispositivos interconectados principalmente a través de ondas electromagnéticas; el tipo de tecnología a implementarse depende de su ámbito de cobertura por lo que se clasifican como se visualiza en la Figura 1.

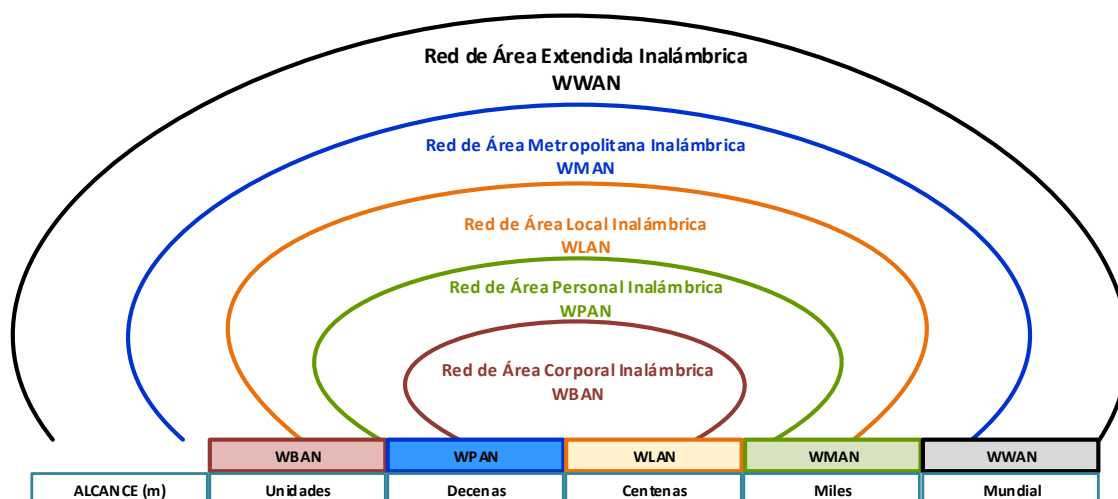


Figura 1. Clasificación de las Redes Inalámbricas por su Ámbito de Cobertura

Para el propósito de este trabajo se abordarán los ámbitos WPAN, WLAN y otras tecnologías (catalogadas por convención en WPAN) como RFID.

2.1 WLAN (Wireless Local Area Network)

Tecnología inalámbrica cuyo ámbito de cobertura son las centenas de metros (habitualmente 100 m). En lo referente a la velocidad de transmisión de datos, la tecnología ha ido evolucionando desde 1 MBps hasta llegar a los 1.3 GBps, con una tendencia creciente a futuro.

Generalmente se trata de una tecnología que despliega una infraestructura de equipos para comunicar un conjunto de dispositivos a través de señales de radio frecuencia.

Un conjunto importante de organismos e industrias han colaborado para normalizar las WLAN, pero ha sido IEEE mediante el estándar IEEE 802.11, el organismo que ha dominado la normalización este tipo de redes.

2.2 WPAN (Wireless Personal Area Network)

Son redes inalámbricas de corto alcance (decenas de metros). Usadas generalmente para interconectar dispositivos periféricos. Habitualmente la velocidad requerida para transmisión de datos entre éstos los dispositivos es relativamente baja por lo que su nivel de consumo de potencia tiene la misma tendencia.

Se trata de redes que usan eficientemente sus recursos, pues no existe un despliegue de infraestructura y protocolos de comunicación complejos, como sucede en otro tipo de tecnologías. Con lo especificado con anterioridad este tipo de tecnologías se emplean en dispositivos móviles pequeños.

El ámbito de la WPAN se asocia con la propiedad de dispositivos electrónicos, por ello se puede indicar este tipo de red es un espacio personal de comunicaciones, dicho espacio se mueve con su propietario y se pueden anexar a éste otros dispositivos.

2.2.1 Grupos de Trabajo WPAN de la IEEE (Institute of Electrical and Electronics Engineers) 802.15: [1]

Grupos establecidos por la IEEE, con investigación y desarrollo de objetivos específicos en torno a tecnologías WPAN. Los grupos de trabajo conforman a su vez grupos de tareas (identificados por una letra), quienes se encargan principalmente de la revisión de algunas particularidades de cada norma. Así para éste y otros numerales se describe brevemente el trabajo desarrollado por cada grupo de trabajo y de tareas.

- **IEEE 802.15.1 (TG1- Task Group 1):** Desarrolla el estándar de Bluetooth 1.1 definiendo el nivel físico y de control de acceso al medio.

- **IEEE 802.15.2 (TG2):** Trabaja sobre mecanismos de resolución de problemas en la coexistencia de las WPAN con las WLAN (Wireless Local Area Network).
- **IEEE 802.15.3 (TG3):** Aborda temas referentes al desarrollo de tecnologías WPAN de alta velocidad sobre aplicaciones multimedia, con bajo consumo de energía y bajos costes. Se desarrollaron básicamente las especificaciones para la capa física y enlace de datos, así como mecanismos de seguridad mediante encriptación de información con AES 128 (Advanced Encryption Standard).
 - **IEEE 802.15.3a (TG3a):** Apunta específicamente a alcanzar velocidades de transmisión en el orden de los cientos de Mbps Tiene los mismos objetivos que el grupo IEEE 802.15.3 pero se añade la estandarización del uso de la banda de radio frecuencia UWB (Ultra Wide Band). Determinó características y criterios de selección de la capa física para WPAN, pero heredó de IEEE 802.15.3 el uso de la capa MAC.
 - **IEEE 802.15.3b (TG3b):** Trabaja sobre la corrección y optimización de la implementación e interoperabilidad de la sub-capa MAC. De hecho, una de sus últimas versiones (IEEE 802.15.3-2005) incluye a la sub-capa MAC (Medium Access Control) para la especificación IEEE 802.15.3-2003.
 - **IEEE 802.15.3.c (TG3c):** Su objetivo es implementar la capa física sobre ondas milimétricas para mejorar el estándar IEEE 802.15.3-2003. Su rango de operación es sobre la banda de frecuencia 57 a 64 GHz con tasas de transmisión muy elevadas por sobre el 1 GBps.
- **IEEE 802.15.4-2003 (TG4):** Su investigación se centra en el desarrollo de soluciones para sistemas con baja transmisión de datos, pero amplia autonomía energética. Un ejemplo de tecnología que se basa en este estándar es ZigBee. Este estándar fue reemplazado por IEEE 802.15.4-2006 y puesto en hibernación por TG4b en marzo de 2004, hasta su publicación final. Entre sus características más relevantes son que, la transferencia de datos se encuentra en el orden de 20, 40 y 250 KBps.
 - **IEEE 802.15.4a-2007 (TG4a):** Es una enmienda a IEEE 802.15.4-2006, que apunta específicamente a la capa física en lo referente a establecer comunicaciones que incorporen facilidades de localización (con alta precisión) en espacios reducidos. Busca mejorar la escalabilidad en velocidad, distancia, coste y consumo de potencia, así como mayor robustez contra interferencias y movilidad.
 - **IEEE 802.15.4b (TG4b):** Realizó mejoras sobre IEEE 802.15.4-2003, cuyo objetivo era la resolución de ambigüedades, reducción de

complejidad innecesaria empleo de nuevos rangos de frecuencia. Justamente una vez aprobado el presente estándar se publicó como IEEE 802.15.4-2006.

- **IEEE 802.15.4c (Enmienda 2):** Propone un conjunto de especificaciones alternativas para la capa física (con sus correspondientes modificaciones en MAC), esto en concordancia de los cambios en la normativa de espectro de frecuencias en China.
- **IEEE 802.15.4d (Enmienda 3):** Establece una modificación a IEEE 802.15.4-2003, que permite, a través de cambios en la capa física y control de acceso al medio, soportar de bandas de frecuencias japonesas. Se posibilita también, la coexistencia con otros protocolos que utilizan la misma banda de frecuencias.
- **IEEE 802.15.4e (TG4e):** Incorporó cambios en la capa MAC, en el estándar IEEE 802.15.4-2006; su objetivo fue dar soporte al mercado industrial y permitir compatibilidad con las modificaciones de WPAN en China.
- **IEEE 802.15.4f (TG4f):** Instaure mejoras a IEEE 802.15.4-2006, a nivel físico orientado al soporte de despliegues masivos de RFID y algunas adiciones asociadas a MAC.
- **IEEE 802.15.4g (TG4g):** Grupo de trabajo conformado para realizar modificaciones sobre la capa física del estándar IEEE 802.15.4 para facilitar aplicaciones de control de procesos a gran escala en algunos ámbitos de las redes.
- **IEEE 802.15.5:** Especificación que define la arquitectura que permite a dispositivos WPAN establecer topologías de malla inalámbrica, interoperables, estables y escalables.
- **IEEE 802.15.6 (TG6):** Estándar que explica las características de las capas MAC y PHY (Physical) para la tecnología WBAN (Redes de Área Corporal Inalámbricas), priorizando la optimización de comunicación entre nodos de bajo consumo y el uso de bandas de frecuencia médicamente recomendadas.
- **IEEE 802.15.7 (TG7):** Incorpora definiciones para las capas física y control de acceso al medio para comunicación de corto alcance mediante espectros de luz visible a velocidades altas (896 MBps).
- **IEEE 802.15.8 (TG8):** Define las capas PHY y MAC para WPAN conocidas como PAC (Peer Aware Communications). PAC está optimizado para comunicaciones P2P sin infraestructura con coordinación totalmente distribuida.

- **IEEE 802.15.9:** Define un marco de intercambio de mensajes basado en IE (Elementos de Información) como método de transporte para datagramas KPM (Key Management Protocol) y directrices para el uso de algunas KPM.
- **IEEE 802.15.10 (TG10):** Se centró en el enrutamiento (L2R -Layer 2 Routing) de paquetes en redes inalámbricas 802.15.4 que cambian dinámicamente.

2.3 Red Inalámbrica AD HOC

Es una red descentralizada, en la que únicamente participan nodos para la transmisión de datos, se trata de una red de pares donde cualquier nodo se puede asociar con otro dentro de su rango de alcance.

Características:

- Su alcance es limitado.
- Los nodos de la red pueden tomar cualquier posición; por lo que sus enlaces no son permanentes.
- Para añadir nodos a la red se requiere únicamente encontrarse dentro del alcance de un nodo perteneciente a esa red.
- Los nodos se conectarán mediante enlaces (links), y la comunicación entre nodos se la realiza mediante rutas (colección de enlaces)
- El reenvío de información se realiza a través de los nodos de la red, lo que implica introducir un retardo adicional por cada nodo que retransmite la información.
- El enrutamiento se lo realiza mediante Clustering, que implica el uso de un algoritmo híbrido y el agrupamiento de nodos en clusters, donde cada nodo calcula sus tablas de enrutamiento según su necesidad y cada clúster calcula sus tablas periódicamente, esto permite optimizar los recursos de red.
- Se deben implementar mecanismos para identificar los servicios con los que cuenta la red.
- La portabilidad hace que los nodos estén sujetos a limitantes energéticas.
- En ocasiones el reenvío de datos se lo realiza mediante flooding (inundación de la red).

2.3.1 Tipos

- **MANETS (Mobile Ad-Hoc Networks):** se trata de una topología lógica mallada conformada por dispositivos móviles (con propiedades de autoconfiguración).
- **Red inalámbrica mesh:** red de comunicaciones organizada en una topología de malla. Combinan una red de infraestructura inalámbrica (con dispositivos intermedios) a las características habituales de una red ad hoc, de esta forma un nodo que se encuentre

fuera de su ámbito de cobertura puede unirse a través de nodos que si pertenezcan a la red.

- **Red de sensores:** conformadas por un gran número de sensores inteligentes, recoge datos en crudo y se les da valor añadido mediante el análisis y procesado de datos. Se trata de redes auto configurables, en donde cada nodo puede actuar como emisor, receptor o enrutador de la información, en estas redes se añade el componente de nodo recolector. Mediante protocolos como AODV (Ad hoc On-Demand Distance Vector), DSDV (Destination-Sequenced Distance Vector Routing), EWWA (Exponentially Weighted Moving Average Algorithms) permite las comunicaciones multi-hop y mensajería broadcast.

2.4 IEEE 802.11-2012

Es una tecnología WLAN empleada en ambientes de hogar y empresariales ya que sus diferentes estándares abarcan una amplia gama de velocidades y alcance.

Los dispositivos que componen la red comparten el medio inalámbrico por lo que se establece varios métodos de control de acceso al medio.

La normativa base define la sub-capa MAC y dos sub-capas físicas, las cuales se desarrollarán más adelante.

Dentro del desarrollo de la presente tecnología es pertinente mencionar que el término Wi-Fi es una marca comercial empleada por Wi-Fi Alliance, institución que se centra principalmente en la certificación de interoperabilidad y no en el desarrollo de estándares por lo que complementa esfuerzos con IEEE [2]. Así también existen otros organismos que colaboran con el avance de la tecnología inalámbrica como UIT-R, FCC, IETF e ISO.

IEEE 802.11 estandariza, para su propósito, los mecanismos de comunicación en la capa física y la sub-capa MAC del modelo OSI (Open System Interconnection), respecto a ésta última sub-capa y previo al desarrollo del tema se definen algunos conceptos útiles; la PDU (Protocol Data Unit) proveniente de la sub-capa LLC (Link Logical Control), que comprende la información añadida por LLC y la información de las capas 3 y 7, en sub-capa MAC se denomina MSDU (MAC Service Data Unit), cuando a ésta se le agrega una cabecera MAC y el campo FCS (Frame Check Sequence) se convierte en MPDU (MAC PDU).

2.4.1 Grupos de estandarización 802.11

Las contribuciones para estandarizar las redes inalámbricas provienen de múltiples ámbitos; pero principalmente de la IEEE quien inició la estandarización en 1991 y ratificó el estándar inicial para 1997 [2]. El grupo de trabajo IEEE 802.11 se encarga del desarrollo

de estándares LAN inalámbrico, la Tabla 1 muestra un resumen de algunos de los principales estándares que se han desarrollado en torno a IEEE 802.11.

Tabla 1. Características de los Principales Estándares IEEE 802.11 [3]

Estándar IEEE	Frecuencia de Operación (GHz)	Velocidad Máxima de Datos (Mbps)	Modulación
IEEE 802.11	2.4	2	DQPSK
IEEE 802.11a	5	54	64-QAM $\frac{3}{4}$
IEEE 802.11b	2.4	11	DQPSK con CCK
IEEE 802.11g	2.4	54	64-QAM $\frac{3}{4}$
IEEE 802.11n	2.4 o 5	600	64-QAM $\frac{5}{6}$
IEEE 802.11ac	5	1300	256-QAM

El estándar original constituye IEEE 802.11 constituye el eje sobre el cual se han realizado mejoras en forma de enmiendas ratificadas. El último documento publicado que incluye varias enmiendas es el IEEE 802.11-2012

2.4.2 Componentes

Para permitir la comunicación inalámbrica entre dispositivos en una WLAN se necesita una infraestructura que se compone de varios dispositivos que se mencionan a continuación:

- **Puntos de acceso inalámbrico (AP):** Inicialmente eran dispositivos semidúplex en el que solo una radio transmitía a la vez, con el progreso de la tecnología estos dispositivos han ido incorporando inteligencia y pareciéndose cada vez más a un switch o incluso a un controlador capaz de dirigir el tráfico en capa 2. Actualmente la mayoría de entornos emplea los ISR (Integrated Services Routers), que implementan funciones de router, switch y punto de acceso en el mismo dispositivo.
- **Estaciones:** Dispositivos que cuentan con interfaces de red inalámbrica compatibles con el estándar de comunicaciones del punto de acceso.
- **Servicio de integración:** Definido en el estándar IEEE 802.11-2012, posibilita el intercambio de MSDU entre el sistema de distribución y una LAN no IEEE 802.11, a través de un puente (típicamente un punto de acceso o controlador WLAN). La función principal del servicio de integración es eliminar el encabezado y el tráiler y luego encapsular la MSDU en una trama de acuerdo a la tecnología a la que se transfiera, este servicio también se encargará de realizar el proceso contrario para transferir datos hacia el entorno IEEE 802.11.

- **Sistema de Distribución (DS):** Componente que se establece en IEEE 802.11-2012, se emplea para crear un ESS (Extended Service Set) mediante la interconexión de un conjunto de BSS (Basic Service Sets) a través de LAN integradas. El SD dispone de:
 - **Medio de Sistema de Distribución (DSM):** medio físico lógico que conecta puntos de acceso, generalmente IEEE 802.3.
 - **Servicios de Sistema de Distribución:** Software colocado en el punto de acceso para proveerle de una inteligencia similar a un switch de red cableada.
- **Sistema de Distribución Inalámbrico (WDS):** Insertado por IEEE 802.11-2012, establece un mecanismo de comunicación inalámbrico con un formato de trama de cuatro direcciones MAC. Este mecanismo viabiliza la comunicación inalámbrica de dos AP situados a una distancia adecuada mediante un backhaul inalámbrico. Otra aplicación de WDS son los AP repetidores que no se conectan al backbone cableado, permitiendo así la transmisión de datos entre dispositivos finales y otro AP que se conecte a una red cableada.
- **Identificador del Conjunto de Servicios (SSID):** Se trata de un identificador (nombre) lógico de la WLAN.

2.4.3 Topologías de Red

Los dispositivos en las redes inalámbricas se pueden asociar de forma lógica, dando lugar a un conjunto de servicios (Service Set). Previo al tratamiento de las topologías se abarca dos conceptos básicos:

- **BSSID (Basic Service Set Identifier):** Identificador de capa enlace de datos de un BSS, que consiste en un número de 48 bits, generalmente una MAC.
- **BSA (Basic Service Area):** Conocido también como celda, representa el área de cobertura de un punto de acceso. Su tamaño depende de las características del AP como potencia, la ganancia de la antena y otros factores como el entorno.

Las WLAN permiten la implementación de los siguientes tipos de topología:

- **BSS (Basic Service Set):** Grupo de dispositivos que se comunican mediante IEEE 802.11, donde el AP es el punto central que permite la comunicación entre estaciones como se observa en la Figura 2. El AP puede conectarse a sistema de distribución cableado mediante un puerto. El BSSID de una BSS es la dirección MAC de la interfaz inalámbrica del punto de acceso. Su área de cobertura es un BSA.

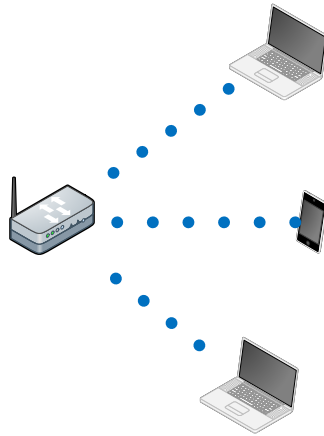


Figura 2. Topología BSS

- **IBSS (Independent BSS):** Conjunto de estaciones que utilizan sus interfaces inalámbricas para comunicarse entre sí sin necesidad de un punto de acceso (Ver Figura 3), en otras palabras, la comunicación entre estaciones es peer-to-peer, considerando que todas las estaciones se conectan mediante en el mismo canal de frecuencia y SSID. Se emplea el BSSID para identificar al IBSS, pero en este caso el identificador es generado aleatoriamente por parte del primer dispositivo que se inicia en la IBSS, el cual se trata d una dirección MAC virtual. Típicamente esta topología no se conecta a un sistema de distribución.

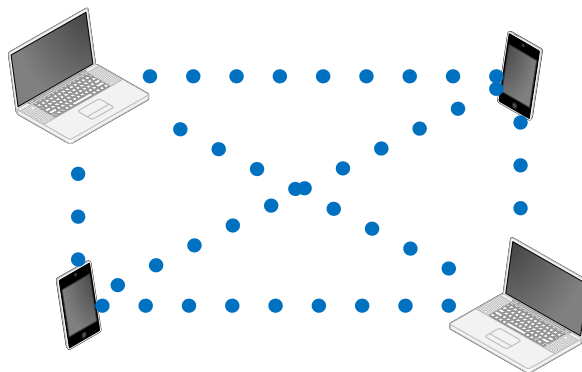


Figura 3. Topología IBSS

- **ESS (Extended Service Set):** Topología que se compone de dos o más BSS interconectados mediante un sistema de distribución, con el fin de proporcionar a sus estaciones un rango de cobertura más amplio, un medio inalámbrico y backbone que actúe como una conexión de capa enlace de datos única e itinerantica sin interrupciones entre celdas de cobertura, aunque éste último parámetro puede irse ajustando de acuerdo a distribución de los AP. La operación coordinada de los AP permite que las estaciones puedan comunicarse entre sí independientemente de su ubicación dentro de la ESS. En la mayoría de aplicaciones de esta topología el SSID es el mismo para todos los AP por lo que se le denomina Identificador del Conjunto de Servicios

Extendido (ESSID); aunque el ESSID sea el mismo el BSSID es único para cada BSA.

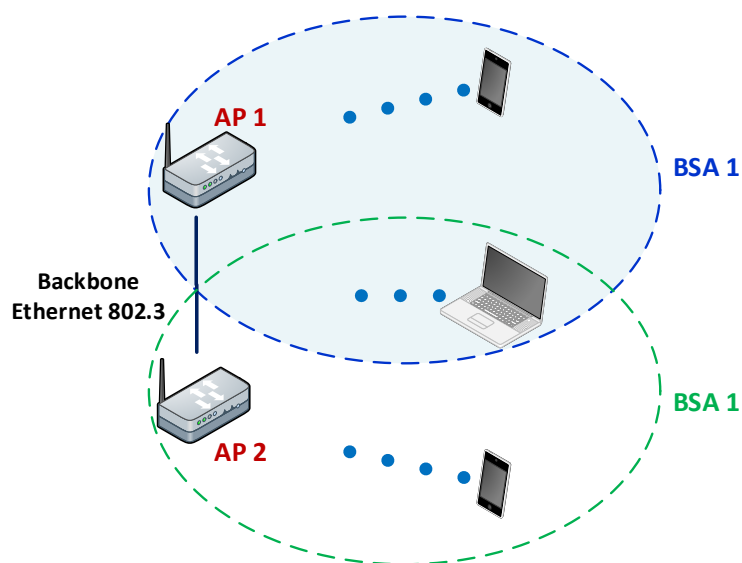


Figura 4. Topología ESS

- **MBSS (Mesh BSS):** Topología que la incluye el estándar IEEE 802.11-2012, se basa en APs que soportan funciones de malla para extender la red a espacios a donde se puede llegar mediante una infraestructura cableada (Ver Figura 5), el conjunto de APs asociados forma un MBSS. La presente topología se compone de:

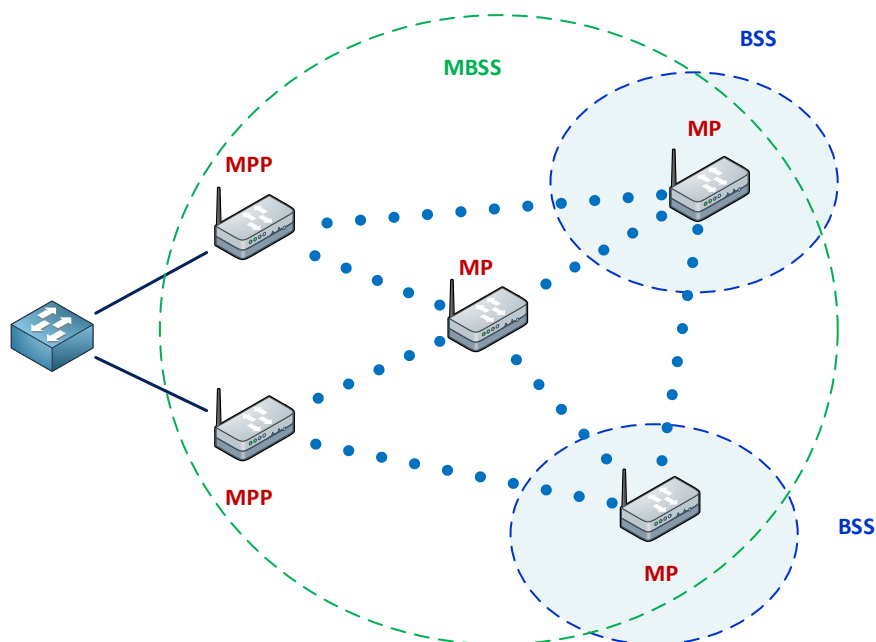


Figura 5. Topología MBSS

- **MPP (Mesh Point Portal):** Se trata de uno o varios APs conectados a un backbone, típicamente cableado
- **MP (Mesh Point):** Son los AP que se enlazan mediante enlaces backhaul para llegar al backbone.

Los nodos con funciones de malla realizan funciones de “enrutamiento” dinámico en capa 2, basado en direcciones MAC mediante el protocolo HWMP (Hybrid Wireless Mesh Protocol).

2.4.4 Stack de Protocolos

El rápido desarrollo de la tecnología inalámbrica propicia el cambio constante de las características de los protocolos. En el presente numeral se abordará las características del protocolo base y que se muestran en la Figura 6, de esta forma se abordará fácilmente las modificaciones al protocolo.

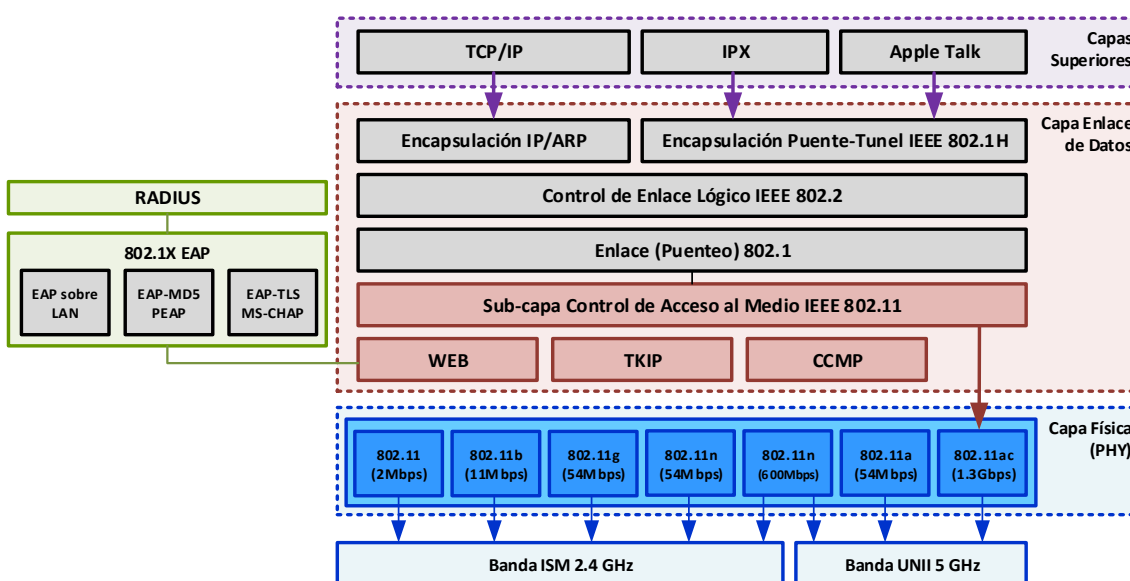


Figura 6. Stack de Protocolos Base IEEE 802.11

Capa Física

Como se visualiza en la Figura 6 para esta capa se han definido múltiples estándares que han ido evolucionando y heredando funciones. La capa física consta de las siguientes sub-capas:

- **PLCP (Physical Layer Convergence Procedure):** Acepta MPDU de la sub-capa MAC mediante el SAP de capa física; ya en capa física las MPDU toman el nombre de PSDU (PLCP Service Data Unit), al añadirse tanto la cabecera y tráiler a la PSDU ésta toma el nombre de PPDU (PLCP Protocol Data Unit). Se encargará también de entregar tramas a la subcapa MAC provenientes del medio.
- **PMD (Physical Medium Dependent):** Transmite y recibe tramas reales IEEE 802.11, al interactuar directamente con el medio aire provee a las tramas modulación y demodulación.

Actualmente el estándar guarda compatibilidad con estándares anteriores por lo que es importante enunciar algunas de las capas físicas que se han definido: FHSS (Frequency-Hopping Spread Spectrum PHY), DSSS (Direct-Sequence Spread Spectrum),

IR (Infrared), las tres mencionadas capas fueron establecidas en el estándar original IEEE 802.11; OFDM (Orthogonal Frequency-Division Multiplexing) definido en el estándar IEEE 802.11a; HR-DSSS (High-Rate DSSS) adoptado para la versión IEEE 802.11b.

Entre las principales funciones de la presente capa están:

- La detección de portadora está dirigida por la sub-capa PMD, así se comprueba si el medio está ocupado o inactivo. En el caso de que una estación no esté transmitiendo o recibiendo tramas PLCP se encarga de, la detección de señales entrantes, censando el medio continuamente; y de realizar una evaluación de canales claros, en donde el PLCP informará a MAC si el medio está inactivo u ocupado para que la sub-capa MAC tome la decisión de enviar tramas.
- Para ejecutar el proceso de transmisión MAC establece los parámetros de transmisión (velocidad y cantidad de bytes), con dicha información las sub-capas PLCP y PMD se encargan de la sincronización y ajustes de comunicación para que los datos puedan ser llevados a través del medio inalámbrico.
- En la recepción de tramas PLCP se encargará de detectar que una trama entrante ha sido recibida correctamente, posteriormente se encargará de informar a MAC datos de la trama antes de ejecutar la desencapsulación.

Subcapa de Control de Acceso al Medio (MAC)

Subcapa que cumple con un conjunto de funciones para la operación del estándar IEEE 802.11; dichas funciones comprenden:

- Posibilitar la transmisión fiable de datos de usuario sobre el aire mediante un protocolo de detección de portadora.
- Se encarga del control de acceso al medio con el objetivo de que las estaciones compartan a misma frecuencia y espacio en el espectro radioeléctrico.
- Interactúa con la subcapa LLC y la capa física.
- Plantea opcionalmente la gestión del uso de energía, enfocándose a estaciones que usan baterías. Los modos de ahorro de energía sugeridos son, SMPS (Spatial Multiplexing Power Save) y PSMP (Power Save Multi-Poll)
- Se permite la fragmentación en caso de que una trama proveniente de otra red supere el tamaño establecido dentro de la red inalámbrica; así también, para superar problemas de ruido e inestabilidad con lo que se disminuye el número de posibles retransmisiones. Cada fragmento consta de un número de secuencia y se confirmará su recepción mediante el protocolo stop&wait.

- Establece mecanismos de recuperación de errores como: acuses de recibo de tramas de datos unicast; ajustes de velocidad frente a retransmisiones excesivas.
- Determina algunos métodos de cifrado, entre los que se tiene a: WEP (Wired Equivalent Privacy), TKIP (Image Temporal Key Integrity Protocol) y CCMP (Counter Mode Cipher Block Chaining Message Authentication Code Protocol, Counter Mode CBC-MAC Protocol).
- Presenta la función opcional de RTS/CTS (Request-to-Send/Clear-to-Send), con el objetivo de minimizar la probabilidad de colisiones entre estaciones ocultas y agregar mejoras al control de acceso al medio, bajo este esquema una estación evita el envío de tramas de datos 802.11 hasta que la estación complete el establecimiento de enlace (handshake) RTS/CTS con otra estación; además se incluye que otras estaciones no pueden acceder al medio mientras la estación que inicia RTS transmite sus datos.

La norma específica tres diferentes tipos de acceso al medio:

- **DCF (Distributed Coordination Function):** Implementa CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) como método de control de acceso al medio. Para censar si el canal está ocupado, se utiliza una combinación de mecanismos físicos y virtuales de detección de portadora.
- **PCF (Point Coordination Function):** Definido en el estándar original e incluye una función de sondeo que indica el momento en que una determinada estación puede transmitir, proporcionando así acceso sincrónico en función del tiempo. Su implementación fue muy limitada.
- **HCF (Hybrid Coordination Function):** Se anexo al estándar mediante IEEE 802.11e para el soporte de calidad de servicio. Incluye dos métodos de acceso:
 - **EDCA (Enhanced Distributed Channel Access):** Agrega a DCF múltiples colas cada una con una prioridad de acceso diferente, es así que con la intervención de EDCAF (EDCA Function) y su representación AIFS (Arbitration Interframe Space), es posible transmitir primero el tráfico con mayor prioridad y luego el tráfico con menor prioridad.
 - **HCCA (HCF Controlled Channel Access):** Mediante HC (Hybrid Coordinator) controla las transmisiones desde las estaciones evitando la contienda por el medio. Añade ciertas mejoras a PCF, como la transmisión de múltiples tramas en lugar de una sola.

Otra función que cumple la subcapa MAC es la incorporación de estaciones a la red, que incluye sondeo de la red, autenticación, asociación con un AP, establecimiento del enlace a cuatro vías (4-way handshake) en caso de usarse TKIP o CCMP.

En la enmienda IEEE 802.11n se incorpora la agregación de tramas para reducir la sobrecarga en el medio, así varias tramas comparten la misma cabecera y tráiler; se definen dos tipos de agregación de tramas:

- **Agregación MSDU (A-MSDU):** agrupa tramas Ethernet, que son una MSDU separada, con lo que se obtiene una MPDU 802.11.
- **Agregación MPDU (A-MPDU):** encapsula varias MPDU, a excepción del campo FCS, en una sola trama 802.11.

Controla la temporización y sincronización de las estaciones debido al uso del canal compartido. El estándar determina cuatro tipos de intervalos de espacio inter-trama (IFS, Interframe Space), que se describen de menor a mayor como sigue:

- **SIFS (Short IFS):** Es el intervalo más corto del estándar, proporciona el nivel de prioridad más alto para el acceso al medio. Está asociado a las tramas ACK, CTS (Clear-to-Send), y la segunda o subsiguientes MSDU de una ráfaga de fragmentos.
- **PCF IFS:** Intervalo de acceso al medio empleado por las estaciones que trabajan mediante el esquema PCF.
- **DCF IFS:** Lo usan las estaciones que operan bajo el esquema DCF para el envío de tramas de datos y control.
- **Extended IFS:** También lo usan las estaciones que trabajan con el esquema DCF para postergar el envío de una trama en caso de que la trama haya sido recibida incorrectamente por un valor de FCS inválido.

Mecanismos de comunicación: Si bien la mayor cantidad de comunicaciones son unicast, el estándar también admite otras formas de comunicación como:

- **Multicast y Broadcast:** Bajo estos esquemas la estación transmisora identifica al grupo de estaciones que puede recibir y procesar una trama. El proceso lo inicia la estación mediante el envío de una trama multicast en formato unicast, subsecuentemente el punto de acceso responderá con un ACK a la estación emisora para posteriormente enviar la trama multicast (o broadcast) al grupo correspondiente, una vez entregadas las tramas a los destinatarios éstos últimos, de acuerdo a lo establecido, no responderán con un ACK.

2.4.5 IEEE 802.11ac-2013

Evolución de IEEE 802.11n creada para superar la velocidad de transmisión de datos de 1 Gbps y operar en frecuencias menores a 6 GHz. IEEE 802.11ac es una iniciativa a cargo del grupo de estudio VHT (Very High Throughput) de IEEE 802.11. Algunas de sus características principales se describen a continuación:

- Usa una de las técnicas pioneras para aumentar la velocidad, MU-MIMO (Multi-User - Multiple Input Multiple Output) que permite el envío de datos de varios clientes a la vez, mediante la asignación, por parte del AP, de uno o varios flujos espaciales a cada estación.
- Soporta las formas de modulación: BPSK, QPSK, 16-QAM, 64-QAM y 256-QAM.
- Establece nuevos tamaños de canal de 80 MHz y 160 MHz.
- Trabaja en la banda de frecuencia no licenciada de 5 GHz, esto en función del tamaño de canal y la forma de codificación que se maneja; a o que se añade la poca interferencia por parte de otras tecnologías en la banda especificada.
- Con todas las innovaciones mencionadas la transmisión de datos alcanza la velocidad de 1.3 Gbps para canales de 80 MHz y 2.6 Gbps en canales de 160 MHz [4].
- Todas las tramas de datos se transmiten usando la forma de agregación MPDU (A-MPDU).

2.4.6 Formato de la PDU

El desarrollo del presente numeral se lo realiza en torno al formato de las PDU definidas en el estándar IEEE 802.11ac-2013.

PDU VHT

El objetivo principal del diseño en capa física fue mantener compatibilidad con estándares anteriores para lo cual se guarda similitud con la trama IEEE 802.11a; además se añade un nuevo encabezado para habilitar las transmisiones MIMO multiusuario. Su estructura se observa en la Figura 7 y se describe subsecuentemente:

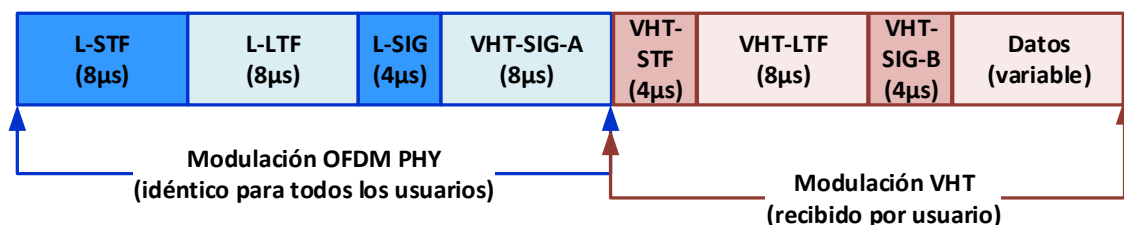


Figura 7. Distribución esquemática de la Trama Física VHT

- **L-STF y L-LTF (Non-HT Short Training Field y Non-HT Long Training Field):** Secuencia de 12 símbolos OFDM usado para la sincronización y selección de una antena por parte del receptor.

- **L-SIG (Non-HT Signal):** En 802.11a se usa en recepción para calcular el tiempo de duración de la trama, ya que contiene la información de velocidad de datos y la longitud de la trama. En ambientes IEEE 802.11ac se establece la velocidad den 6 MBps y una longitud falsa en bytes, así el cálculo de la longitud de la trama por parte de cualquier receptor coincide con la longitud de la trama 802.11ac.
- **VHT-SIG-A y VHT-SIG-B (VHT Signal A y Signal B):** Campos que describen los parámetros usados en la transmisión de la trama encapsulada como el ancho de banda del canal, modulación, codificación, tipo de trama (de usuario único o multiusuario). Cada campo en particular tiene subdivisiones que se detalla como se muestra a continuación:
 - **VHT-SIG-A: La información contenida en este campo la reciben todos los usuarios.** Su representación puede ser de usuario único o multiusuario. La representación para usuario único se divide en VHT-SIG-A1 (Ver Figura 8) y VHT-SIG-A2 (Ver Figura 9) que representan un símbolo OFDM.

Ancho de Banda (2b)	Reservado (1b)	STBC (1b)	ID de Grupo (6b)	Número de Flujos espacio-tiempo (3b)	AID Parcial (9b)	TXPS forbidden (1b)	Reservado (1b)
---------------------	----------------	-----------	------------------	--------------------------------------	------------------	---------------------	----------------

Figura 8. Formato del Sub-campo VHT-SIG-A1

GI Corto (1b)	Desambiguación Corta GI (1b)	Codificación (1b)	Símbolo Extra LDPC (1b)	VHT-MCS (4b)	Beamform med (1b)	Reservado (1b)	CRC (8b)	Cola (6b)
---------------	------------------------------	-------------------	-------------------------	--------------	-------------------	----------------	----------	-----------

Figura 9. Formato del Sub-campo VHT-SIG-A2

- **VHT-SIG-B:** Información que se particulariza para cada usuario. Se encarga de configurar la velocidad de datos y sincronizar la recepción MIMO [4]. Se transmite como un símbolo OFDM y consta de los campos que se muestran en la Figura 10.

Longitud B de Señal VHT (17, 19, 21b)	Reservado (3, 2b)	Cola (6b)
---------------------------------------	-------------------	-----------

Figura 10. Formato del Campo VHT-SIG-B

- **VHT-STF (VHT Short Training Field):** Se encarga de apoyar al receptor a detectar el patrón de repetición y establecer la ganancia del receptor.
- **VHT-LTF (VHT Long Training Field):** Su contenido es una secuencia de símbolos que depende del número de flujos transmitidos, estos símbolos establecen la demodulación del resto de la trama iniciando por VHT Signal B, se usa también para el proceso de estimación de canal.
- **Datos:** Contiene la PDU de capa superior o una trama de agregación; pero a su vez consta de los campos que se muestran en la Figura 11:

Servicio (16b)	PSDU (variable)	PHY pad	Cola (6b)
-------------------	--------------------	---------	--------------

Figura 11. Formato del campo Carga Útil

Algo que se debe tomar en cuenta es que el campo **Service** (Servicio) contiene tres campos que se visualizan en la Figura 12.

Inicialización del Codificador (7b)	Reservado (1b)	VHT-SIG-B CRC (8b)
--	-------------------	-----------------------

Figura 12. Formato del Sub-campo Servicio

Formato General Trama MAC [5] [6]

La especificación IEEE 802.11ac, ha incorporado dos cambios a la trama original, en primera instancia aumenta el tamaño máximo de trama a más de 11000 Bytes [4]; y le da una nueva funcionalidad al campo control HT (High Throughput)

Control de Trama (16b)	Duración /ID (16b)	Dirección 1 (receptor) (48b)	Dirección 2 (emisor) (48b)	Dirección 3 (filtrado) (48b)	Control de Secuencia (16b)	Dirección 4 (opcional) (48b)	Control QoS (16b)	Control HT (32b)	Cuerpo de la Trama (0-91408b)	FCS (32b)
---------------------------	-----------------------	---------------------------------	-------------------------------	---------------------------------	-------------------------------	---------------------------------	----------------------	---------------------	----------------------------------	--------------

Figura 13. Distribución Esquemática de la Trama General MAC

- **Control de Trama:** Se divide en los sub-campos que se muestran en la Figura 14:

Versión de Protocolo (2b)	Tipo (2b)	Subtipo (4b)	A DS (1b)	Desde DS (1b)	Más Fragmentos (1b)	Reprocesar (1b)	Gestión de Potencia (1b)	Más Datos (1b)	WEP (1b)	Orden (1b)
------------------------------	--------------	-----------------	--------------	------------------	------------------------	--------------------	-----------------------------	-------------------	-------------	---------------

Figura 14. Formato del Campo Control de Trama

- **Versión de Protocolo:** Indica la versión del protocolo IEEE 802.11.
- **Tipo y Subtipo:** Identifican si la trama MAC es de control, gestión o datos y sus correspondientes subtipos.
- **A DS (To DS):** Indica si la trama está destinada al sistema de distribución.
- **Desde DS:** Indica si la trama está proviene del sistema de distribución.
- **Más Fragmentos:** Especifica si la presente trama de datos o control es única o se debe esperar más fragmentos.
- **Reprocesar:** Señala si la trama se está retransmitiendo, con lo que se evita procesar tramas duplicadas.
- **Gestión de Potencia:** indica si la estación está en modo ahorro de energía (1) o no (0).
- **Más Datos:** Especifica a la estación receptora si el buffer del AP tiene datos por entregarle.
- **WEP:** Indica si se está usando la forma de cifrado WEP en la carga útil de la trama.

- **Orden:** Definido en 1 si la trama de datos usa la clase de servicio estrictamente ordenada.
- **Duración/ID:** Su contenido adquiere un determinado significado en función de varios aspectos entre los cuales están, si la estación accede al medio en modo ahorro de energía o no.
- **Dirección 1 (receptor):** Dirección MAC de la estación destino.
- **Dirección 2 (emisor):** Dirección MAC de la estación que transmite la trama.
- **Dirección 3 (filtrado):** Identifica al AP que es el destinatario inmediato de la trama
- **Control de Secuencia:** Indica el número de secuencia y fragmento de la trama.
- **Dirección 4 (opcional):** Identifica al AP que transmitió la trama.
- **Control de calidad de servicio:** Se establece en tramas de datos con QoS (Quality of Service), contiene información asociada al flujo de tráfico que se transmite, se divide en los sub-campos que se muestran en la Figura 15.

TID (4b)	EOSP (1b)	ACK Proxy (2b)	MSDU (1b)	Límite TXOP / Tamaño de la Cola / Estado del Buffer (8b)
-------------	--------------	-------------------	--------------	---

Figura 15. Formato del Campo Control QoS

- **Control HT:** Cuando el sub-campo RDG se establece en 0, el presente campo será de tipo HT-Variant con el formato de IEEE 802.11n, mientras que establecido en 1 se trata de un campo VHT-Variant como se indica en la Figura 16; se trata de una serie de sub-campos que comunican la retroalimentación MCS (Modulation and Coding Set) [4], técnica que consiste en el intercambio de parámetros entre estaciones para ajustarse a la mejor velocidad de transmisión de datos.

VHT (1b)	Reservado (1b)	MRQ (1b)	Secuencia MRQ / STBC (3b)	Secuencia MFB / ID de Grupo (3b)	MFB (15b)	ID de Grupo (3b)	Tipo de Código (1b)	Tipo de Retroalimentación (1b)	MFB no Solicitado (1b)	AC (1b)	RDG (1b)
-------------	-------------------	-------------	------------------------------	-------------------------------------	--------------	---------------------	------------------------	-----------------------------------	---------------------------	------------	-------------

Figura 16. Formato del Campo Control HT

- **Cuerpo de la Trama:** Contiene la información de capa superior transportada por la trama.
- **Secuencia de Verificación de Trama:** contiene el CRC (Cyclic Redundancy Check) que se aplica a las cabeceras y cuerpo de la trama.

A la trama descrita se agregan las tramas de: gestión, que entre sus funciones tienen la de asociar dispositivos para establecer redes inalámbricas y las tramas de control, apoyan en la entrega de las tramas de datos. Así también las tramas de datos, gestión y control constan de subtipos.

2.5 IEEE 802.15.4

Establece las especificaciones para la capa física y la sub-capa de control de acceso al medio para LR-WPAN (Low-Rate Wireless Personal Area Network) con dispositivos de alta autonomía energética. Se definió en 2003 pero se modificó en 2004 cuya publicación final se dio para 2006.

El estándar IEEE 802.15.4 se creó en base a un conjunto de criterios, entre los cuales destacan: la implementación de redes con múltiples nodos, baja complejidad, baja velocidad de datos (menores a 250 Kbps), en una banda de frecuencia sin licencia; se añade a lo mencionado el ultra-bajo consumo de energía que permite una fácil instalación y movilidad para los componentes de la red, pero teniendo en cuenta que esto último implica una transmisión en un intervalo de radio comunicaciones corto.

Características:

- Si bien 802.15.4-2006 opera en bandas de radio frecuencia sin licenciamiento, como se visualiza en la Tabla 2 se han establecido tres bandas de frecuencia para su operación.
- Las bandas de frecuencia de los 868 y 915 MHz se caracterizan por tener menor interferencia respecto a la banda de los 2,4 GHz.
- En cuanto al alcance, el estándar indica que depende de los módulos de potencia de salida y el ambiente de operación del dispositivo. Por ejemplo, dispositivos estándar de 0 dBm en ambientes sin interferencia pueden alcanzar hasta 200 metros; mientras que en ambientes con interferencia típicamente se puede alcanzar hasta los 30 metros.
- El bajo consumo de potencia es uno de los ejes del estándar 802.15.4-2006, se define la posibilidad de que los nodos puedan hibernar o apagarse para lograr este propósito; incluso a través de esta característica, se brinda flexibilidad de instalación con la posibilidad plantear fáciles y rápidas modificaciones en la infraestructura de red. Para aprovechar de forma óptima el consumo de energía, el estándar propone que los dispositivos empleen ciclos de trabajo mediante el siguiente esquema, intervalos cortos de transmisión e intervalos largos entre transmisiones.

2.5.1 Componentes de IEEE 802.15.4

Para conformar una WPAN (802.15.4) se necesita dos dispositivos que se comuniquen mediante el mismo canal físico y que se encuentren dentro del mismo POS (Personal Operating Space). Éstos dispositivos pueden ser: FFD (Full-Function Device) o RFD (Reduced-Function Device); se debe considerar que en una WPAN deberá al menos existir un FFD para que coordine la red.

2.5.2 Topologías de Red.

Dependen de los requisitos de aplicación y pueden ser:

Estrella

Distribución en la que un controlador central (coordinador PAN) se comunica con varios dispositivos, el coordinador será responsable de iniciar, terminar o enrutar comunicaciones dentro de la red.

Los dispositivos dentro de la topología PAN tendrán direcciones únicas de 64 bits o en su lugar el controlador PAN le asignará al dispositivo (al momento de incorporarse a la red) una dirección corta de 16 bits; cualquiera sea el caso estas direcciones se utilizan para el establecimiento de comunicaciones.

Para formar una red en estrella se debe establecer un coordinador PAN, activando un FFD; luego de lo cual se escogerá un identificador PAN, para diferenciarla de otras redes existentes dentro de su radio de influencia. De esta forma el coordinador permitirá que otros dispositivos (FFD o RFD) se una a la red.

Peer-to-Peer

Posee un coordinador PAN, pero en este caso dentro del rango de cobertura de la red, cualquier dispositivo puede comunicarse con cualquier otro dispositivo.

Cada PAN tiene un identificador unívoco, el mismo que permite la comunicación entre dispositivos mediante direcciones cortas. Esta infraestructura posibilita también, la transmisión de datos entre distintas redes.

2.5.3 Arquitectura de Protocolos

La arquitectura 802.15.4 se encuentra definida en capas que dan servicio a las capas superiores. La arquitectura se basa en el modelo de referencia OSI (Open System Interconnection). A continuación, se definen las funciones principales de sus capas.

Capa Física (PHY)

Esta capa tiene como función principal la transición y recepción de datos mediante canales de radio; la configuración de algunas de sus funciones se realiza mediante un conjunto de constantes y atributos (*aTurnaroundTime*) se despliega a continuación:

- La activación y desactivación del transceptor de radio, Esta función se lleva a cabo mediante las instrucciones que la sub-capas MAC le da a capa física, indicándole que conmute entre los modos de operación del transceptor de radio (transmisión, recepción e hibernación). Para configurar esta acción se utiliza la constante *aTurnaroundTime*.

- Detección de energía (ED -Energy Detection) en un canal, se comprueba el nivel de potencia de un canal, sato que se usa por la capa de red para escoger el canal.
- Indicador de calidad de enlace (LQI - Link Quality Indicator) para paquetes recibidos.
- Evaluación de canales claros (CCA- Clear Channel Assesment) para CSMA-CA.
- Selección de frecuencia de canal
- Transmisión y recepción de datos.

Proporciona una interfaz entre la sub-capa MAC (mediante el firmware de RF) y el canal de radio físico (a través del hardware de RF). La capa física posee una entidad de gestión PLME (Physical Layer Management Entity), que proporciona las interfaces de servicio de gestión de capas y es la responsable de mantener la PHY PIB (PAN Information Base). Los servicios que ofrece mediante SAP (Service Access Point) son:

- **Servicio de datos:** accesible mediante PD-SAP (PHY Data SAP), proporciona un mecanismo para transporte de MPDU entre entidades de sub-capa MAC de iguales características.
- Por otro lado, su **servicio de gestión** (al que se accede por PLME-SAP) permite el transporte de comandos de gestión entre el MLME (MAC Sublayer Management Entity) y el PLME.

En la Tabla 2 se detalla las bandas de frecuencia formatos de modulación y propagación de este estándar.

Tabla 2. Bandas de Frecuencia y Velocidad de Datos para IEEE 802.15.4

Banda de Frecuencias (MHz)	Rango de Frecuencia (MHz)	Forma de Modulación	Velocidad de Datos (Kbps)
868/915	868-868.6	BPSK	20
	902-928		40
868/915 (opcional)	868-868.6	ASK	250
	902-928		250
868/915 (opcional)	868-868.6	O-QPSK	100
	902-928		250
2450	2400-2483.5	OQPSK	250

Se dispone de 27 canales (numerados del 0 al 26) por página de canal. La página 0 de canal se encuentran disponibles 27 canales en las tres bandas de frecuencia, 16 para la banda de los 2450 MHz, 10 disponibles en la banda de frecuencias de los 915 MHz y 1 canal para 868 MHz. Las páginas 1 y 2 se encuentran útiles 11 canales en dos bandas de frecuencia, 10 canales disponibles para la banda de los 915 MHz con modulación O-QPSK y 1 para 868 MHz mediante modulación ASK (Amplitude-Shift Keying).

Existen 32 páginas por canal disponibles (las páginas de la 3 a la 31 están reservadas para uso futuro).

Sub-capa de Control de Acceso al Medio (MAC)

Además de controlar el acceso al canal de radio físico, debe cumplir con las siguientes funciones:

- Proporcionar mecanismos de asociación y disociación de dispositivos a la PAN.
- Proporcionar control de acceso a canales compartidos (CSMA-CA).
- Generación de redes beacons si el dispositivo es un coordinador.
- Gestión y mantenimiento del mecanismo GTS (Guaranteed Time Slot).
- Facilitar una comunicación fiable entre entidades MAC.

Esta sub-capa proporciona una interfaz entre SSCS (Service-Specific Convergence Sublayer) y PHY. MAC contiene la entidad de gestión MLME que proporciona interfaces de servicio mediante las cuales se pueden invocar funciones de gestión de capas, además mantiene información de objetos gestionados de la sub-capa MAC en la MAC PIB. Los servicios que ofrece mediante acceso SAP son:

- **Servicio de Datos MAC**, al que se accede a través de MCPS-SAP (MAC Common Part Sublayer Service Access Point), éste soporta el transporte de SPDU (SSCS PDU) entre entidades SSCS pares.
- **Servicio de Gestión MAC**, al que se accede mediante MLME-SAP, mismo que permite el transporte de mensajes de gestión entre la capa superior y MLME.

En función del momento que el coordinador indique el inicio de la red, la presente sub-capa tiene dos modos de funcionamiento:

- **Modo Beacon:** Se trata de un modo de operación de la red en el que todos los elementos pueden pasar al modo de bajo consumo energético. Bajo este esquema, el coordinador genera un beacon de sincronización de forma periódica para establecer el momento en que los dispositivos envíen sus datos almacenados y regresen al modo de ahorro de energía. La topología malla no permite este modo de operación.

- **Modo no beacon:** En este modo tanto coordinador y routers deben estar listos para procesar datos en cualquier momento. Desaparece entonces, la sincronización de los nodos para ejecutar el envío de información y el paso al estado de reposo de coordinador y routers.

2.5.4 Formato de la PDU

PPDU (PHY PDU)

En el nivel físico la PDU se compone de tres espacios divididos en cuatro campos, como se observa en la Figura 17 y cuyas funciones generales se presentan a continuación:

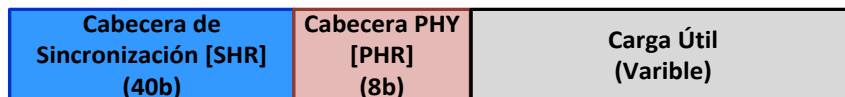


Figura 17. Formato PPDU

- **SHR (Synchronization Header):** Permite que un dispositivo receptor se sincronice y bloquee mediante un flujo de bits. Este campo a su vez se subdivide en (Ver Figura 18):
 - **Preámbulo:** Utilizado por el transceptor para obtener el chip y el símbolo de sincronización de un mensaje entrante.
 - **SDF (Start-of-Frame Delimiter):** Especifica el final del SHR y el inicio de datos de la trama.



Figura 18. Formato Campo SHR

- **PHR (PHY Header):** Establece la longitud de la información de la trama
- **Carga Útil:** Campo de longitud variable que lleva el contenido de capas superiores.

MPDU

A nivel MAC se definen varios tipos de PDU con el fin de reducir la complejidad del protocolo y permitir la transmisión de datos a través del medio inalámbrico.

En primera instancia se define un formato genérico con el fin de simplificar el desarrollo de los tipos de trama que se manejan. Así el **formato genérico de una trama MAC** está compuesto de tres campos principales y varios sub-campos, hay que considerar que según el tipo de trama algunos campos y sub-campos pueden variar.

- **MHR (MAC Header):** se divide en los campos que describen a continuación:
 - **Control de Trama:** Consta de los sub-campos que se describen subsecuentemente y se visualizan en la Figura 19.

Tipo de Trama (3b)	Seguridad Habilitada (1b)	Trama pendiente (1b)	Compresión de PAN ID (1b)	Reservado (3b)	Modo de Direccionamiento de Destino (2b)	Versión de la Trama (2b)	Modo de Direccionamiento de Origen (2b)
--------------------	---------------------------	----------------------	---------------------------	----------------	--	--------------------------	---

Figura 19. Formato del Campo Control de Trama

- **Tipo de trama:** Indica si se transmite una trama beacon, de datos, ACK o una trama de comandos.
- **Seguridad Habilitada:** Bit que indica si la seguridad se encuentra habilitada en la sub-capa MAC.
- **Trama Pendiente:** Se establece en uno cuando se esperan más datos para el dispositivo a quién va dirigida la trama.
- **Compresión PAN ID:** Indica si el dispositivo destino se encuentra en la misma red del dispositivo origen.
- **Versión de trama:** Indica la versión de la trama, para tramas IEEE 802.15.4-2003 se coloca el valor de 00, mientras que para tramas IEEE 802.15.4-2006 el valor será 01.
- **Modos de Direccionamiento Origen y Destino:** Señala el tamaño de las direcciones origen y destino.
 - o **Número de secuencia:** Identifica a la trama.
 - o **Campos de direccionamiento:** contienen los campos, identificador de PAN destino y dirección de destino, identificador de la PAN origen, dirección de origen.
 - o **Encabezado de seguridad auxiliar (opcional):** Contiene la información correspondiente para seguridad.
- **Carga Útil MAC:** contiene la información de datos que se requiere transportar.
- **MFR (MAC Footer):** campo de 16 bits que contiene un CRC.

A continuación, se definen los diferentes tipos de trama que se manejan a este nivel.

Tramas Beacon: Las utiliza el coordinador con el fin de transmitir beacons. Contiene los siguientes campos (Ver Figura 20):

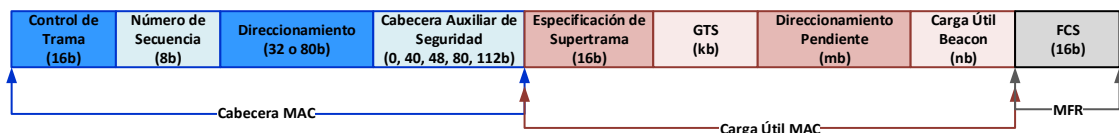


Figura 20. Vista esquemática de la Trama Beacon

- **MHR:** Contiene a su vez los campos control de trama, número de secuencia y opcionalmente el campo de seguridad auxiliar, cuyas funciones ya se han especificado.
 - o **Direccionamiento:** consta únicamente de los campos, identificador de PAN y dirección origen.

- **Carga útil MAC:** aloja varios campos que se enumeran a continuación:
 - **Especificación de Supertrama:** La presente estructura de envío de datos (supertrama) se emplea al trabajar en el modo beacon. Este sub-campo se fracciona como se muestra en la Figura 21.

Orden Beacon (4b)	Orden de Supertrama (4b)	Ranura Final de CAP (4b)	Extensión de Vida Útil de Batería (1b)	Reservado (1b)	Permiso de Asociación (1b)
----------------------	-----------------------------	-----------------------------	---	-------------------	-------------------------------

Figura 21. Formato del campo Especificación de Supertrama

- **GTS:** que se subdivide como se desglosa en los siguientes ítems (ver Figura 22).

Especificación GTS (8b)	Direcciones GTS (0-8b)	Listas GTS (variable)
----------------------------	---------------------------	--------------------------

Figura 22. Estructura del campo GTS

- **Especificación GTS:** cuenta con los sub-campos que se muestra en la Figura 23.

Contador del Descriptor GTS (0-2b)	Reservado (3-6b)	Permiso GTS 7b)
---------------------------------------	---------------------	--------------------

Figura 23. Estructura del sub-campo Especificación GTS

- **Direcciones GTS:** se subdivide en dos campos que se visualizan en la Figura 24.

Máscara de Direcciones GTS (0-6b)	Reservado (7b)
--------------------------------------	-------------------

Figura 24. Formato del sub-campo Direccionamiento Pendiente GTS

- **Lista GTS:** como se ve en la Figura 25 cuenta con tres sub-campos.

Dirección corta del dispositivo (16b)	Ranura de inicio GTS (4b)	Longitud GTS (4b)
--	------------------------------	----------------------

Figura 25. Estructura sub-campo Lista GTS

- **Direccionamiento Pendiente:** Consta de dos sub-campos que se evidencian en la Figura 26.

Especificación de direccionamiento pendiente (8b)	Lista de direcciones (variable)
--	------------------------------------

Figura 26. Formato del campo Direccionamiento Pendiente

- **Especificación de direccionamiento pendiente:** Sub-campo que contiene los espacios que se muestran en la Figura 27.

Número de Direcciones cortas pendientes (0-3b)	Reservado (3b)	Número de direcciones extendidas pendientes (4-6b)	Reservado (7b)
--	----------------	--	----------------

Figura 27. Formato del sub-campo Especificación de direccionamiento pendiente

- **Carga útil beacon:** secuencia establecida por la capa superior para ser transmitida en el marco de la trama beacon.
- **MFR:** cuya función ya se ha determinado previamente.

Tramas de Datos: Se emplean para la transferencia de datos y consta de los campos que se visualizan en la Figura 28.

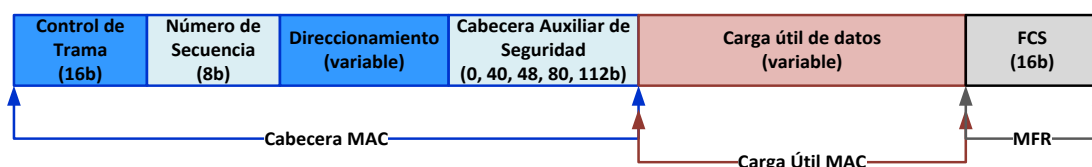


Figura 28. Vista esquemática de la trama de datos

- **MHR:** Consta de los campos control de trama, número de secuencia y opcionalmente el campo de seguridad auxiliar, cuyas funciones ya se han descrito.
 - **Direccionamiento:** consta de las direcciones destino y dependiendo del contenido en el campo control de trama deberá o no llevar las direcciones origen.
- **Carga útil MAC:** o para este caso se denomina carga útil de datos y consta de la información entregada por capas superiores.
- **MFR:** cuya función ya se ha determinado previamente.

Tramas ACK: Se emplean para confirmar la recepción de tramas, en la Figura 29 se puede observar los campos que conforman esta estructura.

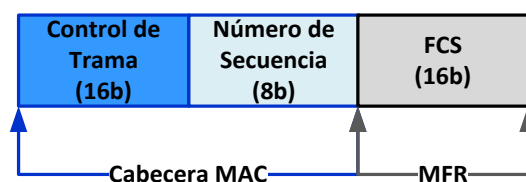


Figura 29. Vista esquemática de la trama ACK

- **MHR:** Este tipo de tramas se fracciona en los campos control de trama, número de secuencia cuyas funciones ya se han puntualizado.
- **MFR:** Cumple el objetivo establecido en la trama genérica.

Trama de comandos MAC: Se utiliza para la transmisión de información de control de entidades MAC pares. Consta de varios campos que se muestran en la Figura 30 y se abordan a continuación:

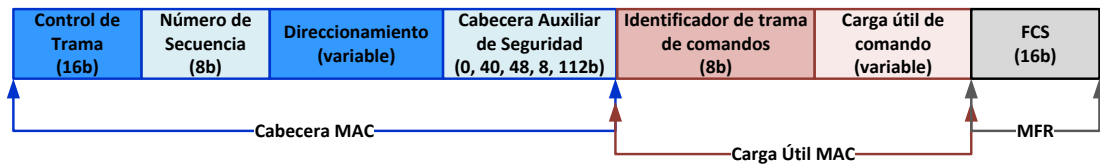


Figura 30. Vista esquemática de la trama de comandos MAC

- **MHR:** se subdivide en: control de trama, número de secuencia y opcionalmente el campo de seguridad auxiliar, cuyas funciones ya se han descrito.
 - **Campos de direccionamiento:** consta de las direcciones destino y dependiendo del contenido en el campo control de trama deberá o no llevar las direcciones origen.
- **Carga Útil MAC:** este campo contiene los sub-campos:
 - **Identificador de la trama de comandos:** Identifica el comando MAC que se está utilizando.
 - **Carga Útil de Comando:** contiene el comando MAC.
- **MFR:** cuya función ya se ha determinado previamente.

2.5.5 Inicialización y mantenimiento de la PAN

La sub-capas MAC lleva a cabo un conjunto de procedimientos, con el fin de asociar y desasociar dispositivos de la red, éstos se especifican en seguida:

- La sub-capas MAC como primer paso le solicitará a la capa física realice una **detección de energía** de uno o varios canales, con lo cual el coordinador podrá elegir el canal con menor cantidad de tráfico para establecer el canal de comunicaciones sobre el mismo.
- **Escaneo activo de canal:** Procedimiento que implica la transmisión continua de beacons. Le permite a un coordinador detectar las redes PAN en su entorno y así escoger un ID único para su uso. En el caso de dispositivos finales o routers este procedimiento les permite unirse a una red, mediante la localización (dentro de su rango de alcance) de un coordinador que transmita beacons. Este tipo de escaneo opera en el modo de red beacon y no beacon.
- Existe la opción de **escaneo pasivo del canal** que implica que los dispositivos finales o routers escuchen el canal en busca de beacons que hayan sido transmitidas por el coordinador. Este escaneo no sirve para el modo de red no beacon.
- En el caso de pérdida de comunicación entre un dispositivo y el coordinador, el dispositivo emitirá notificaciones de orfandad, con el fin de que el coordinador responda a las mismas y así recuperar la sincronización ente ambos.

2.6 Bluetooth

Se trata de un estándar de comunicaciones que define el nivel físico y de control de acceso al medio para WPAN. Establece un '*personal operating space*' omnidireccional dentro del cual se pueden mover dinámicamente los dispositivos.

El estándar ha ido evolucionando entorno a varias versiones: v 1.0 (1999), v 1.0B (1999), v 1.1 (primer estándar funcional - 2002), v 1.2 (2003), v 2.0 (2004), v 2.1 (2007), v 3.0 (2009), v 4.0 (2010), v 4.1 (2013), v 4.2 (2014) y v 5.0 (2017)

2.6.1 Topologías Bluetooth

Éstas pueden ser:

- **Piconet:** Unidad básica de la red Bluetooth, los dispositivos que intervienen son maestro y esclavo, con dos configuraciones posibles un maestro y un esclavo; y un maestro y varios esclavos. Esta infraestructura soporta hasta ocho dispositivos activos y doscientos cincuenta y cinco esclavos en estado 'parking' (inactivo).
- **Scatternet:** Red Bluetooth que tiene una o más piconets. Para que se forme la scatternet un nodo Bluetooth debe convertirse en esclavo de otra piconet, para convertirse en el puente de conexión de ambas redes; el mismo nodo puente puede redistribuir datos entre los miembros de ambas redes. Al ser una colección de piconets admite más de ocho dispositivos. Se debe considerar que el tipo básico de Bluetooth no admite esta configuración.

2.6.2 Stack de Protocolos

Protocolo de Núcleo

Como se muestra en la Figura 31 el protocolo núcleo se divide en dos bloques, *controlador* (hardware) que lo componen los tres niveles inferiores y el bloque *host* que lo conforman los dos niveles superiores conjuntamente con resto de protocolos de aplicación.

- **Radio Frecuencia:** Establece la interfaz de aire, bandas de frecuencia de operación, salto de frecuencia, técnica de modulación y clases de potencia de transmisión [1]
- **Banda Base:** Especifica formato de la PDU (Protocolo Data Unit), el esquema de direccionamiento, detección y corrección de errores es el responsable de manejar los enlaces en una piconet.
- **Link Manager Protocol (LMP):** establece y mantiene la conexión entre dos dispositivos Bluetooth. Define protocolos de autenticación y cifrado. Y la negociación del tamaño de la PDU.

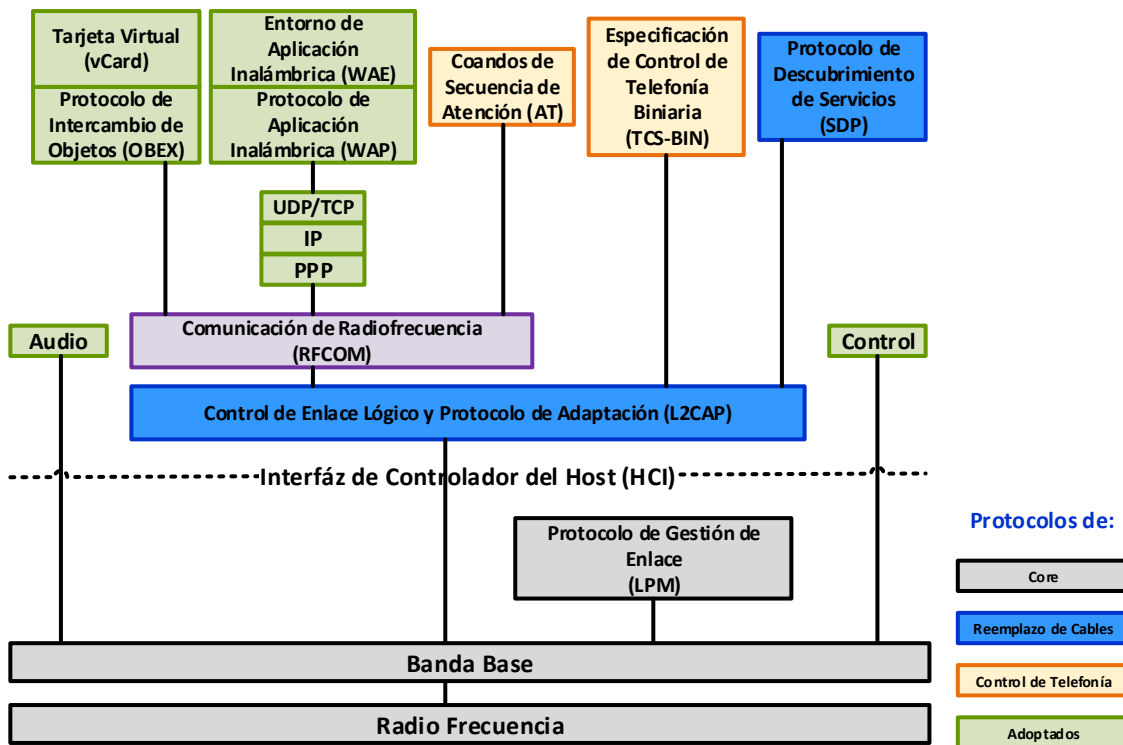


Figura 31. Stack de Protocolos Bluetooth

- **L2CAP (Logical Link Control and Adaptation Protocol):** adapta la PDU de la capa superior al formato de PDU de la capa banda base y viceversa.
- **Service Discovery Protocol (SDP):** Establece las conexiones entre dispositivos Bluetooth, determinando las consultas vinculadas al servicio y la información del dispositivo.

Protocolo de Reemplazo de Cables

RFComm (Radio Frequency Communication) es un protocolo simple de transporte, que provee la emulación transparente de un puerto serial sobre L2CAP, esta emulación permite la transmisión binaria de datos y la emulación de señales EIA-232 (Electronic Industries Alliance - 232) sobre la capa de banda base de Bluetooth. Soporta hasta 60 conexiones simultáneas entre dos dispositivos Bluetooth.

Protocolo de Control de Telefonía

TCS-BIN (Telephony Control Protocol - Binary), protocolo orientado a bit que define la señalización para el establecimiento de llamadas de voz y datos entre dispositivos Bluetooth, y procedimientos de gestión de movilidad para manejo de grupos de dispositivos Bluetooth.

Protocolos Adoptados

Son protocolos que se encuentran ya definidos por otras tecnologías y se adhieren al stack de protocolos de Bluetooth.

Los protocolos adoptados son: PPP (Point to Point Protocol), TCP/UDP/ IP (Transmission Control Protocol/ User Datagram Protocol/ Internet Protocol), OBEX (Object Exchange Protocol) y WAE/WAP (Wireless Application Environment/Wireless Application Protocol). PPP

2.6.3 Arquitectura de Protocolos

Capa Física [1]

Se trata de las especificaciones en banda base y radio definidas en el estándar IEEE 802.15.1.

En una piconet el maestro selecciona el mismo canal con la secuencia de salto de frecuencia y tiempo para la transmisión a todos sus esclavos. Es posible que en una piconet haya nodos maestro/esclavo, lo que supone y la unión de dos piconets en una scatternet.

En lo referente al esquema de **salto de frecuencia**, Bluetooth trabaja con ochenta frecuencias diferentes, con una distancia aproximada entre portadoras de 1 MHz. El ancho de banda de 1 MHz se comparte por ocho dispositivos, tomando en cuenta que los canales lógicos pueden usar el mismo ancho de banda de 80 MHz en simultáneo. La velocidad de salto es 1600 saltos por segundo, de modo que cada canal físico tiene un tiempo uso con una duración de 0.625 ms llamados ranuras (slots) que se enumeran secuencialmente.

Los radios de Bluetooth se comunican mediante TDD (Time Division Duplex), donde la transmisión de datos se lleva a cabo en una dirección a la vez y se alterna en dos direcciones una tras la otra. El acceso a la piconet es TDMA (Time Division Multiple Access), ya que el medio lo pueden compartir dos piconets. En consecuencia, el acceso a la piconet será FH-TDD-TDMA (Frequency Hopping-TDD-TDMA).

Esporádicamente, dos piconets usarán el mismo canal físico durante el mismo intervalo de tiempo, provocando una colisión, pero en esto se resuelve mediante la implementación de ARQ (Automatic Repeat Request). Así, se establece una forma de CSMA (Carrier Sense Multiple Access) entre dispositivos piconets diferentes en la misma scatternet, conocida como FH-CSMA.

Por otra parte, los **enlaces físicos** que se pueden establecer entre maestro y esclavo son:

- **SCO (Synchronous Connection Oriented):** La asignación de ancho de banda es fija entre maestro y esclavo (enlace punto a punto), se reservan dos ranuras consecutivas, una para cada dirección de transmisión. El número máximo de

enlaces que soporta el maestro es tres, en cambio el esclavo soporta dos o tres enlaces de este tipo. No aplica la retransmisión de paquetes SCO.

- **ACL (Asynchronous Connectionless):** Se emplea en la conexión entre maestro y todos los esclavos de la piconet (enlace punto multipunto). El maestro podrá intercambiar paquetes con cualquier esclavo por ranura, incluyendo esclavos que estén conectados por a un enlace SCO, pero tomando en cuenta que solo puede existir una conexión ACL. Aplica la retransmisión de paquetes ACL.

La norma establece cinco **canales lógicos**, que se establecen mediante SCO y ACL, para transportar diferentes tipos de carga útil, estos son: control de enlace (LC, administra el flujo de paquetes sobre el enlace), gestor de enlace (LM), usuario asincrónico(UA), usuario isócrono(UI), usuario sincrónico (US).

Control de Enlace

Incluye los protocolos LMP y L2CAP conjuntamente con sus funciones.

En cuanto a los **canales de control**, para la operación de una piconet se han definido estados de operación para el establecimiento y mantenimiento del enlace:

- **Estado de espera (predeterminado):** el dispositivo se encuentra en baja potencia de consumo, en la que solo ejecuta el reloj nativo.
- **Estado de conexión:** cuando el dispositivo (como maestro o esclavo) pertenece a una piconet.

El estándar también indica la existencia de sub-estados de operación provisionales, para la agregación de nuevos esclavos a la piconet, los cuales son: página, escaneo de página, respuesta del maestro, respuesta del esclavo, consulta, exploración de consulta y respuesta de consulta.

Mecanismos de comunicación: El estándar incorpora además de la transmisión unicast, las siguientes formas de comunicación:

Broadcast: La emisión de tramas de este tipo las lleva a cabo el maestro, estableciendo todos los bits del campo XR en cero, esta PDU se envía a todos los esclavos para su procesamiento, cabe destacar que en este tipo de mensajes el campo ARQN (Automatic Repeat reQuest Number) se encuentra desactivado. Las posibles retransmisiones se controlan activando el campo SEQN (Sequential Numbering scheme). En caso de que el mensaje de broadcast tenga varios fragmentos se encapsula en PDU de Bluetooth y se retransmite “NBC” veces.

En lo que se refiere al manejo de mensajes **multicast** se encuentra bajo el control del protocolo L2CAP. El significado de los términos AM_ADDR, ARQN y SEQN se abordan en el numeral 2.6.4.

2.6.4 Formato de la PDU

Si bien cada capa tiene sus propias PDU que permiten la ejecución de sus funcionalidades, se describirá únicamente el formato de PDU de la capa inferior, ya que aborda el funcionamiento base del protocolo objetivo del presente trabajo.

PDU Banda Base

En esta tecnología los datos son fragmentados y se envían en forma de paquetes por el canal de comunicaciones. Como se visualiza en la Figura 32 principales campos de un paquete Bluetooth son:

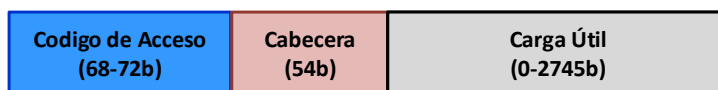


Figura 32. Formato general del paquete Bluetooth.

Código de Acceso: Se emplea para determinar el tiempo de sincronización, compensación, paginación e identificación. Los paquetes que acceden a un mismo canal de una piconet, tendrán el mismo código de acceso al canal. El campo código de acceso se dividirá como en los sub-campos que se muestran en la Figura 33.

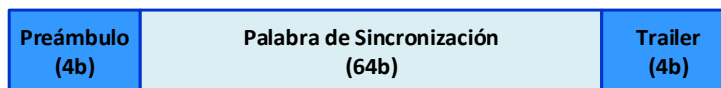


Figura 33. Formato del campo Código de Acceso

Los códigos de acceso pueden ser: CAC (Channel Access Code) identifica unívocamente la piconet; DAC (Device Access Code) usado en paginación y sus correspondientes respuestas; e IAC (Inquiry Access Code) usado para propósitos de identificación.

- **Preámbulo:** Patrón fijo de bits para facilitar las compensaciones DC.
- **Palabra de Sincronización:** Código de 64 bits derivado de una dirección de 24 bits (LAP - Lower Address Part).
- **Trailer:** Conformado con tres más significativos de la palabra de sincronización forman un patrón de bits que se usa para compensar extensiones de DC.

Encabezado: Identifica el tipo de paquete y contiene la información de control. Los campos que contiene la cabecera se visualizan en la Figura 34.

AM_ADDR (3b)	Tipo (4b)	Flujo (1b)	ARQN (1b)	SEQN (1b)	HEC (8b)
-----------------	--------------	---------------	--------------	--------------	-------------

Figura 34. Formato de la Encabezado de Bluetooth

- **AM_ADDR (Active Member Address):** Contiene la dirección temporal de los esclavos de la piconet, permite distinguir los miembros activos y los participantes de la piconet.
- **Tipo:** Identifica el tipo de paquete enviado (dependiendo del enlace físico asociado al paquete) y las ranuras que va a ocupar.
- **Flujo:** proporciona un mecanismo de control de flujo para el tráfico ACL. Cuando el buffer de recepción para el enlace ACL está lleno, se envía un mensaje con Flujo=0, por ende, la estación que recibe el paquete deberá detener temporalmente la transmisión de paquetes ACL en ese enlace.
- **ARQN:** Proporciona un bit de confirmación (acknowledgement) para el tráfico ACL protegido por CRC (Cyclic Redundancy Check). Se devuelve un mensaje de retorno que indica si la protección fue satisfactoria ACK (ARQN=1) o de lo contrario un NAK (ARQN=0), se asume además como NAK cuando no se recibe ningún mensaje.
- **SEQN:** Bit que se va invirtiendo para evitar procesar datos duplicados en recepción.
- **HEC (Header Error Control):** código de detección de errores aplicado a la cabecera del paquete.

Carga útil: de tamaño variable, constituye los datos como tal y de distribuye como se observa en la Figura 35.

Cabecera (8-16b)	Datos (0-2712)	CRC (16b)
---------------------	-------------------	--------------

Figura 35. Formato del campo Carga útil

- **Encabezado (de Carga Útil):** Se establecen en, 8 bits para paquetes de ranura simple y 16 bits para paquetes de ranuras múltiples. Éste sub-campo se divide a su vez en (Ver Figura 36 y Figura 37):
 - **L_CH (Logical Channel):** Identifica al canal lógico.
 - **Flujo:** Se emplea para controlar el flujo en el nivel L2CAP.
 - **Longitud:** Especifica la longitud en bytes únicamente de la carga útil.
- **Datos:** Información de usuario
- **CRC:** corresponden a 16 bits de código de redundancia cíclica para todas las cargas de datos, excepto para el paquete AUX1 que consta de 30 Bytes de información sin CRC o FEC (Frame Check Sequence), en datos de alta velocidad.

Se debe considerar que si se transporta voz la especificación banda base no se especifica el sub-campo encabezado; en contraposición a los paquetes ACL y SCO en los que si se define éste sub-campo.

L_CH (2b)	Flujo (1b)	Longitud (5b)
--------------	---------------	------------------

Figura 36. Sub-campo Encabezado de carga útil - paquetes de ranura simple.

L_CH (2b)	Flujo (1b)	Longitud (9b)	Indefinido (4b)
--------------	---------------	------------------	--------------------

Figura 37. Sub-campo Encabezado de carga útil - paquetes de ranura múltiple

2.6.5 Bluetooth Core Specification Version 3.0 High Speed

La versión Bluetooth 3.0 + HS (High Speed) alcanza una velocidad de 24 Mbps (en la práctica 22-26 Mbps), debido a que emplea AMP (Alternate MAC/PHY) para transferir grandes cantidades de datos. La velocidad de este estándar para dispositivos ad-hoc, representa una gran ventaja el momento de transferir grandes cantidades de datos a gran velocidad.

Opera en la banda ISM (Industrial, Scientific and Medical) a 2.4 GHz, pudiendo soportar hasta 79 (23 para España) canales con un ancho de banda de 1 MHz cada uno.

Dentro del estándar se han definido tres principales clases de potencia, en la Tabla 1, en el respectivo apartado, se puede observar las clases de potencia con su respectiva área de cobertura.

Trabaja con la modulación GFSK (Gaussian Frequency Shift Keying).

Incluye Unicast Connection Less, que proporciona a los dispositivos una mejor respuesta, respecto a anteriores versiones, y una notable duración de batería.

Alternative MAC/PHY

Se permite emplear MAC y PHY para la transferencia de datos sobre Bluetooth 3.0, donde Bluetooth Radio se utiliza para la señalización inicial y períodos de transferencia de datos limitadas, pero cuando se necesita transmitir grandes cantidades de datos se emplea MAC/PHY 802.11, generalmente asociados con Wi-Fi.

Con esta característica se reduce el consumo de energía mediante el uso de radio de alta velocidad en los escenarios mencionados.

Datos sin conexión unicast.

Esquema que permite el envío de datos (preferiblemente en pequeñas cantidades) sin establecer explícitamente la configuración del canal L2CAP. Reduce la latencia y la potencia de consumo en dispositivos con conexión intermitente.

2.7 ZigBee

Es un estándar que define un conjunto de protocolos para la implementación de redes WPAN. Para el desarrollo de la tecnología e implementación de estándares cuenta con el apoyo de Zigbee Alliance, conformada por representantes de la industria de todo el mundo.

El estándar especifica que ZigBee opera sobre las capas PHY y MAC de 802.15.4, que brinda un medio adecuado para entornos con baja relación señal a ruido. Además, se establece el bajo consumo de energía, bajo coste, corto alcance y velocidad de datos menor a 250 KBps (típica de 20 KBps).

Opera sobre las bandas de frecuencia ISM de los 868 MHz en Europa, 915 MHz en Estados Unidos y 2.4 GHz en todo el mundo. La forma de modulación para la banda de los 2.4 GHz es DSSS (Direct Sequence Spread Spectrum)

El método de control de acceso al medio es CMSA/CA (Carrier Sense Multiple Access with Collision Avoidance), con la posibilidad de usar TDMA en aplicaciones de baja latencia.

La ZigBee Alliance ha definido varias versiones del estándar: ZigBee 1.0 (2004), ZigBee 1.1 (2006), ZigBee PRO y ZigBee (2007), ZigBee PRO y ZigBee 3.0 (2015)

2.7.1 Tipos de Dispositivos

Dentro una red ZigBee existen tres tipos de dispositivos:

- **Coordinador:** Permite el establecimiento de la PAN a la que se anexarán los demás elementos de la topología, además se encargará de tareas de seguridad, seleccionar el canal y el ID de la PAN (de 16 bits), asignación de dirección de 16 bits a los dispositivos de su red; dentro del espacio PAN el coordinador es único.
- **Router:** Elemento que posibilita la extensión de la red. Puede asociarse con el coordinador u otro router ZigBee, viabilizando el enrutamiento de mensajes mediante múltiples saltos.
- **Dispositivo Final:** elemento básico de la red que no soporta dispositivos secundarios.

Tanto coordinador y router no pasan al modo reposo y se recomienda que su alimentación sea mediante línea eléctrica.

Para la conformación de la red el coordinador, no pertenezca a otra red Zigbee, buscará un canal de radio que no interfiera con las frecuencias WLAN que se encuentren en uso; posteriormente el coordinador asigna un ID PAN a la red que inicia con él su

operación, también asignará su propia dirección de red (0x0000); a continuación, el coordinador está listo para recibir solicitudes de adhesión de routers y dispositivos finales. A los pasos anteriores se añade, la exploración PAN, en la que el coordinador envía una trama broadcast beacon de respuesta en el canal restante; mediante el coordinador se recibirá el ID PAN de router y dispositivos finales cercanos, este procedimiento también permite conocer si routers y dispositivos finales pueden unirse o no a la red.

Tanto routers como dispositivos finales se pueden unir a la red de dos formas: MAC association (implementado por el dispositivo subyacente a la capa MAC) y Network re-join (implementado por la capa de red). Una puntualización respecto a MAC association es que permite la asociación entre coordinador y router / dispositivo final, o router y dispositivo final, y router y otros routers.

2.7.2 Topologías de Red

ZigBee hereda las topologías de 802.15.4 añadiendo las topologías de malla y árbol. Una red ZigBee puede alojar hasta 255 dispositivos; pero además se pueden crear hasta 255 clusters con lo que se puede tener hasta 64770 nodos. Según el estándar la distribución de los nodos permite la implementación de varios tipos de topologías (Ver Figura 39, Figura 38 y Figura 40), como se describe a continuación:

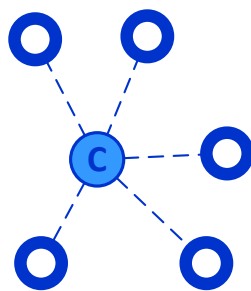


Figura 38. Topología Estrella

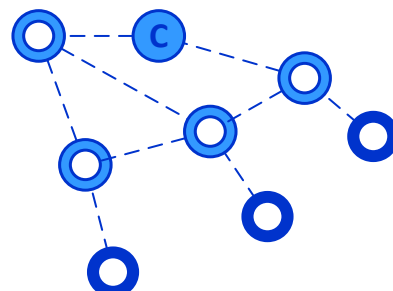


Figura 39. Topología Malla

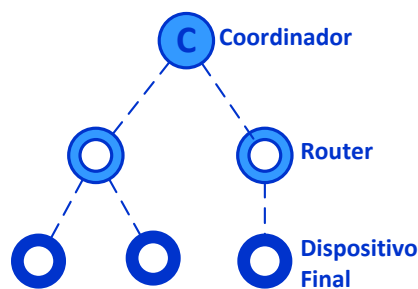


Figura 40. Topología Árbol



- **Estrella:** Un coordinador ejerce el control sobre la red, por lo que se encarga de iniciar y mantener comunicaciones con y entre dispositivos. El área de cobertura de esta topología está limitada por el alcance que tenga el coordinador.

- **Malla:** Además de poseer un coordinador, la red se puede ampliar a través del uso de routers. Permite mantener disponibilidad incluso si varios nodos caen en la infraestructura.
- **Árbol:** Se trata de una extensión de la topología en malla en la que se emplea una estructura y enrutamiento jerárquico, para la transmisión de información de usuario y control. Esta topología el coordinador y los routers pueden alojar otros routers o dispositivos finales.

Además de lo mencionado, el estándar define la característica de multi-hop que permite a la red abarcar una gran superficie.

2.7.3 Stack de Protocolos

El stack de protocolos de ZigBee como se observa en la Figura 41, se compone de capas que brindan un conjunto de servicios; una entidad de servicios expone una interfaz a la capa superior mediante un SAP y cada SAP soporta un conjunto de primitivas de servicio para brindar una funcionalidad. De acuerdo al estándar ZigBee opera sobre las capas PHY y MAC de IEEE 802.15.4, las cuales se abordaron en el numeral 2.5.3, por lo cual en este espacio únicamente se definirán las dos capas superiores de ZigBee.

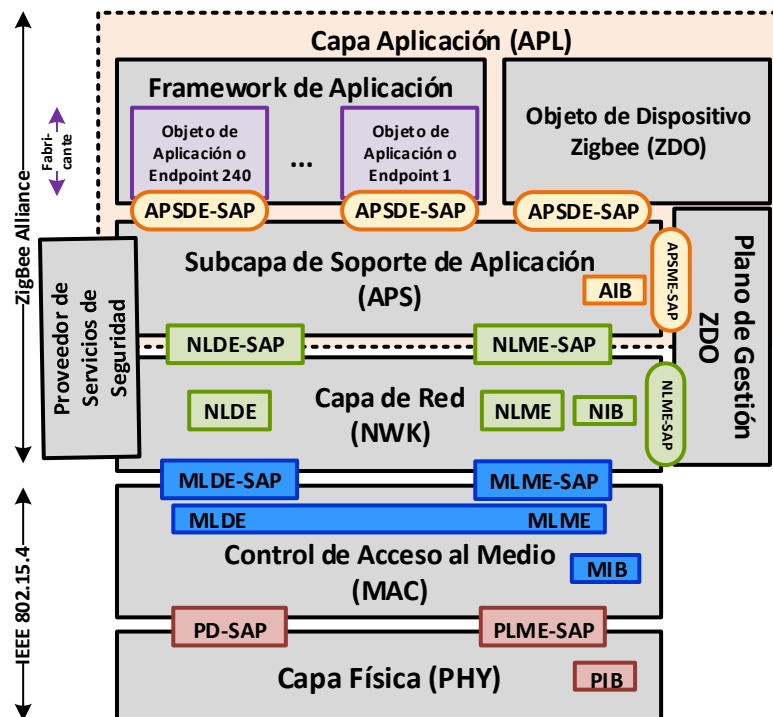


Figura 41. Stack de Protocolos ZigBee e IEEE 802.15.4

Capa Red

Es la interfaz entre la capa aplicación y la sub-capas MAC. Para la interacción con la capa aplicación provee dos entidades de servicios fundamentales:

- **Servicio de Datos:** mediante NLDE (Network Layer Data Entity) permite la transmisión de datos a través de su SAP asociado. Permite que la capa aplicación transporte APDUs (Application PDU) entre dispositivos de la misma red. Proporciona además dos servicios: generación de la NPDU (Network PDU) y la especificación de la topología de enrutamiento.
- Mientras que, el **servicio de gestión** lo realiza mediante NLME (Network Layer Management Entity), permitiendo que la capa aplicación interactúe con el stack. Provee los servicios de configuración de un nuevo dispositivo, inicialización de una nueva red, incorporación y abandono de una red, direccionamiento, descubrimiento de vecinos, descubrimiento de ruta, recepción de control. Se encarga además del mantenimiento la base de datos de objetos gestionados o NIB (Network Information Base).

Se debe considerar que entre la capa de red y la capa aplicación existen dos SAP, uno para cada servicio, así también existen dos SAP entre la capa de red y la sub-capas MAC. Además de las de las interfaces mencionadas, existe una interfaz implícita entre NLME y NLDE, que permite a NLME utilizar el servicio de datos de la red (NLDE).

Los servicios de red incluyen algunas primitivas entre la capa aplicación y MAC y entre los servicios de la propia capa; dichas primitivas permitirán ejecutar las siguientes funcionalidades: descubrimiento de la red, formación de la red, permitir la incorporación de dispositivos, inicializar el router, incorporación a la red, incorporación de dispositivos directamente a la red, abandono de la red, restablecimiento de un dispositivo, recepción de sincronización e información de mantenimiento base.

Mecanismos de comunicación: la forma de comunicación entre dispositivos se divide en tres categorías broadcast, multicast y unicast; el mecanismo de comunicación por defecto es unicast.

- **Broadcast:** en esta forma de comunicación el mensaje será transmitido a todos los dispositivos que están en un mismo canal de frecuencia, independiente de su dirección o identificador PAN. La transmisión sobre IEEE 802.15.4, se realiza con el direccionamiento corto por lo que la dirección de destino se establece en 0xFFFF, si bien para IEEE 802.15.4 el identificador PAN se puede establecer en 0xFFFF, ZigBee siempre establece el identificador PAN ya que no permite broadcast a través de múltiples redes.
Mediante la capa red, la sub-capas APS de cualquier dispositivo en una red ZigBee puede iniciar una transmisión broadcast. Para confirmar la recepción del mensaje de broadcast se emplea el mecanismo de acuse de recibo pasivo, en donde el coordinador y routers de la red comprueban si sus dispositivos vecinos han transmitido correctamente el mensaje; este procedimiento

consiste en que luego de que un dispositivo difunde un mensaje, entra en modo recepción y esperará hasta que el paquete sea re-difundido por cualquiera de los dispositivos vecinos. Un mensaje de redifusión confirma que un dispositivo vecino ha recibido y transmitido un mensaje de broadcast con éxito.

Coordinadores y routers mantienen una BTT (Broadcast Transaction Table) conocido como BTR (Broadcast Transaction Record) que registra todos los mensajes de broadcast que se transmiten, en este registro se incluye el número de secuencia y la dirección de origen del paquete de broadcast, la BTR tiene un tiempo de vida determinado.

- **Multicast:** esta forma de comunicación consiste en el envío de un mensaje a un conjunto de dispositivos pertenecientes a una misma red. El esquema de transmisión consiste en dispositivos miembro de un grupo multicast identificado por un ID de 16 bits; con la ventaja de que cada dispositivo puede pertenecer a varios grupos multicast y mantener una lista de miembros del grupo en una tabla denominada *nwkGroupIDTable*.

Existen dos modos de generación y envío de mensajes en esta forma de comunicación:

- **Modo miembro:** un dispositivo miembro genera un mensaje multicast y lo envía a los miembros de su grupo multicast.
- **Modo no-miembro:** un dispositivo no miembro del grupo multicast envía el mensaje a un miembro del grupo para que éste lo transmita a los miembros del grupo.

Capa Aplicación

Ésta capa se compone de dos sub-capas, las cuales se describen a continuación, incluye también los objetos de aplicación (endpoints) definidos por el fabricante.

- **APS (Application Support Sub-layer):** Se encarga mediante NLDE-SAP de proporcionar una interfaz entre la capa red y la capa aplicación, mediante un conjunto de servicios para el uso de ZDO (ZigBee Device Object) y de los objetos de aplicación definidos por el fabricante; servicios se ofrecen a través de dos entidades:
 - **Entidad de datos:** brinda el servicio de transmisión de datos mediante su SAP (APSDE-SAP), a la capa de red y al mismo tiempo a ZDO y los objetos de aplicación admitiendo el transporte de PDU de capa aplicación entre dispositivos en la misma red.
 - **Entidad Gestora:** mediante APSME-SAP proporciona el servicio de gestión para que la capa aplicación interactúe con el stack de protocolos. Se encarga, además del mantenimiento de las tablas de

vinculación, que trata de la capacidad de combinar dos dispositivos en función de sus servicios y necesidades, y el reenvío de mensajes entre dispositivos enlazados. Otra a de sus responsabilidades es el mantenimiento de la base de datos de objetos gestionados AIB (APS Information BASE).

La sub-capas APS, además de las formas convencionales de comunicación soporta direccionamiento indirecto, bajo este esquema un dispositivo de recursos limitados conectado puede comunicarse sin conocer la dirección de destino; para lo cual el dispositivo limitado envía éstas transmisiones al coordinador ZigBee quién se encarga de examinar la dirección de origen, las direcciones de endpoint y el identificador de clúster desde su tabla de enlace para enviar el mensaje a la dirección de destino o endpoint correspondiente.

- **Framework de aplicación:** Es el entorno en el que se alojan los objetos de aplicación (endpoints) que permiten controlar y administrar las capas de protocolo en los dispositivos ZigBee. Puede haber un total de 240 objetos de aplicación diferentes, identificados por una dirección de endpoint del 1 al 240. Un objeto de aplicación contiene un perfil de aplicación (programa) desarrollado por un fabricante o por ZigBee Alliance. Cada perfil de aplicación tiene un identificador de perfil de 16 bits, emitido por ZigBee Alliance. Así también cada perfil de aplicación tiene sus componentes principales:
 - **Clusters:** identificado unívocamente por un identificador de clúster de 16 bits; se trata de un conjunto de atributos agrupados, los atributos almacenan datos o valores de estado.
 - **Descriptor de dispositivo:** proveen información sobre el dispositivo y se identifican mediante un valor de 16 bits. El perfil de aplicación de ZigBee usa el concepto de estructura de datos de descriptor en el que se tiene un valor de 16 bits que actúa como puntero (descriptor de datos) a la ubicación de los datos de aplicación. Al detectarse la presencia de un dispositivo por parte de otro, se envían las descripciones de dispositivo del nuevo componente. Las secciones del descriptor de dispositivo son: descriptor de nodo, descriptor de potencia del nodo, descriptor simple, descriptor complejo y descriptor de usuario.
- **ZDO (ZigBee Device Object):** provee una interfaz entre la sub-capas APS y el Framework de aplicación. Así también, cumple con las siguientes funciones:

- Satisfacer los requerimientos comunes de todas las aplicaciones que operan en el stack de protocolos ZigBee.
- Inicializar APS, la capa red, y el proveedor de servicios de seguridad.
- Monitorear la información de configuración de las aplicaciones finales para establecer e implementar el descubrimiento y administración de seguridad, red y enlace.
- Presenta interfaces públicas a los objetos de aplicación en el framework de aplicación, para que los objetos de aplicación controlen las funciones del dispositivo y la red.
- El estándar define un perfil para ZDO (para todos los dispositivos ZigBee) denominados ZDP (Perfil de Dispositivo ZigBee), que al igual que en el Framework de aplicación, contienen clústers y descriptores de dispositivos, con la diferencia de que los clústers no utilizan atributos debido a que el ZDO tiene sus propios atributos de configuración. En este tipo de perfiles se definen clústers obligatorios y opcionales; los primeros se deben implementar en todos los dispositivos ZigBee.
- Especificar el papel de un componente en la red (como coordinador o dispositivo final); descubrir los dispositivos en la red y determinar qué servicios de aplicación proporcionan.
- Iniciar y/o responder a solicitudes de enlace y establecer una relación segura entre dispositivos de red.

2.7.4 Formato de la PDU

Formato General NPDU

La PDU en la capa red consta, como se observa en la Figura 42 de dos módulos básicos, que se describen a continuación:



Figura 42. Vista esquemática de la NPDU

- **Cabecera de Red:** consta de los siguientes campos:
 - **Control de Trama:** se sub-divide en los siguientes apartados (Ver Figura 43):

Tipo de Trama (2b)	Versión de Protocolo (4b)	Descubrimiento de Ruta (2b)	Bandera Multicast (1b)	Seguridad (1b)	Dirección IEEE destino (1b)	Dirección IEEE origen (1b)	Reservado (3b)
--------------------	---------------------------	-----------------------------	------------------------	----------------	-----------------------------	----------------------------	----------------

Figura 43. Formato del campo Control de Trama.

- **Tipo de Trama:** Secuencia de dos bits que indica el tipo de PDU que se está manejando (datos o comandos de red)
 - **Versión del protocolo:** indica el número de versión del protocolo ZigBee que se está utilizando.
 - **Descubrimiento de ruta:** determina la forma de enrutamiento para la PDU; controla las operaciones de descubrimiento de ruta.
 - **Bandera Multicast:** indica si la PDU se envía mediante multicast, cuando éste sub-campo encuentra establecido en 1.
 - **Seguridad:** Indica si las opciones de seguridad se encuentran activas para la PDU.
 - **Ruta de origen:** se establece en uno si el campo ruta de origen de subtrama se incluye en la PDU.
 - **Dirección IEEE destino y origen:** si estos sub-campos se encuentran establecidos en uno indican que la PDU incluye las direcciones IEEE de 64 bits en los campos opcionales de dirección IEEE destino y origen.
- o **Dirección de destino:** Campo obligatorio q alberga los 16 bits de la dirección del dispositivo destino o la dirección de broadcast (0xFFFF). La dirección de red de un dispositivo será la misma que la dirección corta MAC IEEE 802.15.4, esto último aplica también a la dirección de origen.
 - o **Dirección de origen:** Campo obligatorio que contiene la dirección del dispositivo que origina la trama.
 - o **Radio:** Especifica el alcance de la transmisión de radio e irá disminuyendo en 1 por cada salto que cruce el paquete.
 - o **Número de Secuencia:** Campo que se va incrementando en uno cuando se transmite una nueva trama.
 - o **Control Multicast (opcional):** se incluye si la PDU es de multicast, se fracciona en los sub-campos que se muestran en la Figura 44:

Modo Multicast (2b)	Radio no-miembro (3b)	Radio no-miembro máximo (3b)
-------------------------------	---------------------------------	--

Figura 44. Formato del campo Control Multicast

- **Modo multicast:** señala si el paquete está siendo enviado en modo no-miembro (00) o modo miembro (01).
- **Radio no-miembro:** restringe el número de veces que una PDU multidifusión es difundida por dispositivos no miembro

e irá disminuyendo en uno cada vez que la PDU sea difundida por dichos dispositivos.

- **Radio no-miembro máximo:** usado por los dispositivos miembro cada vez que éstos dispositivos re difunden el paquete, copiando el contenido del sub-campo radio no miembro en el presente sub-campo.
- o **Ruta de origen de sub-trama:** En el enrutamiento de origen, el dispositivo que genera el paquete le indica a éste cual es la ruta que debe tomar para llegar al destino. Este campo se divide como se observa en la Figura 45.

Conteo de Envío (8b)	Índice de envío (8b)	Lista de envío (variable)
-------------------------	-------------------------	------------------------------

Figura 45. Formato del campo Ruta de Origen de Sub-trama

- **Conteo de envío:** indica el número de veces que se ha retransmitido el paquete.
 - **Índice de envío:** Se coloca en cero en el origen y va aumentando en uno por cada salto por el que pase la PDU.
 - **Lista de envío:** Este sub-campo contiene la lista de direcciones cortas de los nodos que se utilizarán para la transmisión del paquete mediante enrutamiento en origen
- **Carga útil de red:** contiene la información específica a transmitirse.

Cabe indicar que en esta capa se han definido otros formatos para paquetes de datos y comandos de red.

Formato General APS

La APSDE (Entidad de Datos de Sub-capa Aplicación) es la encargada de proporcionar el servicio de datos a objetos de aplicación y a ZDO, es decir recibe los datos que deben ser transmitidos desde un objeto de aplicación o ZDO, a dichos datos añade los encabezados correspondientes para crear una trama de datos APS. Cabe mencionar que dentro de los datos que se reciben de ZDO se incluye los comandos ZDP que tienen un formato normado.

En este nivel la PDU consta de dos espacios fundamentales que se desglosan a continuación y se observan en la Figura 46.

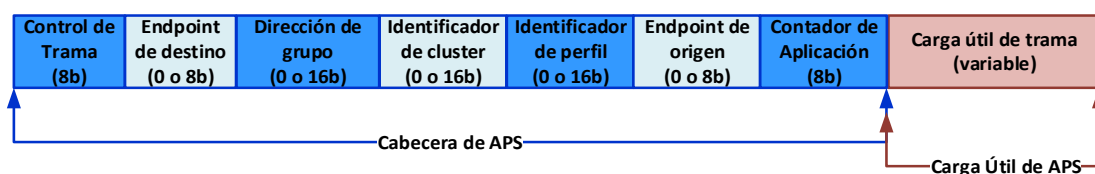


Figura 46. Vista esquemática de la APS

- **Cabecera de Aplicación:** en este espacio se establecen los siguientes campos:
 - **Control de Trama:** consta de los sub-campos que se muestran en la Figura 47.

Tipo de Trama (2b)	Modo de entrega (2b)	Modo de direccionamiento indirecto (1b)	Seguridad (1b)	Respuesta ACK (1b)	Reservado (1b)
---------------------------	-----------------------------	--	-----------------------	---------------------------	-----------------------

Figura 47. Formato del campo Control de Trama

- **Tipo de Trama:** indica si la PDU es de datos, comando o acuse de recibo.
- **Modo de entrega:** establece la opción de transmisión de la PDU (unicast, direccionamiento indirecto, multicast y broadcast). En caso de que el modo de entrega sea direccionamiento indirecto, no se incluyen los campos dirección origen o destino.
- **Modo de direccionamiento indirecto:** cuando este bit se encuentra establecido, indica que la PDU va dirigida al coordinador ZigBee, en este escenario se excluye el campo endpoint de destino.
- **Seguridad:** Se trata de un campo que lo determina el SSP (Proveedor de Servicios de Seguridad).
- **Solicitud de Acuse de Recibo:** indica al receptor que debe enviar un acuse de recibo como confirmación de recepción de la PDU.
- **Endpoint de destino:** indica el endpoint del receptor final de la PDU.
- **Dirección de grupo:** cuando éste campo se ha especificado la PDU se entregará a todos los dispositivos finales que son miembros del grupo.
- **Identificador de clúster:** Contiene el identificador de clúster que se utiliza para la fase de establecimiento del enlace por lo que solo estará presente durante dicho proceso.
- **Identificador de perfil:** consta del identificador de perfil deseado ZigBee y se utilizará durante el filtrado de mensajes en cada dispositivo que recibe la PDU.
- **Endpoint de origen:** indica el endpoint del transmisor inicial de la PDU.
- **Contador de aplicación:** campo que va incrementando en uno cuando se envía una nueva PDU.

En este nivel el estándar ZigBee ha determinado otros formatos para trama de datos, comando y acuse de recibo (ACK).

2.8 RFID (Radio Frequency Identification)

En términos generales se trata de una serie de tecnologías de identificación que usa mecanismos de comunicación inalámbrica para dar seguimiento a objetos físicos mediante la asignación de una identidad electrónica única.

RFID ha existido por más de medio siglo, pero su desarrollo surgió por el interés de instituciones gubernamentales de los Estados Unidos y por algunas transnacionales, a lo que se añade, la reducción de costes y tamaño de sus componentes, así como la mejora en la transmisión de datos a todo nivel. Es así que RFID se ha desplegado a un conjunto importante de aplicaciones.

A diferencia de otras tecnologías de identificación que requieren un alineamiento adecuado entre sus componentes, RFID permite la identificación de objetos (etiquetas) a distancia y sin necesidad de una alineación específica de los diferentes componentes; otra ventaja es que con RFID se pueden escanear varios objetos al mismo tiempo así también sus componentes se encuentran disponibles en una amplia variedad de factores de forma lo que permite una adaptación ágil a cualquier entorno.

La normativa que rige a RFID es la implantada por EPCglobal e ISO.

El avance de la presente tecnología se divide en las siguientes eras: propietaria, de cumplimiento, de empresas habilitadas con RFID, de las industrias habilitadas con RFID y la del Internet de las Cosas; en cuanto a ésta última era se destaca la asignación de una identidad digital a objeto físico, como un componente habitual que permite vincular el objeto al internet.

2.8.1 Componentes del Sistema

Un sistema RFID consta de los elementos que se observan en la Figura 48 y se describen a continuación:

- **Etiqueta:** dispositivo colocado en el elemento que se desea rastrear, contienen información que se envía al lector mediante radio frecuencia. Disponen de los siguientes componentes:
 - **Antena:** transmite la información referente a la etiqueta.

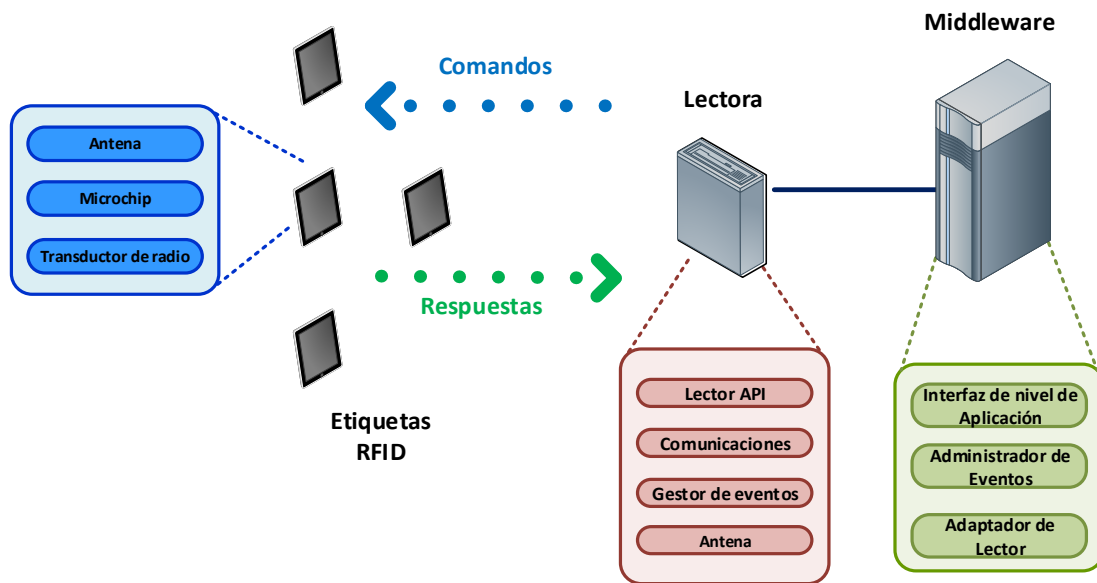


Figura 48. Componentes sistema RFID

- **Transductor de radio:** convierte la información que envía la antena.
- **Microchip:** es una memoria donde se almacena el identificador y demás datos. Algunas versiones de etiquetas no poseen este elemento por lo que pueden almacenar información con un máximo de 24 bits. Las etiquetas pueden ser escritas una sola vez (de lectura) o reescritas muchas veces (de múltiple escritura).

Las etiquetas además de lo mencionado las etiquetas disponen de diversas formas de empaquetamiento, permitiendo que su uso se aplique a múltiples ámbitos. En lo que se refiere a potencia de alimentación existen etiquetas, pasivas que se activan por las señales que emita un lector; activas que usan las señales del lector para comunicaciones, pero para alimentar al microchip y otros elementos disponen de una batería; bidireccionales que disponen de su propia potencia para ejecutar sus comunicaciones, que pueden ser con otras etiquetas prescindiendo del lector.

- **Lector:** dispositivo de borde de la red, que reconoce etiquetas y extrae la información que contienen, para enviarla al middleware. Está formado de los siguientes subsistemas:
 - **Lector API (Application Programming Interface):** interfaz de configuración, gestión y supervisión del lector; además permite a los programas conectarse para capturar evento referente a la lectura de etiquetas RFID.
 - **Comunicaciones:** gestiona las funciones de red.
 - **Gestor de eventos:** permite determinar qué observaciones son eventos y que eventos ameritan el envío de alertas. Entiéndase por observación

al reconocimiento de una etiqueta por parte de un lector; una observación distinta la descrita constituye un acontecimiento y el análisis de observaciones es el filtrado de eventos.

- **Antena:** se trata de una o más antenas con sus interfaces de lógica y soporte que establecen la comunicación entre lector y las etiquetas a través de ondas electromagnéticas.

Al igual que las etiquetas, los lectores disponen de un amplio conjunto de factores de forma, lo que conjuntamente con características como alcance, capacidad de almacenamiento, compatibilidad con etiquetas, permiten su uso en diversos ámbitos.

- **Middleware RFID:** software con el que se comunica el lector para informar a aplicaciones de la presencia de dispositivos etiquetados. Los escenarios de trabajo en donde se emplean tecnologías RFID, son propensos a inundación de información, pues se debe vincular la identificación de las etiquetas con información relevante; y son justamente los middlewares quienes mediante estandarización manejan adecuadamente la inundación de información. Sus funciones se extienden al filtrado de eventos y la administración de lectores. A continuación, se describen sus componentes:

- **Adaptador de Lector:** proporciona un interfaz entre los lectores y el sistema; encapsula las interfaces de lectores propietarios para que sean transparentes a los desarrolladores de aplicaciones.
- **Administrador de eventos:** se encarga de recopilar datos de lectura de radiofrecuencia sin procesar desde los lectores (u otras fuentes) los consolida y filtra basado en reglas (filtros) preestablecidos de nivel de aplicación.
- **Interfaz de nivel de aplicación:** consta de una API para configurar, gestionar y supervisar el middleware y los lectores bajo su control. así también, proporciona un mecanismo estandarizado que permite a las aplicaciones registrarse y recibir eventos RFID filtrados de un conjunto de lectores. EPCglobal se ha encargado de publicar el estándar ALE (Eventos de Nivel de Aplicación) para la gestión de eventos en la funcionalidad de RFID.

En algunos ambientes además de los componentes descritos se pueden añadir elementos que brindan servicios de almacenamiento de información como el centro de datos empresarial que consta de las aplicaciones empresariales que presentan resultados del conjunto de datos recogidos; y del bus de servicios empresariales que trata de las diferentes formas de integración de aplicaciones.

2.8.2 Radios de operación

Para propiciar una comunicación apropiada tanto lector como etiqueta deberán estar operando a la misma frecuencia. La banda en la que los sistemas RFID trabajan es en la banda no licenciada ISM, en la Tabla 3 se muestra los espectros de radio frecuencia más utilizadas por RFID, en conjunto con algunas características importantes.

Tabla 3. Frecuencias de Operación y Distancias de Lectura se Sistemas RFID [7]

Nombre	Rango de Frecuencias	Rango de Longitud de Onda	Frecuencias ISM [MHz]	Rango de Lectura para Etiquetas Pasivas [m]
Baja Frecuencia (LF)	30 - 300 KHz	1 - 10 Km	0.125 - 0.135	< 0,5
Alta Frecuencia (HF)	3 - 30 MHz	10 -100 m	6.78 8.11 13.56 27.12	< 3
Ultra-alta Frecuencia (UHF)	300 MHz -3 GHz	10 cm -1 m	433 869 915	< 9
Microondas	1 - 300 GHz	1 mm - 30 cm	2440 5800	> 10

El desarrollo del presente trabajo en cuanto a la tecnología RFID se centrará básicamente en el espectro UHF específicamente en la frecuencia de operación de los 433 MHz.

Los sistemas RFID que trabajan en UHF, tienden a operar con velocidades de transferencia y lectura más altas respecto a LF y HF, por lo que su aplicación se orienta a ambientes abiertos y con una importante densidad de etiquetas.

2.8.3 Estándares

Entre los organismos que han implementado estándares para el manejo de los entornos RFID se tiene a, EPCglobal e ISO/IEC.

2.8.4 ISO/IEC 18000-7 [8]

Estándar ratificado y publicado en 2004 para la tecnología RFID, implementa una regulación de parámetros para la interfaz aire de dispositivos que funcionan como etiquetas activas en la banda de los 433 MHz. Su propósito es establecer una especificación común para dispositivos RFID, fomentando su compatibilidad e interoperabilidad con otros estándares ISO/IEC y dispositivos RFID.

El sistema en el que se enfoca la norma es aquel compuesto por los elementos tradicionales lectora (interrogadora) y etiqueta, donde la etiqueta se coloca en el elemento a gestionar y la lectora controla el protocolo, comunicándose con las etiquetas, es así que las lee, almacena sus datos y asegura la entrega y validez del mensaje. Los sistemas definidos por ISO/IEC 18000 incorporan a más de las funciones mencionadas: la escritura de datos, manejo de múltiples etiquetas, detección de errores, entre otros.

Se establece que un sistema RFID de acuerdo a esta normativa debe:

- Trabajar en la banda de frecuencia de 433.92 MHz.
- Con un ancho de banda de 500 KHz.
- A una potencia de transmisión máxima de 5.6 dBm EIRP (Equivalent Isotropically Radiated Power) o 3.45 dBm ERP (Effective Radiated Power), equivalente a 2.2 mW.

Capa Física

A continuación, se mencionan las características para establecer la comunicación entre lectora y etiqueta mediante la frecuencia UHF de banda estrecha.

- Frecuencia de portadora 433.92 MHz \pm 20 ppm,
- Tipo de modulación FSK,
- Derivación de frecuencia \pm 50 KHz,
- Tasa de modulación 27.7 KHz,
- Señal de arranque 30 KHz.

Señal de arranque, es un tono de sub-portadora de 30 KHz en un período de 2.5 a 2.7 s.; dicha señal la transmite la lectora durante 2.5 s como mínimo para activar a las etiquetas que se encuentran dentro del rango de alcance, una vez que las etiquetas detectan la señal pasan al estado *listo* (ready) en espera de comandos enviados por la lectora.

Esta forma de comunicación es maestro-esclavo, permite transmisiones de respuesta múltiple por parte de las etiquetas, es así que para controlar la respuesta múltiple se usa el algoritmo de “Recopilación de Etiquetas y Arbitraje de Colisiones” (Tag Collection and Collision Arbitration).

Capa Enlace de Datos

La transmisión de datos entre lectora y etiqueta se realizan en forma de paquete, el cual está formado de preámbulo, carga útil y un último período lógico pequeño. El fin del preámbulo y el inicio del primer byte de datos se evidencian por los últimos dos impulsos del preámbulo. Los bytes de datos se transmiten con el formato de código

Manchester, enviándose primero los bytes más significativos, pero considerando que dentro de un byte el primer bit es el menos significativo.

2.8.5 Formato de la PDU

Formato General del Paquete en la Capa Enlace de Datos

El paquete se divide en cuatro espacios principales, los cuales se desarrollan seguidamente:

- **Preámbulo:** Compuesto por 20 impulsos con un período 60 μ s (30 μ s en alto y 30 μ s en bajo) seguido por un impulso de sincronización final de 42 μ s en alto y 54 μ s en bajo, que identifica la dirección de comunicación.
- **Carga Útil:** Codificado mediante Manchester (impulso descendente en el centro del tiempo 1L e impulso ascendente 0L, bit de parada codificado como 0L); este espacio está compuesto por 8 bits de datos y un bit de parada. Donde el período de bit será 36 μ s, con un período total de 324 μ s.
- **CRC:** Se calcula con un valor inicial de 0xFFFF sobre todos los bytes de datos (a excepción del preámbulo) de acuerdo con el polinomio CCITT (Comité Consultatif International Télégraphique et Téléphonique) ($x^{16}+x^{12}+x^5+1$). El CRC se agrega a los datos en forma de 2 bytes.
- **Período Final del Paquete:** Período de 36 μ s de pulso lógico en bajo que se transmite luego del CRC.

Formato del Mensaje Lectora - Etiqueta

Este mensaje se fracciona en los campos que se muestran en la Figura 49 y que se desglosan a continuación:

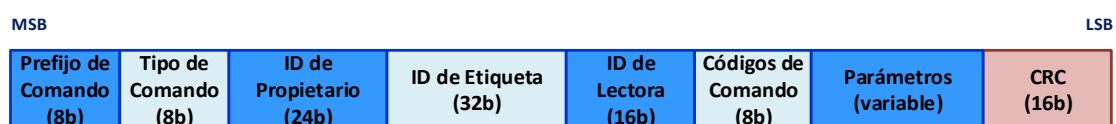


Figura 49. Vista esquemática del Mensaje Lectora -Etiqueta

- **Tipo de Comando:** Se utiliza para indicar la presencia (estableciendo en 1) de los campos ID de etiqueta e ID de propietario en el paquete de datos actual, consta de los sub-campos que se muestran en la Figura 50.

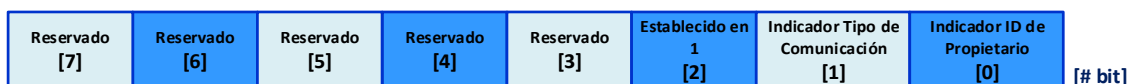


Figura 50. Formato del campo Tipo de Comando

- **ID de Propietario:** Permite la comunicación entre la lectora con las etiquetas que hayan sido establecidas con un ID de propietario específico. Este ID se

puede programar arbitrariamente y cambiarse a posterior en la memoria no volátil de cada etiqueta.

- **ID de Etiqueta:** Es un número entero de 32 bits que identifica unívocamente a cada etiqueta y que se le asigna durante su fabricación, por lo que no se puede cambiar y consecuentemente es solo de lectura. La estructura del ID de etiqueta se encuentra en la norma ISO/IEC 15963.
- **ID de Lectora:** Es un número entero de 16 bits programado en la memoria no volátil de la lectora, pudiéndose cambiar a conveniencia. Se utiliza para enrutar eficientemente respuestas de etiquetas a través de la red de lectoras.
- **Códigos de Comando:** Campo de 8 bits, cuyos 7 bits menos significativos identifican su función base mientras que el octavo bit se establece en 0 para lectura y 1 para escritura. Algunos de los comandos que especifica son: estatus, número de modelo, revisión de firmware, establecer contraseña, ente otros.

Los campos ID de etiqueta, Id de propietario y Parámetros se encuentran presentes en función de lo que establezca el campo Código de Comando.

Mecanismos de comunicación: La tecnología RFID permite diversos tipos de comunicaciones, para lo cual la norma ISO/IEC 18000-7 utiliza algunos campos del mensaje lectora – etiqueta descritos con anterioridad.

- **Unicast:** En este tipo de comunicación la lectora que se comunica con una etiqueta debe especificar el ID de etiqueta por lo que se establece en 1, el bit 1 del campo tipo de comando.
- **Broadcast:** Este esquema permitirá a la lectora comunicarse con todas las etiquetas, dentro de su rango de cobertura; por lo que el bit 1 del campo tipo de comando se establecerá en 0, en consecuencia, el campo ID de etiqueta se omitirá en el paquete de datos. En cambio, sí se establece en 1 el bit 0 del campo tipo de comando, este indica la presencia del campo ID de propietario dentro del mensaje, por lo que las etiquetas que contengan ese ID de propietario responderán a la comunicación con la lectora.

Formato del Mensaje Etiqueta – Lectora

Dependiendo del tipo de mensaje enviado de lectora a etiqueta los mensajes que respuesta etiqueta a lectora pueden ser:

Formato de Mensaje de Respuesta de Broadcast

Mensaje de respuesta a un mensaje Broadcast enviado por la lectora. Los comandos de Broadcast se utilizan para recoger el ID de etiqueta, ID de usuario o bloques cortos

de datos de un grupo de etiquetas seleccionado mediante el algoritmo de recolección de grupo. Se divide en los campos que se muestran en la Figura 51.

Estado de la Etiqueta (16b)	Longitud del Mensaje (8b)	ID de Lectora (16b)	ID de Etiqueta (32b)	ID de Propietario (24b)	ID de Usuario (0-128b)	Datos (0-variable)	CRC (16b)
-----------------------------	---------------------------	---------------------	----------------------	-------------------------	------------------------	--------------------	-----------

Figura 51. Formato Mensaje Respuesta de Broadcast

- **Estado de Etiqueta:** Indica condiciones como formato de respuesta, tipo de etiqueta e indicador de alerta. Contiene los sub-campos que se muestran en la Figura 52.

Campo Modo [15-12]	Reservado [11-9]	Indicador ACK [8]	Reservado [7-6]	Tipo de Etiqueta [5-3]	[2]	Indicador de ID de Usuario [1]	Indicador de Batería [0]	[# bit]
--------------------	------------------	-------------------	-----------------	------------------------	-----	--------------------------------	--------------------------	---------

Figura 52. Formato Campo Estado de Etiqueta

- **Longitud del Mensaje:** Especifica la longitud del mensaje en bytes incluyendo los códigos CRC.
- **ID de Lectora:** Valor entero entre 1 y 65535.
- **ID de Etiqueta:** Identificador preestablecido por el fabricante.
- **ID de Propietario:** Identificador único asignado a la corporación.
- **ID de Usuario:** Campo opcional establecido por el usuario.
- **Datos:** Su contenido depende del tipo de comando enviado por la lectora para recopilar información de las etiquetas, este contenido varía entre el ID de usuario o datos específicos alojados la memoria de la etiqueta.

Formato de Mensaje de Respuesta Unicast

Mensaje enviado por la etiqueta a la lectora, en respuesta a un comando unicast el cual necesita el ID de etiqueta para obtener información de la misma. La distribución de sus campos se observa en la Figura 53.

Estado de la Etiqueta (16b)	Longitud del Mensaje (8b)	ID de Lectora (16b)	ID de Etiqueta (32b)	Código de Comando (8b)	Parámetros (variable)	CRC (16b)
-----------------------------	---------------------------	---------------------	----------------------	------------------------	-----------------------	-----------

Figura 53. Formato Mensaje de Respuesta Unicast.

Los campos Estado de Etiqueta, Longitud del Mensaje, ID de Lectora e ID de Etiqueta tienen el mismo significado que en el mensaje de respuesta broadcast.

- **Código de Comando:** Contiene el código de comando recibido.
- **Parámetros:** Incluye los parámetros establecidos por el campo código de comando.

2.9 Plataformas Hardware

2.9.1 Samsung ARTIK™ 530 [9]

Samsung ARTIK es una plataforma IoT integrada (hardware y software) que ofrece productos y servicios seguros, interoperables e inteligentes. Ofrece un conjunto de módulos para el desarrollo de IoT, entre los que se encuentra ARTIK™ 530 (Ver Figura 54); que entre sus características principales presenta:

- Procesador Quad Core 32-bit, Cortex® A-9 @ 1.2 GHz, para procesamiento local de datos
- ARM MALI™ GPU (motor multimedia), para tratamiento de audio y video.
- Admite los estándares IEEE 802.11a/b/g/n, ZigBee (sobre 802.15.4), Bluetooth 4.2 (Ble + Classic) y Thread.
- Incluye DRAM de 512 MB y flash de 4 GB.
- Cuenta con tres tarjetas principales:
 - **Tarjeta Interposer:** entre sus componentes principales comprende, módulo ARTIK™ 530; botones de encendido y reset; puertos HDMI (High-Definition Multimedia Interface), Ethernet, USB OTG (On-The-Go); adaptadores de antena 802.11/Bluetooth y Zigbee, etc.
 - **Tarjeta Plataforma:** Se interconecta a la parte inferior de la tarjeta interposer para proveerle circuitos y funcionales adicionales. Consta de los siguientes elementos principales, switch de encendido/apagado, adaptador DC 5 V para alimentación, UART (Universal Asynchronous Receiver-Transmitter), conector para la tarjeta interfaz, entre otros.
 - **Tarjeta interfaz:** Se conecta a las tarjetas interposer y plataforma mediante el conector de expansión de la tarjeta plataforma; permite la interconexión con interfaces externas mediante 80 pines programables divididos en dos instancias (J2 y J3) que permiten la interacción con elementos externos.
- Incluye el Sistema Operativo Fedora® con herramientas multimedia, conectividad, gráficos, administración de energía y librerías de seguridad.
- Se utiliza en aplicaciones: de automatización, multimedia, inteligentes entre otras.
- Permite la conexión mediante interfaces y conectores adecuados de elementos externos como cámara, tarjeta de sensores, touchscreen LCD (Liquid Cristal Display) u OLED (Organic Light-Emitting Diode).
- Cuenta con elementos de seguridad, que brindan autenticación de seguridad punto a punto para la transferencia de datos.

- Mediante hardware externo posibilita el uso de interfaces Ethernet 10/100/1000 Base-T MAC. Otras interfaces con las que cuenta están GPIO, UART, I²C, USB, SPI, HSIC, ADC, etc.
- Para incorporar capacidades Thread o ZigBee sobre IEEE 802.15.4 el módulo cuenta con un transceptor compatible a 2.4 GHz, a lo que debe añadirse una antena para el establecimiento de radio comunicaciones.
- ARTIK trabaja con el procesador principal sobre un módulo coprocesador de red NPC (Procesador de Comandos ZigBee)
- Una vez que dos módulos se unan a la red pueden enviarse mensajes entre sí.

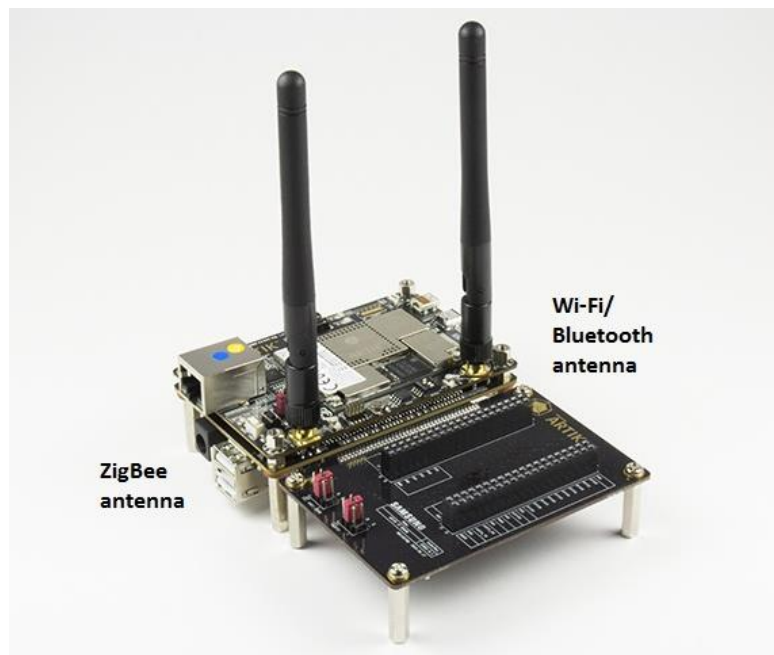


Figura 54. ARTIK™ 530 [10]

2.9.2 Arduino

Se trata de una empresa dedicada a la construcción de una plataforma integrada de hardware y software de código abierto, empleadas para desarrollar proyectos con microcontroladores.

Hardware

Consta del microcontrolador y una placa base que le brinda a éste un entorno de alimentación, funcionamiento y conexión a un computador. En lo que respecta al microcontrolador se basa principalmente en la serie megaAVR, desde la serie ATmeg168 hasta la serie ATmega2560. La placa incorpora componentes que proporcionan al microcontrolador:

- **Funcionamiento:** Oscilador de cristal para ajustar la frecuencia de operación hasta 16 MHz.

- **Conexión con un computador:** Puerto USB para la programación del microcontrolador, o en su defecto un puerto RS-232.
- **Alimentación:** Regulador de potencia, que brinda a la placa alimentación directa vía puerto USB, o mediante un conector de alimentación externa. El voltaje de entrada recomendado se encuentra entre los 7 V y 12 V.
- **Interconexión a otros componentes:** Dos filas de conectores hembra de 0.1 pulgadas que permiten la conexión de los pines del microcontrolador a elementos externos.

Algunas de las placas más utilizadas se citan a continuación [11]:

- **Arduino Uno:** Evolución de la primera placa de Arduino, se trata de un modelo básico que se fundamenta en el microcontrolador ATmega328, que funciona a una frecuencia de 16 MHz, con una memoria flash de 32 KB, 14 pines de entrada/salida, consta de un puerto USB tipo B.
- **Arduino Mega 2560:** Placa que basa su operación en el microcontrolador ATmega2560, con frecuencia de trabajo de 16 MHz, memoria flash de 256 KB, 54 pines para entrada/salida, incorpora cuatro puertos de comunicación serial, un SPI (Serial Peripheral Interface) y un puerto USB. Consta de un conjunto importante de periféricos.
- **Arduino Ethernet:** Desarrollada para el ámbito de la IoT cuenta con una interfaz ethernet, basada en el microcontrolador ATmega328, presenta 14 pines de entrada/salida, necesita un cable FTDI (Future Technology Devices International) para la carga de datos, incorpora lector de tarjetas micro SD, cuenta con un conector para alimentación externa.
- **LilyPad Arduino:** Se trata de una placa portátil de pequeñas dimensiones (50 mm de diámetro aproximadamente) del ámbito de los wearables (WBAN), trabajan a 8 MHz de frecuencia, se fabrican en diferentes modelos por lo que se basan microcontroladores distintos, así también de acuerdo ello varía el número de pines de entrada/salida, forma de conexión al computador entre otros aspectos.
- **Arduino Yún:** Se trata de una placa que consta de las características similares a placa IoT, pero añade un dispositivo Linux® con soporte Ethernet y Wi-Fi. Consta de un microcontrolador ATmega32u4, opera a la frecuencia de 16 MHz. El dispositivo Linux®, se compone de un chip Atheros AR9331 que trabaja a una frecuencia de 400 MHz, posee una flash de 16 MB y una ranura micro SD. En cuanto a la alimentación la recibirá a través de un puerto mini USB incorporado.
- **Arduino Mini, Micro y Nano:** Son placas de pequeñas dimensiones (de 2 × 0.75 pulgadas), incorporan los microcontroladores ATmega32u4 para la placa

micro y ATmega328 en las placas mini y nano, en lo referente a pines de entrada/salida nano y mini presentan 14 pines y la tarjeta micro 20 pines, todas operan a la frecuencia de 16 MHz y constan de una memoria de 32 KB.

La Figura 55 muestra la estructura de un Arduino Nano, mientras que la Tabla 4 define la distribución de pines.



Figura 55. Arduino Nano [12]

Tabla 4. Distribución de Pines Arduino Nano

No. PIN	Nombre	Dirección	Descripción
1-2, 5-16	D0(TR1), D1 (RX00) D2-D13	Entrada/Salida	Entradas Digitales
3, 28	Reset	Entrada	Reseteo (Activo en Bajo)
4, 29	GND	PWD	Tierra
17	3V3	Salida	Salida +3.3V (desde FTDI)
18	AREF	Entrada	Referencia ADC
19-26	A0-A7	Entrada	Entradas Analógicas
27	+5V	Entrada o Salida	Salida de 5V (del regulador) Entrada 5V (alimentación externa)
30	VIN	PWR	Alimentación de Voltaje

Software

Se trata de un IDE (Integrated Development Environment), que consiste en un editor de código y un compilador integrado, fundamentado en el lenguaje C. Herramientas que son comunes para cualquier familia de placas Arduino.

El programa de instalación se encuentra disponible para OS X, Windows y Linux®. En cuanto al entorno de desarrollo consta de una barra de herramientas, la sección de

edición de código y un área de mensajes que permite al usuario conocer el estado del desarrollo o si existe un error en el mismo.

A los elementos mencionados se añade una gran comunidad Online de Arduino dedicada dar soporte al desarrollo de proyectos que se construyen en base a esta plataforma.

2.9.3 XBee Radio

Se trata de hardware de comunicación creado por la compañía Digi International. Estas soluciones integradas que posibilitan la interconexión y comunicación entre dispositivos. Su diseño fue pensado para aplicaciones que manejen gran cantidad de datos, baja latencia y sincronización predecible.

Los protocolos sobre los que Digi ha desarrollado módulos XBee son propietarios, IEEE 802.11 y ZigBee; pero el desarrollo del presente numeral se centrará sobre este último protocolo, por lo que subsecuentemente se enumeran algunas de las familias XBee:

- XBee y XBee-PRO, 802.15.4
- XBee and XBee-PRO, ZB
- XBee and XBee-PRO, DigiMesh 2.4

Los módulos XBee se fabrican en versión normal y PRO, la ventaja de versión PRO, frente a versión normal, radica en que poseen mejores características en cuanto a potencia en el transmisor y sensibilidad en el receptor; con el fin de ampliar el área de cobertura de dichos dispositivos.

Los dispositivos XBee se dividen en series, en función del chipset que utilizan, es así que la serie 1 soporta únicamente comunicaciones punto a punto mientras que las series 2 y 2.5 soportan algunas formas de comunicación como redes de malla.

Uno de los factores a considerar, al momento de elegir un módulo XBee, es el tipo de antena que se emplea, enseguida se establecen algunas características para antenas de la familia ZB:

- **WHIP:** Consiste en un alambre soldado al módulo XBee, con una capacidad de radiación omnidireccional.
- **CHIP:** Se trata de un chip de cerámica adaptado casi a la misma altura de la placa del módulo XBee, en este caso la señal radiada se atenúa en múltiples direcciones.
- **U.FL.:** Conector para adaptar una antena externa que se conecta al módulo XBee mediante un cable adaptador (pigtail).

- **RPSMA:** Conector de antena externa que se conecta directamente al módulo XBee.
- **PCB:** Se encuentra incrustada en el módulo. Disponible solo para la versión PRO.

En la Tabla 5 se muestra la distribución de pines de XBee:

Tabla 5. Distribución de Pines XBee [13]

No. PIN	Nombre	Dirección	Dirección por Defecto
1	VCC	NA	NA
2	DOUT	Salida	Salida
3	DIN/CONFIG	Entrada	Entrada
4	DIO12	Ambas	Deshabilitado
5	RESET	Ambas	Colector abierto con pullup
6	RSSI PWM/DIO10	Ambas	Salida
7	DIO11	Ambas	Entrada
8	Reservado	NA	Deshabilitado
9	DTR/SLEEP_RQ/DIO08	Ambas	Entrada
10	GND	NA	NA
11	DIO4	Ambas	Deshabilitado
12	CTS/DIO7	Ambas	Salida
13	ON/SLEEP	Salida	Salida
14	VREF	Entrada	NA
15	ASSOCIATE/DIO5	Ambas	Salida
16	RTS/DIO6	Ambas	Entrada
17	ADE3/DIO3	Ambas	Deshabilitado
18	ADE2/DIO2	Ambas	Deshabilitado
19	ADE1/DIO1	Ambas	Deshabilitado
20	ADE0/DIO0	Ambas	Deshabilitado

Recursos Adicionales

A continuación, se abordan algunos de los principales elementos para configurar implementar infraestructuras con XBee:

- **Adaptadores USB:** Placas que permiten la comunicación entre un computador y el módulo XBee para poder configurarlo y actualizarlo. Entre los adaptadores más comunes se tiene: SparkFun XBee, Adafruit XBee Adapter Kit, New Micros XBee Dongle, Gravitech XBee to USB Adapter.
- **Arduino Board Adapter Hack:** Placa que admite la configuración XBee desde el software de código abierto de Arduino, además admite la utilización de algunas características X-CTU (XBee Configuration and Test Utility). En la Tabla 6 se presenta la forma de conexión de pines entre XBee y la placa Arduino:

Tabla 6. Pines de Conexión entre XBee y Arduino Board Adapter Hack [14]

XBee	Arduino Board Adapter Hack
VCC o 3.3V	3V3
TX o DOUT	RX o 0
RX o DIN	TX o 1
GND	GND

Existen otros elementos de soporte, algunos de ellos opcionales, para el despliegue de la red como: breadboard, Digi Evaluation Board (tarjeta de evaluación), Breakout Boards (adaptadores conectar XBee al breadboard).

2.9.4 Kits de Desarrollo RFID [15]

La tecnología RFID cuenta con múltiples dispositivos que se pueden integrar, a continuación, se describe los elementos de un kit comercial que se adecúa al propósito del presente trabajo:

Lector Serial L-RX202

Se trata de un lector de largo alcance con alta sensibilidad (de 0.5 – 137 m al aire libre) que opera con etiquetas RFID activas. Otras de sus características son:

- Frecuencia de operación 433 MHz, con forma de modulación ASK
- Admite la conexión con un PC (o Sistema Middleware RFID) mediante una interfaz RS-232 o RS-485.
- Incorpora la posibilidad de conectarse a una red de acceso cableada mediante un conector serie a Ethernet.
- Se puede configurar mediante el puerto micro USB y USB con alimentación.
- Permite el ajuste de rango de alcance
- Soporta una variedad de antenas que pueden operar a distintos rangos de frecuencia dependiendo de la necesidad de alcance.

- El lector puede actuar como de forma particular o como parte de una red de lectores; para esto último se permite el uso de más de 255 direcciones de receptor únicas.
- Los rectores dispuestos en red se alimentan y conectan mediante cable UTP Cat. 5.
- Modos de transmisión de datos:
 - Auto-Sondeo de Hardware: los datos de las etiquetas se proporcionan automáticamente desde la red de lectores.
 - Software de sondeo: Los datos de la etiqueta se solicitan individualmente por un lector inteligente o un sistema Middleware RFID.
- Cuenta con la certificación de FCC (Federal Communications Commission) y ETSI (European Telecommunications Standards Institute)
- Su ámbito de aplicación principal es el rastreo de elementos activos.

Etiqueta TGP Personnel

Entre sus características destacadas están:

- Etiqueta activa que envía un ID único y datos en intervalos de tiempo ajustables.
- Frecuencia de transmisión 433.92 MHz. Rango de alcance 1-100 metros dependiendo de la lectora escogida y número de etiquetas en el sistema.
- Ultra-bajo consumo de energía que permite larga duración de la batería, se incluye alertas de finalización de vida útil de la batería.
- Transmisor de ultra largo alcance
- Presenta funciones opcionales, configurables como: Alerta anti-manipulación instantánea, alerta instantánea de movimiento, soporte de otras tecnologías RFID (como HID y UHF GEN2).
- Admite un algoritmo para evitar colisiones.
- Se debe tomar precaución de no colocar la etiqueta directamente en superficies metálicas.
- Tamaño 86 × 54 × 5 mm

Micro Etiqueta TGM-A

Sus principales características se citan a continuación:

- Etiqueta activa que envía un ID único y datos en intervalos de tiempo ajustables.
- Frecuencia de transmisión 433.92 MHz. Rango de alcance 1-50 metros dependiendo de la lectora escogida y número de etiquetas en el sistema.

- Ultra-bajo consumo de energía que permite larga duración de la batería, anexa alertas de finalización de vida útil de la batería.
- Incorpora funciones opcionales, como: Alerta anti-manipulación instantánea, alerta instantánea de movimiento.
- Anexa un algoritmo para evitar colisiones.
- Incluye una antena omnidireccional que permite su lectura en cualquier dirección sin importar si se coloca cerca de superficies metálicas.
- Puede adherirse a un activo, mediante elementos adicionales propios de la marca.
- Tamaño 29 × 25 × 7 mm

Etiqueta TGFL-A Key Fob con Pulsador

Entre sus principales propiedades están:

- Indican activamente la presencia de una red de lectoras de modelos como, L-RX201, L-RX202 o L-RX211
- El pulsador permite enviar señales separadas a las lectoras para el control de sistemas conectados a la red.
- Se alimenta mediante batería interna que determina directamente su vida útil.
- Emite una señal de radio frecuencia con intervalo preestablecido de 15 segundos.

Elementos Adicionales

- Antena Stub a 433 MHz
- Cable para conexión a puerto Serial
- Demo y aplicación de ajuste de rango
- Software analizador
- Versión de prueba de 30 días del software RF-Watch, Homeseer Software, Indigo Software for OS-X

Una vez que se ha expuesto un análisis completo de las tecnologías involucradas en el presente estudio, la Tabla 7. Comparación de los Estándares Bluetooth 3.0, ZigBee 3.0, IEEE 802.11ac y RFID muestra un resumen útil para el desarrollo de los siguientes apartados concernientes al presente trabajo.

Tabla 7. Comparación de los Estándares Bluetooth 3.0, ZigBee 3.0, IEEE 802.11ac y RFID

Tecnología	Alcance (m)	Velocidad de datos sobre el aire (Mbps)	Frecuencia de Operación (GHz)	Potencia de consumo (mW)	Ancho de Banda mínimo Requerido (MHz)	Número máximo de dispositivos Asociados
Bluetooth 3.0	~10 ⁶	24 ¹	2.4 - 2.485 ¹	2.5 ¹ (Clase 2)	15 ² (dinámico)	1 maestro con 7 esclavos (Piconet) ³
ZigBee/ 802.15.4	13 ⁴	0.25 ⁴	2.400 - 2.4835 ⁵	1 ⁴	3 ² (estático)	Decenas ⁶
IEEE 802.11ac	200- 300 (m ²) ⁷	1300 ⁸	5 ⁸	Depende del tiempo de transmisión de datos ⁹ ~11000-15000 ¹⁰	80 ⁸	30-60 (clientes) ¹¹
RFID	<9 ¹²	0.0277 ¹³	0.3-3 ¹² (0.433)	2.2 ¹⁴ (para transmisión)	0.5 ¹⁴	-

¹ Fuente: <http://gizmodo.com/5232940/giz-explains-everything-bluetooth-and-why-bluetooth-30-is-better>

² Fuente: http://ftp1.digi.com/support/documentation/019-0162_D.pdf, p. 13

³ Fuente: <https://www.quora.com/Why-does-Bluetooth-only-allow-7-simultaneous-slaves>

⁴ Fuente: https://rua.ua.es/dspace/bitstream/10045/1109/7/Informe_ZigBee.pdf, p. 5

⁵ Fuente: <http://bit.ly/2qbfBOD>

⁶ Fuente: <http://bit.ly/2oF6zNp>

⁷ Fuente: <http://bit.ly/2pi7Lpw>

⁸ Fuentes: http://www.arubanetworks.com/assets/poster_wave2.pdf, <http://bit.ly/2oF6upk>

⁹ Fuente: http://www.arubanetworks.com/assets/wp/WP_80211acInDepth.pdf, p. 27

¹⁰ Fuentes: <http://bit.ly/1OO5rdm>, <http://bit.ly/2osUDi5>

¹¹ Fuente: <http://bit.ly/2oIQ2En>

¹² Fuente: <http://bit.ly/2oIWQ4L>, p. 121

¹³ Fuente: International Standard ISO/IEC 18000-7: Information Technology - Radio-frequency identification for item management, Part 7: Parameters for active air interface communications at 433 MHz p. 16

¹⁴ Fuente: http://tel_archives.ofca.gov.hk/en/ad-comm/rsac/paper/rsac2_2007.pdf, p. 2

3 Entornos Museísticos y Tecnologías Inalámbricas

3.1 Análisis de Entornos Museísticos

Los museos son instituciones de carácter público o privado que se encargan de preservar bienes culturales de diverso tipo con el fin de presentarlos al público.

De acuerdo con la temática de colecciones que presentan y otros aspectos determinados por ICOM (Consejo Internacional de Museos) los museos pueden ser de:

- Arte
- Generales, Especializados, Monográficos y Mixtos
- Historia
- Etnografía, Antropología y Artes Populares
- Ciencias Naturales
- Científicos y Técnica Industrial
- Otras Variaciones

Así también, algunos grupos presentan subdivisiones que permiten describir y orientar mejor sus contenidos y objetivos; este es el caso de los Museos Generales, Especializados, Monográficos y Mixtos, en los cuales se incluyen los museos: al aire libre, jardines, reservas y parques naturales.

En función de las colecciones que se presenten en los museos las exposiciones pueden ser: permanentes, temporales e itinerantes; así también se debe considerar que puede haber elementos dentro de una colección que deberán rotar con el fin de que se puedan preservar adecuadamente.

3.1.1 Museografía

Según la ICOFOM (Comité Internacional del ICOM para la Museología), la museografía es “el conjunto de técnicas desarrolladas para llevar a cabo las funciones museales y particularmente las que conciernen al acondicionamiento del museo, la conservación, la restauración, la seguridad y la exposición” [16].

Dentro de este contexto el diseño museográfico se refiere a la exhibición de piezas; para lo cual un museógrafo (o curador) elabora un guion (documento que organiza la colección de un museo, incluyendo su descripción y demás elementos de apoyo para su exhibición), en base a los recursos, espacio y demás elementos que permitan una difusión adecuada al público.

3.1.2 Proyecto Museográfico

La exhibición de colecciones museísticas implica varios aspectos que deben considerarse:

- Elementos de diseño, para lo cual es necesario conocer las características de las piezas de una colección (información que deberá incluir el guion); en el caso de esculturas se debe conocer sus medidas, peso, tipo de material de composición y requerimientos de montaje para su conservación; con el objetivo de especificar los elementos necesarios para su exhibición.
- El museógrafo usará la información proporcionada por el guion para elaborar una distribución adecuada de las piezas museísticas en el espacio asignado; considerando los accesos, fuentes de luz natural, salidas de aire acondicionado, dispositivos tecnológicos que se deban incorporar, entre otros aspectos que estimen importantes.
- El diseño se debe plasmar realizar con el equipo de museística en planos y maquetas con el fin de evitar cualquier tipo de inconvenientes en el momento del montaje.
- La preservación de las piezas es uno de los principales aspectos a tomar en cuenta, por lo que en el diseño deberá contemplar el uso de sistemas de seguridad, sistemas contra incendios, control de iluminación, control de humedad y temperatura.
- Ya en el montaje como tal se debe considerar, la escala (referente al nivel de colocación de las piezas para su observación), la distribución, ubicación de bases y paneles, vitrinas.

En lo que respecta al funcionamiento de un museo de entorno abierto la preservación de piezas implica considerar los siguientes aspectos:

- Mantener un inventario actualizado de las piezas museísticas, incluyendo fotografías de las mismas.
- Inspeccionar constantemente cada una de las piezas, para detectar irregularidades.
- Aspectos ambientales como humedad, agentes contaminantes y otros factores deberán ser considerados para evitar los múltiples riesgos de deterioro a los que están expuestos los elementos museísticos.

3.1.3 Distribución de Piezas Museísticas (Definición del Área de Trabajo)

El montaje, que corresponde a la ubicación, colocación y ensamblaje de las piezas museísticas, depende directamente del museo, el museólogo, el tipo de exposición y la colección museística. Para el presente desarrollo se considera un museo desplegado en

un espacio abierto que presenta esculturas, cuya distancia varía entre 10 y 30 metros como se muestra en la Figura 56.



Figura 56. Distribución de Piezas Museísticas en Entornos Abiertos [17]

3.2 Elección de la Tecnología

Al tratarse de un ambiente museográfico en espacio abierto, existen algunos factores que se deben tomar en cuenta para el empleo de una tecnología en particular; en función de lo cual, en este sub-numeral se evalúa y se compara cada una de las tecnologías descritas en los apartados 2.4, 2.6, 2.7 y 2.8.

3.2.1 Factores de Evaluación

Previo a la evaluación de las tecnologías se presenta las características principales consideradas que incidirán en la elección de la tecnología más adecuada, las cuales se las ha dividido en dos grupos, como se describe a continuación.

Características del entorno:

- **Rango de Identificación:** La señal del dispositivo fijo deberá ser identificada, a un rango de distancia entre los 50 centímetros y 10 metros, por el dispositivo móvil para que se ejecute la reproducción de audio.
- **Fuentes de interferencia:** Considerar si la tecnología se adapta a ambientes abiertos, con altos niveles de interferencia (principalmente asociados a factores climáticos y ruido ambiente), esto incidirá directamente en la identificación de la señal entre los dispositivos.

Características tecnológicas:


- **Velocidad de datos:** Tener la certeza de que se cuenta con una velocidad adecuada para el establecimiento de la conexión y transmisión de señales entre los dispositivos.
- **Consumo de Potencia:** Factor principal a considerar, ya que de esto depende la autonomía y movilidad de los equipos para su operación en museo en ambiente abierto. Se considerará el máximo consumo de potencia, cuando el radio del dispositivo utilice la tecnología en evaluación.

- **Exactitud:** Es importante que los dispositivos devuelvan la información correcta, correspondiente a la pieza museística a la que se ha acercado un visitante con un dispositivo móvil, considerando que en el ambiente existirán señales de dispositivos relacionados con los elementos museográficos y dispositivos de otras infraestructuras.
- **Coste:** Revisar brevemente la inversión en cuanto a equipos y demás elementos que conformen el despliegue de la infraestructura.

3.2.2 Evaluación de Wi-Fi (IEEE 802.11a/b/g/n)

En cuanto a esta tecnología se plantea el uso de Arduino Nano conjuntamente con Intel Edison como elementos principales del dispositivo fijo, por otro lado, el módulo ARTIK™ 530 sería el dispositivo móvil; las características destacadas de estos elementos se presentan en la Tabla 8; considerar que los valores que se presentan se asocian a IEEE 802.11 en sus versiones a/b/g y n, compatibles con las características que poseen los dispositivos mencionados.

Tabla 8. Características de Evaluación, IEEE 802.11a/b/g/n

Factor de Evaluación	 Wi-Fi (IEEE 802.11a/b/g/n)	
	Fijo	Móvil
Dispositivo	Arduino Nano + Intel Edison	ARTIK™ 530
Cobertura (m)	10 – 200 ¹⁵	
Fuentes de Interferencia	Líneas eléctricas de alta tensión, teléfonos inalámbricos, dispositivos que emitan señales en las bandas de 2.4 y 5 GHz. ¹⁶	
Velocidad de Datos (Mbps)	54 – 248 ¹⁷	
Consumo de Potencia (mW)	~ 640 + ~ 1400 ≈ 2040 ¹⁸	~ 2500 ¹⁹
Coste (\$)	~ 12 ²⁴ + ~ 75 ²⁰ ≈ 87	~ 180 ²¹

3.2.3 Evaluación de Bluetooth

La Tabla 9 considera valores asociados (algunos de ellos aproximados) a los dispositivos fijo y móvil, en este caso el dispositivo fijo se conforma esencialmente de

¹⁵ Fuente: <https://developer.artik.io/documentation/developer-guide/wireless-iot/>

¹⁶ Fuente: <https://support.apple.com/es-es/HT201542>

¹⁷ Fuente: <http://bit.ly/2oF6zNp>

¹⁸ Fuente: <http://intel.ly/2rdDXIC>, <http://intel.ly/2sKZcSv>, <http://bit.ly/2sXjDwq>


¹⁹ Fuente: <http://bit.ly/2qUAoe5>

²⁰ Fuente: <https://www.sparkfun.com/news/1603>

²¹ Fuente: <https://www.arrow.com/en/products/sip-kitnxd001/samsung-electronics>

una tarjeta Arduino Nano a la que se anexa un módulo Bluetooth HC-06; en cuanto al dispositivo móvil se trata del módulo ARTIK™ 530.

Tabla 9. Características de Evaluación, Bluetooth 3.0


Factor de Evaluación	 Bluetooth 3.0	
Dispositivo	Fijo	Móvil
		Arduino Nano + Módulo Bluetooth HC-06
Cobertura (m)	9 ²²	0.5 – 20 ¹⁵
Fuentes de Interferencia	Líneas eléctricas de alta tensión, dispositivos que emitan señales en las bandas de 2.4 GHz. ¹⁶	
Velocidad de Datos (Mbps)	1 ²³	1 – 3 ¹⁵
Consumo de Potencia (mW)	~ 240 ²³ + ~ 1400 ¹⁸ ≈ 1640	~ 2500 ¹⁹
Coste (\$)	~ 12 ²⁴ + ~ 9 ²⁵ ≈ 21	~ 180 ²¹

Se debe recordar que, el módulo Bluetooth HC-06 solo puede operar en modo esclavo en cualquier topología Bluetooth.

3.2.4 Evaluación de ZigBee

Los elementos que se plantean para esta tecnología son, en la parte fija consta básicamente de una tarjeta Arduino Nano en conjunto con un módulo XBee ZB, mientras que como dispositivo móvil se mantiene a ARTIK™ 530; la Tabla 10 presenta las características más relevantes para su evaluación.

Tabla 10. Características de Evaluación, ZigBee

Factor de Evaluación	 ZigBee	
Dispositivo	Fijo	Móvil
		Arduino Nano + XBee ZB
Cobertura (m)	120 (outdoor) ²⁶	10 – 200 ¹⁵
Fuentes de Interferencia	Tecnologías inalámbricas que operan en la misma frecuencia como IEEE 802.11 b/g, Bluetooth, teléfonos inalámbricos y hornos microondas ²⁷ .	
Velocidad de Datos (KBps)	250 ²⁶	250 ¹⁵

²² Fuente: <http://bit.ly/2rSc4cj>

²³ Fuente: <http://bit.ly/2r3c19A>

²⁴ Fuente: <http://amzn.to/2s1yWWH>

²⁵ Fuente: <http://amzn.to/2sebiXt>

²⁶ Fuente: <http://bit.ly/2sfp6kx>, p. 10, 11

²⁷ Fuente: <http://bit.ly/2s7NIHK>, Chapter 8 - ZigBee Coexistence


Consumo de Potencia (mW)	$\sim 264^{26} + \sim 1400^{18} \approx 1664$	$\sim 2500^{19}$
Coste (\$)	$\sim 12^{24} + \sim 55^{28} \approx 67$	$\sim 180^{21}$

En cuanto a la potencia de consumo es importante considerar los dispositivos finales XBee ZB, reciben acuses de recibo del coordinador, que le permiten conocer si existen datos dirigidos hacia él, en caso de no tener datos por procesar el dispositivo entra en modo de reposo.

3.2.5 Evaluación de RFID

La Tabla 11 considera valores asociados (algunos de ellos aproximados) a los dispositivos fijo y móvil; para el presente escenario los dispositivos fijos son la etiqueta TGP Personnel y la micro etiqueta TGM-A; en cuanto al dispositivo móvil corresponde al lector serial L-RX202 conectado al módulo ARTIK™ 530. Si bien, en la evaluación (Tabla 11) de RFID se considera únicamente al lector, en lo que corresponde al parámetro consumo de potencia, se anexa el valor correspondiente al módulo ARTIK™ 530.

Tabla 11. Características de Evaluación, a 433 MHz

Factor de Evaluación	 RFID a 433 MHz		
	Fijo		Móvil
Dispositivo	Etiqueta TGP Personnel	Micro Etiqueta TGM-A	Lector Serial L-RX202
Cobertura (m)	1 - 100 ²⁹	1 - 50 ³⁰	20 (Antena AN-200) ³⁴
Fuentes de Interferencia	Dispositivos que emitan señales en las bandas de 433 MHz (controladores de sensores, sensores inalámbricos de un determinado sistema, domótica). ³¹		
Velocidad de Datos (Kbps)	Configurables por el usuario ³²		9.38 - 112.5 ³
Consumo de Potencia (mW)	Ultra Bajo consumo de potencia que permite una vida útil de la batería de 3 años aproximadamente ³³		$\sim 960^{34} + \sim 2500^{19} \approx 3460$
Exactitud	Alta resolución en un rango < 0.5 m		
Coste (\$)	$\sim 36^{35}$	$\sim 39^{36}$	$\sim 500^{37} + \sim 180 \approx 680$

²⁸ Fuente: <http://amzn.to/2rLd2VB>

²⁹ Fuente: <http://wavetrend.co/wave-content/uploads/TGP.pdf>

³⁰ Fuente: http://www.fastrfid.com/wavetrend/tgm_microtag.pdf

³¹ Fuente: <http://bit.ly/2rB8NNW>

³² Fuente: <http://www.wavetrend.co/product/personnel-tag-tgp/>

³³ Fuente: <http://bit.ly/2rUqSY1>, <http://bit.ly/2sfFGAN>

³⁴ Fuente: <http://www.wavetrend.net/downloads/information-sheets/readers/RX202.pdf>

³⁵ Fuente: <http://bit.ly/2qWfBmp>

³⁶ Fuente: <http://www.wavetrend.co/product/personnel-tag-tgp/>

³⁷ Fuente: <http://bit.ly/2r1NkiF>

La mayoría de los dispositivos presentados constan de un modo de reposo o ahorro de energía cuando su radio no se encuentra activo, pero IEEE 802.11 es la excepción en cuanto a esta particularidad.

3.2.6 Elección de la Tecnología a Implementarse

Para escoger tecnología radio que mejor se adapte a un museo en entorno abierto, se presenta la comparativa que se muestra en la Figura 57 y Figura 58 en las que se ha considerado tres aspectos fundamentales de las tecnologías radio: velocidad de datos (eje x), rango de alcance (eje y) y potencia de consumo (tamaño del círculo) mientras la radio se encuentra activa, las demás características, de los módulos IoT, establecidas en los numerales 3.2.2, 3.2.3, 3.2.4 y 3.2.5 se considerarán en los párrafos subsiguientes.

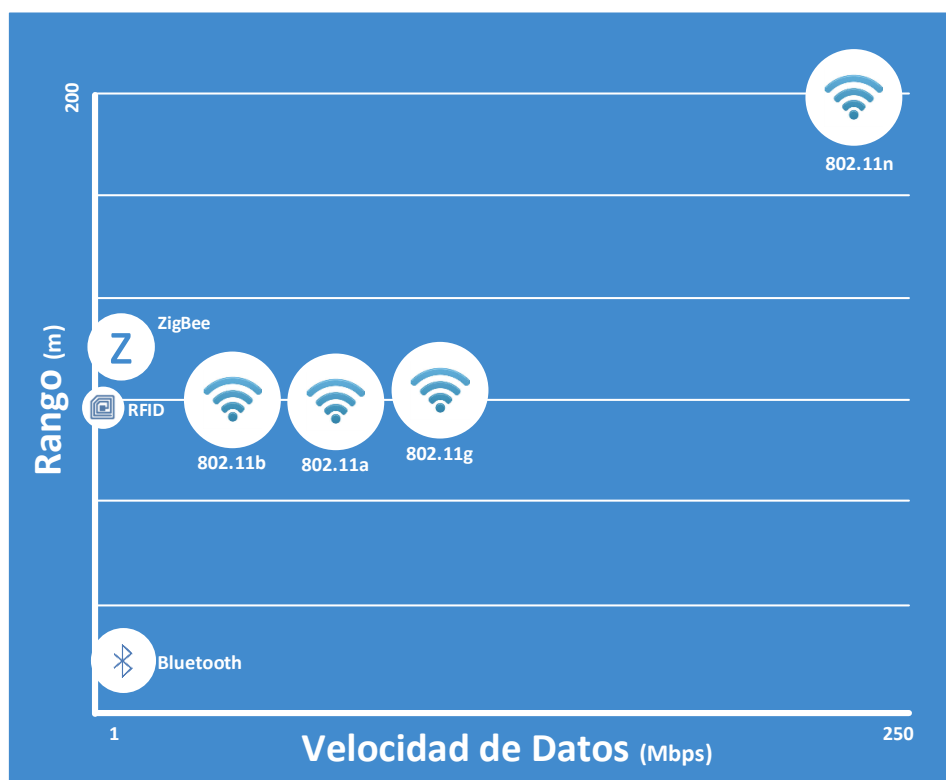


Figura 57. Comparativa entre Tecnologías (Dispositivos Fijos)

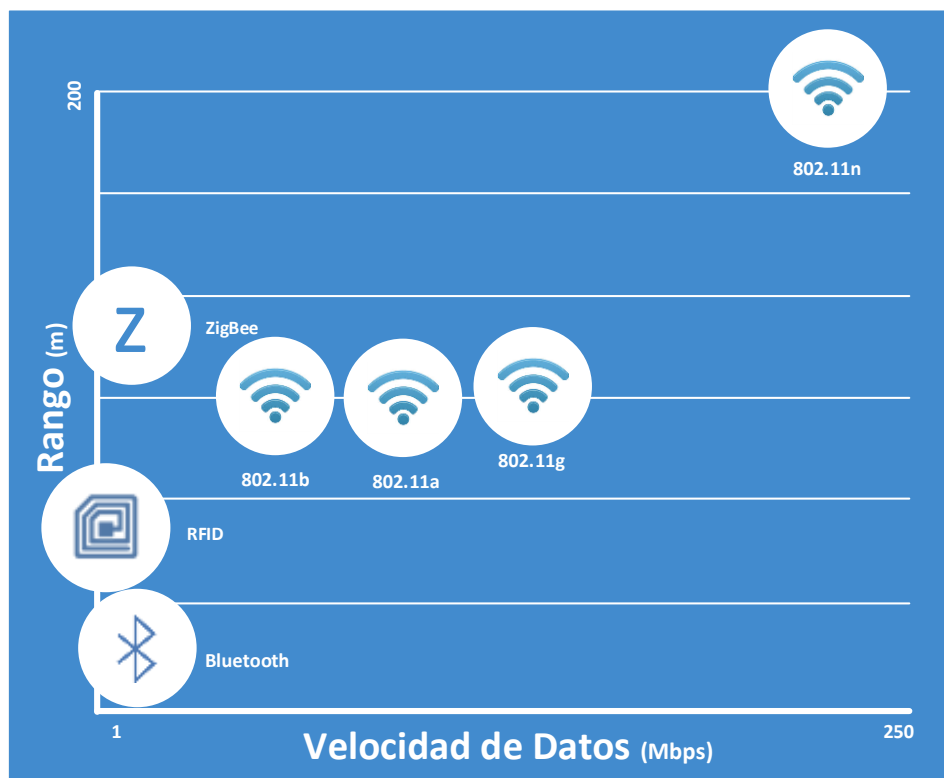


Figura 58. Comparativa entre Tecnologías (Dispositivos Móviles)

La Figura 57 y Figura 58, ofrecen una visión general de las tecnologías escogidas para el presente estudio; considerando que la velocidad de datos no incide fuertemente, en lo que corresponde al desarrollo del sistema (los justificantes se tratarán en el Capítulo 4) se debe tomar en cuenta lo siguiente:

- La tecnología IEEE 802.11 (en cualquiera de sus versiones) se ha desarrollado con la finalidad de transferir grandes flujos de datos, ejemplo de aquello es la presente comparativa a la se puede anexar la versión IEEE 802.11 ac que como se muestra en la Tabla 7 alcanza velocidades de 1.3 Gbps. El desarrollo de la presente tecnología en torno a la velocidad de datos incluye los nodos IoT que se pueden conectar a redes más grandes, pero alcanzar esta característica implica un consumo de potencia mayor que se relaciona directamente con el crecimiento de la velocidad. Respecto a las demás características, el coste de esta tecnología en lo referente a dispositivos fijos es mucho mayor, el rango de alcance no ofrece mayores ventajas al presente trabajo ya que la comunicación entre dispositivos oscila entre los 0.5 y 10 metros, y en lo referente a la interferencia por parte de otros dispositivos es muy alta debido al auge de las comunicaciones Wi-Fi. En función de lo manifestado no se perfila como una tecnología adecuada para el sistema que se desarrolla.
- Considerando que, los visitantes a cualquier tipo de museos tienden a pasar mucho tiempo en las instalaciones para apreciar las piezas museísticas, la

elección de una tecnología adecuada se centra en el consumo de potencia, como se visualiza en las figuras sujeto de la presente comparativa, la tecnologías de menor consumo en promedio son Bluetooth y RFID, en tanto que ZigBee si bien tiene un consumo de potencia similar, evidencia una desventaja para el presente desarrollo, pues al tener un rango de alcance mayor reconocerá mayor cantidad de dispositivos y por ende empleo de potencia será mayor; a lo indicado se suma un coste más alto del dispositivo fijo; en consecuencia, por las motivaciones mencionadas se descarta a ZigBee.

- De acuerdo a lo manifestado, quedarían por evaluar Bluetooth y RFID; en función de las gráficas comparativas la tecnología más adecuada sería Bluetooth ya que se ajusta más al entorno de trabajo en cuestión, así como su baja velocidad y alcance mínimo, respaldan un reducido consumo de potencia; dentro de la misma temática, cabe indicar que el kit RFID propuesto opera con Tag Activo, característica que le permite cubrir distancias considerables, lo que implicaría (en un escenario real) un consumo de energía superior a Bluetooth. Desde otro ámbito, y realizando un balance de coste general de los dispositivos RFID con el coste de los dispositivos Bluetooth, los primeros son mucho mayores por lo que su viabilidad en cuanto a este aspecto es poco probable.

En función de lo manifestado la tecnología a implementarse es Bluetooth; pero considerando las ventajas que presenta RFID se realizará una revisión tecnológica, que se profile como una opción alternativa a la que se implementará.

4 Diseño e Implementación del Sistema

Mediante el diseño e implementación de un sistema de Internet de las Cosas para un entorno museístico, se pretende que los visitantes de museos tengan una experiencia de realidad aumentada que aporte significativamente su estadía en este entorno; del mismo modo se busca dotar a los entornos museísticos de un sistema portable, eficiente y de bajo coste que los acerque a las nuevas tecnologías y sume al valor cultural de las colecciones que se exhiben.

4.1 Análisis de Requerimientos

Como se ha evidenciado en el numeral 3.1 en los ambientes de museo en espacios abiertos, las piezas museísticas se encuentran distribuidas de tal forma que se hace necesario contar con un sistema de audioguía, para que los visitantes conozcan detalles acerca de un determinado elemento.

Un sistema de Internet de las Cosas para entornos museales de espacio abierto debe cumplir con varios requerimientos, algunos de los cuales se han descrito previamente. Dichos requerimientos serán el eje sobre el cual se desarrolla el diseño y la posterior implementación del sistema; a continuación, se establecen algunos de estos requerimientos:

4.1.1 Modelo Básico de la Red

El tipo de red a implementarse es una red PAN para Bluetooth, compuesta de varios maestros y esclavos, distribuidas en varias Piconets basadas en comunicaciones machine-to-machine.

4.1.2 Dimensionamiento de Tráfico

Desde la perspectiva de la cantidad de datos a intercambiarse, el planteamiento que se muestra en la Figura 59, corrobora gráficamente lo establecido para el diseño; en el sistema como tal, no se realiza un intercambio de datos entre dispositivos, solo se ejecuta el reconocimiento de los nodos por parte del módulo receptor; en función de lo cual únicamente se necesita conocer la tasa de transferencia efectiva para un número determinado de reconocimientos, esto mediante los siguientes datos:

- Se acude al estándar [18], el cual indica que un paquete de **'Advertising data and Scan Response'** tiene un **tamaño** máximo de **31 Bytes**.
- El **índice de simultaneidad** se asocia con el número de dispositivos escaneados por parte del módulo receptor, se aproxima la cantidad de

dispositivos a un valor de **10**, asociados con aquellos que pertenecen al museo y otros de los visitantes.

- De acuerdo a varios estudios consultados [19] [20], el tiempo de ida y vuelta en una configuración Bluetooth estándar, **Round-Trip Time (RTT)**, se ha determinado en un valor promedio de **100 ms**, esto.
- La **velocidad de datos** se establece en **1 Mbps** (valor tomado de la Tabla 9)

Con los datos previos se procede a realizar el cálculo de la tasa de transferencia efectiva por paquete, mediante la siguiente ecuación:

$$C_{AD\&SR} = \frac{31 \text{ Byte}}{1 \text{ paquete}} * \frac{1 \text{ paquete}}{100 \times 10^{-3} \text{ segundos}} * \frac{8 \text{ bits}}{1 \text{ Byte}} * \frac{1 \text{ Kbits}}{1024 \text{ bits}} = 2.42 \text{ Kbps}$$

Una vez obtenida la capacidad por paquete se calcula la tasa de transferencia total, a través de la ecuación subsiguiente:

$$C_{Total} = 2.42 * 10 = 24.21 \text{ Kbps}$$

La respuesta obtenida pone de manifiesto que el enlace del sistema, que es de 1 Mbps, soportará fácilmente el número de reconocimientos considerados.

4.1.3 Consumo de Potencia (Batería)

Cuando se trata de grandes instalaciones, como un museo en entorno abierto, las personas tienden a estar mucho tiempo; de acuerdo a varios sitios web consultados en promedio las personas pasan 4 horas en un museo. Es importante entonces que los dispositivos involucrados tengan un consumo de energía apropiado, lo que se corrobora retomando algunos datos presentados en la Tabla 9, en donde se manifiesta que en promedio el módulo receptor tiene un consumo de 2,5 W y los nodos 1,6 W.

4.1.4 Distancia entre Dispositivos

Como ya se ha ido manifestando, la distancia entre dispositivos fijos y móviles oscilará entre 50 centímetros y 10 metros.

4.2 Diseño de la Red de Área Personal

4.2.1 Distribución de Dispositivos

El presente diseño se particulariza a una unidad del sistema, que se trata de una PAN distribuida en una topología de Piconet que está compuesta tal como se observa en la Figura 59 en su forma más fundamental, de dos Módulos IoT:

- **Módulo Receptor (dispositivo móvil):** consiste principalmente del Módulo ARTIK™ 530, este lo portará el visitante del museo para que emita el contenido asociado a una pieza.

- **Nodos (dispositivos fijos):** que se componen entre otros elementos del Módulo Bluetooth HC-06, una placa Arduino Nano, un acelerómetro MMA8451 y un Zumbador; a cada pieza museística se le incluirá un dispositivo fijo.

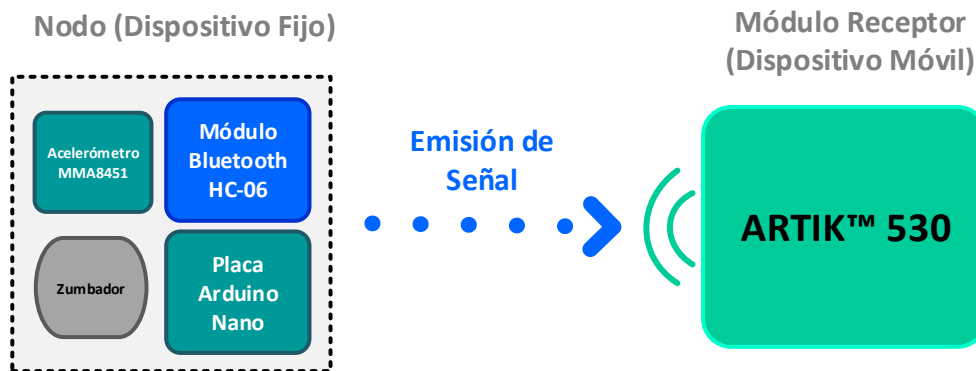


Figura 59. Elementos Básicos del Sistema

El objetivo del sistema es conseguir que los módulos IoT establezcan comunicación entre peers, esto implica que la emisión continua de señal del dispositivo fijo sea identificada, a una distancia mínima (diez metros) por la escucha continua de señales del dispositivo móvil con lo cual éste empezará a difundir información al usuario referente a la pieza museística a que se ha acercado.

4.2.2 Planteamiento de Requerimientos por Etapas para el Desarrollo del Software

Debido a la extensión de este apartado, se plantean las siguientes etapas de implementación:

Primera Etapa

El módulo receptor (ARTIK™ 530) debe localizar el nodo, que se encuentra emitiendo señales, entre una distancia que oscile entre los 0.5 y 10 metros.

- **Adicionales Opcionales Primera Etapa:**
 - De acuerdo a la investigación realizada, la medición de la distancia no es una de las capacidades del presente dispositivo, se pretende localizar a los dispositivos mediante el nivel de potencia RSSI (Received Signal Strength Indication), de la misma forma que la distancia se planteará niveles de potencia para empezar a emitir el contenido multimedia
 - Se sugiere que la identificación de los dispositivos se realice por nombre para que, si una pieza museística se cambia de posición, el nodo permanezca en la misma ubicación y mediante el cambio del nombre, mediante una configuración sencilla, se asocie el nodo a otra pieza museística

Segunda Etapa (Opcional)

Se comprobará que el dispositivo escaneado se encuentre en el repositorio del módulo receptor, con el fin de evitar la emisión de contenido frente a la presencia de dispositivos bluetooth que no pertenecen al sistema.

Tercera Etapa

Finalmente se presentará una lista con el nombre del dispositivo e información asociada al mismo.

4.3 Montaje y Programación del Módulo Receptor (Dispositivo Móvil)

4.3.1 Componente de Hardware

ARTIK™ 530 es el módulo receptor (dispositivo móvil), que portarán los visitantes al museo y mediante el cual tendrán acceso a la información relacionada con las piezas museísticas. La información técnica relacionada con este dispositivo se trató en el numeral 2.9.1.

4.3.2 Programación del Componente

Como se ha manifestado previamente el módulo ARTIK™ 530, incorpora la funcionalidad de Bluetooth, a través de un conjunto de librerías asociadas al software instalado, Fedora® Linux®.

Inicialmente, se aborda la configuración y uso de las capacidades Bluetooth para posteriormente desarrollar un código funcional en lenguaje C.

Diseño de Solución, utilizando la Línea de Comandos de Bluetooth.

En la línea de comandos de Linux® se encuentra disponible la aplicación de 'bluetoothctl', que se utiliza para el escaneo de dispositivos y su emparejamiento. Para ingresar a la aplicación se digita el siguiente comando.

```
[root@artik ~]# bluetoothctl
```

Una vez en la aplicación se accede a una línea de comandos propia de Bluetooth se despliega un mensaje describiendo al controlador, antes de iniciar alguna acción se activa el módulo ARTIK™ 530 como agente Bluetooth y luego se solicita el acceso al agente con los siguientes comandos:

```
[bluetooth]# agent on
Agent registered

[bluetooth]# default-agent
Default agent request successful
```

Cuando se han ejecutado los comandos indicados previamente, se procede a escanear los dispositivos cercanos mediante el comando que se muestra a continuación:

```
[bluetooth]# scan on
```

El resultado que se despliega, con la ejecución del comando 'scan on' se muestra en la Figura 60

```
[bluetooth]# scan on
Discovery started
[CHG] Controller 70:2C:1F:23:F9:6E Discovering: yes
[NEW] Device 60:AF:6D:3D:95:52 Lola (GalaxyS4)
[NEW] Device F0:1F:E5:3C:3C:D1 MIO GLOBAL-FUSE
[NEW] Device 88:33:14:7C:E3:51 88-33-14-7C-E3-51
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -94
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -84
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -95
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -84
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -95
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -84
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -93
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -85
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -97
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -84
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -97
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -82
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -94
[CHG] Device 60:AF:6D:3D:95:52 RSSI: -95
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI: -85
[bluetooth]# scan off
[CHG] Device 88:33:14:7C:E3:51 RSSI is nil
[CHG] Device F0:1F:E5:3C:3C:D1 RSSI is nil
[CHG] Device 60:AF:6D:3D:95:52 RSSI is nil
Discovery stopped
[CHG] Controller 70:2C:1F:23:F9:6E Discovering: no
[DEL] Device 60:AF:6D:3D:95:52 Lola (GalaxyS4)
[DEL] Device F0:1F:E5:3C:3C:D1 MIO GLOBAL-FUSE
[DEL] Device 88:33:14:7C:E3:51 88-33-14-7C-E3-51
```

Figura 60. Dispositivos Escaneados, mediante 'scan on'

Como se observa el dispositivo inicia la búsqueda de dispositivos Bluetooth que puede detectar en sus cercanías. El indicador [NEW] que aparece al inicio de cada línea señala que se ha detectado un nuevo dispositivo.

Diseño de Solución, Implementación de un Programa en Lenguaje C con BlueZ

Previo al inicio del desarrollo de un programa en C es necesario instalar un conjunto de software en el sistema operativo. Mediante los comandos que se establecen subsecuentemente:

```
[root@artik ~]# dnf install bluez
[root@artik ~]# dnf install bluez-libs
[root@artik ~]# dnf install bluez-libs-devel
```

La instalación que se ha llevado a cabo, permite el uso el empleo de un poderoso stack de comunicaciones, denominada BlueZ, consta de extensas API que permiten el aprovechamiento de los recursos locales de Bluetooth; además proporciona soporte a las capas y protocolos básicos de Bluetooth, permite el procesamiento de datos multihilo,

brinda compatibilidad con varios dispositivos Bluetooth, proporciona una interfaz de socket estándar para todas las capas, es totalmente compatible con Linux® Fedora®. Será entonces, esta herramienta la que se utilizará en la presente implementación de un código en lenguaje C.

En la primera implementación, que se muestra en la Figura 61, se busca mostrar la dirección del dispositivo Bluetooth “BD_ADDR” conjuntamente con su nombre, que para el tratamiento en el código y en las librerías correspondientes se trata de ‘bdaddr’ y ‘name’ respectivamente.

```

/*Librería estándar que defina variables, macros para utilizar las funciones de entrada/salida
estándar
*/
#include <stdio.h>
//Librería que contiene funciones C para la gestión de memoria dinámica, control de procesos.
#include <stdlib.h>
// Librería que define diversas constantes y tipos simbólicos, y declara diversas funciones.
#include <unistd.h>
//Librería para la creación de sockets
#include <sys/socket.h>
//Librerías BlueZ para escanear, leer, escribir en dispositivos bluetooth
#include <bluetooth/bluetooth.h>
#include <bluetooth/hci.h>
#include <bluetooth/hci_lib.h>

/*La función main, se declara con los parámetros de entrada:
  argc (argument counter) y **argv (argument values)
*/
int main(int argc, char **argv)
{
    /*Declaración del puntero ii de tipo inquiry_info,
    | ii se ha inicializado en NULL
    */
    inquiry_info *ii = NULL;
    //Declaración de variables enteras
    int max_rsp, num_rsp;
    int dev_id, sock, len, flags;
    int i;
    /*Declaración de variables que almacenan una
    | cadena e inicialización de valores en 0
    */
    char addr[19] = { 0 };
    char name[248] = { 0 };

    /* Llamado a la función 'hci_get_route' a la que se le pasa un valor nulo, cuyo
    | resultado se le asigna a la variable 'dev_id'. Esta función retorna la dirección
    */
    dev_id = hci_get_route(NULL);

    /*Se abre un socket Bluetooth con el 'dev_id' obtenido. Esto representa una conexión
    | al microcontrolador en el adaptador Bluetooth local especificado.
    */
    sock = hci_open_dev( dev_id );
    /*Como hci.c devuelve, 'dev_id = -1' en caso de que no se haya obtenido la dirección y a
    | 'sock' se le asigna el valor de -1 en caso de que no se haya obtenido el
    | 'device descriptor' - 'dd'; se evalúa en función de estos resultados y se imprime una
    | cadena [la de 'error'] previa al error como tal; en este caso se devuelve el error de
    | finalización sin éxito
    */
    if (dev_id < 0 || sock < 0) {
        perror("opening socket");
        exit(1);
    }

    //Inicialización de variables
    len = 8;
    max_rsp = 255;

```

```

/* Se asigna a 'flags' el valor de 'IREQ_CACHE_FLUSH' establecido
como una constante en 0x0001 (dentro de hci.h)
*/
flags = IREQ_CACHE_FLUSH;
/* Se realiza el alojamiento en memoria luego de realizar
la conversión de tipo a 'inquiry_info*'
*/
ii = (inquiry_info*)malloc(max_rsp * sizeof(inquiry_info));

/*hci_inquiry realiza el descubrimiento de dispositivos bluetooth cercanos y devuelve una
lista de los dispositivos detectados y alguna información básica en la variable 'ii';
todo esto se asigna a la variable 'num_rsp'.
'hci_inquiry', requiere un número de recurso por lo que se usa 'dev_id' devuelto por
'hci_get_route'.
*/
num_rsp = hci_inquiry(dev_id, len, max_rsp, NULL, &ii, flags);
/*Si 'num_rsp' devuelve ret = -1 (según hci_inquiry en hci.c) entonces se
cumple esta condición y se pasa un error 'hci_inquiry'
*/
if( num_rsp < 0 ) perror("hci_inquiry");

// Se ejecuta un for para devolver todos los dispositivos encontrados
for (i = 0; i < num_rsp; i++) {
    // Las direcciones de bluetooth se almacenan y manipulan como estructuras bdaddr_t
    ba2str(&(ii+i)->bdaddr, addr);

    //llamada a la función 'memset', para inicializar el 'name' en 0
    memset(name, 0, sizeof(name));

/* Una vez encontrados los dispositivos cercanos y sus direcciones, el programa determina
los nombres asociados con esas direcciones y las presenta al usuario
*/
if (hci_read_remote_name((sock, &(ii+i)->bdaddr, sizeof(name), name, 0) < 0)
    strcpy(name, "[unknown]");
    printf("%s %s\n", addr, name, rssi);
}

//Se libera la memoria
free( ii );
//Se cierra el socket
close( sock );
return 0;
}

```

Figura 61. Código en C para Escanear Dispositivos Bluetooth Cercanos [21]

Si bien en código se encuentra comentado, es necesario incorporar información adicional:

- La función **'main'**, se declara con parámetros de entrada donde:
 - **argc (argument counter)**: es el número de parámetros ingresados, que se han especificado por línea de comandos incluido el nombre del ejecutable que es el primer parámetro
 - ****argv (argument values)**: es un array de cadenas que contiene los argumentos especificados por línea de comandos
- Se asigna a **'flags'** el valor de **'IREQ_CACHE_FLUSH'** establecido como una constante en 0x0001 (dentro de hci.h), con esta asignación la memoria caché de los dispositivos detectados se vacía antes de realizar la consulta actual, de lo contrario, si las banderas están establecidas en 0, los resultados de las consultas pueden ser devueltas incluso si los dispositivos no están en el rango.
- **'hci_inquiry'** realiza el descubrimiento de dispositivos bluetooth cercanos y devuelve una lista de los dispositivos detectados y alguna información básica en la variable 'ii'; todo esto se asigna a la variable 'num_rsp'. 'hci_inquiry',

requiere un número de recurso por lo que se usa 'dev_id' devuelto por 'hci_get_route'. La consulta dura $1.28 * \text{'len'}$ segundos, por lo que se establece 'max_rsp' de 255 para una consulta estándar de 10.24 segundos.

- Las direcciones de Bluetooth se almacenan y manipulan como estructuras `bdaddr_t`. `ba2str`, toma un paquete de 6 Bytes y lo transforma en una cadena `XX:XX:XX:XX:XX:XX` (Paso de estructuras por referencia)

Para realizar la compilación del código (guardado como `scan.c` en el módulo ARTIK™ 530) se invoca a gcc (GNU Compiler Collection) y el enlace con Bluetooth, como se indica a continuación:

```
[root@artik pruebas]# gcc -o scan scan.c -l bluetooth
```

Una vez que se ejecuta el programa devuelve la información de la dirección del dispositivo Bluetooth y su nombre asociado; dicho resultado se muestra en la Figura 62.

```
[root@artik pruebas]# ./scan  
C8:FF:28:BC:4A:9B  DESKTOP-KHOBT6
```

Figura 62. Dispositivo escaneado mediante, programa en C con BlueZ

Se desarrollaron varios códigos, a partir del código presentado previamente, con la finalidad de obtener el RSSI, debido a que en código anterior se utilizó la estructura 'inquiry_info' mediante la cual en el puntero 'ii' se alojó la información básica de los dispositivos, específicamente se guarda la información que se especifica en la Figura 63, correspondiente a la librería 'hci.h'; en función de lo cual se utilizó la estructura 'inquiry_info_with_rssi' que contiene la variable 'rssi'.

Una vez que se ha empleado la estructura, se declara una variable 'rssi' de tipo 'int8_t' para facilitar su obtención.

```
typedef struct {  
    bdaddr_t      bdaddr;  
    uint8_t      pscan_rep_mode;  
    uint8_t      pscan_period_mode;  
    uint8_t      pscan_mode;  
    uint8_t      dev_class[3];  
    uint16_t     clock_offset;  
} __attribute__((packed)) inquiry_info;
```

Figura 63. Estructura 'inquiry_info' de la librería 'hci.h' [22]

```

typedef struct {
    bdaddr_t      bdaddr;
    uint8_t       pscan_rep_mode;
    uint8_t       pscan_period_mode;
    uint8_t       dev_class[3];
    uint16_t      clock_offset;
    int8_t        rssi;
} __attribute__((packed)) inquiry_info_with_rssi;

```

Figura 64. Estructura 'inquiry_info_with_rssi' de la librería 'hci.h' [22]

El alojamiento en memoria, del puntero `ii`, para luego de realizar la conversión de tipo se lo realiza con la estructura 'inquiry_info_with_rssi*'.
 Para obtener, como tal el valor de 'rssi' anexa a lo mencionado la estructura condicional 'if' con la función 'hci_read_rssi', como se muestra en la Figura 65.

```

if (hci_read_rssi(sock, htob(ii->con_info->handle), &rssi, 10000)<0){
    strcpy(name, "[unknown]");
    printf("%s \n" rssi);
}

```

Figura 65. Estructura 'if' con la función 'hci_read_rssi', para obtener el 'rssi'

Una vez que se pasa el código al módulo ARTIK™ 530 se presentan varios errores que se muestran en la Figura 66.

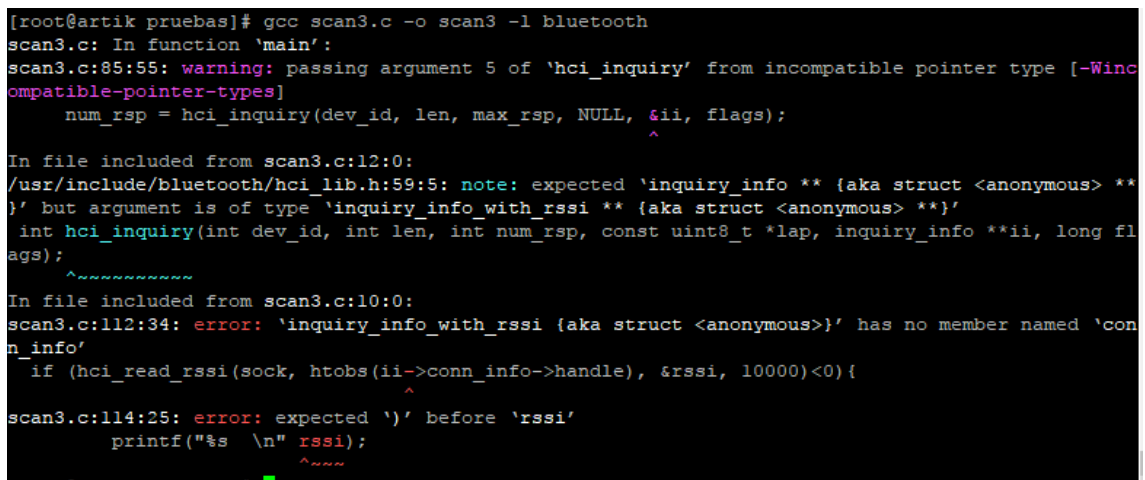


Figura 66. Errores desplegados al ejecutar el código modificado.

Se podría pensar que modificar el código inicial y obtener el 'rssi' de la misma forma que se lo hace con el 'bdaddr' es una solución viable, pero se debe considerar que 'inquiry_info' es un parámetro, representado por el puntero 'ii', que se pasa a la función 'hci_inquiry'; y en este caso ni la estructura que se ha presentado en la Figura 63 ni la función 'hci_inquiry' de la librería 'hci_lib.h' (ver Figura 67) contienen el parámetro 'rssi' para devolverlo. En función de lo cuál a pesar de los errores que se presenta se ha optado por la estructura 'inquiry_info_with_rssi' y la función 'hci_read_rssi' (ver Figura 68).

```
int hci_inquiry(int dev_id, int len, int num_rsp, const uint8_t *lap, inquiry_info **ii, long flags);
```

Figura 67. Función 'hci_inquiry' de la librería 'hci_lib.h'

```
int hci_read_rssi(int dd, uint16_t handle, int8_t *rssi, int to);
```

Figura 68. Función 'hci_read_rssi' de la librería 'hci_lib.h'

Diseño de Solución, Implementación de un Programa en Lenguaje C con Paquetes de Desarrollo SDK

En función de los desperfectos y la complejidad que representa el encontrar una solución próxima al desarrollo previamente planteado, se intenta de forma no exitosa, mediante los paquetes de desarrollo SDK implementar un código funcional que cumpla con el propósito planteado para el presente trabajo.

La actual propuesta, previo a su operación, requiere:

- Verificar si el dispositivo ARTIK™ 530 tiene el módulo SDK instalado, a través del comando que se despliega subsecuentemente:

```
[root@artik ~]# dnf info libartik-sdk-base
```

- Como la tarjeta ARTIK™ 530 presenta una configuración en el repositorio adecuado, se procede a actualizar el SDK a la versión más reciente, así también se actualizar el paquete 'libartik-sdk-systemio'

```
[root@artik ~]# dnf update libartik-sdk-base libartik-sdk-systemio
```

- Ahora bien, se instalarán algunos paquetes de desarrollo para que el SDK se pueda vincular al programa que se va a desarrollar, en este caso se instalan los siguientes paquetes:

```
[root@artik ~]# dnf install libartik-sdk-systemio-devel  
[root@artik ~]# dnf install libartik-sdk-bluetooth-devel
```

- Posteriormente, se deben instalar los archivos de encabezado que describen a la API, por ejemplo, para trabajar con Artik Cloud se debe instalar el paquete que se indica a continuación:

```
[root@artik ~]# dnf install libartik-sdk-connectivity-devel
```

El programa que se presenta se encuentra distribuido en tres directorios los cuales contienen los ficheros que se enumeran a continuación:

- **App:** main.c
- **Hardware:** hardLibrary.c, hardLibrary.h
- **Middleware:** lista_dispositivos_cercanos_Bluetooth.c,
lista_dispositivos_emparejados_Bluetooth.c,
lista_dispositivos_todos_Bluetooth.c, middleware.c, middleware.h.

Cuando se ha llevado a cabo la ejecución de los puntos anteriores, se incluye en el encabezado de algunos de los ficheros del programa las librerías SDK con sus respectivos subdirectorios, la Figura 69 muestra las librerías que se han incluido para: `hardLibrary.c`, `lista_dispositivos_cercanos_Bluetooth.c`, `lista_dispositivos_emparejados_Bluetooth.c`, `lista_dispositivos_todos_Bluetooth.c`.

```
#include </usr/include/artik/base/artik_module.h>
#include </usr/include/artik/base/artik_loop.h>
#include </usr/include/artik/bluetooth/artik_bluetooth.h>
```

Figura 69. Librerías SDK incluidas en el Programa con Paquetes SDK

En esta implementación se busca obtener una lista con el `BD_ADDR` y el nombre de los dispositivos Bluetooth: cercanos, emparejados y del conjunto de dispositivos (todos); esto mediante la implementación de las rutinas y librerías que se muestran en la: Figura 70, Figura 71, Figura 72, Figura 73, Figura 74, Figura 75, Figura 76 y Figura 77; el presente código fue desarrollado a partir de la implementación previa realizada con BlueZ [21].

```
// Librerías necesarias
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>

#include "../Middleware/middleware.h"

/*
-----
Main de la aplicacion, para probar el correcto funcionamiento
-----
*/

int main(){
    m_lista_dispositivos_todos_Bluetooth (stdout);
}
```

Figura 70. Código en C correspondiente a 'main.c'

```

.
.
#include "hardLibrary.h";

static artik_bluetooth_module *bt = NULL;
static int numD;
static artik_bt_device *devB = NULL;

/*
-----
Obtención de los dispositivos bluetooth ya conocidos, funciones que permiten realizarla.
-----
*/

artik_error lista_dispositivos_emparejados_Bluetooth_F(artik_bt_device *devices, int *num){
    artik_error ret = S_OK;
    int num1 = 0;
    int num2 = 0;

    bt = (artik_bluetooth_module *)artik_request_api_module("bluetooth");
    if (!bt) { return -1;}

    ret = bt->get_devices(&devices, &num1);
    if (ret != S_OK)
        goto exit;

    ret = bt->get_connected_devices(&devices+num1, &num2);
    if (ret != S_OK)
        goto exit;
    *num=num1+num2;

exit:
    if (devices && (num1 > 0 || num2 > 0)) {
        .
        .
    }
}

```

Figura 71. Fragmento de Código en C relacionado con 'hardLibrary.c'

```

.
.
#define HARDLIBRARY_H

artik_error lista_dispositivos_emparejados_Bluetooth_F(artik_bt_device*, int*);
static void scan_callback(artik_bt_event, void*, void*);
static void on_timeout_callback(void*);
.
.

```

Figura 72. Fragmento de Código en C correspondiente a 'hardLibrary.h'

```

.
.
#include "../Hardware/hardLibrary.h";

int main (int argc, char** argv) {
    int pipe = atoi(argv[1]);

    artik_bt_device *devices = NULL;
    int *num = 0;
    artik_error ret = S_OK;

    ret = lista_dispositivos_cercanos_Bluetooth_F (devices, num, 20000);

    fprintf(pipe, "{closeDevices:[");

    for (i = 0; i < *num; i++) {
        fprintf(pipe, "{deviceAddr:%s,\n",
            .
            .
        }
    }
}

```

Figura 73. Fragmento de Código en C relacionado con 'lista_dispositivos_cercanos_Bluetooth.c'

```

.
.
#include "../Hardware/hardLibrary.h";

int main (int argc, char** argv) {
    int pipe = atoi(argv[1]);

    artik_bt_device *devices = NULL;
    int *num = 0;
    artik_error ret = S_OK;

    ret = lista_dispositivos_emparejados_Bluetooth_F (devices, num);

    fprintf(pipe, "{closeDevices:[]");

    for (i = 0; i < *num; i++) {
        fprintf(pipe, "{deviceAddr:%s,\n",
            .
            .
    }
}

```

Figura 74. Fragmento de Código en C correspondiente a 'lista_dispositivos_emparejados_Bluetooth.c'

```

.
.
#include "../Hardware/hardLibrary.h";

int main (int argc, char** argv) {
    int pipe = atoi(argv[1]);

    artik_bt_device *devices = NULL;
    int *num = 0;
    artik_error ret = S_OK;

    ret = lista_dispositivos_emparejados_Bluetooth_F (devices, num);

    fprintf(pipe, "{closeDevices:[]");

    for (i = 0; i < *num; i++) {
        fprintf(pipe, "{deviceAddr:%s,\n",
            devices[i].remote_address ? devices[i].
            remote_address : "(null)");
        fprintf(pipe, "ID:%s),\n",
            devices[i].remote_name ? devices[i].
            remote_name : "(null)");
    }

    *devices = NULL;
    *num = 0;

    ret = lista_dispositivos_cercanos_Bluetooth_F (devices, num, 20000);

    for (i = 0; i < *num; i++) {
        fprintf(pipe, "{deviceAddr:%s,\n",
            .
            .
    }
}

```

Figura 75. Fragmento de Código en C relacionado con 'lista_dispositivos_todos_Bluetooth.c'

```

.
.
void m_lista_dispositivos_cercanos_Bluetooth (int pipe) {
    // Creación de un proceso aparte para evitar bloqueo
    int pid;
    pid = fork();
    switch(pid){
        case -1: // Si pid es -1 quiere decir que ha habido un error
            fprintf(pipe, "No se ha podido crear el proceso\n");
            break;
        case 0:
            // Cuando pid es cero quiere decir que es el proceso hijo
            // Llamada al conjunto de funciones
            char str[16];
            snprintf(str, 16, "%d", pipe);
            execlp("lista_dispositivos_cercanos_Bluetooth", , 0);
            fprintf(pipe, "No se ha podido realizar el exec\n");
            break;
        default: // Cuando es distinto de cero es el padre
            break;
    }
}

void m_lista_dispositivos_emparejados_Bluetooth (int pipe) {
    // Creación de un proceso aparte para evitar bloqueo
    int pid;
    pid = fork();
    switch(pid){
        case -1: // Si pid es -1 quiere decir que ha habido un error
            fprintf(pipe, "No se ha podido crear el proceso\n");
            break;
        case 0:
            // Cuando pid es cero quiere decir que es el proceso hijo
            // Llamada al conjunto de funciones
            char str[16];
            snprintf(str, 16, "%d", pipe);
            .
            .
            .
    }
}

```

Figura 76. Fragmento de Código en C correspondiente a 'middleware.c'

```

.
.
#define MIDDLEWAREY_H

void m_lista_dispositivos_cercanos_Bluetooth (int);
void m_lista_dispositivos_emparejados_Bluetooth (int);
.
.

```

Figura 77. Fragmento de Código en C relacionado con 'middleware.h.'

El uso de las librerías SDK en el programa, implica que se debe especificar las 'flags' de compilación apropiadas para que el compilador encuentre los archivos de cabeceras y librerías en las ubicaciones adecuadas. La compilación gcc se lleva a cabo de la siguiente forma:

```
[root@artik pruebas]# gcc main.c -o main -I/usr/include/artik/base -I/usr/include/artik/bluetooth -lartik-sdk-base
```

Se han añadido los siguientes parámetros, por las razones que se citan:

- **-I<path>**: Especifica la ruta en la que el compilador debe buscar los archivos de encabezado incluidos
- **lib<libname>**: Indica al compilador que se vincule con la librería compartida 'lib<libname>.so'

- **-l<path>**: especifica bajo que ruta el enlazador debe buscar las librerías compartidas a ser vinculadas.

Como se muestra en la Figura 78, la compilación no ha sido exitosa debido a un error asociado con el método 'm_lista_dispositivos_todos_Bluetooth' al que se le pasa un argumento del tipo 'struct _IO_FILE' (stdout), cuando se le debería pasar un entero.

```
[root@artik App]# gcc main.c -o main -I/usr/include/artik/base -I/usr/include/artik/bluetooth -lartik-sdk-base
main.c: In function 'main':
main.c:23:42: warning: passing argument 1 of 'm_lista_dispositivos_todos_Bluetooth' makes integer from pointer without a cast [-Wint-conversion]
   m_lista_dispositivos_todos_Bluetooth (stdout);
                                         ^~~~~~
In file included from main.c:14:0:
../Middleware/middleware.h:7:6: note: expected 'int' but argument is of type 'struct _IO_FILE **'
   void m_lista_dispositivos_todos_Bluetooth (int);
                                         ^~~~~~
/tmp/ccsFGRK5.o: In function 'main':
main.c:(.text+0x18): undefined reference to `m_lista_dispositivos_todos_Bluetooth'
collect2: error: ld returned 1 exit status
```

Figura 78. Compilación de 'main.c'

Para avanzar en el análisis del posible error se modificó el 'main.c', en lugar de pasar como parámetro 'stdout' se colocó un '0', siendo el resultado el que se distingue en la Figura 79.

```
[root@artik App]# gcc main.c -o main -I/usr/include/artik/base -I/usr/include/artik/bluetooth -lartik-sdk-base
/tmp/cc2s35Fr.o: In function 'main':
main.c:(.text+0xc): undefined reference to `m_lista_dispositivos_todos_Bluetooth'
collect2: error: ld returned 1 exit status
```

Figura 79. Compilación de 'main.c', modificado con parámetro 0

Inclusive, se volvió a modificar el 'main.c' para probar con '1', pero como se puede ver en la Figura 80 se mantiene el error anteriormente desplegado.

```
[root@artik App]# vi main.c
[root@artik App]# gcc main.c -o main -I/usr/include/artik/base -I/usr/include/artik/bluetooth -lartik-sdk-base
/tmp/ccMkGJ38.o: In function 'main':
main.c:(.text+0xc): undefined reference to `m_lista_dispositivos_todos_Bluetooth'
collect2: error: ld returned 1 exit status
```

Figura 80. Compilación de 'main.c', modificado con parámetro 1

Del mismo modo se probó modificando el 'middleware.h' y el 'middleware.c', como se visualiza en la Figura 81 y Figura 82 correspondientemente, comentando los parámetros de entrada; pero como se observa en la Figura 83 el error persiste.

```
void m_lista_dispositivos_cercanos_Bluetooth (int);
void m_lista_dispositivos_emparejados_Bluetooth (int);
void m_lista_dispositivos_todos_Bluetooth (/*int*/);

#endif /* MIDDLEWARE_H */
```

Figura 81. Modificación de 'middleware.h'

```
void m_lista_dispositivos_todos_Bluetooth (/*int pipe*/) {
    // Creación de un proceso aparte para evitar bloqueo
    int pid;
    pid = fork();
    switch(pid) {
```

Figura 82. Modificación de 'middleware.c'

```
[root@artik App]# gcc main.c -o main -I/usr/include/artik/base -I/usr/include/artik/bluetooth -Iartik-sdk-base
/tmp/ccdjLSUn.o: In function `main':
main.c:(.text+0x8): undefined reference to `m_lista_dispositivos_todos_Bluetooth'
collect2: error: ld returned 1 exit status
```

Figura 83. Compilación de 'main.c', cambiando 'middleware.h' y 'middleware.c'

Se probó también compilar utilizando el programa 'pkg-config', para recuperar automáticamente las 'flags' de compilación desde el sistema implementado; por lo que la compilación se ejecuta según se establece inmediatamente:

```
[root@artik pruebas]# gcc main.c -o main `pkg-config --libs -cflags libartik-sdk-base libartik-sdk-bluetooth`
```

El resultado de esta última forma de compilar se muestra en la Figura 84; persistiendo el error previamente desplegado (Ver Figura 84).

```
[root@artik App]# gcc main.c -o main `pkg-config --libs -cflags libartik-sdk-base libartik-sdk-bluetooth`
Unknown option -cflags
/tmp/ccGCrEYD.o: In function `main':
main.c:(.text+0x8): undefined reference to `m_lista_dispositivos_todos_Bluetooth'
collect2: error: ld returned 1 exit status
```

Figura 84. Compilación de 'main.c' utilizando 'pkg-config'

Se probó compilar las clases de otros directorios, pero aquello tampoco solucionó el problema.

Como se visualiza en la Figura 76 correspondiente al fichero 'middleware.c' se han utilizado procesos pesados como 'fork()', el cual es una función que crea un "proceso hijo" exactamente igual al "proceso padre"; si 'fork()' se ejecuta exitosamente devuelve: al padre el 'pid' del proceso creado y al hijo el valor de '0', para este último caso se llama al conjunto de funciones.

Además de los inconvenientes que se han presentado, se ha evidenciado que los procesos pesados introducen retardos de 3 a 4 segundos, por lo que se descarta esta opción y se opta entonces por un script en la Bash, el cual tiene una respuesta más inmediata.

Diseño de Solución, Implementación de un Script de Bash

Mediante esta alternativa se pretende conseguir los resultados adecuados, para lo cual se utiliza el comando asociado a la Bash 'hcitool', que se emplea para configurar conexiones Bluetooth o transmitir algún comando especial a los dispositivos que cuentan

con esta tecnología. Para obtener el BD_ADDR y el nombre se ejecuta el comando 'hcitool' con la opción 'scan', como se muestra en la Figura 85.

```
[root@artik ~]# hcitool scan
Scanning ...
    20:16:03:22:61:68      54A54F
    20:16:03:22:39:63      54A54C
    20:16:03:22:43:57      54A54A
    20:16:03:22:17:52      54A54B
```

Figura 85. Ejecución y resultados, comando 'hcitool scan'

El script obtenido empleando la herramienta antes citada, se despliega en la Figura 86.

```
#!/bin/bash
# scan5.sh: Escaneo de Dispositivos Bluetooth Cercanos
echo 'Iniciando el escaneo BLUETOOTH'
# Se reinicia el servicio Bluetooth
service bluetooth restart
# Se reinicia el mecanismo de comunicación entre procesos
service dbus restart
while true
do
hcitool scan
done
```

Figura 86. Script de Bash

El script incluye el 'while true' para que se ejecute continuamente espacio establecido en el script.

Con el script completo se procede a cambiar los permisos de acceso al fichero 'scan5.sh' para que se pueda ejecutar, con lo cual se digita el siguiente comando:

```
[root@artik pruebas]# chmod +x scan5.sh
```

Una vez llevado a cabo los pasos anteriores se procede con la ejecución del script y se visualizan los resultados que se muestran en la Figura 87.

4.4 Montaje y Programación de los Nodos (Dispositivos Móviles)

A cada pieza museística se le incluirá un nodo (dispositivo fijo) que será reconocido por el módulo receptor.

```
[root@artik pruebas]# ./scan5.sh
Redirecting to /bin/systemctl restart bluetooth.service
Redirecting to /bin/systemctl restart dbus.service
PolicyKit daemon disconnected from the bus.
We are no longer a registered authentication agent.
Scanning ...
AC:C1:EE:0F:2C:68 n/a
Scanning ...
AC:C1:EE:0F:2C:68 n/a
Scanning ...
AC:C1:EE:0F:2C:68 n/a
Scanning ...
AC:C1:EE:0F:2C:68 n/a
Scanning ...
AC:C1:EE:0F:2C:68 n/a
Scanning ...
AC:C1:EE:0F:2C:68 n/a
Scanning ...
20:16:03:22:17:52 54A54B
Scanning ...
20:16:03:22:17:52 54A54B
Scanning ...
20:16:03:22:17:52 54A54B
Scanning ...
20:16:03:22:17:52 54A54B
Scanning ...
20:16:03:22:17:52 54A54B
Scanning ...
20:16:03:22:17:52 54A54B
Scanning ...
20:16:03:22:17:52 54A54B
Scanning ...
```

Figura 87. Ejecución y resultados, script 'scan5.sh'

4.4.1 Componentes de Hardware

La Figura 88 muestra los componentes principales de un módulo fijo o nodo y subsecuentemente se describe la funcionalidad de cada elemento.

- **Módulo Bluetooth HC-06:** Emite la señal (continua) de radio desde el nodo para que pueda ser detectado por el módulo receptor. Posee un Firmware que permite la introducción de una cantidad limitada de comandos.
- **Placa Arduino Nano:** Gestiona la comunicación entre el Módulo HC-06 y los demás elementos del nodo. Algunas características como su estructura y distribución de pines se desarrollaron el numeral 2.9.2.
- **Acelerómetro MMA8451 (Arduino):** Detecta movimientos y desplazamientos en las piezas museísticas, con el fin de que éstas se mantengan en su sitio. Son capaces de detectar fuerzas estáticas (como la gravedad) o dinámicas (vibraciones y movimiento) de aceleración.

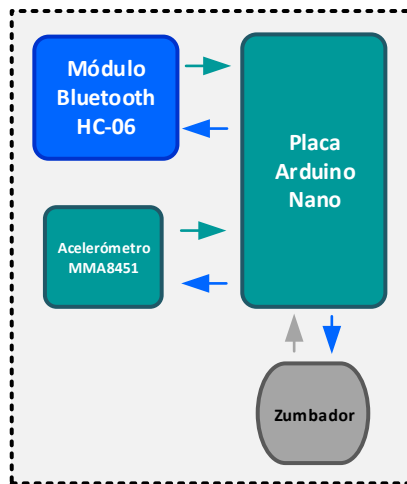


Figura 88. Componentes de un Nodo

- **Zumbador:** Su función es producir un sonido electroacústico continuo, alertará al visitante en caso de haber superado la distancia mínima (0.5 m) entre el módulo receptor y el nodo. Está compuesto de un electroimán y una lámina metálica de acero; se puede conectar a diferentes circuitos y con lo que se obtiene distintos tonos.

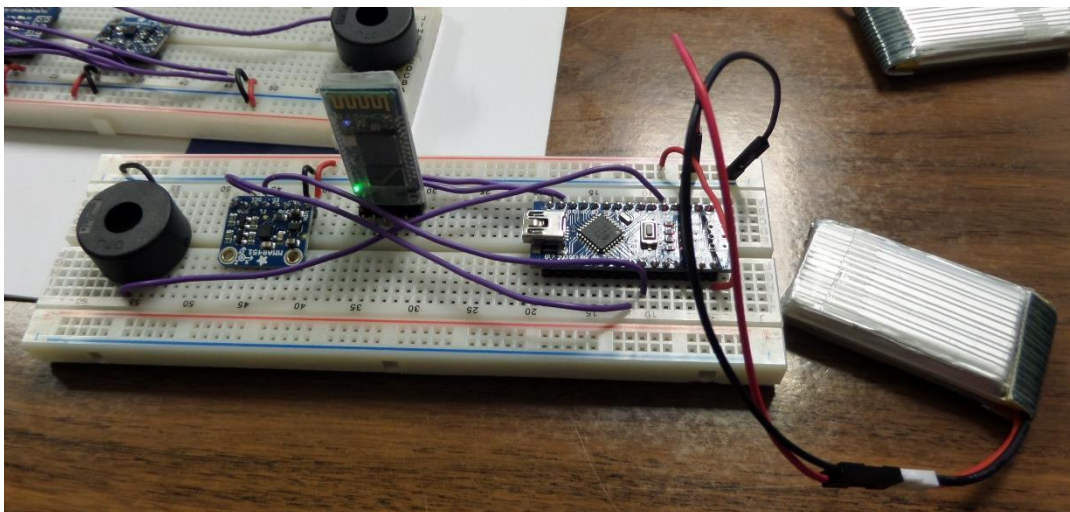


Figura 89. Montaje Nodo

4.4.2 Programación de Componentes

La implementación de código para el funcionamiento en conjunto de los componentes del nodo, se lo ha focalizado en el módulo Arduino Nano. El lenguaje empleado para la programación de Arduino está basado en C++, que es una adaptación de avr-libc quien proporciona una librería de C de alta calidad para utilizar con GCC.

Módulo Bluetooth HC-06

La configuración de este módulo se la realiza mediante la placa Arduino Nano, el código que se muestra en la Figura 90 permite a Arduino Nano simular una comunicación serial, para establecer los siguientes parámetros en el módulo Bluetooth HC-06:

- Nombre asociado al dispositivo bluetooth, mediante la variable 'ssid'.
- La tasa en baudios para llevar a cabo la comunicación entre Arduino Nano y el módulo HC-06, a través de la variable 'baudios'.
- Para realizar el emparejamiento del dispositivo se establece además una contraseña, para el propósito se plantea la variable 'password'.

```

// Librería para simular un puerto UART en el Arduino
#include <SoftwareSerial.h>
#define LONG_READ 20

// Pines donde conectaremos el modulo HC-06
SoftwareSerial mySerial(10, 11); // RX, TX
unsigned char cont;

// Opciones de configuración:
char ssid[10]   = "54A54A"; // Nombre para el modulo Bluetooth.
char baudios   = '5';      /* 1=1200, 2=2400, 3=4800, 4=9600, 5=19200, 6=38400,
                          7=57600, 8=115200, 9=230400, A=460800, B=921600, C=1382400*/
char password[5] = "abcd"; // Contraseña para el emparejamiento del modulo.

char id_buffer [LONG_READ];

void leeRespuesta () {
  cont = 0;
  while (mySerial.available())
  {
    delay(1);
    id_buffer[cont] = mySerial.read();
    cont++;
  }
}

//Código inicial que se ejecuta al pulsar RESET
void setup() {
  Serial.begin(19200);
  cont = 0;
  // set the data rate for the SoftwareSerial port
  mySerial.begin(38400);
  Serial.println("Sistema inicializado");

  // Tiempo de espera:
  pinMode(13,OUTPUT);
  digitalWrite(13,HIGH);
  delay(10000);
  digitalWrite(13,LOW);

  // Se inicia la configuración:
  mySerial.print("AT");
  leeRespuesta();
  delay(1000);
  Serial.println("AT");
  Serial.println(cont);
  Serial.println(id_buffer);

  // Se ajusta el nombre del Bluetooth:
  mySerial.print("AT+NAME"); mySerial.print(ssid);

```

Figura 90. Fragmento de Código, configuración Módulo Bluetooth HC-06 con Arduino Nano

Algo importante que se debe considerar es que cuando el módulo ARTIK™ 530 identifique al módulo Bluetooth HC-06, el parpadeo del LED (del módulo HC-06, ver Figura 89) se mantiene, esto debido a que no se ejecuta una vinculación o conexión entre dichos componentes.

Placa Arduino Nano

La Figura 91 despliega el desarrollo establecido para que la placa Arduino Nano, obtenga los datos correspondientes del acelerómetro, para conocer si la pieza museística se mantiene o no en su sitio.

```
#include <Wire.h>
#include <Adafruit_MMA8451.h>
#include <Adafruit_Sensor.h>

Adafruit_MMA8451 mma = Adafruit_MMA8451();

void setup(void) {
  Serial.begin(9600);

  Serial.println("Adafruit MMA8451 test!");

  if (! mma.begin()) {
    Serial.println("Couldnt start");
    while (1);
  }
  Serial.println("MMA8451 found!");

  mma.setRange(MMA8451_RANGE_2_G);

  Serial.print("Range = "); Serial.print(2 << mma.getRange());
  Serial.println("G");
}

void loop() {
  // Read the 'raw' data in 14-bit counts
  mma.read();
  Serial.print("X:\t"); Serial.print(mma.x);
  Serial.print("\tY:\t"); Serial.print(mma.y);
  Serial.print("\tZ:\t"); Serial.print(mma.z);
  Serial.println();

  /* Get a new sensor event */
  sensors_event_t event;
  mma.getEvent(&event);

  /* Display the results (acceleration is measured in m/s^2) */
  Serial.print("X: \t"); Serial.print(event.acceleration.x); Serial.print("\t");
  Serial.print("Y: \t"); Serial.print(event.acceleration.y); Serial.print("\t");
  Serial.print("Z: \t"); Serial.print(event.acceleration.z); Serial.print("\t");
  Serial.println("m/s^2 ");

  /* Get the orientation of the sensor */
  uint8_t o = mma.getOrientation();

  switch (o) {
    case MMA8451_PL_PUF:
      Serial.println("Portrait Up Front");
      break;
    case MMA8451_PL_PUB:
      Serial.println("Portrait Up Back");

```

Figura 91. Fragmento de Código, configuración Acelerómetro mediante Arduino Nano

Mediante el código que se despliega en la Figura 92 se controla el sonido emitido por el zumbador, ante la presencia de un módulo receptor.

```

// Librería para simular un puerto UART en el Arduino
#include <SoftwareSerial.h>

#define LONG_BUFFER 15
#define PIN_BZZ 2

// Estructuras de datos necesarias
SoftwareSerial mySerialBT(11,12); // RX, TX
char msg_buffer [LONG_BUFFER];

void setup() {
  mySerialBT.begin(9600);
  pinMode(PIN_BZZ, OUTPUT);
  digitalWrite(PIN_BZZ, LOW);
}

void loop() {
  if (mySerialBT.available()) {
    for (unsigned char j=0;j<LONG_BUFFER;j++)
    {
      delay(1);
      msg_buffer[j] = mySerialBT.read();
    }
    switch (msg_buffer [9]) {
      case 'Y':
        digitalWrite(PIN_BZZ, HIGH);
    }
  }
}

```

Figura 92. Fragmento de Código, configuración Zumbador con Arduino Nano

4.5 Revisión Tecnológica con RFID

Como se conoce y de acuerdo a los estándares abordados en el numeral 2.8, en lo que concierne a esta alternativa, su topología consiste básicamente en un lector que envía comandos a las etiquetas y en función de aquello recibe respuestas, es importante mencionar que la configuración de las etiquetas se limita un conjunto reducido de funciones. Es así que, esta revisión tecnológica se centrará básicamente en la comunicación entre el lector Serial L-RX202 y el módulo ARTIK™ 530. Debido a la similitud entre tecnologías Bluetooth y RFID (ver Figura 57 y Figura 58) y por tratarse de una primera revisión tecnológica se mantendrán los datos referentes a los requerimientos establecidos en el numeral 4.1.

4.5.1 Interconexión del Hardware

Interfaz UART ARTIK™ 530 con Conector RS-232 del Lector Serial L-RX202

En el caso del ARTIK™ 530 posee la interfaz UART de conexión externa [23], que se ubica en:

- **Nombre del Dispositivo Linux:** /dev/ttyAMA4
- **Conexión Externa:** En los pines 33, 35 de Test Header o Tx-Rx en la Tarjeta interfaz.

Los pines de conexión RS-232 para el Lector Serial L-RX202 [24] son:

- **Conexión Externa:** Puerto Izquierdo, Pin 1 y 2 Rx y Tx correspondientemente.

En el lado del lector Serial L-RX202 la interconexión de pines se la realiza mediante el Patch Cord de 8 pines incorporado y el adaptador RJ-45 a serial RS-232 con conector a fuente de alimentación, como se muestra en la Figura 93. En el lado del ARTIK™ 530 y ya con el DB9 a disposición se interconecta los pines de Tx y Rx como se muestra en la Figura 94.

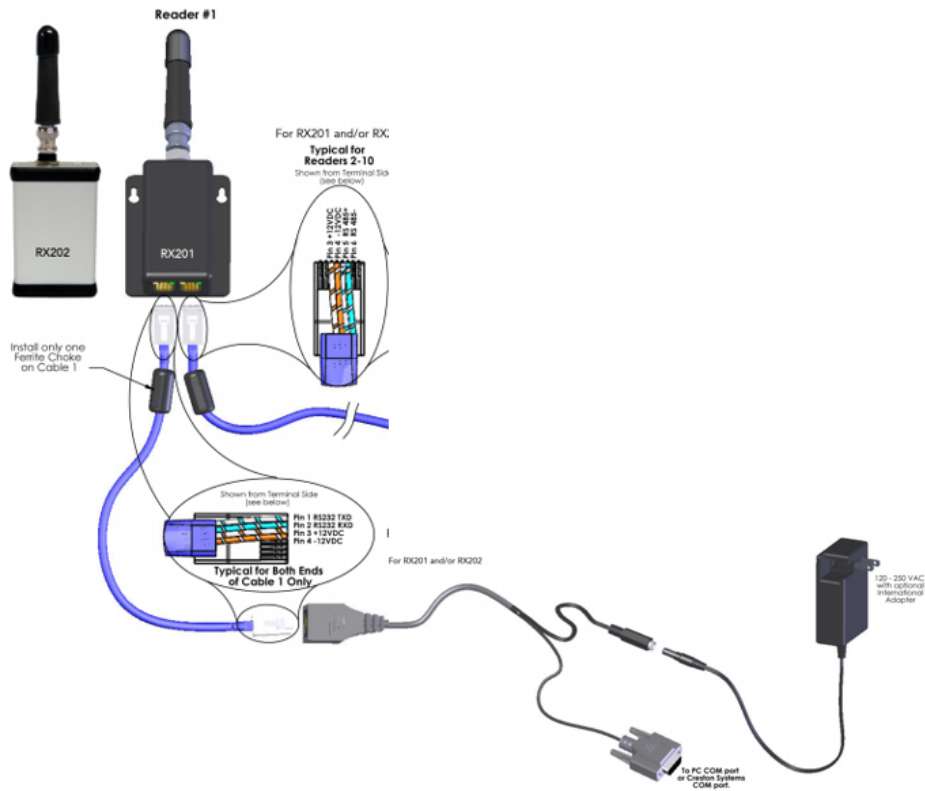


Figura 93. Cableado para la Conexión del Lector Serial L-RX202 [25]

Lector Serial L-RX202
DB9 Hembra

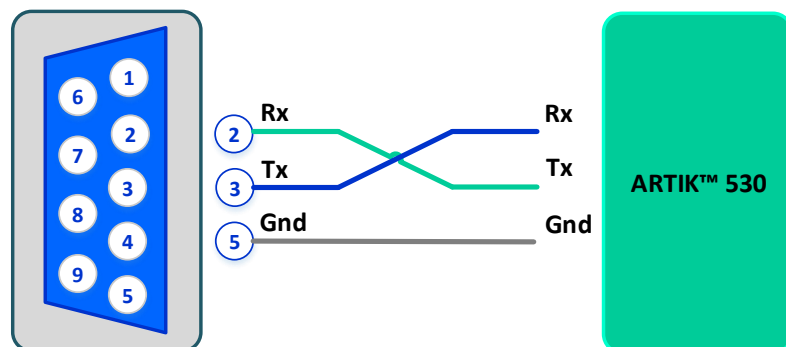


Figura 94. Interconexión entre DB9 del Lector Serial L-RX202 y ARTIK™ 530

4.5.2 Configuración para la Interconexión

Para llevar a cabo la lectura de datos por parte de ARTIK™ 530 de los datos provenientes del Lector Serial L-RX202 se plantean las siguientes opciones:

Mediante la Línea de Comandos [23]

La línea de comandos de Linux® Fedora® con la que cuenta el dispositivo permite, mediante la ejecución de los siguientes comandos, leer y configurar los datos que llegan al puerto serial:

```
[root@artik ~]# stty -F /dev/ttyAMA4
speed 115200 baud; line = 0;
.
.
```

Para configurar la tasa de baudios y confirmar la configuración se ejecuta los comandos que se muestran a continuación:

```
[root@artik ~]# stty -F /dev/ttyAMA4 9600
[root@artik ~]# stty -F /dev/ttyAMA4
speed 9600 baud; line = 0;
```

En cuanto se ha configurado el puerto, la interfaz permitirá realizar procedimientos de lectura/escritura. Mediante la conexión ya establecida se puede ejecutar el siguiente comando para leer los datos que provienen del Lector Serial L-RX202:

```
[root@artik ~]# cat /dev/ttySAC3
```

Sobre la Línea de comandos de ARTIK™ 530 se deben visualizar los datos provenientes del Lector y con el tratamiento adecuado de los mismos, se los interpretará adecuadamente para procesarlos.

Utilizando un Programa en Lenguaje C

Por tratarse de una revisión tecnológica este punto únicamente presenta un fragmento de código (Ver Figura 95) estructurado para la lectura de los pines de Tx y Rx del mismo dispositivo, con el objetivo de evidenciar que el lenguaje C se puede utilizar para lectura datos mediante comunicación serial.

```

static artik_serial_handle handle = NULL;
static const char* current_test = NULL;
#define MAX_RX_BUF 64
static void signal_handler(int signum)
{
    if (signum == SIGALRM) {
        fprintf(stderr, "TEST: %s failed, timeout expired\n", current_test);
        exit(-1);
    }
}
static void set_timeout(const char *test_name, unsigned int seconds)
{
    current_test = test_name;
    signal(SIGALRM, signal_handler);
    alarm(seconds);
}
static void unset_timeout()
{
    alarm(0);
    signal(SIGALRM, SIG_DFL);
}
static void forward_data(void *param, unsigned char *buf, int len)
{
    if (buf != NULL)
        fprintf(stdout, "Forward read: %s\n", buf);
    else {
        artik_serial_module *serial = (artik_serial_module *)artik_request_api_module("serial");
        artik_loop_module *loop = (artik_loop_module *)artik_request_api_module("loop");
        serial->unset_received_callback(handle);
        loop->quit();
        artik_release_api_module(serial);
        artik_release_api_module(loop);
    }
}
static artik_error test_serial_loopback(int platid)

```

Figura 95. Fragmento de Código en C, comunicación Serial Loopback [26]

5 Experimentación y Resultados

En este capítulo se detalla el conjunto de pruebas realizadas con los módulos IoT operando en un ambiente real. Para cumplir con este propósito se realizó una implementación tipo prototipo según lo que se desglosa en los siguientes numerales.

5.1 Despliegue y Experimentación

5.1.1 Instalación del Prototipo

De conformidad con lo establecido en el numeral 4.5, correspondiente al montaje y programación de los nodos, se procedió a instalar a los mismos (acoplados en una placa) en objetos que se asemejen a las piezas que conforman el ambiente de museística, un ejemplo de aquello es el que se observa en la Figura 96.

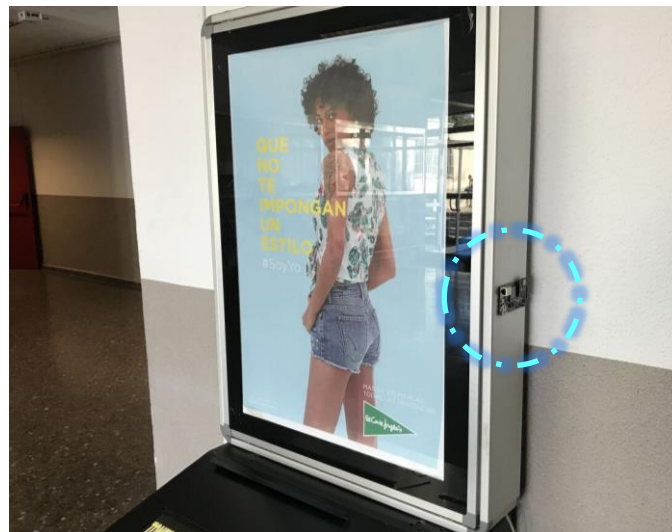


Figura 96. Nodo Instalado en un Objeto Real.

En cuanto al módulo receptor, como se ha reiterado en varias ocasiones, este lo portará el visitante, por lo que no se hace ejecución una instalación como tal.

5.1.2 Escenario de Pruebas

El despliegue de pruebas se lo realizó en las instalaciones del edificio Bloque B de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid, específicamente en los pasillos de dichas instalaciones, algunos de los puntos en los que se desarrollaron las pruebas se muestran en la Figura 97.

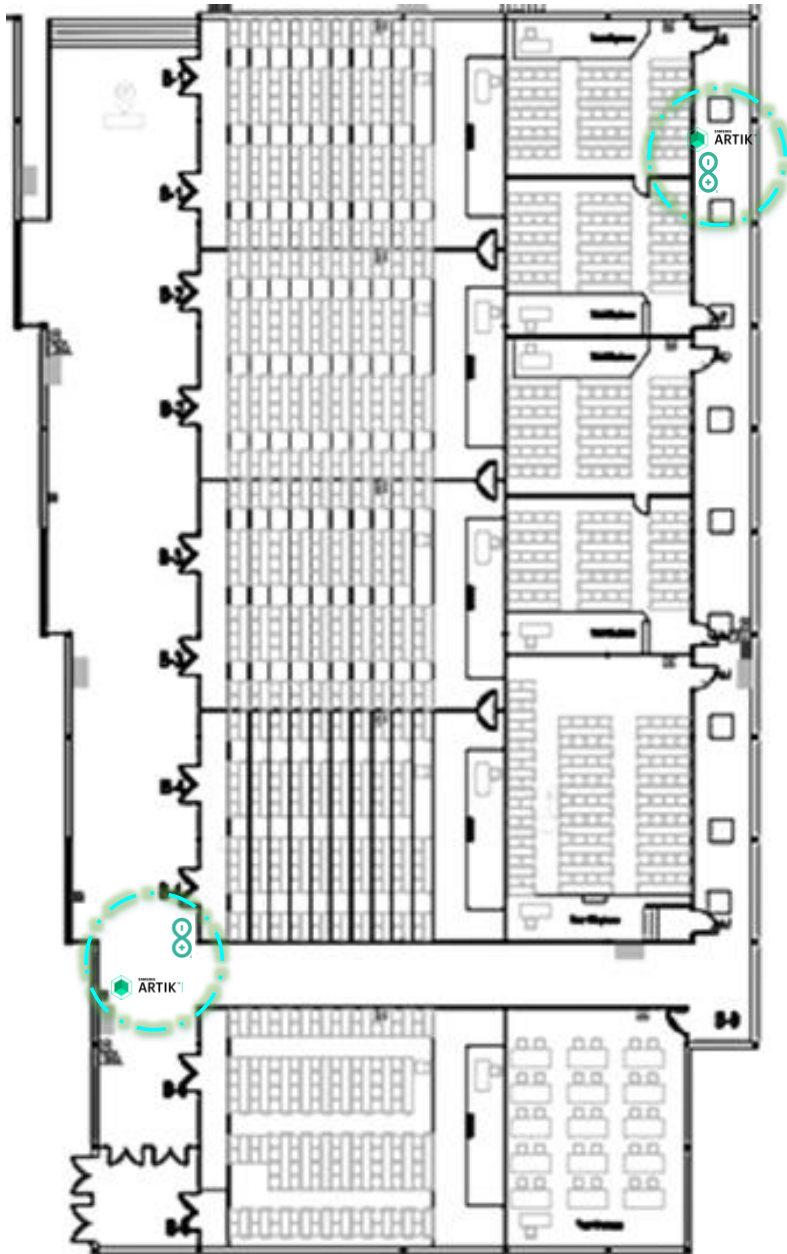


Figura 97. Escenario de Pruebas, ETSI Telecomunicación - Bloque B

5.1.3 Pruebas de Campo

Ya con los diferentes módulos implementados y desplegados sobre el escenario seleccionado, se procedió a verificar la interacción entre los mismos (en el siguiente numeral se revisarán los resultados correspondientes), una muestra de estas pruebas se visualiza en la Figura 98 y Figura 99



Figura 98. Prueba de Campo 1



Figura 99. Prueba de Campo 2

5.2 Resultados Obtenidos

Una vez que se llevó a cabo la experimentación con el módulo receptor y los nodos en el del edificio Bloque B de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid, se puede determinar las siguientes aseveraciones:

- El sistema a través de los módulos implementados se ajusta adecuadamente a un ambiente real, ya que a pesar de la presencia de agentes que inciden sobre el funcionamiento de los módulos; como los elementos que conforman a los objetos donde se instala los nodos, la presencia señales de diferente naturaleza entre otros aspectos, el sistema se desenvuelve con total normalidad al igual que en el ambiente de laboratorio.
- La tecnología que se ha escogido para la implementación del sistema se ajusta adecuadamente al entorno de pruebas, esto debido a que:
 - La topología y la correspondiente distribución de los equipos, no evidencia inconvenientes que sugieran un cambio de la misma
 - Se verificó que la tasa de transferencia efectiva calculada es adecuada para el escaneo de los nodos por parte del módulo receptor, ya que no se presenta saturación en el canal de comunicaciones.
- En el contexto, de la distancia entre dispositivos planteada, de 0.5 a 10 metros, el escaneo se llevó a cabo sin contratiempos, ya que se detectó correctamente los nodos pertenecientes al sistema. De igual forma se verificó la emisión de sonidos por parte del zumbador frente a una proximidad excesiva del módulo receptor.
- En cuanto al uso de batería, los nodos mantuvieron un nivel de consumo adecuado pero el módulo ARTIK™ 530 parecía tener un mayor consumo ya que presentaba un ligero calentamiento en la batería, sería entonces este un punto a tomar en cuenta.
- Se confirma que la elección de la forma de programación establecida para los equipos es la mejor ya que no existen retardos como cuando se utilizó procesos pesados, los que introducían retardos de entre 3 y 4 segundos.

6 Conclusiones y Líneas Futuras de Investigación

En el presente capítulo se sintetiza los puntos más destacados del estudio, diseño e implementación del presente trabajo, por lo que se desglosan las conclusiones resultantes de todo el proceso realizado. A lo mencionado, se añade un conjunto de líneas futuras que permitan optimizar la presente implementación e incentivar el desarrollo de nuevas aplicaciones con las tecnologías que han sido sujeto de análisis.

6.1 Conclusiones

La realidad aumentada añade información de origen virtual, a la información física reciba por los usuarios de forma natural a través de un conjunto de dispositivos, para los cuales la movilidad es una de sus características principales. Es así que las tecnologías IoT constituyen una solución de implementación adecuada para este tipo de aplicaciones, ya que debido a sus características no solo permiten desplegar funcionalidades complejas, sino que también son tecnologías accesibles, portables y móviles; en función de lo manifestado se ha pensado en la fusión de estas tendencias tecnológicas para plantear un desarrollo aplicado a ambientes de museística.

Para presente trabajo se realiza un estudio adecuado de las tecnologías inalámbricas: IEEE 802.11, Bluetooth, ZigBee y RFID. El estudio involucra fundamentos, componentes, topologías, protocolos de operación, entre otros aspectos; con la información recopilada se adquieren las bases para conocer, entre otras funcionalidades, la interacción de componentes, la dinámica de intercambio de mensajes; lo cual constituye una base que permite a posterior desplegar un diseño adecuado del sistema que involucre tanto a compenentres de hardware y como de software.

El avance y accesibilidad que se tiene actualmente a una diversidad de dispositivos IoT ha permitido indagar sobre objetos que anteriormente las tecnologías de red no tenían acceso; facilitando así la generación y el acceso a los datos de dichos objetos que aparentemente sin la intervención humana o de tecnologías previas no se podía obtener información.

Las tecnologías inalámbricas como: IEEE 802.11, Bluetooth, ZigBee y RFID, se pueden aplicar a ambientes de PAN, debido a la mejora constante que se realiza sobra cada tecnología y al constante despliegue de aplicaciones en distintos ámbitos.

La exposición de una comparativa conforme a elementos clave de evaluación tecnológica que no deja de lado las características del entorno museal para el montaje del proyecto, ha permitido en el presente trabajo optimizar y focalizar la implementación

mediante la tecnología Bluetooth; es así que se han evidenciado basados en parámetros técnicos, las potencialidades de la tecnología escogida frente a otras.

El módulo receptor, ARTIK™ 530 referente al trabajo presentado, cuenta con ventajas como las capacidades estandarizadas de hardware y software IEEE 802.11, Bluetooth y ZigBee, a esto se suma las funcionalidades añadidas de tarjeta interface que permite entre muchas otras cosas, la comunicación serial a través la interfaz UART para así conectarse con otras tecnologías como RFID. Cuenta también con las herramientas multimedia suficientes, asociadas al sistema operativo Fedora®, para el tratamiento de audio y video.

La funcionalidad de los nodos se ha ampliado, de la emisión de señales para su detección por parte del módulo receptor, a capacidades como la detección de movimientos y desplazamientos de la pieza museística, función establecida por el acelerómetro en conjunto con la placa Arduino Nano; y la emisión de sonidos por parte del zumbador frente a una proximidad excesiva por parte del módulo receptor.

El uso de herramientas de código libre, que pueden ser modificadas de acuerdo a las necesidades del proyecto, ha permitido el desarrollo de la programación de los componentes del sistema. Sobre el módulo receptor se instaló y actualizó varias herramientas para el despliegue del código, tales como BlueZ, con lo cual se posibilitó el uso y aprovechamiento de los recursos Bluetooth a través del uso de estructuras como 'inquiry_info' de la librería 'hci.h' con la que, mediante herramientas de lenguaje C, se obtuvo los parámetros de 'bdaddr' y 'name' (nombre y dirección del dispositivo respectivamente), estos mismos resultados se los consiguió a través de un script de la Bash que utiliza principalmente el comando 'hcitool'. La programación de los nodos se centró en el módulo Arduino Nano, mediante el lenguaje de programación C++ que es una adaptación de avr-libc para su uso con GCC; la placa Arduino Nano, mediante el código adecuado, configura el módulo Bluetooth HC-06, controla el sonido del zumbador y registra la ubicación original de la pieza museística a través del acelerómetro.

Emplear paquetes de desarrollo SDK, implica el uso de sus correspondientes librerías, por lo que su ubicación debe estar disponible para el compilador, es así que será importante especificar correctamente las 'flags' de compilación. Así también el uso de paquetes SDK, ha evidenciado que los procesos pesados introducen retardos de 3 a 4 segundos, por lo que se descarta esta opción y se opta entonces por el script en la Bash, el cual tiene una respuesta más inmediata.

La incorporación de las nuevas tecnologías a casi todos los ámbitos es un hecho inevitable, son entonces los dispositivos IoT una alternativa eficiente que permite a

ámbitos tradicionales como la museística involucrarse con las nuevas tendencias tecnológicas, de esta forma se brinda a estas instituciones una nueva y atractiva forma de promoción cultural haciendo más atractiva la experiencia museística.

Una de las fortalezas del sistema es su dinámica de comunicación, pues no existe una transferencia de datos constante entre nodos y módulos receptores, el escaneo e identificación de dispositivos es lo único que se necesita para desencadenar las tareas multimedia.

6.2 Líneas Futuras de Investigación

El despliegue y funcionamiento del sistema, depende directamente del consumo de potencia y la autonomía de alimentación energética para que se garantice un funcionamiento adecuado del conjunto de dispositivos, por ende, sería importante complementar la investigación realizada referente a esta característica que rige el funcionamiento de los dispositivos. En lo que respecta a los nodos se sugiere buscar una opción para que ingresen al modo de bajo consumo de energía y en lo que corresponde al módulo receptor, buscar alternativas que permitan tener un menor consumo de potencia con la alimentación mediante batería; con estas medidas se obtendría un mejor desempeño de los módulos receptores en el ambiente museal y un mayor tiempo de vida útil tanto de las baterías como de los dispositivos en general.

Para superar los inconvenientes derivados del uso de las herramientas de lenguaje C utilizadas, se debe depurar la comunicación entre procesos; empezar identificando el 'pid' del proceso padre y el conjunto de funciones derivadas a las que se llama permita obtener mayor información del error que se está presentando.

La incorporación al sistema de la funcionalidad de reproducción de audio, para complementar el trabajo realizado, es viable pues ARTIK™ 530 tiene un micrófono monoaural incorporado y su programación tanto para grabación como para reproducción de audio se encuentra disponible a través de herramientas como el programa utilitario precargado 'amixer' o mediante el despliegue de programas con lenguaje Python.

Un aspecto a reforzar dentro del sistema implementado es la seguridad, a pesar de que ARTIK™ 530 brinda autenticación de seguridad punto a punto para la transferencia de datos, sería importante indagar sobre la posibilidad de que esta característica propia de ARTIK se extienda a los dispositivos que no pertenecen a la marca Samsung, otro arista a investigar dentro de esta temática sería que el escaneo de los dispositivos sea detectado únicamente por módulos receptores pertenecientes al sistema, evitando así posibles intrusiones.

Con la finalidad de mejorar las capacidades de los componentes a través de su programación mediante código, cabría la alternativa de utilizar un lenguaje de alto nivel como Python, sería importante indagar en las particularidades y opciones de las librerías PyBluez disponibles para Fedora®; pero se debe considerar la posibilidad de que este lenguaje de programación no se adapte a los nodos.

La tendencia de los proyectos desarrollados con dispositivos IoT es conectarse con la nube con la finalidad principal de intercambiar y procesar sus datos; un ejemplo de aquello son los módulos ARTIK™, que cuentan con la posibilidad de conectarse a la herramienta SAMSUNG ARTIK™ Cloud. Sería interesante indagar opciones de conexión a la nube a los diferentes dispositivos independientemente de la marca comercial a la que pertenezcan; para dicho propósito existen protocolos como CoAP (Constrained Application Protocol) que ofrece un modelo de programación REST (Representational State Transfer), que puede sustituir a las aplicaciones web; así también el protocolo MQTT (Message Queue Telemetry Transport) permite un menor consumo de ancho de banda por parte los dispositivos IoT.

Bibliografía

- [1] IEEE, «Index of /15/pub,» [En línea]. Available: <http://www.ieee802.org/15/pub/>. [Último acceso: Febrero 2017].
- [2] D. A. Westcott y D. D. Coleman, CWNA: Certified Wireless Network Administrator Official Study Guide: Exam CWNA-106, Indianapolis: Sybex, 2014, pp. Fundamentals, History of WLAN.
- [3] D. Hucaby, CCNA Wireless 200-355 Official Cert Guide, Indianapolis: Cisco Press, 2016, pp. Chapter 2 RF Standards, IEEE 802.11 Standards.
- [4] M. S. Gast, 802.11ac: A Survival Guide, Gravenstein Hwy N, Sebastopol: O'Reilly Media, Inc., 2013.
- [5] Cisco Networking Academy®, «Curso: Introducción a Redes,» 2013.
- [6] P. Roshan y J. Leary, 802.11 Wireless LAN Fundamentals, Indianapolis: Cisco Press, 2003.
- [7] F. Thornton y P. Sanghera, How to Cheat at Deploying and Securing RFID, Burlington: Syngress, 2011, p. 121.
- [8] International Organization for Standardization/International Electrotechnical Commission, «International Standard ISO/IEC 18000-7: Information Technology - Radio-frequency identification for item management, Part 7: Parameters for active air interface communications at 433 MHz,» ISO copyright office, Ginebra, Suiza, 2004.
- [9] Samsung Electronics Co., Ltd., «Samsug ARTIK,» 2017. [En línea]. Available: <https://www.artik.io/modules/artik-530/>. [Último acceso: Abril 2017].
- [10] SAMSUNG Artik, «Where do the antennas go?,» 1995-2017. [En línea]. Available: <https://developer.artik.io/documentation/artik/getting-started/antenna-locations.html>. [Último acceso: Abril 2017].
- [11] F. Perea, Arduino Essentials, Birmingham: Packt Publishing Ltd., 2015.

- [12] F. Perea, *Arduino Essentials*, Birmingham: Packt Publishing, 2015.
- [13] C. Bell, *Beginning Sensor Networks with Arduino and Raspberry Pi*, New York: Apress, 2013.
- [14] R. Faludi, *Building Wireless Sensor Networks*, Sebastopol: O'Reilly Media, Inc., 2010.
- [15] iAutomate.com, «RFID Development Kit - Great for Proof of Concept,» [En línea]. Available: <http://www.iautomate.com/products/rfid-development-kit-great-for-proof-of-concept.html>. [Último acceso: Abril 2017].
- [16] ICOFOM, «Conceptos claves de Museología,» 2009. [En línea]. Available: <http://bit.ly/1gdP2fp>. [Último acceso: Abril 2017].
- [17] ArchDaily de México S.A., Tatiana Bilbao, «Propuesta Museo Juan Soriano / Tatiana Bilbao,» Daniela Cruz, 30 Junio 2014. [En línea]. Available: <http://www.archdaily.mx/mx/623274/propuesta-museo-juan-soriano-tatiana-bilbao>. [Último acceso: Mayo 2017].
- [18] Grupo de Trabajo IEEE 8021.15.1, «BLUETOOTH SPECIFICATION Version 4.0 [Vol 3],» 30 Junio 2010. [En línea]. Available: https://www.libelium.com/forum/libelium_files/bt4_core_spec_adv_data_reference.pdf. [Último acceso: Junio 2017].
- [19] J. Sewook, A. Chang y M. Gerla, «Performance Comparison of Overlaid Bluetooth Piconets (OBP) and Bluetooth Scatternet».
- [20] J. Wåhslén, I. Orhan y T. Lindh , «Local Time Synchronization in Bluetooth Piconets,» *International Conference on Body Sensor Networks*, 2011.
- [21] A. Huang, «An Introduction to Bluetooth Programming,» 2008. [En línea]. Available: <https://people.csail.mit.edu/albert/bluez-intro/c404.html>. [Último acceso: Mayo 2017].
- [22] GitHub, Inc., «GitHub, pauloborges/bluez,» 2017. [En línea]. Available: <https://github.com/pauloborges/bluez/blob/master/lib/hci.h>. [Último acceso: Junio 2017].

- [23] SAMSUNG Artik, «ARTIK Modules | Developers,» 1995-2017. [En línea]. Available: <https://developer.artik.io/documentation/developer-guide/gpio/kernel-gpio.html>. [Último acceso: Junio 2017].
- [24] WAVETREND, *RX202 User manual*, 2012.
- [25] iAutomate.com, «iAutomate.com,» 2015. [En línea]. Available: <https://www.iautomate.com/content/rx201-rx202-block-diagram.pdf>. [Último acceso: Junio 2017].
- [26] SAMSUNG Artik, «serial_test/artik_serial_test.c,» 1995-2017. [En línea]. Available: https://developer.artik.io/documentation/api/serial_test_2artic_serial_test_8c-example.html. [Último acceso: Junio 2017].