REPÚBLICA DEL ECUADOR
Escuela Politécnica Nacional
"E SCIENTIA HOMINIS SALUS"

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.

- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.

- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los de

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

MEDICIÓN DEL CONSUMO DE ENERGÍA EN UN NODO SENSOR INALÁMBRICO EN LA TRANSMISIÓN DE VIDEO SOBRE IPV6 EN APLICACIONES DE DOMÓTICA.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

IVÁN PATRICIO ONTANEDA GUAMÁN

ivan.ontaneda@hotmail.com

DIRECTOR: ING. CARLOS ROBERTO EGAS ACOSTA, MSc.

carlos.egas@epn.edu.ec

Quito, junio 2017

DECLARACIÓN

Yo, Iván Patricio Ontaneda Guamán, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____

Iván Patricio Ontaneda Guamán

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Iván Patricio Ontaneda Guamán, bajo mi supervisión.

Ing. Carlos Egas, MSc.
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Agradezco a Dios, por haberme guiado durante estos años por el camino del bien y permitirme culminar una etapa más en mi formación académica.

A mi madre Martha por todo su amor y apoyo incondicional a lo largo de toda mi vida. Y por haber trabajado muy duro para que pudiera culminar mi carrera con éxito. Siempre gracias.

A mi abuelita María por su generosidad, humildad, consejos. Y sobre todo por estar siempre conmigo.

A mi hermana Verónica por su compañía y confianza en todo momento.

A la Escuela Politécnica Nacional y a mis profesores por los conocimientos impartidos durante toda mi carrera universitaria.

Finalmente, un agradecimiento especial al MSc. Carlos Egas, director del presente proyecto, gracias por sus conocimientos, por su apoyo y su confianza para llevar a cabo este gran proyecto.

DEDICATORIA

A mi Dios Padre todo poderoso por haberme guiado por el camino correcto y estar siempre cuando más lo necesité.

A mi madre, por su infinito esfuerzo y apoyo.

A mi abuelita María, que con su paciencia y generosidad siempre estuvo en el momento indicado para brindarme sus consejos y su apoyo incondicional.

CONTENIDO

DECLARACIÓN ..... **I**

CERTIFICACIÓN ..... **II**

AGRADECIMIENTOS ..... **III**

DEDICATORIA ..... **IV**

CONTENIDO ..... **V**

ÍNDICE DE FIGURAS ..... **X**

ÍNDICE DE TABLAS ..... **XIII**

ÍNDICE DE SEGMENTOS DE CÓDIGO ..... **XIV**

RESUMEN ..... **XV**

PRESENTACIÓN ..... **XVI**

CAPÍTULO I **1**

1 MARCO TEÓRICO **1**

1.1 LA DOMÓTICA 1

1.1.1 APLICACIONES DE LA DOMÓTICA 1

1.2 REDES LAN INALÁMBRICAS 2

1.2.1 TOPOLOGÍAS DE REDES LAN INALÁMBRICAS 3

1.3 IEEE 802.11 4

1.4 TECNOLOGÍA WIFI (802.11 b/a/g/n) 6

1.4.1 IEEE 802.11b 6

1.4.2 IEEE 802.11a 6

1.4.3 IEEE 802.11g 7

1.4.4 IEEE 802.11n 7

1.5 IPv6 8

1.5.1 BENEFICIOS DE IPv6 8

1.5.2 SINTAXIS 10

1.5.3 IPv6 SOBRE IEEE 802.11 10

1.6 MÓDULO INTEL EDISON 11

1.6.1 COMPONENTES DEL MÓDULO INTEL EDISON 11
1.6.2 FIRMWARE DEL MÓDULO INTEL EDISON 14
1.6.3 ENTORNOS DE DESARROLLO INTEGRADO 15
1.6.4 TARJETAS DE DESARROLLO 15
1.6.4.1 Tarjeta Intel Breakout 15
1.6.4.1.1 Jumpers 16
1.6.4.1.2 Fuente de alimentación 18
1.6.4.1.3 Polarización 19
1.6.4.1.3.1 Voltaje de arranque - Señal DCIN 20
1.6.4.1.4 Pulsadores 21
1.6.4.1.5 Leds 22
1.6.4.1.6 Cables de conexión 22
1.6.4.1.7 Baterías alcalinas 23
1.6.4.1.8 Manipulación 23
1.7 SISTEMAS OPERATIVOS 23
1.7.1 SISTEMA OPERATIVO LINUX 24
1.7.2 SISTEMA OPERATIVO WINDOWS 24
1.7.3 SISTEMA OPERATIVO ANDROID 24
1.7.3.1 Android con soporte IPv6 25
1.7.4 COMPARACIÓN ENTRE SISTEMAS OPERATIVOS 26
1.8 CÁMARAS WEB 26
1.8.1 ESTÁNDAR UVC 27

CAPÍTULO II 28

2 IMPLEMENTACIÓN DEL PROTOTIPO PARA LA TRANSMISIÓN DE VIDEO SOBRE IPV6 28

2.1 IMPLEMENTACIÓN Y CONFIGURACIÓN DE LA RED DE TRANSPORTE IPv6…. 29
2.1.1 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DE LA RED 29
2.1.1.1 Router 29
2.1.1.2 PuTTY 30
2.1.1.3 Gateway 31
2.1.2 IMPLEMENTACIÓN DE LA RED DE TRANSPORTE IPv6 31

2.1.3 CONFIGURACIÓN DE LA RED DE TRANSPORTE IPv6.....33
2.1.3.1 Configuración OSPFv3.....34
2.1.3.2 Configuración de servidor DHCPv6.....35
2.1.3.3 Conexión y configuración del Gateway.....36
2.2 CONFIGURACIÓN DE LA PC Y DEL DISPOSITIVO MÓVIL EN LA RED IPv6… 38
2.2.1 CONFIGURACIÓN DE LA PC Y LA LAPTOP.....38
2.2.2 CONFIGURACIÓN DEL DISPOSITIVO MÓVIL.....39
2.2.3 PRUEBAS DE CONECTIVIDAD.....40
2.2.3.1 Prueba de conectividad entre la Pc y la red IPv6.....40
2.2.3.2 Prueba de conectividad entre la laptop y la red IPv6.....40
2.2.3.3 Prueba de conectividad entre el dispositivo móvil y la red IPv6...41
2.3 CONFIGURACIÓN DEL DISPOSITIVO INTEL EDISON.....41
2.3.1 REQUERIMIENTOS DE CONFIGURACIÓN.....41
2.3.1.1 Ubuntu 14.04.....42
2.3.1.2 Intel Platform Flash Tool Lite.....42
2.3.2 CONFIGURACIÓN DEL DISPOSITIVO.....43
2.3.2.1 Ensamblaje del dispositivo Intel Edison.....43
2.3.2.2 Instalación del firmware.....43
2.3.2.3 Configuración de acceso al dispositivo Intel Edison.....45
2.3.2.4 Configuración Wi-Fi.....46
2.3.2.4.1 Connman.....46
2.3.2.5 Configuración SSH.....48
2.3.2.6 Configuración de la alimentación externa.....48
2.3.2.7 Configuración del gestor de paquetes.....49
2.3.2.8 Configuración de la cámara web.....50
2.4 IMPLEMENTACIÓN DE LA APLICACIÓN.....51
2.4.1 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN.....51
2.4.1.1 Sublime Text 3.....51
2.4.1.2 Node.JS.....52
2.4.1.2.1 Npm.....52
2.4.2 DESARROLLO DE LA APLICACIÓN.....53
2.4.2.1 Archivo de configuración.....55

2.4.2.2 Servidor web 55
2.4.2.3 Servidor WebSocket.....58
2.4.2.4 Implementación de video.....60
2.4.2.5 Página web 62
2.4.2.5.1 Página de inicio.....62
2.4.2.5.2 Página del computador 64
2.4.2.5.3 Página del dispositivo móvil 66
2.5 IMPLEMENTACIÓN DEL CÓDIGO EN EL DISPOSITIVO INTEL EDISON… 66

CAPÍTULO III.....69

3 PRUEBAS DE FUNCIONAMIENTO Y MEDICIÓN DEL CONSUMO DE ENERGÍA 69

3.1 PRUEBAS DE FUNCIONAMIENTO 69
3.1.1 PRUEBA 1 DE FUNCIONAMIENTO EN LA PC.....69
3.1.2 PRUEBA 2 DE FUNCIONAMIENTO EN LA LAPTOP.....71
3.1.3 PRUEBA 3 DE FUNCIONAMIENTO EN EL DISPOSITIVO MÓVIL 72
3.2 MEDICIÓN DEL CONSUMO DE ENERGÍA.....75
3.2.1 REQUERIMIENTOS PARA LA MEDICIÓN.....75
3.2.1.1 Equipos de medición 75
3.2.1.2 Tiempos de operación.....76
3.2.1.3 Resoluciones de trasmisión.....76
3.2.1.4 Distancia al nodo gateway.....76
3.2.2 ESCENARIO DE MEDICIÓN 77
3.2.3 RESULTADOS OBTENIDOS.....77
3.2.3.1 Estado de reposo 77
3.2.3.2 Aplicación inicializada 77
3.2.3.3 Transmisión de video hacia la PC 78
3.2.3.4 Transmisión de video hacia la Laptop 78
3.2.3.5 Transmisión de video hacia el Celular 79
3.2.4 ANÁLISIS DE RESULTADOS.....79
3.2.4.1 Estado de reposo 79
3.2.4.1.1 Tiempo de descarga de la batería.....80

3.2.4.2 Aplicación inicializada 80
3.2.4.2.1 Tiempo de descarga de la batería.....80
3.2.4.3 Transmisión de video hacia la PC 81
3.2.4.3.1 Tiempo de descarga de la batería.....82
3.2.4.4 Transmisión de video hacia la laptop.....83
3.2.4.4.1 Tiempo de descarga de la batería.....83
3.2.4.5 Transmisión de video hacia el celular.....84
3.2.4.5.1 Tiempo de descarga de la batería.....85

CAPÍTULO IV 86

4 CONCLUSIONES Y RECOMENDACIONES 86

4.1 CONCLUSIONES 86
4.2 RECOMENDACIONES 87

REFERENCIAS BIBLIOGRÁFICAS 89

ANEXOS.....95

ÍNDICE DE FIGURAS

CAPÍTULO I

Figura 1.1. Campos de aplicación de la domótica. 2
Figura 1.2. Topología de infraestructura....4
Figura 1.3. Topología ad-hoc. 4
Figura 1.4. Módulo de Cómputo Intel Edison 11
Figura 1.5. Diagrama de bloques Intel Edison....13
Figura 1.6. Módulo Intel Edison....14
Figura 1.7. Tarjeta Intel Breakout....15
Figura 1.8. Diagrama de bloques tarjeta Intel Breakout....16
Figura 1.9. Jumper de Tarjeta Intel Breakout 17
Figura 1.10. Conector Tipo Barril 18
Figura 1.11. Formas de alimentación DC....19
Figura 1.12. Distribución de Polarización de la Tarjeta Intel Breakout....20
Figura 1.13. Botón de encendido 21
Figura 1.14. Leds....22
Figura 1.15. Cable de conexión. 22
Figura 1.16. Insertando el Módulo Intel Edison en la Tarjeta Intel Breakout. 23
Figura 1.17. Cámara web.....27

CAPÍTULO II

Figura 2.1. Diagrama esquemático del prototipo a implementarse. 28
Figura 2.2. Red a implementarse. 29
Figura 2.3. Router Cisco 1841. 30
Figura 2.4. Interfaz gráfica del emulador PuTTY. 30
Figura 2.5. Router D-Link DIR-600....31
Figura 2.6. Esquema de conexión de los routers. 32
Figura 2.7. Esquema de conexión de los Gateways.....33
Figura 2.8. Diagrama de red a ser implementado. 33
Figura 2.9. Comando para verificar conectividad. 35
Figura 2.10. Conexión del router con la PC.....36
Figura 2.11. Página de configuración del router. 37

Figura 2.12. Configuración Wireless Network.....37
Figura 2.13. Dirección IPv6 de la PC. 38
Figura 2.14. Dirección IPv6 de la laptop.....39
Figura 2.15. Selección de red. 39
Figura 2.16. Verificación de dirección IPv6. 40
Figura 2.17. Ping entre la PC y la red IPv6. 40
Figura 2.18. Ping entre la laptop y la red IPv6.....41
Figura 2.19. Ping entre el dispositivo móvil y la red IPv6.....41
Figura 2.20. Interfaz gráfica Flash Tool Lite. 42
Figura 2.21. Componentes del dispositivo.....43
Figura 2.22. Dispositivo Intel Edison. 43
Figura 2.23. Descarga del firmware más reciente. 44
Figura 2.24. Instalación del firmware inicial.....44
Figura 2.25. Firmware instalado.....45
Figura 2.26. Ejecución de inicio.....45
Figura 2.27. Configuración del gestor de conexiones.....47
Figura 2.28. Ejecución de connmanctl. 47
Figura 2.29. Interfaz wlan0.....47
Figura 2.30. Acceso vía SSH. 48
Figura 2.31. Actualización Opkg.....49
Figura 2.32. Conexión de la batería y cámara web. 50
Figura 2.33. Interfaz Sublime Text 3. 51
Figura 2.34. NodeJs.....52
Figura 2.35. NPM.....52
Figura 2.36. Flujograma del proyecto edi-cam. 53
Figura 2.37. Flujograma de la aplicación.....54
Figura 2.38. Flujograma del servidor web. 57
Figura 2.39. Conexión Websocket 58
Figura 2.40. Flujograma de implementación de video. 62
Figura 2.41. Página de inicio.....63
Figura 2.42. Página del computador. 64
Figura 2.43. Página del dispositivo móvil. 66
Figura 2.44. Código subido a la plataforma GitHub. 67

Figura 2.45. Ejecución del comando git clone. 67
Figura 2.46. Instalación de paquetes npm. 68
Figura 2.47. Ejecución del script server.js. 68

CAPÍTULO III

Figura 3.1. Componentes de la Prueba 1. 70
Figura 3.2. Prueba de funcionamiento en la PC. 70
Figura 3.3. Componentes de la Prueba 2. 71
Figura 3.4. Prueba de funcionamiento en la laptop. 72
Figura 3.5. Componentes de la Prueba 3. 73
Figura 3.6. Prueba de funcionamiento en el dispositivo móvil. 74
Figura 3.7. Equipos de medición. 75
Figura 3.8. Distancia entre dispositivos. 76
Figura 3.9. Escenario y componentes para la medición. 77

ÍNDICE DE TABLAS

CAPÍTULO I

Tabla 1.1. Estándares IEEE 802.11. 8
Tabla 1.2. Características del Hardware. 12
Tabla 1.3. Comparación de los sistemas operativos. 26

CAPÍTULO II

Tabla 2.1. Direccionamiento de red. 33

CAPÍTULO III

Tabla 3.1. Tiempos de Operación. 76
Tabla 3.2. Resoluciones. 76
Tabla 3.3. Distancias al nodo Gateway. 77
Tabla 3.4. Valores de medición en transmisión de video hacia la PC. 78
Tabla 3.5. Valores de medición en transmisión de video hacia la laptop. 78
Tabla 3.6. Valores de medición en transmisión de video hacia el celular. 79
Tabla 3.7. Consumo de potencia hacia la PC. 81
Tabla 3.8. Resultados de consumo. 82
Tabla 3.9. Consumo de potencia hacia la laptop. 83
Tabla 3.10. Resultados de consumo. 83
Tabla 3.11. Consumo de potencia hacia el celular. 84
Tabla 3.12. Resultados de consumo. 85

ÍNDICE DE CÓDIGOS

CAPÍTULO II

Código 2.1. Configuración global OSPFv3. 34
Código 2.2. Configuración de la interfaz. 34
Código 2.3. Valor DUID. 35
Código 2.4. Configuración pool de direcciones IPv6. 36
Código 2.5. Configuración pool de direcciones IPv4. 36
Código 2.6. Archivo de configuración. 55
Código 2.7. Servidor web. 56
Código 2.8. Puertos. 57
Código 2.9. Variables de video. 59
Código 2.10. Servidor WebSocket. 59
Código 2.11. Archivo ffmpeg. 61
Código 2.12. Servicio de archivos estáticos. 62
Código 2.13. Enlace de páginas. 64
Código 2.14. Decodificación de video. 65
Código 2.15. Conexión WebSocket. 65

RESUMEN

El presente trabajo de titulación tiene como objetivo la medición del consumo de energía por la transmisión de video sobre IPv6 en aplicaciones de domótica en un nodo sensor que utiliza batería. En este caso, la plataforma Intel Edison será el nodo sensor encargado de procesar y transmitir el video sobre la red.

En el primer capítulo se detalla de forma teórica conceptos de domótica, redes de área local inalámbricas, IEEE 802.11, tecnología WiFi, IPv6, características del módulo Intel Edison, tarjeta Intel Breakout, baterías alcalinas, sistemas operativos y cámaras web.

En el segundo capítulo se describe los requerimientos, implementación y configuración de la red de transporte IPv6, la configuración de los dispositivos finales y la tarjeta Intel Edison en la red IPv6, la descripción del proyecto edi-cam, las modificaciones realizadas para transmitir video sobre la red IPv6 y la implementación del código en el dispositivo Intel Edison.

En el tercer capítulo se detalla las pruebas de funcionamiento en los diferentes dispositivos y la medición del consumo de energía cuando se transmite video sobre la red IPv6.

En el cuarto capítulo se plantean las conclusiones y recomendaciones referidas al presente trabajo de titulación.

Adicionalmente se incluye la sección de Anexos, adjuntos en forma digital, donde se presenta la configuración de la red IPv6 y los códigos utilizados en la aplicación desarrollada.

PRESENTACIÓN

El vertiginoso avance de la tecnología experimentado en los últimos años ha contribuido de una manera eficaz al desarrollo de la domótica en aspectos tan cotidianos como la iluminación, climatización, seguridad, comunicación, etc. Es por eso que muchas aplicaciones nuevas aparecen con el pasar de los días.

Sin embargo, aún no existe en el mercado nacional aplicaciones de domótica que utilicen para su funcionamiento el protocolo IPv6 (Internet Protocol version 6). De igual manera en el mercado internacional tampoco existen muchas aplicaciones de domótica que utilicen IPv6, esto se debe a que todavía no se ha generalizado el uso de este protocolo en las redes del hogar.

Por otra parte, el uso de baterías como fuente de alimentación de los nodos sensores inalámbricos es común debido a que estos deben permanecer situados en lugares alejados y en funcionamiento por largos periodos de tiempo. Lo que limita el tiempo de actividad de este tipo de nodos.

En la actualidad la mayoría de proveedores de servicio de Internet del país, ya tienen implementado IPv6 en sus redes de transporte y se prevé que para el próximo año darán IPv6 a los usuarios. Es por esto que es indispensable empezar a aprovechar las ventajas que ofrece IPv6 como protocolo de red para el desarrollo de futuras aplicaciones.

Con el desarrollo de este trabajo de titulación se presenta una opción para implementar una aplicación que trabaje con el protocolo IPv6 en el campo de la domótica utilizando para ello la plataforma Intel Edison y el uso de baterías. A si mismo se realizan mediciones para calcular el consumo de energía cuando se está transmitiendo video sobre la red IPv6. Además se deja abierta la posibilidad que en base a este trabajo de titulación, se implementen nuevos proyectos junto con la plataforma Intel Edison y el protocolo IPv6.

CAPÍTULO I

1 MARCO TEÓRICO

1.1 LA DOMÓTICA

La domótica es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda para mejorar la calidad de vida del usuario. El término domótica etimológicamente proviene de la unión de la palabra latina “domus” que significa casa y de la palabra griega “tica” que significa automática.

Un sistema domótico es capaz de recoger información proveniente de nodos sensores, procesar y emitir órdenes a los actuadores [1]. Además, el sistema puede acceder a redes exteriores como internet.

La domótica permite dar solución a los requerimientos sociales y a las nuevas tendencias de nuestra forma de vida, facilitando el diseño de viviendas y hogares polifuncionales y flexibles.

1.1.1 APLICACIONES DE LA DOMÓTICA [1], [2]

En el campo de la domótica, las aplicaciones se pueden ordenar en las siguientes categorías:

- *Gestión Energética:* La gestión energética es considerada una de las aplicaciones más trascendentales de la domótica. Se apoya en tres principios fundamentales los cuales son: el ahorro energético, la eficiencia energética y la generación energética. La domótica ofrece una gestión eficiente del uso de todos los tipos de energía.
- *Confort:* Desde el punto de vista de la domótica, el confort es básicamente la gestión de diferentes dispositivos tales como el control de la iluminación, control del clima, control de aberturas, entre otros, para convertir la vivienda en un lugar más confortable.

- *Seguridad:* Es una de las aplicaciones más importantes de la domótica. Está compuesta por una red de seguridad encargada de vigilar automáticamente las personas, animales y bienes, así como también de incidencias y averías.
- *Accesibilidad:* Esta aplicación de la domótica ha tomado mucha importancia en los últimos años. Busca posibilitar el acceso de cualquier persona a cualquier entorno utilizando para ello mecanismos que se ajusten más a sus necesidades.
- *Comunicación:* Esta aplicación busca garantizar la comunicación mediante el control y la supervisión remota de la vivienda a través del celular, Tablet, PC, entre otros, y aumentar la interactividad entre las personas y el hogar.

La Figura 1.1, muestra los diferentes campos de aplicación de la domótica.

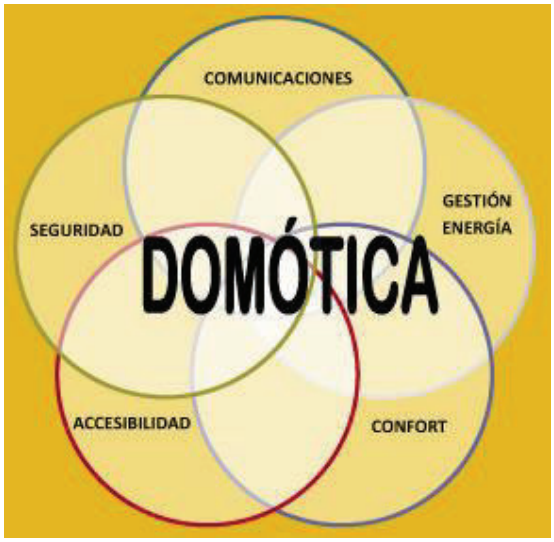


Figura 1.1. Campos de aplicación de la domótica [1].

1.2 REDES LAN INALÁMBRICAS [3]

Hoy en día, las redes LAN (*Local Area Network*, Red de Área Local) inalámbricas se han convertido en una alternativa a la hora de disponer de una red que no tenga limitaciones en cuanto a su ubicación. Las conexiones inalámbricas pueden reemplazar una infraestructura con cables disminuyendo los gastos enormemente. Este hecho puede ser de interés en numerosos escenarios, por ejemplo en las normativas de construcción, en donde el uso de cables puede estar restringido.

El aumento de usuarios móviles que ha tenido el nuevo milenio ha causado que las redes LAN inalámbricas tomen más importancia para su uso. El acceso portátil a las redes inalámbricas se hace a través de equipos portátiles, dispositivos móviles, y NIC[1] (*Network Interface Card*, Tarjeta de interfaz de red), inalámbricas. El hecho de no tener que utilizar más cables para instalar nuevos dispositivos ha facilitado las conexiones en redes domésticas. Además permite al usuario desplazarse a distintos lugares (salas de reunión, salas de espera, cafeterías, aulas, etc.), sin perder el acceso a los datos de la red. Sin el acceso inalámbrico, se tendría que utilizar cables de diferentes longitudes y disponer de conexiones de red, lo cual sería muy incómodo [4].

En varios lugares, el acceso a internet está disponible a través de zonas activas de redes inalámbricas públicas. Las estaciones de buses, los centros comerciales, los aeropuertos, los restaurantes y otras áreas comunes son dotadas de equipamiento necesario para ofrecer este servicio. En todos estos escenarios anteriormente mencionados las redes LAN inalámbricas funcionan a alta velocidad, para ello hacen uso de los estándares 802.11.

1.2.1 TOPOLOGÍAS DE REDES LAN INALÁMBRICAS [3]

Las redes LAN inalámbricas se implementan utilizando dos topologías básicas. Se utilizan distintos términos para nombrar estas topologías, tales como administradas y no administradas, infraestructura y ad-hoc, y alojadas y par a par. Se utilizará los términos infraestructura y ad-hoc.

Una topología de infraestructura está conformado por una LAN de cable y una LAN inalámbrica conectadas a través de una estación base, denominada AP (*Access Point*, Punto de Acceso). El punto de acceso sirve de controlador central de la red LAN inalámbrica, coordina la transmisión y recepción de múltiples dispositivos inalámbricos dentro de un área específica. El número de dispositivos y el área dependen del estándar de conexión inalámbrico que se esté utilizando [5]. Se pueden utilizar varios puntos de acceso según el área que se desea cubrir, ya sea un AP para dar cobertura a una zona pequeña o varios puntos de acceso para una zona grande. En la Figura 1.2 se presenta una topología de infraestructura.

[1] NIC: Es el dispositivo electrónico que permite a un terminal acceder a una red y compartir recursos.

Figura 1.2. Topología de infraestructura.

En una topología ad-hoc los propios dispositivos inalámbricos crean la LAN, cada dispositivo se puede comunicar directamente con los demás dispositivos de la red sin necesidad de pasar por un controlador centrar ni puntos de acceso. Esta topología es práctica en lugares pequeños que no necesitan de otra red. Algunos ejemplos en donde se podrían utilizar este tipo de topología serian un domicilio sin LAN de cable o una sala de reuniones donde los equipos usados se reúnen para compartir ideas [3]. En la Figura 1.3 se presenta una topología ad-hoc

Figura 1.3. Topología ad-hoc.

1.3 IEEE 802.11 [6], [7]

Tras el auge de la tecnología Ethernet como el estándar que determina las características físicas y eléctricas que debe poseer una LAN, surgieron

inconvenientes en cuanto a la conectividad donde el cableado era inadecuado para soportar redes LAN cableadas. Es por eso que el deseo de tener una "Ethernet inalámbrica" se hizo más notable y fue la motivación para crear IEEE 802.11. El trabajo sobre el primer estándar de comunicaciones inalámbricas de datos comenzó en 1990 cuando IEEE 802.11 se inició como un grupo de trabajo dentro de IEEE 802. De esta manera los dispositivos de difícil acceso a una conectividad cableada podrían hacerlo de forma inalámbrica dentro de la red. Otra motivación para formar el grupo de trabajo IEEE 802.11 fue la asignación de la banda de 2,4 GHz para dispositivos sin licencia por parte de la FCC[2] (*Federal Communications Commission*, Comisión Federal de Comunicaciones), en 1986.

El estándar IEEE 802.11 define la capa física y la capa de enlace de datos al igual que el resto de los estándares IEEE 802. El primer estándar de este protocolo fue IEEE 802.11 Legacy. Especifica velocidades de transmisión 1 y 2 Mbit/s empleando como medio físico la transmisión de señales infrarrojas y utiliza CSMA/CA[3] (*Carrier Sense Multiple Access With Collision Avoidance*, Acceso Múltiple Detección de Portadora y Prevención de Colisiones), para acceso al canal. A lo largo de su desarrollo la norma ha sido confundida con diferentes nombres es por eso que es llamado de varias maneras tales como Wi-Fi (*Wireless-Fidelity*, Fidelidad Inalámbrica), WLAN (*Wireless Local Area Network*, Red de Área Local Inalámbrica) y IEEE 802.11x [8].

Wi-Fi es una marca de la Alianza Wi-Fi[4] que tiene como objetivo hacer cumplir los requerimientos para la interoperabilidad entre los equipos basados en el estándar IEEE 802.11. En la actualidad el nombre de Wi-Fi se usa en vez de IEEE 802.11. WLAN es utilizado para describir una red de área local inalámbrica. Este es un nombre alternativo usado por el IEEE para designar al estándar 802.11. Y finalmente IEEE 802.11x es usado habitualmente para referirse a todo el grupo de estándares dentro del IEEE 802.11 (a, b, g, n, etc.) [8].

---

[2] FCC: Agencia estatal independiente de los Estados Unidos encargada de la regulación de telecomunicaciones interestatales e internacionales.
[3] CSMA/CA: Permite que múltiples estaciones utilicen el mismo medio para la transmisión. Cada equipo anuncia antes de transmitir para evitar colisiones
[4] Alianza Wi-Fi: Es una organización sin fines de lucro que promueve la tecnología Wi-Fi y certifica los productos Wi-Fi.

El estándar 802.11 para redes WLAN incluye una serie de revisiones. Estas revisiones se diferencian principalmente en las técnicas de modulación a usarse, la gama de frecuencias a soportar y la QoS (*Quality of Service*, Calidad de Servicio) [8].

1.4 TECNOLOGÍA WIFI (802.11 b/a/g/n) [9]

La tecnología WiFi está compuesta por los estándares inalámbricos IEEE 802.11a, 802.11b, 802.11g y 802.11n. Muchos productos como por ejemplo teléfonos inteligentes, enrutadores inalámbricos, Laptops, impresoras inalámbricas, utilizan las tecnologías 802.11 para su funcionamiento. Los estándares son usados para la implementación de redes de área local inalámbricas (WLAN) rápidas a ultra rápidas en la actualidad. En esta sección se describirá las características más importantes de los estándares b, a, g y n.

1.4.1 IEEE 802.11b

La revisión IEEE 802.11b logra una velocidad de transmisión de hasta 11 Mbit/s y utiliza CSMA/CA para el control de acceso al medio. Funciona en la banda libre de 2.4GHz y se basa en la misma técnica de modulación DSSS especificada en el estándar original. Un dispositivo basado en esta revisión puede operar hasta 11 Mbit/s, y reducirá automáticamente su velocidad de transmisión hasta 5.5, 2 o 1 Mbit/s cuando el receptor empiece a detectar errores debido a la interferencia o a la atenuación del canal. Las ventajas de utilizar este estándar son: bajo costo, el alcance de la señal es buena y no se obstruye fácilmente. Entre las desventajas se tiene: más lento cuando trabaja a máxima velocidad y puede tener interferencias con los electrodomésticos que trabajan en la misma banda de frecuencia [8] [10].

1.4.2 IEEE 802.11a

La revisión IEEE 802.11a utiliza la técnica de modulación OFDM[5] (*Orthogonal Frequency-Division Multiplexing*, Multiplexación por división de frecuencia ortogonal), lo que permite obtener velocidades de transmisión de hasta 54 Mbit/s. Emplea CSMA/CA para el control de acceso al medio y opera en la banda de 5GHz.

[5] OFDM: Es una técnica de modulación que se consigue dividiendo el canal en un conjunto de subportadoras que se dividen en grupos en función de la necesidad de cada uno de los usuarios.

Este estándar no es compatible con IEEE 802.11b debido a que utilizan bandas de frecuencia distintas, sin embargo existen equipos que pueden trabajar con ambos estándares. La revisión IEEE 802.11a no ha alcanzado la difusión que tiene IEEE 802.11b, por haber llegado más tarde al mercado, además la banda en la que trabaja no está disponible aun en todos los países. Las ventajas de utilizar esta revisión son: eficiente a velocidad máxima y la banda de frecuencias en la que trabaja está regulada, de esta manera se evita la interferencia de la señal con otros dispositivos. Entre las desventajas se tiene: costo más alto, por eso se encuentra generalmente en las redes de negocios y el alcance de la señal es corta por ende se puede obstruir fácilmente [8] [10].

1.4.3 IEEE 802.11g

La revisión IEEE 802.11g opera en la banda de 2.4GHz. Utiliza la técnica de modulación OFDM y alcanza una velocidad de transmisión teórica máxima de 54 Mbit/s o cerca de 24.7 Mbit/s de velocidad real de transferencia. La interoperabilidad que tiene con 802.11b es una de las razones fundamentales de su masiva aceptación. Los equipos que trabajan bajo este estándar llegaron al mercado tan pronto como se ratificó la norma en Junio de 2003. A pesar de su mayor aceptación 802.11g tiene el mismo problema de 802.11b con respecto a la interferencia puesto que funciona en el mismo rango de frecuencia ya saturado por dispositivos como hornos microondas, dispositivos bluetooth y teléfonos inalámbricos. Las ventajas de utilizar esta revisión son: eficiente a velocidad máxima y el rango de señal es bueno y no se obstruye fácilmente [8] [10].

1.4.4 IEEE 802.11n [11]

La revisión IEEE 802.11n también conocida como "Wireless N" fue diseñada para mejorar el estándar IEEE 802.11g. Introduce el concepto MIMO (*Multiple-Input Multiple-Output*). MIMO implica utilizar varios transmisores y múltiples antenas receptores para aumentar la velocidad de transferencia y el alcance. Los grupos de estándares de la industria ratificaron el IEEE 802.11n en 2009 con especificaciones que proporcionan hasta 300 Mbit/s de velocidad de transmisión en la red. Las ventajas de utilizar esta revisión son: eficiente a velocidad máxima, mayor alcance de la señal y es más resistente a la interferencia de la señal de fuentes externas.

Entre las desventajas se tiene: su costo es mayor y el uso de múltiples señales puede interferir en gran medida con las redes 802.11b/g cercanas.

En la Tabla 1.1 se puede observar un cuadro comparativo entre los principales estándares IEEE 802.11.

Tabla 1.1. Estándares IEEE 802.11 [9].

Protocolo	Fecha de Lanzamiento	Frecuencia de Operación	Velocidad de Datos (Típica)	Velocidad de Datos (Max)
IEEE 802.11b	1999	2.4 GHz	6.5 Mbit/s	11 Mbit/s
IEEE 802.11a	1999	5 GHz	25 Mbit/s	54 Mbit/s
IEEE 802.11g	2003	2.4 GHz	25 Mbit/s	54 Mbit/s
IEEE 802.11n	2007	2.4 GHz	200 Mbit/s	500 Mbit/s

1.5 IPv6

Ante la escasez de direcciones IPv4 (*Internet Protocol version 4*, Protocolo de Internet versión 4) y en previsión de que los equipos que se conectan a una red serán mucho más numerosos a futuro, surge el protocolo IPv6 (*Internet Protocol version 6*, Protocolo de Internet versión 6).

1.5.1 BENEFICIOS DE IPv6 [12]

IPv6 permitirá el crecimiento del Internet para las próximas décadas. Este nuevo protocolo aumenta las capacidades de la versión actual (IPv4).

Los beneficios de utilizar este nuevo protocolo se pueden agrupar de la siguiente forma:

- *Más espacio de direcciones*: IPv6 amplía el espacio de direcciones IP a 128. El espacio de direcciones de 128 bits es lo suficientemente grande como para tener 155.000 millones de direcciones IPv4 en cada metro cuadrado sobre la superficie de la Tierra, incluyendo los océanos [13].

- *Innovación*: IPv6 elimina el uso de NAT[6] (*Network Address Translation*, Traducción de Direcciones de Red), para permitir el uso de nuevas aplicaciones tales como VoIP[7] (*Voice Over IP*, Voz sobre IP), videoconferencia de buena calidad, etc.
- *Configuración Automática Sin Estado*: Con IPv6 no es necesario configurar manualmente las direcciones de los servidores o utilizar un DHCP[8] (*Dynamic Host Configuration Protocol*, Protocolo de Configuración Dinámica de Host), debido a que no hay ningún "estado" (información de configuración) que se pueda perder.
- *Renumbering*: Cambiar direcciones IP para un grupo de hosts se vuelve más sencillo utilizando IPv6, todo lo que se requiere es que los routers dejen de anunciar el prefijo antiguo y empiecen a anunciar uno nuevo. Los hosts crearán automáticamente nuevas direcciones para ellos y notarán que las direcciones viejas ya no están "actualizadas".
- *Eficiencia:* El procesamiento IPv6 es más eficiente que el procesamiento IPv4. Las mejoras de eficiencia en IPv6 incluyen lo siguiente:
  - La cabecera IPv6 tiene una longitud fija.
  - La cabecera IPv6 está optimizada para procesar hasta 64 bits a la vez (32 en IPv4).
  - Ya no se requiere que los enrutadores fragmenten los paquetes sobredimensionados, simplemente pueden señalar a la fuente para enviar paquetes más pequeños.
  - Todas las difusiones para funciones de descubrimiento fueron reemplazadas por multicasts[9]. Sólo los hosts que están escuchando activamente las multicasts se interrumpen, en lugar de todos los hosts, como ocurre con las emisiones.

[6] NAT: Proceso necesario cuando se intenta establecer comunicación entre dos puntos de red que no son compatibles.
[7] VoIP: Voz sobre el protocolo de Internet.
[8] DHCP: Contiene un grupo de direcciones IP que se asignan dinámicamente al nodo solicitante.
[9] Multicast: Es el envío de la información en múltiples redes a múltiples destinos simultáneamente.

1.5.2 SINTAXIS [14]

IPv6 dispone de un espacio de direccionamiento de 128 bits divididos en ocho campos de dos bytes cada uno. Cada campo debe estar escrito con cuatro dígitos hexadecimales y separado de otro usando (:) como se indica a continuación:

2001:0000:0000:0000:AC10:0700:2DDC:FE01

La dirección anterior se puede adaptar a un formato más sencillo cuando se tienen ceros innecesarios, es decir, la dirección quedaría como:

2001:0:0:0:AC10:700:2DDC:FE01

Y más aún cuando existe un intervalo de ceros, un formato más simple sería:

2001::AC10:700:2DDC:FE01

Cabe resaltar que el símbolo (::) se debe utilizar una sola vez en la representación de una dirección IPv6.

Para indicar el número de red en una dirección IPv6 se utiliza un prefijo representado de la siguiente manera:

Dirección IPv6 / Longitud del prefijo

Por ejemplo: 2001:0:0:0:8:: / 60, indicaría un prefijo de 60 bits para una subred.

1.5.3 IPv6 SOBRE IEEE 802.11

En la práctica, las LAN inalámbricas que son utilizables con IPv4 también se pueden utilizar con IPv6. No obstante, hay algunas diferencias prácticas, especialmente con respecto a la multidifusión que se trabaja en IPv6. La transmisión en el aire tiene una tasa de error relativamente alta, por lo que todas las tramas unicast[10] IEEE 802.11 son reconocidas y, si es necesario, se retransmiten en la capa de enlace de datos.

Sin embargo, no existe una manera razonable para que los receptores reconozcan la recepción multicasts o broadcasts[11]. Esto significa que el punto de acceso

[10] Unicast: Es el envío de información desde un único emisor a un único receptor.
[11] Broadcast: Transmisión de datos que serán recibidos por todos los dispositivos en una red.

transmite multicasts a una velocidad “segura” y espera que sean recibidas de la mejor manera. Se transmite multicasts a velocidades bajas si es necesario, para acoplarse a implementaciones 802.11 que están limitadas a 2 Mbps (IEEE 802.11 Legacy). En teoría, se puede decir que IPv6 sobre IEEE 802.11 es menos fiable que IPv4, porque IPv6 está diseñado para trabajar en multicasts que es más eficiente [12].

1.6 MÓDULO INTEL EDISON

Intel Edison es una plataforma de computación ultra pequeño desarrollado por Intel® y que es capaz de realizar las funciones de cualquier computador [15].

La primera versión de la placa Intel Edison fue lanzada en enero del 2014 con una aplicación para la vigilancia de bebes denominada Nursery 2.0[12].

Tras algunos cambios, en septiembre del mismo año, se presentó la segunda versión del módulo Intel Edison cuyo tamaño se asimila a la de una tarjeta SD[13], con componentes en ambos lados de la placa. En la Figura 1.4 se muestra el módulo Intel Edison.

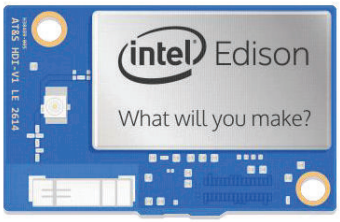


Figura 1.4. Módulo de Cómputo Intel Edison [16].

1.6.1 COMPONENTES DEL MÓDULO INTEL EDISON [17]

La plataforma Intel Edison está diseñada para reducir las barreras de entrada para cualquier persona que realice prototipos y produzca IoT (*Internet of things*, Internet de las cosas) y productos de computación portátiles. Intel Edison es un módulo que se interconecta con sistemas de usuario final a través de un conector de 70 pines. La Tabla 1.2 muestra los principales componentes del sistema.

[12] Nursery 2.0: Fue la primera aplicación desarrollada con el uso de la placa Intel Edison.
[13] SD: Es una tarjeta pequeña que permite guardar información en dispositivos portátiles.

Tabla 1.2. Características del Hardware [17].

Componente	Descripción
Procesador	Modelo del procesador: Intel® Atom™ Frecuencia del procesador: 500 MHz Litografía del procesador: 22 nm Modelo de microcontrolador: Intel Quark Frecuencia microcontroladores: 100 MHz
RAM	1 GB de memoria LPDDR3[14] POP[15]
Almacenamiento Interno	4 GB eMMC[16] para almacenar el sistema de archivos y los datos de usuario
Energía	TI[17] SNB9024 PMIC[18]
Wireless	Banda Dual (2.4 y 5 GHz) IEEE 802.11a/b/g/n
Bluetooth	Bluetooth version 4.
Antena	Antena integrada de chips de doble banda o u.FL[19] para antena externa.
Conector	Hirose[20] de 70 pines serie DF40.
Tamaño	35.5 × 25.0 × 3.9 mm
Voltaje de Entrada DC	3.15 a 4.5 V
Entradas / Salidas	40 GPIO[21] de uso general que puede configurarse como: • Tarjeta SD: 1 Interfaz. • UART[22]: 2 Controladores. • I²C[23]: 2 controladores. • SPI[24]: 1 controlador con 2 selecciones de chip • I2S[25]: 1 controlador. • GPIO: 14 adicionales.

[14] LPDDR3 (*Low Power Dance Dance Revolution)*: Memoria de baja potencia que ofrece una mayor velocidad de datos, mayor ancho de banda y eficiencia de energía.
[15] POP (*Package-on-Package*): Paquetes en paquetes.
[16] eMMC: Es una memoria flash basada en NAND como las de los pendrive o las de las tarjetas SD de las cámaras.
[17] TI: Texas Instruments.
[18] SNB9024 PMIC: Circuito integrado de administración de energía.
[19] u.FL: Conector de antena que se utiliza en aplicaciones donde el espacio es reducido.
[20] Hirose: Conector de 70 pines utilizado en el módulo de cómputo Intel Edison
[21] GPIO (*General Purpose Input/Output*): Es un pin genérico en un chip, cuyo comportamiento se puede controlar por el usuario en tiempo de ejecución.
[22] UART (*Universal Asynchronous Receiver-Transmitter*): Es el dispositivo que controla los puertos y dispositivos serie.
[23] I²C (*Inter-Integrated Circuit*): Es un bus serie de datos.
[24] SPI (*Serial Peripheral Interface*): Es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.
[25] I2S (*Integrated Interchip Sound*): Es un estándar eléctrico de bus serial usado para interconectar circuitos de audio digital.

USB[26] 2.0	1 controlador OTG[27]
Clocks	19.2 MHz, 32 kHz

En la Figura 1.5 se muestra el diagrama de bloques básico del módulo Intel Edison.

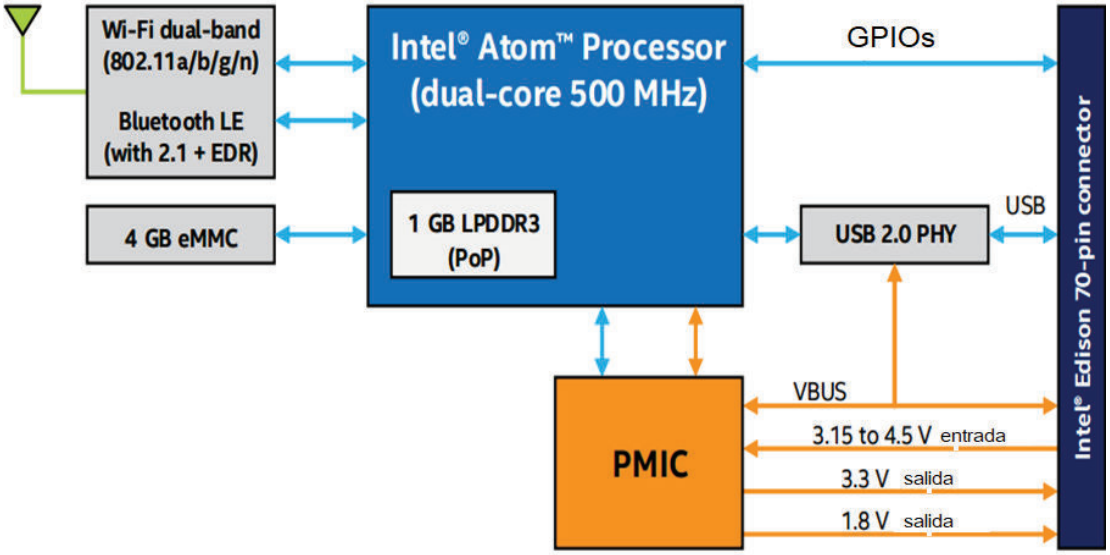
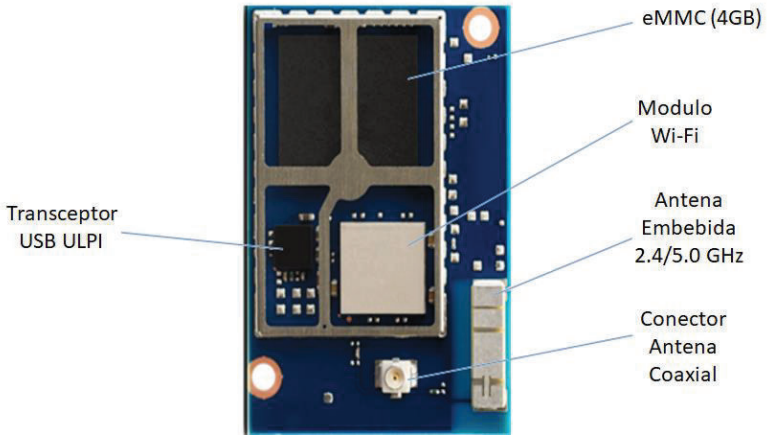


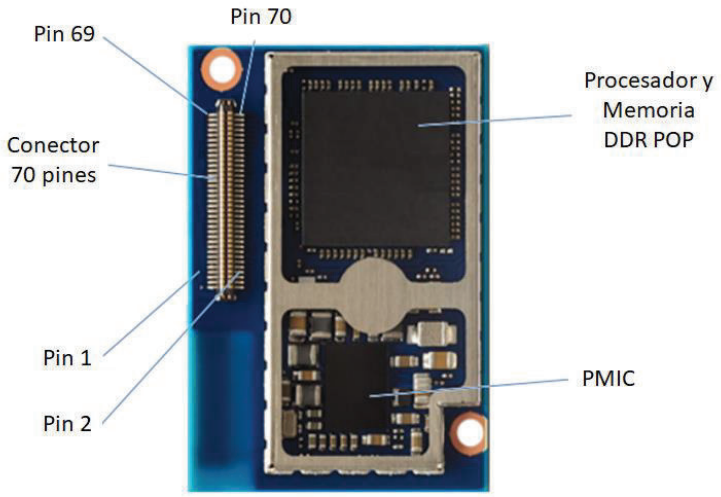
Figura 1.5. Diagrama de bloques Intel Edison [17].

En la Figura 1.6 se muestra la vista superior y vista inferior respectivamente del módulo Intel Edison. Se especifican además las partes que están descritas en la Figura 1.5. Diagrama de bloques Intel Edison.



a) Vista Superior

[26] USB (*Universal Serial Bus*): Dispositivo de almacenamiento de datos.
[27] OTG (*On-The-Go*): Es una tecnología que permite a los dispositivos con puertos USB más flexibilidad a la hora de gestionar la conexión.



b) Vista Inferior

Figura 1.6. Módulo Intel Edison [17].

1.6.2 FIRMWARE DEL MÓDULO INTEL EDISON

El firmware es un conjunto de instrucciones de un programa informático que se encuentra registrado en una memoria ROM, flash o similar. Estas instrucciones fijan una lógica primaria que ejerce el control de los circuitos de alguna clase de dispositivo [18].

El firmware predeterminado en el módulo Intel Edison es una versión de Linux construida con el proyecto Yocto [19].

Yocto es un proyecto de colaboración de código abierto que proporciona plantillas, herramientas y métodos para ayudar a crear sistemas personalizados basados en Linux para productos embebidos. Utiliza el gestor de paquetes OPKG[28] para su configuración y actualización. [20].

Para instalar el firmware en el módulo Intel Edison se puede hacer uso del sistema operativo Windows, Linux o Mac OS X. Dependiendo del ambiente de trabajo que se utilice se deberá descargar los programas gratuitos en la página oficial del módulo Intel Edison.

[28] OPKG (*Open PacKaGe management*): Sistema de gestión de paquetes ligero.

1.6.3 ENTORNOS DE DESARROLLO INTEGRADO

Entre los entornos de desarrollo con los que puede trabajar el módulo Intel Edison se tiene:

- *Arduino*: Es una interfaz intuitiva con códigos de una variedad de fabricantes.
- *JavaScript y Node.js*: Es una tecnología basada en el motor de JavaScript V8 para permitir la ejecución de programas hechos en JavaScript en un ámbito independiente del navegador [21].
- *C/C++*: Lenguaje de programación que permite la manipulación de objetos.
- *Python*: Es un lenguaje de programación orientado a objetos. Puede realizar cualquier tipo de programa, desde aplicaciones de Windows, servidores de red o páginas web [22].

1.6.4 TARJETAS DE DESARROLLO

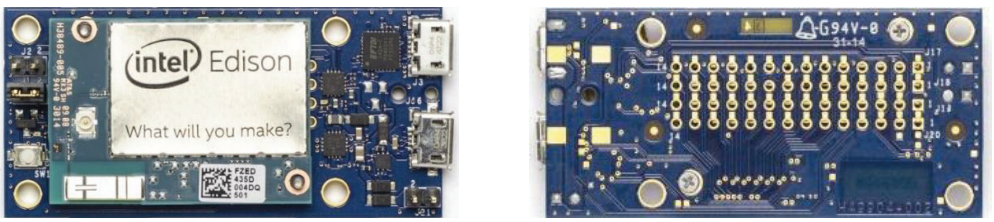
El módulo Intel Edison puede conectarse a varias placas con el fin de poder interactuar con el usuario final. Estas placas se denominan tarjetas de desarrollo.

Las tarjetas de desarrollo que actualmente se utilizan junto con el módulo son:

- Tarjeta Arduino.
- Tarjeta Intel Edison Breakout

1.6.4.1 Tarjeta Intel Breakout [23]

La tarjeta de desarrollo Intel Breakout fue diseñada para la creación de prototipos con hardware y software libre. Su tamaño es del doble del área del módulo Intel Edison. La Figura 1.7, muestra la vista superior e inferior de la tarjeta conectada al módulo Intel Edison.



a) Vista Superior b) Vista Inferior

Figura 1.7. Tarjeta Intel Breakout [24].

La Figura 1.8, muestra el diagrama en bloques de la tarjeta Intel Breakout.

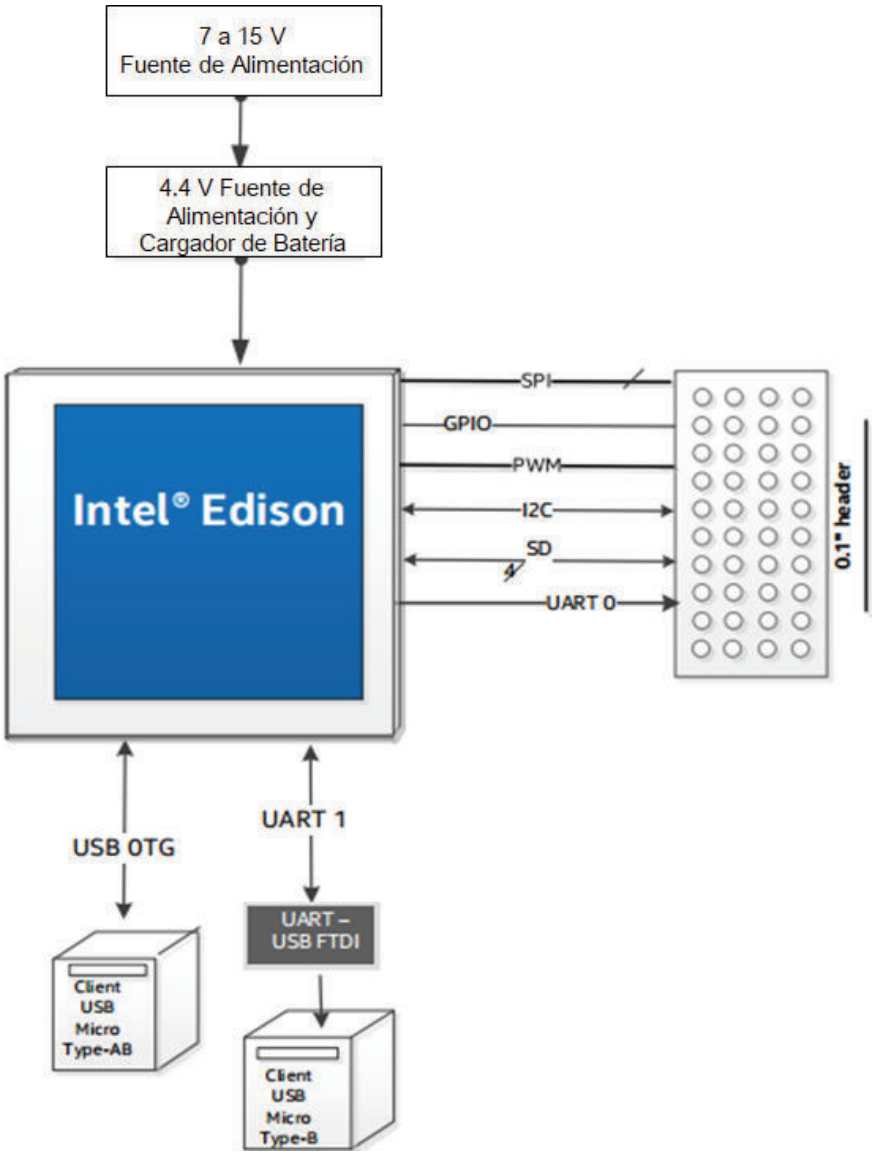
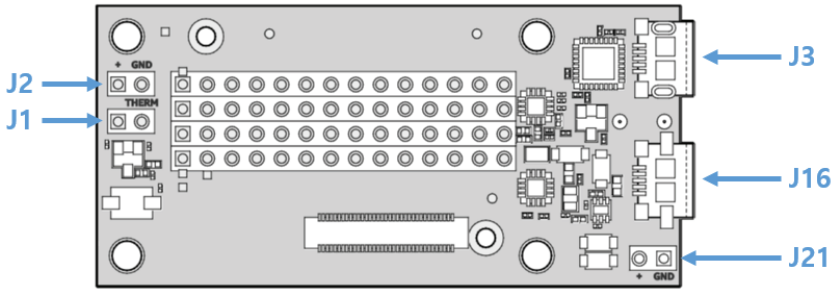


Figura 1.8. Diagrama de bloques tarjeta Intel Breakout [23].

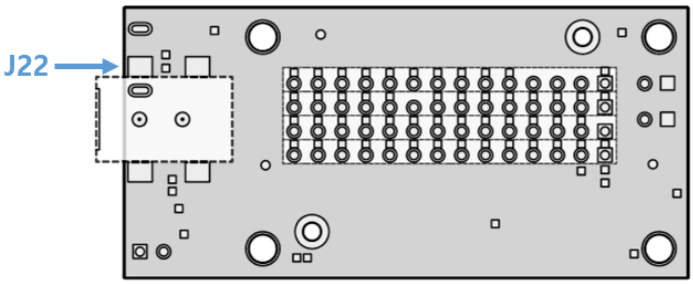
El diagrama de bloques de la tarjeta Intel Breakout está compuesta por la fuente de alimentación DC, cargador de batería, interruptor de alimentación USB OTG, puerto UART 1, puerto USB OTG y puertos de entrada/salida.

1.6.4.1.1 Jumpers [23]

Un jumper o puente es un elemento que permite cerrar el circuito electrónico del que forma parte dos conexiones. Los jumpers (J) de la tarjeta Intel Breakout se muestran en la Figura 1.9.



a) Parte Superior



b) Parte Inferior

Figura 1.9. Jumper de Tarjeta Intel Breakout [25].

Los jumpers más utilizados se describen a continuación:

- *J2*: Conector de batería. La tarjeta Intel Breakout puede funcionar con una batería de iones de litio recargable (3.7V). Cuando se aplica energía a través de *J21*, *J22* o *J3*, la batería de iones de litio se recarga. Sin embargo se debe tener precaución debido a que la tarjeta no dispone de un circuito para controlar el nivel de recarga de la batería de iones de litio.
- *J3*: Convertidor micro-USB FTDI[29] (*Future Technology Devices International*) serie a USB. Se utiliza para realizar una conexión serie a través de la terminal.
- *J16*: Puerto compatible con USB, micro AB, y OTG. Si se conecta un cable micro A en este puerto, la tarjeta Breakout funcionará como host; Si se conecta un cable micro B en este puerto, la tarjeta Breakout funcionara como un dispositivo USB.

[29] FTDI: Es una empresa privada escocesa de dispositivos semiconductores, especializada en tecnología Universal Serial Bus.

- *Modo Host*: El módulo Intel Edison se comporta como una computadora. Un periférico USB (mouse, teclado o webcam) puede ser conectado a la tarjeta. En este modo debe suministrarse alimentación externa a través de *J21* o *J22.*
- *Modo Dispositivo*: El módulo Intel Edison se comporta como un periférico de PC en la computadora. Este modo se utiliza para:
  - Suministrar tensión de 5V a la tarjeta.
  - Leer/escribir en la memoria flash integrada desde la computadora como una unidad de disco.
  - Programar la tarjeta Intel Edison a través del USB mediante el Arduino IDE[30] (*Integrated Development Environment*).
  - Programar la tarjeta Intel Edison utilizando los IDEs Intel® XDK e Intel® System Studio IoT Edition.

- *J21*: Entrada principal de alimentación de 7 a 15 Vdc. La polaridad se muestra en la Figura 1.9 a).
- *J22*: Conector de alimentación (no instalado). Se puede utilizar un conector de barril de 2.5mm que se observa en la Figura 1.10 y soldarlo en la parte inferior de la tarjeta Intel Breakout que se muestra en la Figura 1.9 b). La tensión de entrada es de 7 a 15 Vdc.

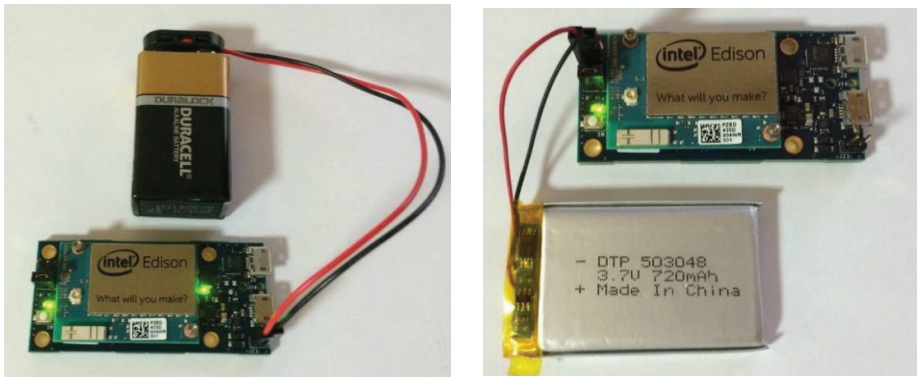
Figura 1.10. Conector Tipo Barril [25].

1.6.4.1.2 Fuente de alimentación [23]

El módulo Intel Edison es un dispositivo de baja potencia, su voltaje de alimentación está en el rango de 3.15 a 4.5 V. En cuanto a corriente, no consume más de 200mA. Durante la transmisión Wi-Fi se pueden producir picos de corta duración de 600mA. La alimentación puede ser suministrada de tres maneras:

[30] IDE: Es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

- Baterías: El dispositivo Intel Edison puede funcionar usando una batería de 9V conectado al jumper *J21* como se explicó en la sección 1.6.4.1.1. La corriente suministrada debe ser superior a los 150mAh para el correcto funcionamiento de la placa. También se puede utilizar una batería de iones de litio conectado en el jumper *J2* como se observa en la Figura 1.11 a).
- Cable USB: Cuando se configura el módulo Intel Edison en modo dispositivo visto en la sección 1.6.4.1.1, la alimentación se suministra mediante un cable Micro-USB tipo B conectado a la PC.
- Fuente externa: Si el módulo Intel Edison está configurado en modo host, la alimentación se suministra por el jumper *J22* como se explicó en la sección 1.6.4.1.1. Se debe soldar el conector que se observa en la Figura 1.11 b). El voltaje de entrada está en el rango de 7 a 15 V.



a) Baterías

b) Cable USB

Figura 1.11. Formas de alimentación DC [25].

1.6.4.1.3 Polarización [23]

En la Figura 1.12 se presenta el diagrama de polarización de la tarjeta Intel Breakout.

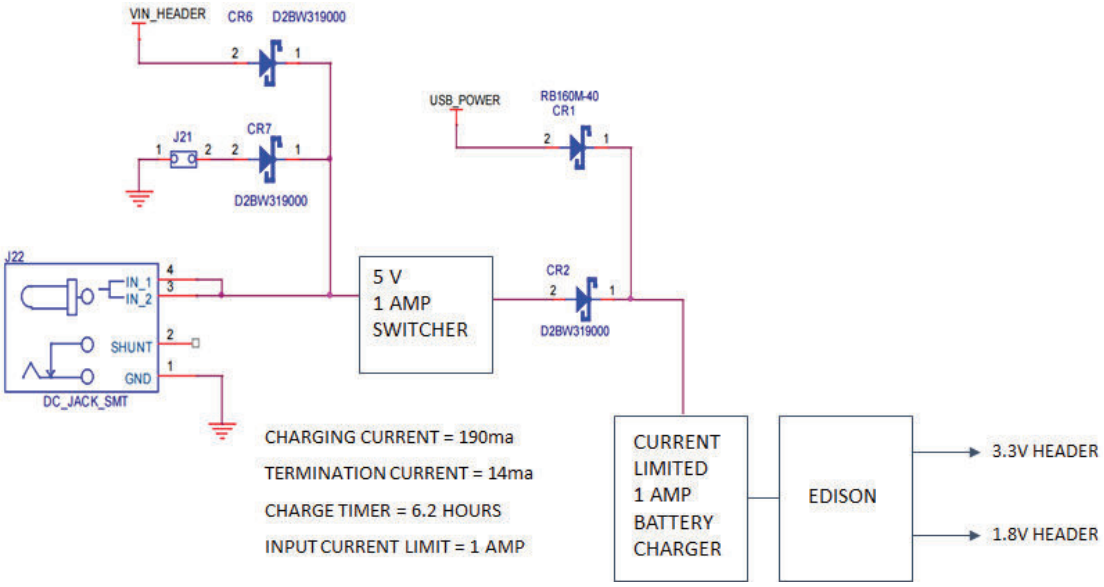


Figura 1.12. Distribución de Polarización de la Tarjeta Intel Breakout [23].

Cuando se conecta un adaptador de alimentación externa, el voltaje se dirige a un convertidor DC-DC y la tensión disminuye a 5V. Esta energía va a un IC (*Integrated Circuit*, Circuito Integrado), que limita la tensión de salida a 4.4V con el fin de proveer un rango seguro de potencia V_{SYS}[31] (Voltaje del sistema) para el módulo Intel Edison. La gama de potencias V_{SYS} es de 3.15 a 4.5 V. Esto permite al módulo Intel Edison funcionar con una batería estándar de iones de litio. El IC del módulo está configurado para limitar la corriente a 1A, programado para cargar a 190mA y cargar baterías estándar de iones de litio con voltaje máximo de carga de 4.2V.

La tarjeta Intel Breakout posee un circuito integrado de carga que limita la entrada de corriente y se apaga cuando el sistema esta con exceso de temperatura. La batería debe tener al menos una capacidad de 150mAh para su correcto funcionamiento.

1.6.4.1.3.1 Voltaje de arranque - Señal DCIN [23]

DCIN es una señal que indica si la tarjeta Intel Breakout se encuentra alimentado desde una batería o desde una fuente de alimentación externa. Además establece el nivel de voltaje de arranque requerido en V_{SYS}. Para que el arranque se produzca, el voltaje debe subir de 2.5 a 3.5 V en 10ms; De lo contrario se aborta el arranque.

[31] V_{SYS}: Potencia de entrada del sistema en la tarjeta Intel Edison.

El voltaje mínimo absoluto para asegurar la funcionalidad Wi-Fi y Bluetooth es 3.15V.

1.6.4.1.4 Pulsadores [23]

Un botón o pulsador es un dispositivo utilizado para realizar cierta función. Los botones son de diversas formas y tamaños y se encuentran en todo tipo de dispositivos. La tarjeta Intel Breakout dispone de un botón de encendido, el cual se muestra en la Figura 1.13.

Figura 1.13. Botón de encendido [25].

El botón de encendido (SW1) está configurado por el software. Si se mantiene pulsado el botón de encendido producirá resultados diferentes en función del estado actual del módulo Intel Edison y la duración de pulsado.

- Cuando el dispositivo Intel Edison está completamente apagado, manteniendo pulsado el botón de encendido durante tres segundos hará encender el dispositivo y arrancará el módulo de cómputo Intel Edison.
- Cuando el dispositivo Intel Edison está en funcionamiento, manteniendo pulsado el botón de encendido durante más de dos segundos, pero menos de siete segundos hará que el dispositivo entre en modo AP.
- Cuando el dispositivo Intel Edison está en funcionamiento, manteniendo pulsado el botón de encendido durante diez segundos o más hará que el módulo de cómputo Intel Edison se apague.

1.6.4.1.5 Leds [23]

Un LED[32] es un componente optoelectrónico que es usado como indicador en muchos dispositivos. La tarjeta Intel Breakout dispone de dos LEDs. En la Figura 1.14 se muestra la ubicación de cada uno de ellos.

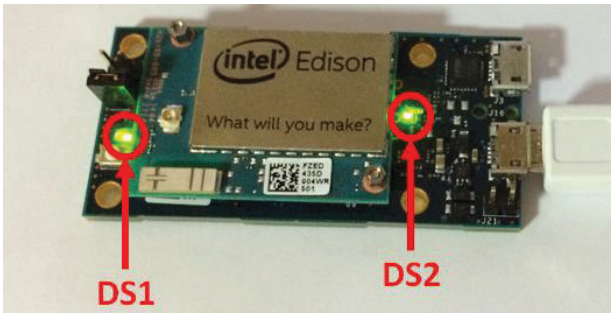


Figura 1.14. Leds [25].

- *Led de restablecimiento (DS1)*: Se encenderá cuando el procesador Intel Edison está en marcha y en restablecimiento.
- *Led de carga (DS2)*: Se enciende cuando se está cargando una batería previamente conectada.

1.6.4.1.6 Cables de conexión

Un cable es un conductor de electricidad que generalmente está recubierto de un material aislante o protector. Su grosor dependerá del nivel de tensión al que este expuesto. Son necesarios dos cables para conectar la tarjeta Intel Breakout a la PC. Los cables deben contener en un extremo un conector USB y en el otro, un conector micro USB Tipo B como se muestra en la Figura 1.15.

Figura 1.15. Cable de conexión.

[32] LED (*Light-Emitting Diode*): Diodo Emisor de Luz.

1.6.4.1.7 Baterías alcalinas

Son fuentes de energía que proporcionan voltaje de corriente continua. Una batería es un conjunto de varias celdas electroquímicas individuales en el que la energía de la reacción química se trasforma en corriente eléctrica. Como se mencionó en la sección 1.6.4.1.2 es necesario utilizar una batería de 9V.

1.6.4.1.8 Manipulación [23]

Al acoplar el módulo Intel Edison a la tarjeta Intel Breakout se debe manipular el modulo por los bordes del PCB (*Printed Circuit Board*, Plaqueta de Circuito Impreso*)* y ejercer presión únicamente sobre el conector de 70 pines y en la esquina de la placa como se muestra en la Figura 1.16.

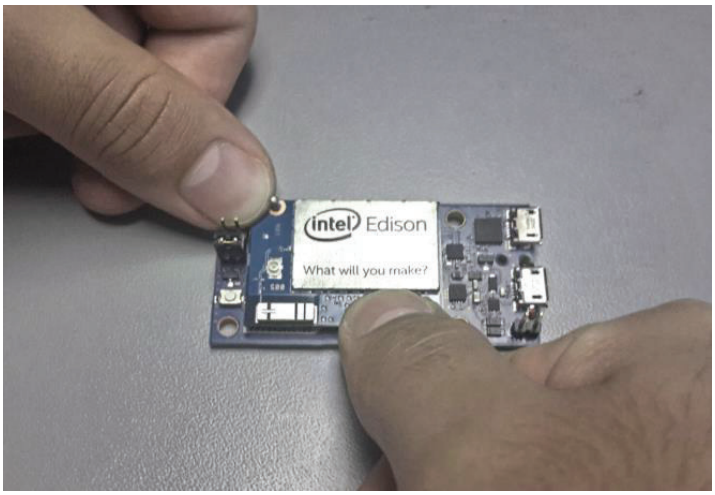


Figura 1.16. Insertando el Módulo Intel Edison en la Tarjeta Intel Breakout [23].

1.7 SISTEMAS OPERATIVOS [26]

Un OS (*Operating System*, Sistema Operativo) es el software más importante de un ordenador o dispositivo móvil. Para que funcionen los programas y aplicaciones, cada ordenador de uso general debe tener un sistema operativo. El OS también es responsable de la seguridad, asegurándose de que los usuarios no autorizados no tengan acceso al sistema.

Las funciones principales que cumple cualquier sistema operativo son: inicializar el hardware del ordenador, administrar los recursos de memoria utilizada en cada

aplicación, gestionar dispositivos periféricos en base a rutinas básicas, controlar la ejecución de aplicaciones y mantener la integridad del sistema.

1.7.1 SISTEMA OPERATIVO LINUX [27]

Linux es un sistema operativo de software libre. Fue desarrollado por Linus Torvalds en 1991 cuando era estudiante de informática de la Universidad de Helsinki (Finlandia). El sistema operativo Linux está conformado por un núcleo denominado Linux y un conjunto de herramientas GNU (*GNU is not Unix*). Su nombre formal es GNU/Linux debido a que el sistema se distribuye bajo licencia pública. Proporciona además una interfaz de comandos y una interfaz gráfica.

1.7.2 SISTEMA OPERATIVO WINDOWS [28]

Fue desarrollado por Microsoft desde 1981. La primera versión de Microsoft Windows fue lanzada en 1985 como una extensión gráfica del MS-DOS (*MicroSoft Disk Operating System*). Años después, se desarrolla la primera versión independiente del MS-DOS denominada Windows 95, la cual esta provista de una interfaz gráfica basada en un prototipo de ventanas lo que le hace más amigable al usuario.

La última versión de este sistema operativo fue lanzada en el 2015 denominada Windows 10. Esta versión cuenta con funcionalidades para interactuar con diferentes dispositivos tales como computadores, teléfonos móviles y Tabletas.

Windows es el sistema operativo más utilizado debido a su fácil uso, variedad de programas y configuraciones que dispone su sistema.

1.7.3 SISTEMA OPERATIVO ANDROID [29]

Android es el sistema operativo para dispositivos móviles más utilizado en el mundo. Fue desarrollado inicialmente por Android Inc., y posteriormente comprado por la compañía de Google en el 2005. Este sistema operativo está basado en GNU Linux. Cuenta además con su propia plataforma de descarga llamada Google Play, la cual permite descargar diferente tipo de contenido entre aplicaciones, juegos y herramientas varias.

En la actualidad se han desarrollado 14 versiones de este sistema operativo los cuales se mencionan a continuación [30]:

- Android 1.0 (Apple Pie).
- Android 1.1 (Banana Bread).
- Android 1.5 (Cupcake).
- Android 1.6 (Donut).
- Android 2.0/2.1 (Eclair).
- Android 2.2 (Froyo).
- Android 2.3 (Gingerbread).
- Android 3.0 (Honeycomb).
- Android 4.0 (Ice Cream Sandwich).
- Android 4.1 (Jelly Bean).
- Android 4.4 (KitKat).
- Android 5.0 (Lollipop).
- Android 6.0 (Marshmallow).
- Android 7.0 (Nougat).

1.7.3.1 Android con soporte IPv6

El protocolo IPv6 como se vio en la sección 1.5, es soportado desde la versión Android 5.0 (Lollipop). Desde esta versión, IPv6 viene instalado por defecto en el sistema operativo [31].

Aunque IPv6 implementa la configuración automática sin estado revisado en 1.5.1, para brindar un mayor control en cuanto a las asignaciones de direcciones y amplitud en la configuración de servicios de red se hace uso de DHCPv6 (*Dynamic Host Configuration Protocol version 6*, Protocolo de Configuración de Host Dinámico versión 6).

En la actualidad DHCPv6 no está soportado en el sistema operativo Android, por lo cual es necesario implementar otro mecanismo para que los dispositivos móviles pueden conectarse a una red IPv6 automáticamente.

1.7.4 COMPARACIÓN ENTRE SISTEMAS OPERATIVOS

La Tabla 1.3 presenta un cuadro comparativo de los sistemas operativos revisados.

Tabla 1.3. Comparación de los sistemas operativos.

Sistema Operativo	Características
[29]	• Código libre. • Multi-tarea y multi-usuario. • Utilizado en la mayoría de dispositivos inteligentes. • Aplicaciones gratuitas y de pago. • Multimedia. • Seguro y confiable frente a virus.
[27]	• Código y software libre. • Multi-tarea, multi-usuario y multi-plataforma. • Buena velocidad de procesamiento. • Diferentes tipos de distribuciones. • Seguro y confiable frente a virus. • Administración por línea de comandos.
[28]	• Software propietario. • Multi-tarea y multi-usuario. • Gran cantidad de software desarrollado. • Mayor velocidad de procesamiento. • Vulnerable frente a virus. • Actualizaciones constantes.

Como se observa en la Tabla anterior, Android comparte similares características junto con Linux, sin embargo, el primero se utiliza en su mayoría en dispositivos móviles, mientras que el segundo, hace uso de computadoras para su uso. El cambio, sistema operativo Windows es el más desarrollado y más usado en PCs en la actualidad.

1.8 CÁMARAS WEB [32]

La cámara web (webcam) es una pequeña cámara digital de bajo costo que capta imágenes y las transmite a través de un ordenador. Se originó en 1991 en el

departamento de informática de la Universidad de Cambridge (Inglaterra). Las primeras webcam se comercializaron en 1994.

En general, una cámara web está formada por un lente, un sensor de imagen y la circuitería necesaria para manejarla. Algunas cámaras utilizan un sensor CMOS (*Complementary Metal-Oxide-Semiconductor*) integrado en la placa electrónica como un chip de silicio para ahorrar costos y espacio. Las webcam se conectan al ordenador por medio de conexiones USB. En la Figura 1.17 se observa una cámara web de la empresa Logitech.

Figura 1.17. Cámara web [32].

1.8.1 ESTÁNDAR UVC [33]

El estándar UVC (*USB video device class*, Clase de Dispositivo de Video USB) describe los dispositivos USB que no precisan ningún controlador específico para funcionar, puesto que utilizan un controlador UVC estándar válido para todos los fabricantes. El usuario solo tiene que conectar la cámara al ordenador para poder acceder automáticamente a su contenido [34].

Las cámaras web fueron los primeros dispositivos en soportar el estándar UVC, luego se sumaron las videocámaras digitales, conversores de vídeo analógicos, sintonizadores de televisión digital y cámaras de imagen fija que soportan la transmisión de vídeo tanto para la entrada y salida de video.

El estándar UVC 1.1 que incorpora el dispositivo, garantiza su funcionamiento con cualquier sistema operativo y plataforma (Windows, Linux, Mac etc.). Basta con enchufar la cámara para obtener las imágenes, sin necesidad de instalar ningún controlador [34].

CAPÍTULO II

2 IMPLEMENTACIÓN DEL PROTOTIPO PARA LA TRANSMISIÓN DE VIDEO SOBRE IPv6

En este capítulo se realiza la implementación del prototipo para la transmisión de video sobre IPv6 utilizando el módulo Intel Edison, la tarjeta Intel Breakout, una cámara web, una batería de 9V, un dispositivo móvil, una laptop, una PC y un sistema que simula la red de transporte IPv6. En la Figura 2.1 se presenta un diagrama esquemático referencial del prototipo a implementarse.

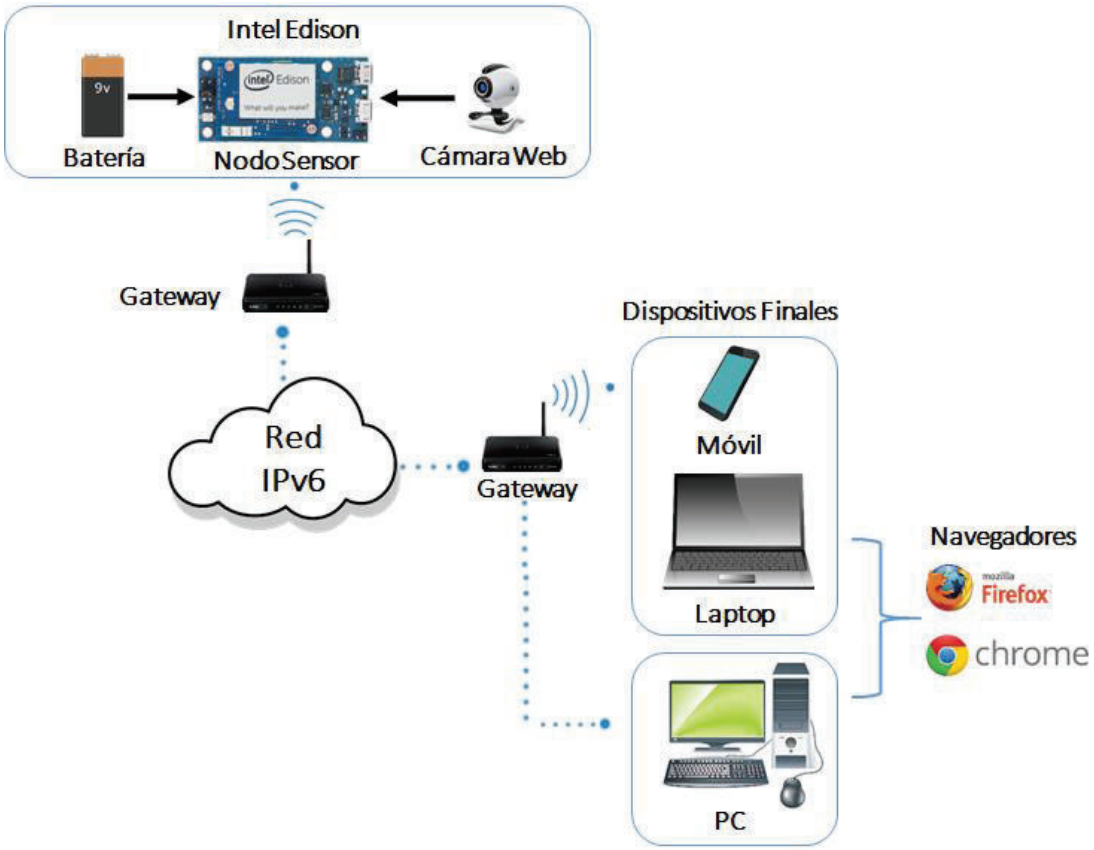


Figura 2.1. Diagrama esquemático del prototipo a implementarse.

Inicialmente se implementa un sistema para simular la red de transporte IPv6, luego se diseña el programa en el nodo sensor para transmitir video sobre la red IPv6 y finalmente se integra todo el sistema para comprobar su funcionamiento.

2.1 IMPLEMENTACIÓN Y CONFIGURACIÓN DE LA RED DE TRANSPORTE IPv6

2.1.1 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DE LA RED

Los siguientes requerimientos son los que se van a cumplir en la implementación del sistema que simulará la red de transporte IPv6.

- La red debe ser implementada en base a una topología tipo Bus o lineal.
- Debe permitir la conectividad entre el host local y los equipos remotos.
- La red debe ser capaz de asignar automáticamente direcciones IPv6 a los dispositivos que se conectan a esta.

La red a implementarse se muestra en la Figura 2.2.

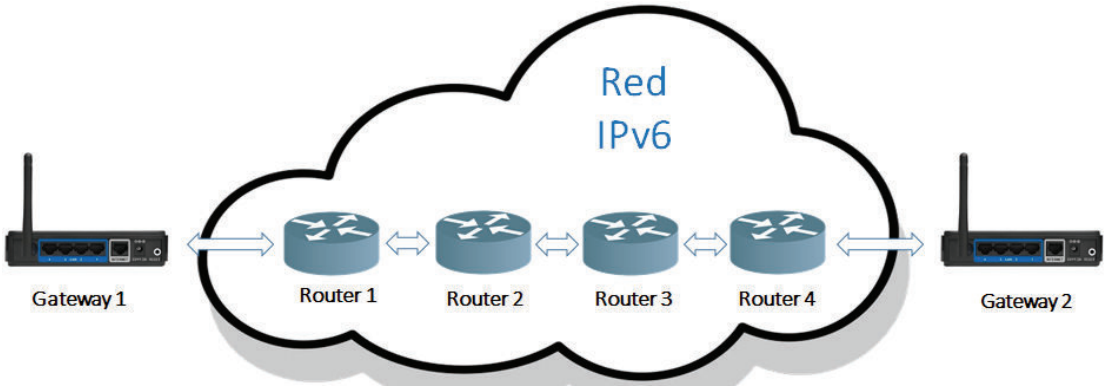


Figura 2.2. Red a implementarse.

Para el desarrollo del sistema que simulará la red de transporte IPv6 se dispondrá de cuatro routers, dos Gateway inalámbricos y cables, directos y cruzados.

2.1.1.1 Router

Se hará uso del router Cisco 1841 que se ilustra en la Figura 2.3.

a) Parte delantera

b) Parte trasera

Figura 2.3. Router Cisco 1841 [35].

Este router posee las siguientes características: dos puertos WAN Ethernet, cuatro puertos 10/100BASE-T[33], tarjetas de interfaz WIC (sólo datos), VWIC (voz y datos), HWIC y AIM (módulos de integración avanzada), un puerto para backup sobre dial RDSI (Red Digital de Servicios Integrados), un puerto USB 1.1, 128 MB de RAM y 32 MB de FLASH [35].

2.1.1.2 PuTTY [36]

Para la configuración de los routers Cisco 1841 se utilizará el emulador PuTTY. Su interfaz gráfica se presenta en la Figura 2.4.

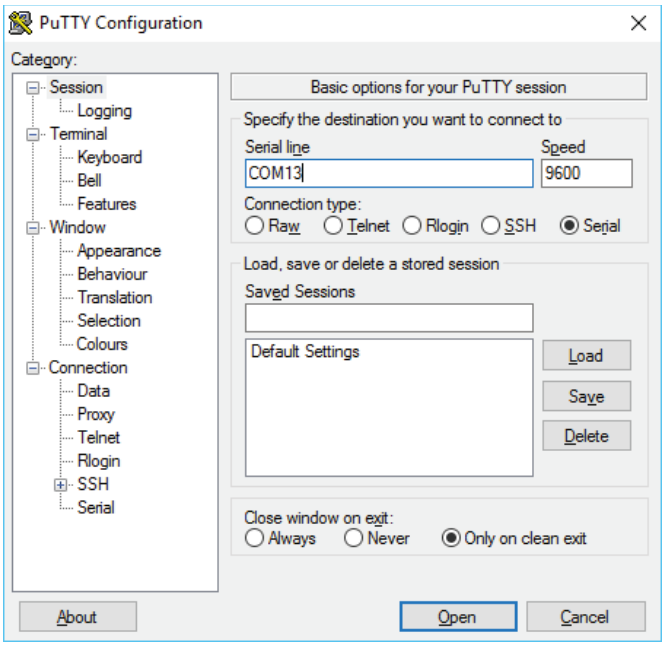


Figura 2.4. Interfaz gráfica del emulador PuTTY.

[33] 100BASE-T: Estándar Fast Ethernet de 100 Mbit/s.

PuTTY es una interfaz sencilla y manejable. Está disponible de forma gratuita para Windows y tiene soporte para IPv6.

2.1.1.3 Gateway

Para tener una conexión inalámbrica en la red IPv6, se utiliza el router D-Link DIR-600 que se muestra en la Figura 2.5.

a) Parte delantera

b) Parte trasera

Figura 2.5. Router D-Link DIR-600 [37].

Entre las características principales que ofrece el router DIR-600 se tiene: un puerto WAN 10/100Base-TX, cuatro puertos LAN 10/100Base-TX, una antena con tecnología Wireless N para obtener la mejor cobertura, velocidad inalámbrica de hasta 150 Mbps, certificación de la especificación WPS (*Wi-Fi Protected Setup*, Configuracion segura de Wi-Fi), motor de QoS opcional para priorizar la voz y los vídeos, configuración fácil con el asistente de D-Link Click'n Connect disponible en 18 idiomas [37].

2.1.2 IMPLEMENTACIÓN DE LA RED DE TRANSPORTE IPv6

Para implementar la red de transporte IPv6 se procede a conectar los cuatro routers utilizando cables cruzados. En la Figura 2.6 se observa el esquema de conexión entre los routers.

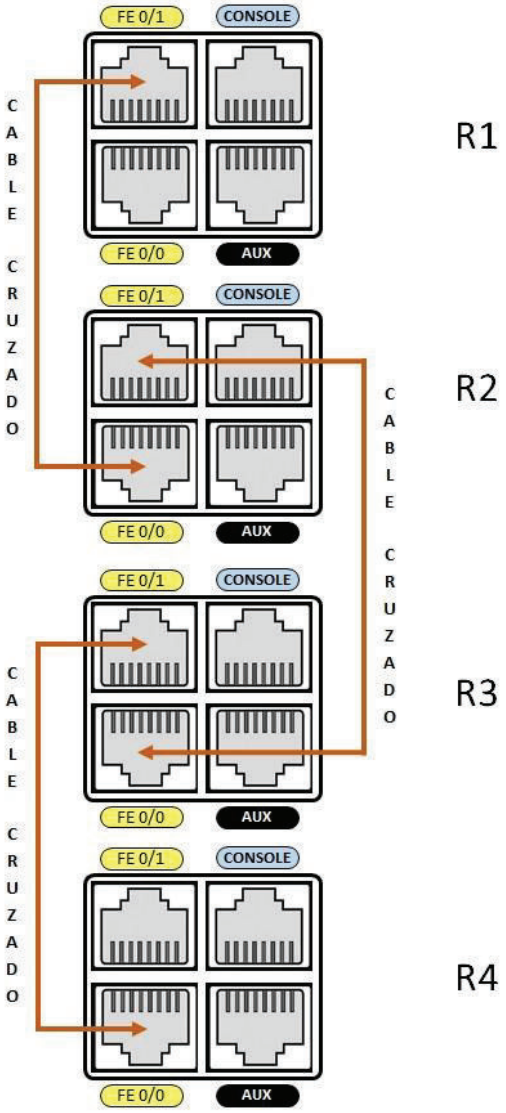


Figura 2.6. Esquema de conexión de los routers.

Los routers se conectan de la siguiente manera:

- R1 se conecta a R2 desde el puerto FE 0/1 al puerto FE 0/0 respectivamente.
- R2 se conecta a R3 desde el puerto FE 0/1 al puerto FE 0/0 respectivamente.
- R3 se conecta a R4 desde el puerto FE 0/1 al puerto FE 0/0 respectivamente.

Desde el puerto FE 0/0 de R1 se conecta el Gateway 1 y desde el puerto FE 0/1 de R4 se conecta el Gateway 2 utilizando un cable directo. En la Figura 2.7 se observa el esquema de conexión de los Gateways.

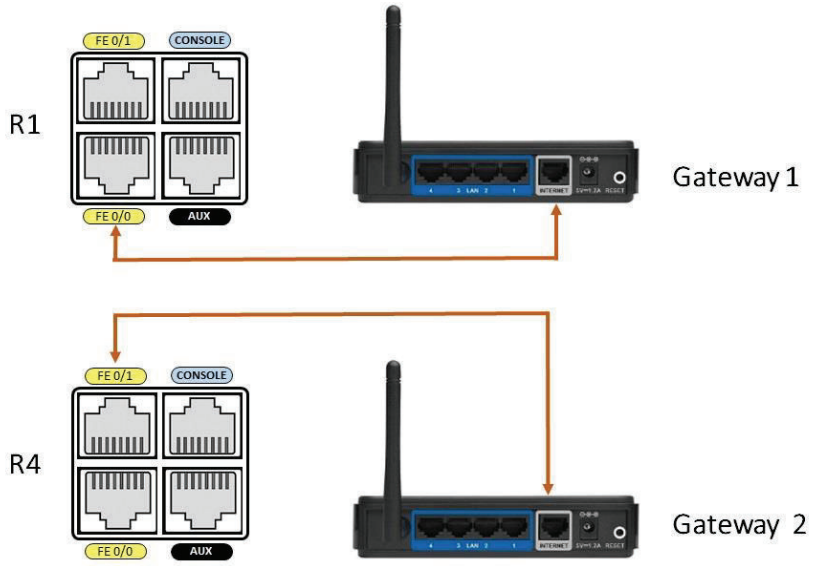


Figura 2.7. Esquema de conexión de los Gateways.

2.1.3 CONFIGURACIÓN DE LA RED DE TRANSPORTE IPv6

En la Figura 2.8 se muestra el diagrama de red a ser implementado.

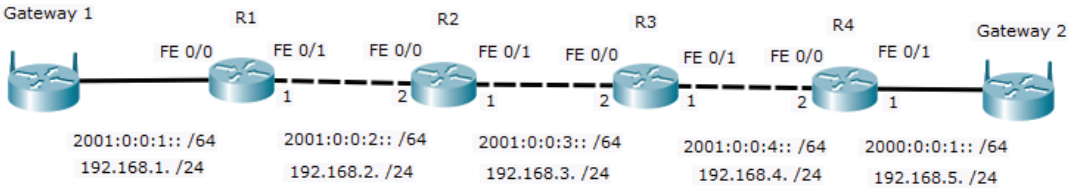


Figura 2.8. Diagrama de red a ser implementado.

En la Tabla 2.1 se muestra el direccionamiento asignado.

Tabla 2.1. Direccionamiento de red.

Dispositivo	Interface	Dirección IPv6	Prefijo	Dirección IPv4	Máscara
R1	FE 0/0	2001:0:0:1::1	64	192.168.1.1	24
	FE 0/1	2001:0:0:2::1	64	192.168.2.1	24
R2	FE 0/0	2001:0:0:2::2	64	192.168.2.2	24
	FE 0/1	2001:0:0:3::1	64	192.168.3.1	24
R3	FE 0/0	2001:0:0:3::2	64	192.168.3.2	24
	FE 0/1	2001:0:0:4::1	64	192.168.4.1	24
R4	FE 0/0	2001:0:0:4::2	64	192.168.4.2	24
	FE 0/1	2000:1::1	64	192.168.5.1	24

La implementación de la red se basa principalmente en la configuración del protocolo OSPFv3[34] y configuración del servidor DHCPv6, por lo que a continuación se detallan los segmentos de código más importantes. El código completo se encuentra en el ANEXO A.

2.1.3.1 Configuración OSPFv3

OSPFv3 es un protocolo de enrutamiento por estado del enlace que trabaja con direcciones IPv6. No posee soporte para direcciones IPv4, razón por la cual se implementa también el protocolo OSPFv2[35] para poder adaptar la plataforma Intel Edison a la red. La configuración se realiza en los cuatro routers.

Desde el modo de configuración se habilita el ruteo para IPv6 con el comando `ipv6 unicast-routing`, luego se ingresa al proceso OSPF para asignar un ID al router. Por comodidad se asigna el ID según el número del router. El Código 2.1, muestra la configuración global de OSPFv3.

```
R1(config)#ipv6 unicast-routing
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
```

Código 2.1. Configuración global OSPFv3.

El proceso OSPFv3 se hace a nivel de interfaz con el comando `ipv6 ospf IDENTIFICADOR area #ÁREA`, donde:

- *IDENTIFICADOR*: Puede ser un nombre o un número. En este caso es 1,
- *#ÁREA:* Área a la que se asocia la red. En este caso es 0.

Dentro de cada interfaz de R1 se configura una dirección IPv6, luego se digita el comando anteriormente explicado como se indica en el Código 2.2. Este proceso se debe realizar en los cuatro routers.

```
R1(config)# interface fastEthernet 0/0
R1(config-if)#ipv6 address 2001:0:0:1::1/64
R1(config-if)#ipv6 ospf 1 area 0
R1(config-if)#no shutdown
```

Código 2.2. Configuración de la interfaz.

[34] OSPFv3 (*Open Shortest Path First version 3*): Protocolo de enrutamiento OSPF para IPv6.
[35] OSPFv2 (*Open Shortest Path First version 2*): Protocolo de enrutamiento OSPF para IPv4.

Mediante el comando `show ipv6 ospf neighbor` que se ilustra en la Figura 2.9 se puede verificar la conectividad de la red. Donde:

- *Neighbor ID:* Es el número del router más próximo de la red.
- *Interface:* Es la interfaz por donde se llega al siguiente router.

```
R1#show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
2.2.2.2           1   FULL/DR         00:00:33    3               FastEthernet0/1

R2#show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
3.3.3.3           1   FULL/DR         00:00:37    3               FastEthernet0/1
1.1.1.1           1   FULL/BDR        00:00:33    4               FastEthernet0/0

R3#show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
4.4.4.4           1   FULL/BDR        00:00:39    3               FastEthernet0/1
2.2.2.2           1   FULL/BDR        00:00:35    4               FastEthernet0/0

R4#show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
3.3.3.3           1   FULL/DR         00:00:34    4               FastEthernet0/0
```

Figura 2.9. Comando para verificar conectividad.

2.1.3.2 Configuración de servidor DHCPv6

Un servidor DHCPv6 asigna direcciones IPv6 automáticamente y otras configuraciones a los dispositivos de una red.

Se realizará la configuración del servidor DHCPv6 en los routers de borde (R1 y R4). Además se configura el servidor DHCPv4 debido al inconveniente que poseen los dispositivos móviles revisado en la sección 1.7.3.1.

Se configura el router para asignar direcciones estáticas mediante la asociación de la dirección IPv6 con el identificador único de DHCP (DUID[36]). El valor DUID se observa digitando el comando que se muestra en el Código 2.3

```
R1#show ipv6 dhcp
This device's DHCPv6 unique identifier (DUID): 000300010015FAC98D54
```

Código 2.3. Valor DUID.

[36] DUID: Es utilizado por un cliente para obtener una dirección IP de un servidor DHCPv6.

Desde el modo de configuración se procede a digitar el comando `ipv6 dhcp pool Pool-Name`, donde: *Pool*-Name será el nombre del pool de direcciones IPv6, luego se utiliza el valor DUID junto con el comando que se muestra en el Código 2.4. para asignar las direcciones IPv6.

```
R1(config)#ipv6 dhcp pool EDISON
R1(config-dhcpv6)#prefix-delegation 2001:0:0:1::23F6:33BA/64
000300010015FAC98D54
R1(config-dhcpv6)#prefix-delegation pool EDISON
R1(config-dhcpv6)#dns-server 2001:0:0:1::19
```

Código 2.4. Configuración pool de direcciones IPv6.

Para la configuración del protocolo DHCPv4 se sigue el mismo procedimiento en cuanto al pool de direcciones. En el Código 2.5. se muestra los comandos utilizados para su configuración.

```
R1(config)#ip dhcp pool EDISON
R1(dhcp-config)#network 192.168.1.0 255.255.255.0
R1(dhcp-config)#default-router 192.168.1.1
R1(dhcp-config)#dns-server 80.58.0.33
```

Código 2.5. Configuración pool de direcciones IPv4.

2.1.3.3 Conexión y configuración del Gateway

Para que los dispositivos finales puedan tener acceso inalámbrico a la red implementada, se hará uso de dos routers D-Link DIR 600. Las características de este router se revisaron en la sección 2.1.1.3. A continuación se explica el procedimiento necesario para configurar un router en modo punto de acceso.

En primer lugar se conecta el router con la PC con un cable directo desde el puerto LAN del router hacia la PC como se ilustra en la Figura 2.10.

Figura 2.10. Conexión del router con la PC.

Para ingresar a la página de configuración del router, se abre un navegador web y se digita la dirección 192.168.0.1. A continuación se llena el campo de usuario con *admin*, el campo de contraseña queda en blanco y al final se escriben las letras del recuadro como se muestra en la Figura 2.11.

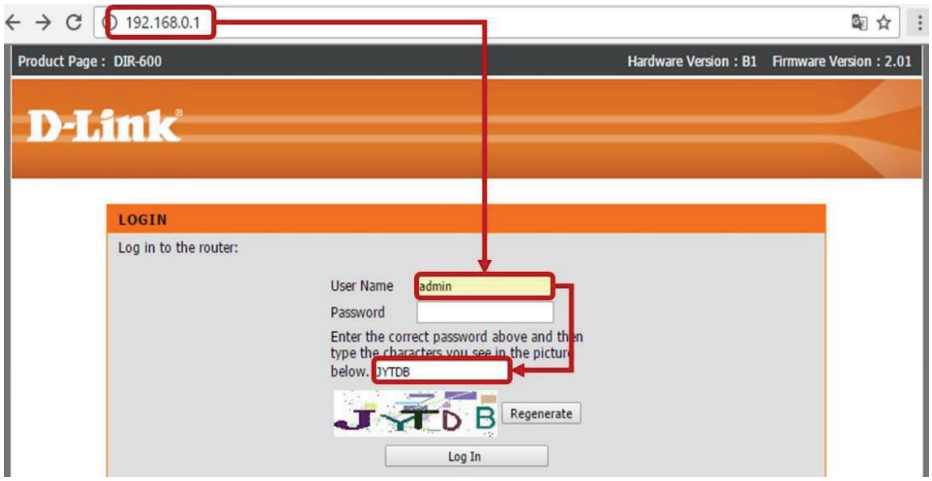


Figura 2.11. Página de configuración del router.

En la parte de *Wireless Network Settings* se configura el nombre de la red y la contraseña respectiva. La Figura 2.12, muestra la configuración inalámbrica.

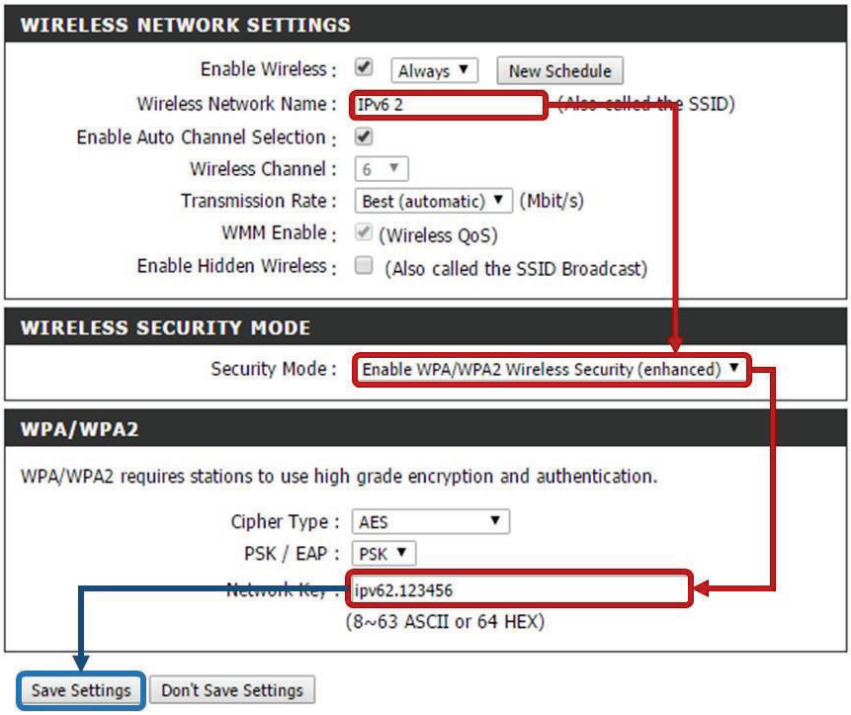


Figura 2.12. Configuración Wireless Network.

Finalmente en la parte de *Internet Setup* se selecciona el casillero para habilitar el modo de punto de acceso.

2.2 CONFIGURACIÓN DE LA PC Y DEL DISPOSITIVO MÓVIL EN LA RED IPv6

Una vez implementada la red, se procede a configurar la PC, la laptop y el dispositivo móvil en la red IPv6.

La PC y la laptop podrán trabajar con el sistema operativo Windows o Linux especificados en la sección 1.7. En este caso, se realizará la configuración de la PC en el OS Windows 7 y de la laptop en Ubuntu 14.04.

El dispositivo móvil trabajará con el sistema operativo Android 6.0.1, el cual es apto para soportar el protocolo IPV6 especificado en la sección 1.7.3.1.

2.2.1 CONFIGURACIÓN DE LA PC Y LA LAPTOP

Para configurar la PC se debe conectar un cable Ethernet desde el puerto de red de la PC hacia uno de los puertos LAN del Gateway. Una vez hecho esto, la PC reconocerá la red y asignará automáticamente una dirección IPv6. La Figura 2.13, muestra la configuración de red mediante la ejecución del comando `ipconfig` en el sistema operativo Windows.

```
C:\Users\CCNA v5>ipconfig

Configuración IP de Windows


Adaptador de Ethernet Conexión de área local:

   Sufijo DNS específico para la conexión. . :
   Dirección IPv6 . . . . . . . . . . . . : 2001::1:bcdf:6cf7:11ee:e049
   Dirección IPv6 temporal. . . . . . . . : 2001::1:d968:beb3:56a3:2294
   Vínculo: dirección IPv6 local. . . . . : fe80::bcdf:6cf7:11ee:e049%10
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.1.4
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : fe80::215:faff:fec9:8d54%10
                                       192.168.1.1
```

Figura 2.13. Dirección IPv6 de la PC.

Para configurar la laptop, se hará uso de la señal inalámbrica del Gateway y la conexión WiFi que dispone la laptop. A continuación, se selecciona la red y finalmente se digita la contraseña establecida. La laptop obtendrá automáticamente

una dirección IPv6. Para verificar la dirección IPv6 se utiliza el comando `ifconfig`. La Figura 2.14, muestra la interfaz *wlan0* de la laptop.

```
wlan0     Link encap:Ethernet  HWaddr 9c:4e:36:62:2b:8c
          inet addr:192.168.1.6  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::9e4e:36ff:fe62:2b8c/64 Scope:Link
          inet6 addr: 2001::1:302f:c614:90ba:59bc/64 Scope:Global
          inet6 addr: 2001::1:9e4e:36ff:fe62:2b8c/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:144 errors:0 dropped:0 overruns:0 frame:0
          TX packets:298 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17567 (17.5 KB)  TX bytes:44191 (44.1 KB)
```

Figura 2.14. Dirección IPv6 de la laptop.

2.2.2 CONFIGURACIÓN DEL DISPOSITIVO MÓVIL

La configuración del dispositivo móvil se realiza mediante la conexión WiFi que dispone el celular y la señal inalámbrica del Gateway. A continuación, se selecciona la red “IPv6 2” y se digita la contraseña correspondiente como se muestra en la Figura 2.15.

Figura 2.15. Selección de red.

Una vez hecho esto, el Gateway asignará automáticamente una dirección IPv6 al dispositivo móvil. Posteriormente, se procede a revisar la dirección asignada en el dispositivo revisando los ajustes de red como se muestra en la Figura 2.16.

Figura 2.16. Verificación de dirección IPv6.

2.2.3 PRUEBAS DE CONECTIVIDAD

Se realiza pruebas de conectividad de la PC, laptop y del dispositivo móvil hacia la red IPv6 utilizando el comando `ping`. El objetivo de `ping` es determinar si un host destino, identificado con una determinada IP, es accesible desde otro host [38].

Para verificar la conectividad en la red IPv6 se ejecuta el comando `ping -6` desde el CMD (*Command Prompt*, Símbolo del Sistema) de Windows y `ping6` desde la terminal de Linux y Android.

2.2.3.1 Prueba de conectividad entre la Pc y la red IPv6

La Figura 2.17, muestra la prueba de conectividad exitosa entre la Pc y la red IPv6.

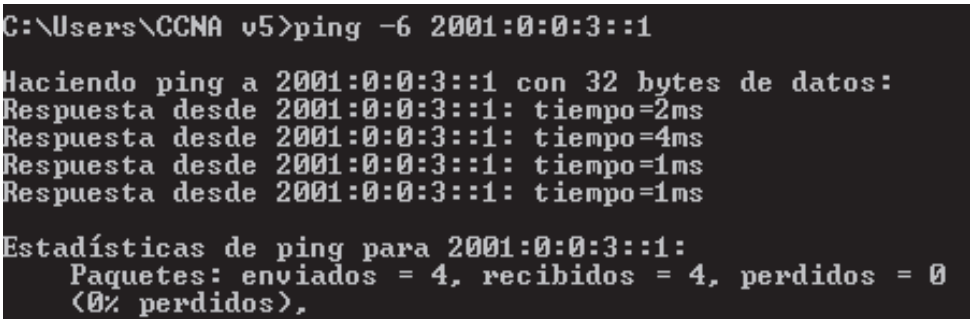


Figura 2.17. Ping entre la PC y la red IPv6.

2.2.3.2 Prueba de conectividad entre la laptop y la red IPv6

La Figura 2.18, muestra la prueba de conectividad exitosa entre la laptop y la red IPv6.

```
sonykec@ivan-PC:~$ ping6 2001:0:0:4::1
PING 2001:0:0:4::1(2001:0:0:4::1) 56 data bytes
64 bytes from 2001:0:0:4::1: icmp_seq=1 ttl=62 time=70.4 ms
64 bytes from 2001:0:0:4::1: icmp_seq=2 ttl=62 time=6.32 ms
64 bytes from 2001:0:0:4::1: icmp_seq=3 ttl=62 time=10.6 ms
^C
--- 2001:0:0:4::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
```

Figura 2.18. Ping entre la laptop y la red IPv6.

2.2.3.3 Prueba de conectividad entre el dispositivo móvil y la red IPv6

Para realizar esta prueba, se debe descargar previamente en el dispositivo móvil cualquier aplicación que simule una terminal de consola. Una vez hecho esto, se siguen los procedimientos anteriormente descritos. La Figura 2.19, muestra la prueba de conectividad exitosa entre el dispositivo móvil y la red IPv6.

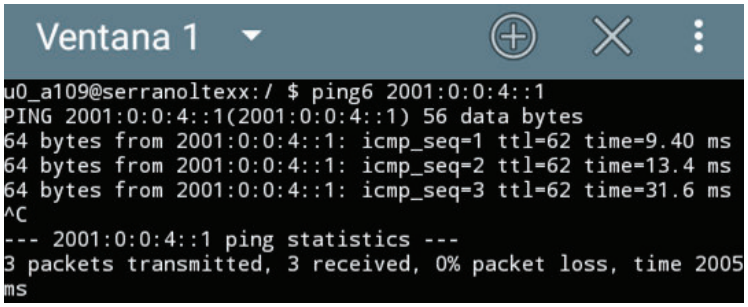


Figura 2.19. Ping entre el dispositivo móvil y la red IPv6.

2.3 CONFIGURACIÓN DEL DISPOSITIVO INTEL EDISON

2.3.1 REQUERIMIENTOS DE CONFIGURACIÓN

Los siguientes requerimientos son los que se van a cumplir en la configuración del dispositivo Intel Edison.

a) El dispositivo Intel Edison debe obtener una dirección IPv6 al conectarse a la red IPv6.
b) El dispositivo Intel Edison debe permitir comunicarse vía SSH[37] (*Secure Shell*) con la laptop para su programación y configuración.
c) La tarjeta Intel Breakout debe permitir el uso de una batería de 9V para su funcionamiento.

[37] SSH: Protocolo que sirve para acceder a máquinas remotas a través de una red.

d) La tarjeta Intel Breakout debe permitir el uso de una cámara web.

2.3.1.1 Ubuntu 14.04 [39]

El sistema operativo seleccionado para programar y configurar el dispositivo Intel Edison es Linux Ubuntu 14.04 de 32 bits.

Ubuntu es un sistema operativo desarrollado en GNU/Linux, se distribuye como software libre y es ideal para laptops, PCs y servidores. Este sistema operativo cuenta además con un entorno visual y con todas las aplicaciones que el usuario pueda necesitar, ya sea desde procesadores de texto hasta herramientas de programación.

Se utiliza este sistema operativo debido a que es estable y fácil de usar.

2.3.1.2 Intel Platform Flash Tool Lite [40]

Intel Platform Flash Tool Lite es una aplicación gratuita que sirve para instalar el firmware a los dispositivos basados en Intel. En la Figura 2.20 se ilustra la interfaz gráfica de esta aplicación.

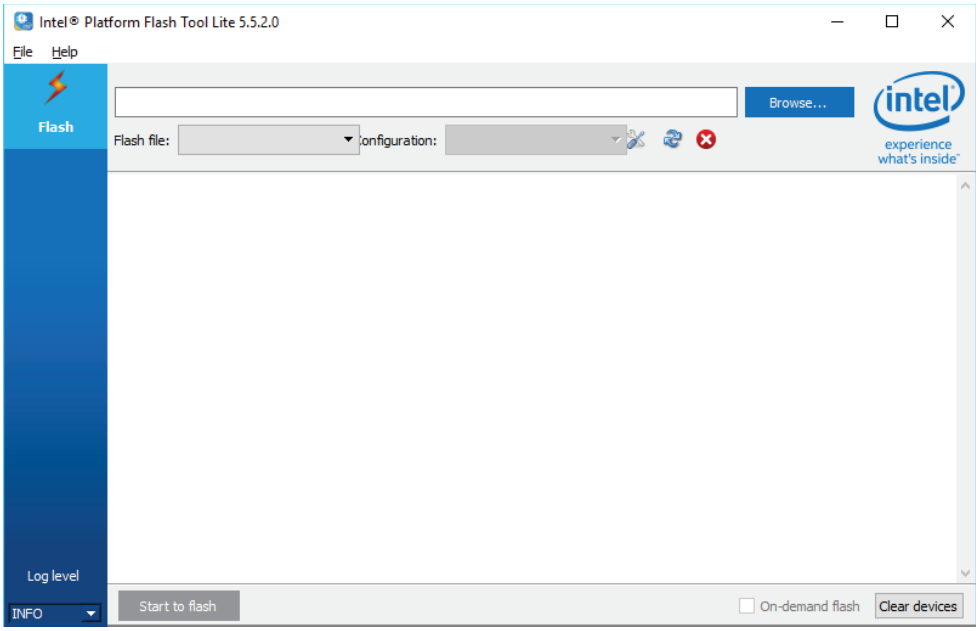


Figura 2.20. Interfaz gráfica Flash Tool Lite.

Debido a que el dispositivo Intel Edison trabajará con una versión de Linux, se utilizara el Intel Platform Flash Tool Lite 5.5.2 para Ubuntu.

2.3.2 CONFIGURACIÓN DEL DISPOSITIVO

2.3.2.1 Ensamblaje del dispositivo Intel Edison

El primer paso consisten en ensamblar el dispositivo. Para ello se dispondrá del módulo Intel Edison, la tarjeta Intel Breakout y dos tuercas hexagonales como se observa en la Figura 2.21.

Figura 2.21. Componentes del dispositivo.

A continuación, se adapta el módulo Intel Edison con la tarjeta Intel Breakout a través del conector de 70 pines especificado en la sección 1.6, luego se ajusta en las esquinas del módulo mediante las tuercas hexagonales. El dispositivo ensamblado se presenta en la Figura 2.22.

Figura 2.22. Dispositivo Intel Edison.

2.3.2.2 Instalación del firmware

Para llevar acabo la instalación del firmware es necesario descargar la versión más reciente, el cual se encuentra disponible en la página oficial del módulo Intel Edison. El Release 3.5 por ser el firmware más actual a la fecha, es el que se va utilizar en esta instalación. Se procede a descargar el software como se muestra en la Figura 2.23.

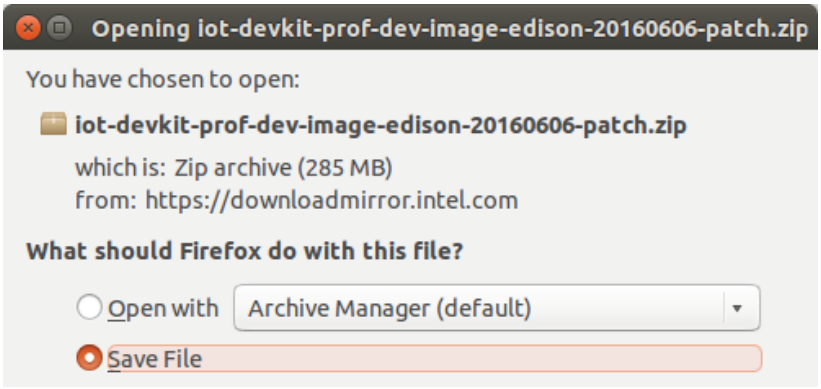


Figura 2.23. Descarga del firmware más reciente.

Con la ayuda de Flash Tool Lite especificado en la sección 2.3.1.2 se procede a instalar el firmware en el dispositivo Intel Edison.

Una vez abierta la aplicación, se carga el archivo descargado anteriormente, se selecciona el OS adecuado, en este caso Linux y se da clic en *start to flash*, segundos después se conecta un cable micro-USB Tipo B a USB desde el jumper *J16* hacia la PC con el fin de establecer la conexión entre los dispositivos. La Figura 2.24, muestra el instante que la aplicación solicita la conexión de la tarjeta.

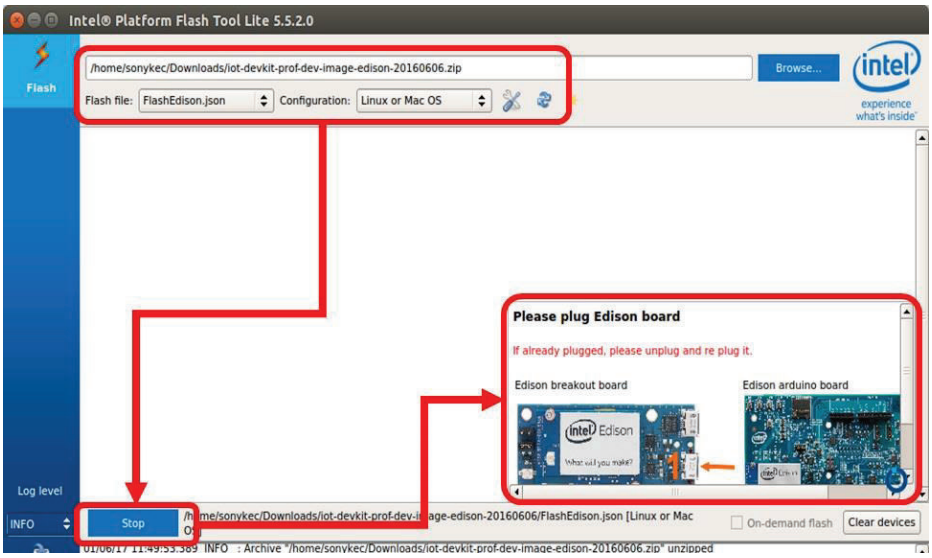


Figura 2.24. Instalación del firmware inicial.

Una vez finalizado el proceso de instalación se espera 2 minutos para desconectar la tarjeta de la PC. La Figura 2.25, muestra la parte final de la instalación.

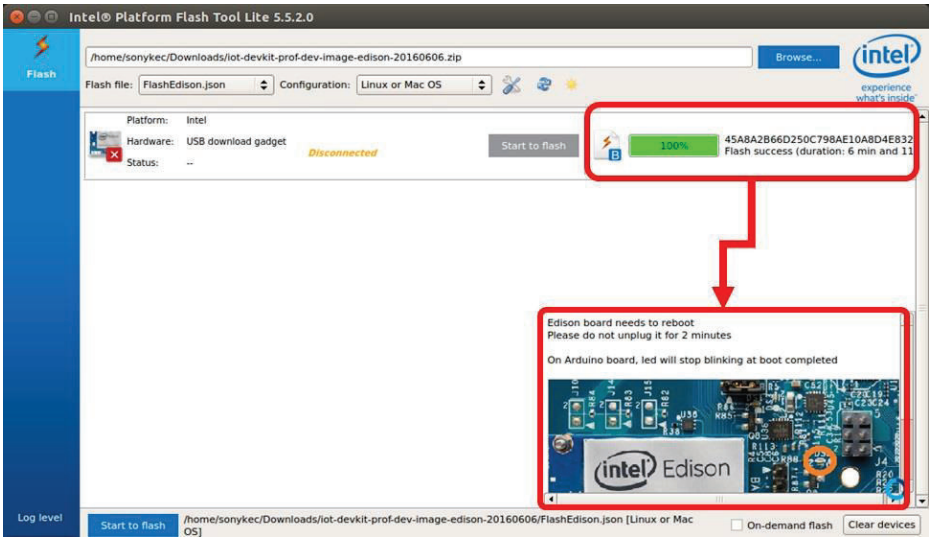


Figura 2.25. Firmware instalado.

2.3.2.3 Configuración de acceso al dispositivo Intel Edison

Para la configuración del dispositivo Intel Edison se utiliza dos cables micro-USB a USB, los cuales se conectan desde los jumpers *J3* y *J16* a la PC. Esta conexión hará que la tarjeta empiece a funcionar en *Modo Dispositivo* el cual se especificó en la sección 1.6.4.1.1.

A continuación, dentro de la terminal se digita el comando `sudo screen /dev/ttyUSB0 115200` que permitirá acceder al sistema, donde *ttyUSB0* es el dispositivo serial conectado y *115200* es la velocidad con la que trabaja.

Una vez ejecutado este comando, se digita `root` para iniciar sesión en el dispositivo y entrar a la terminal de Edison como se ilustra en la Figura 2.26.

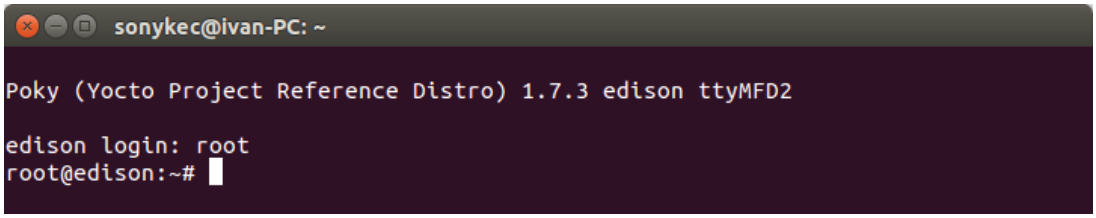


Figura 2.26. Ejecución de inicio.

Desde la terminal de Edison se realizará las configuraciones que se mencionan a continuación.

2.3.2.4 Configuración Wi-Fi

Existen dos métodos para configurar la conexión Wi-Fi en el dispositivo Intel Edison.

La primera consiste en ejecutar desde la terminal de Edison el comando `configure_edison --wifi`; conforme avanza la configuración se debe seguir los siguientes pasos:

- Elegir la red inalámbrica seleccionando el número que aparece en la terminal de Edison.
- Digitar la contraseña de la red inalámbrica seleccionada.

Una vez terminado este proceso, la tarjeta Intel Edison se conectara a la red inalámbrica.

El problema con esta configuración es que la tarjeta Intel Edison no logra reconocer direcciones IPv6. Es por eso que se seguirá otro método para configurar el dispositivo a la red IPv6 implementada.

2.3.2.4.1 Connman

Connman es un gestor de conexiones daemon[38] que permite administrar las conexiones de red a través de Wi-Fi. Para configurar el dispositivo Intel Edison se hará uso de este gestor de conexiones que se encuentra incluido en el sistema.

Para empezar a utilizar *connman* es necesario eliminar todas las redes que se encuentren en la interfaz *wlan0* del dispositivo. Para ello se ejecuta desde la terminal de Edison el comando `wpa_cli -i wlan0 remove_network all`, luego se inicializa y habilita el *connman* mediante los comandos `systemctl start connman` y `systemctl enable connman`. Todo este procedimiento se puede observar en la Figura 2.27.

[38] Daemon: Es un programa que constantemente está disponible y permanece inactivo hasta que es invocado para cumplir una tarea.

```
root@edison:~# wpa_cli -i wlan0 remove_network all
OK
root@edison:~# systemctl start connman
root@edison:~# systemctl enable connman
root@edison:~#
```

Figura 2.27. Configuración del gestor de conexiones.

Para ejecutar comandos de usuario que serán necesarios para configurar el dispositivo, se ingresa al modo interactivo que posee *connman* ejecutando desde la terminal de Edison el comando `connmanctl`. Una vez dentro, se habilita Wi-Fi y se busca las redes inalámbricas disponibles mediante los comandos `enable wifi` y `scan wifi`, luego se ejecuta el comando `services` para desplegar las redes inalámbricas disponibles como se observa en la Figura 2.28.

```
connmanctl> services
    INTERNET_CNT         wifi_784b87a27d93_494e5445524e455420434e54_managed_psk
    IPv6                 wifi_784b87a27d93_49507636_managed_psk
    CASA                 wifi_784b87a27d93_43415341_managed_psk
    MARCOS               wifi_784b87a27d93_4d4152434f53_managed_psk
```

Figura 2.28. Ejecución de connmanctl.

Para acceder a la red IPv6 que se muestra en la Figura 2.28, se registra un agente de usuario ejecutando el comando `agent on`, luego con el comando `connect wifi_784b87a27d93_49507636_managed_psk` se procede a conectarse a la red seleccionada. Posteriormente se digita la contraseña de dicha red para terminar la configuración.

Para evitar que el dispositivo Intel Edison olvide la configuración al momento de reinicio se ejecuta en la terminal de Edison el comando `wpa_cli save`.

Con el comando `ifconfig` se verifica que la dirección IPv6 este configurado en la interfaz *wlan0* del dispositivo tal como se ilustra en la Figura 2.29.

```
wlan0     Link encap:Ethernet  HWaddr 78:4b:87:a2:7d:93
          inet addr:192.168.5.12  Bcast:192.168.5.255  Mask:255.255.255.0
          inet6 addr: fe80::7a4b:87ff:fea2:7d93/64 Scope:Link
          inet6 addr: 2000:1::7a4b:87ff:fea2:7d93/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Figura 2.29. Interfaz wlan0.

La dirección IPv6 del dispositivo es `2000:1::7a4b:87ff:fea2:7d93`.

2.3.2.5 Configuración SSH

SSH es un protocolo que facilita la comunicación segura entre dos sistemas y que permite a los usuarios acceder a un host de manera remota [41].

Para comunicarse remotamente con el dispositivo Intel Edison a través de la red, se debe configurar previamente una contraseña.

La contraseña se configura ejecutando desde la terminal de Edison el comando `configure_edison --password`. De esta manera se habilita el uso del dispositivo vía SSH, el cual será necesario utilizar cuando la tarjeta Intel Edison sea alimentada desde una batería externa.

Para acceder al dispositivo haciendo uso de SSH se ejecuta en la terminal el comando `ssh root@2000:1::7a4b:87ff:fea2:7d93`.

La primera vez que se accede por esta vía, el dispositivo solicitara un permiso de autenticación. Una vez concedido el permiso se digita la contraseña establecida y la tarjeta estará lista para ser usada.

El procedimiento de acceso vía SSH se observa en la Figura 2.30.

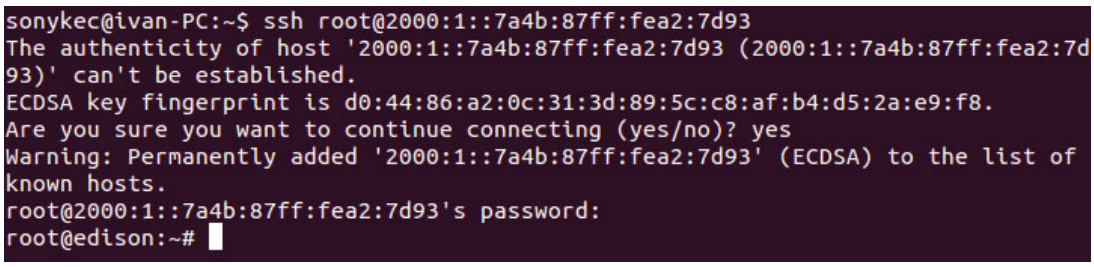


Figura 2.30. Acceso vía SSH.

2.3.2.6 Configuración de la alimentación externa

Para utilizar el dispositivo Intel Edison con alimentación externa se hará uso de una batería de 9V.

Las características que debe poseer la batería así como también su configuración con el dispositivo se especificó en la sección 1.6.4.1.2.

2.3.2.7 Configuración del gestor de paquetes

Un gestor de paquetes mantiene un registro del software que está instalado en un dispositivo, y permite instalar, actualizar y eliminar software de una manera sencilla [42].

El sistema operativo del dispositivo Intel Edison se basa en Yocto Linux y utiliza *opkg* como gestor de paquetes.

Para configurar el gestor de paquetes se ejecuta desde el terminal de Edison el comando `nano /etc/opkg/base-feeds.conf`. Dentro del archivo `base-feeds.conf` se digita y guarda las siguientes líneas:

```
src/gz all http://repo.opkg.net/edison/repo/all
src/gz edison http://repo.opkg.net/edison/repo/edison
src/gz core2-32 http://repo.opkg.net/edison/repo/core2-32
```

De esta manera se agregan paquetes útiles, tales como el controlador UVC que servirá para la configuración de la cámara web y la herramienta Git[39] que se utilizara para clonar el código de la aplicación a la tarjeta Intel Edison.

A continuación, se procede a descargar la actualización *opkg* mediante el comando `opkg update` como se muestra en la Figura 2.31.

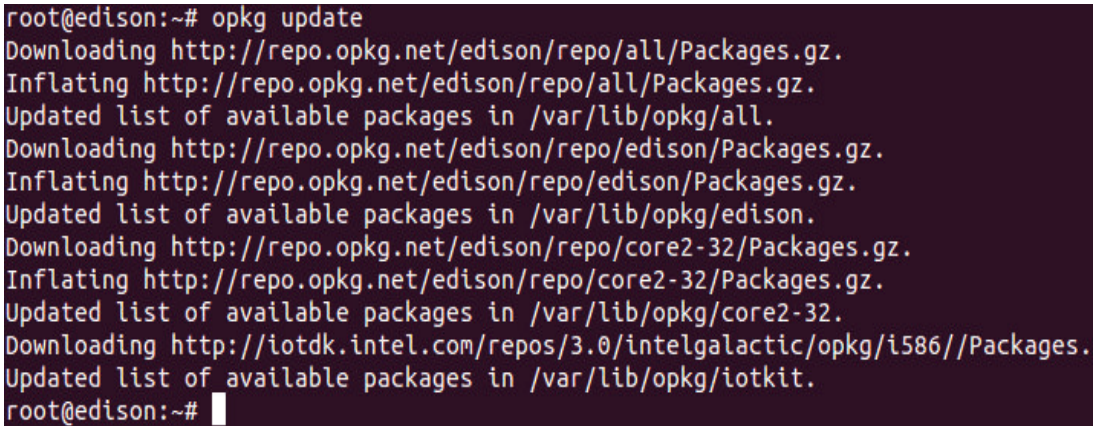


Figura 2.31. Actualización Opkg.

[39] Git: Software de control de versiones.

2.3.2.8 Configuración de la cámara web

Se necesitará una cámara web que soporte el estándar UVC revisado en la sección 1.8.1. En este caso se utiliza la cámara web marca Zense.

Entre sus características se encuentran: una lente de 3 megapíxeles CMOS, una resolución de 2048x1536 para fotos y 640x480 para video, enfoque manual y ángulo de visión de 60/360/90 grados, tres luces LED y soporte UVC.

Para que la cámara web funcione con la tarjeta Intel Edison, se debe conectar una batería de 9V en jumper *J21* y la cámara en el jumper *J16* como se ilustra en la Figura 2.32.

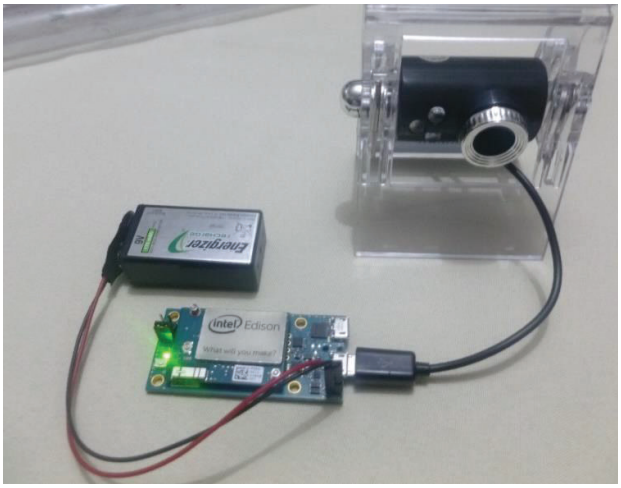


Figura 2.32. Conexión de la batería y cámara web.

Para comprobar que la cámara web cumpla con el formato UVC se accede a la tarjeta Intel Edison vía SSH revisado en la sección 2.3.2.5. Una vez dentro, se desconecta y conecta la cámara y se ejecuta desde la terminal de Edison el comando `dmesg -c`, el cual desplegara una serie de mensajes entre ellos estará el mensaje `USB Video Class driver`, confirmando que la cámara web soporta el formato UVC.

Para acceder y utilizar la cámara web adaptada al dispositivo Intel Edison se debe incluir en la línea de comandos `/dev/video0`.

2.4 IMPLEMENTACIÓN DE LA APLICACIÓN

2.4.1 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN

Los siguientes requerimientos son los que se van a cumplir en la implementación de la aplicación.

e) El dispositivo Intel Edison debe permitir transmitir video sobre la red IPv6 a través de la cámara web que se adapta a la tarjeta Intel Breakout.
f) El video transmitido por la red IPv6 debe visualizarse en el dispositivo móvil, laptop y PC a través de una página web.

2.4.1.1 Sublime Text 3 [43]

Sublime Text 3 es un editor de texto de código fuente ideal para los programadores que utilizan Java[40], JavaScript[41], HTML[42], C++[43], entre otros lenguajes de programación.

Posee una interfaz gráfica amigable para el usuario y un increíble rendimiento. Actualmente se encuentra en fase beta y se puede evaluar de forma gratuita. Para su uso continuo es necesario adquirir una licencia. La Figura 2.33, muestra el logotipo de Sublime Text.

Figura 2.33. Interfaz Sublime Text 3.

Se eligió este editor de texto por las características que ofrece en cuanto a rendimiento y eficiencia. Además se puede descargar de forma gratuita y utilizar la versión de evaluación que es plenamente funcional y no tiene fecha de caducidad.

---

[40] Java: Es un lenguaje de programación de propósito general orientado a objetos.
[41] JavaScript: Es un lenguaje de programación que permite crear scripts en páginas HTML.
[42] HTML (*HyperText Markup Language*): Es un lenguaje de programación para páginas web.
[43] C++: Es un lenguaje de programación basado en el lenguaje C.

2.4.1.2 Node.JS [44]

Node.js es una plataforma para el desarrollo de aplicaciones escalables construida sobre V8[44], el intérprete de JavaScript de Google Chrome.

Los scripts[45] de *Node.js* están diseñados para correr del lado del servidor para maximizar el rendimiento y la eficiencia, es de código abierto, y se puede ejecutar en Windows, Linux o Mac OS.

Esta plataforma viene incluida en las actuales distribuciones de Linux. Para verificar la versión de *Node.js* instalada en el dispositivo Intel Edison se ejecuta desde el terminal de Edison el comando `node -v`. La Figura 2.34, muestra el logotipo de esta plataforma.

Figura 2.34. NodeJs [44].

Se eligió *Node.js* debido a sus características en cuanto al rendimiento y eficiencia en el desarrollo de aplicaciones en tiempo real.

2.4.1.2.1 Npm

El *npm* (*Node Package Manager*) es el manejador de paquetes de *Node.js*. Este paquete viene instalado en las versiones más recientes de *Node.js*. Para verificar la versión *npm* instalada en el dispositivo Intel Edison se ejecuta desde el terminal de Edison el comando `npm -v`. La Figura 2.35, muestra el logotipo de *npm.*

Figura 2.35. NPM [44].

[44] V8: Es un motor de código abierto para JavaScript creado por Google.
[45] Script: Es un conjunto de instrucciones.

2.4.2 DESARROLLO DE LA APLICACIÓN

Para el desarrollo de la aplicación se utilizó el código del proyecto edi-cam [45]. El proyecto edi-cam está disponible en GitHub, una plataforma web de uso gratuito que sirve para almacenar proyectos utilizando el sistema de control de versiones Git [46].

Se realizaron modificaciones al código del proyecto edi-cam con el fin de cumplir los requerimientos expuestos en la sección 2.4.1. En la Figura 2.36 se presenta el diagrama de flujo del funcionamiento del proyecto edi-cam.

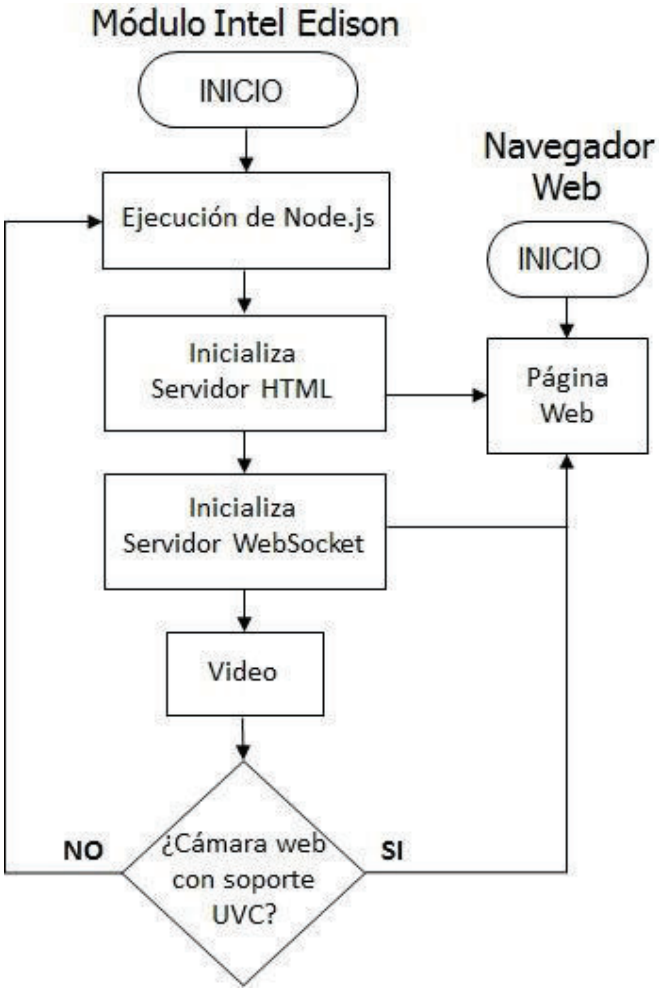


Figura 2.36. Flujograma del proyecto edi-cam.

El proyecto edi-cam está dividido en diferentes bloques que inician con la ejecución de node.js, el servidor HTML, el servidor WebSocket, el script de video y la página web.

Los bloques mencionados anteriormente, se detallan a continuación junto con las modificaciones que se realizaron en el bloque de video y en la página web. El código completo modificado para la aplicación de este proyecto se encuentra en su totalidad en el ANEXO B.

En la Figura 2.37 se presenta el diagrama de flujo de la aplicación modificada para la transmisión de video sobre la red IPv6.

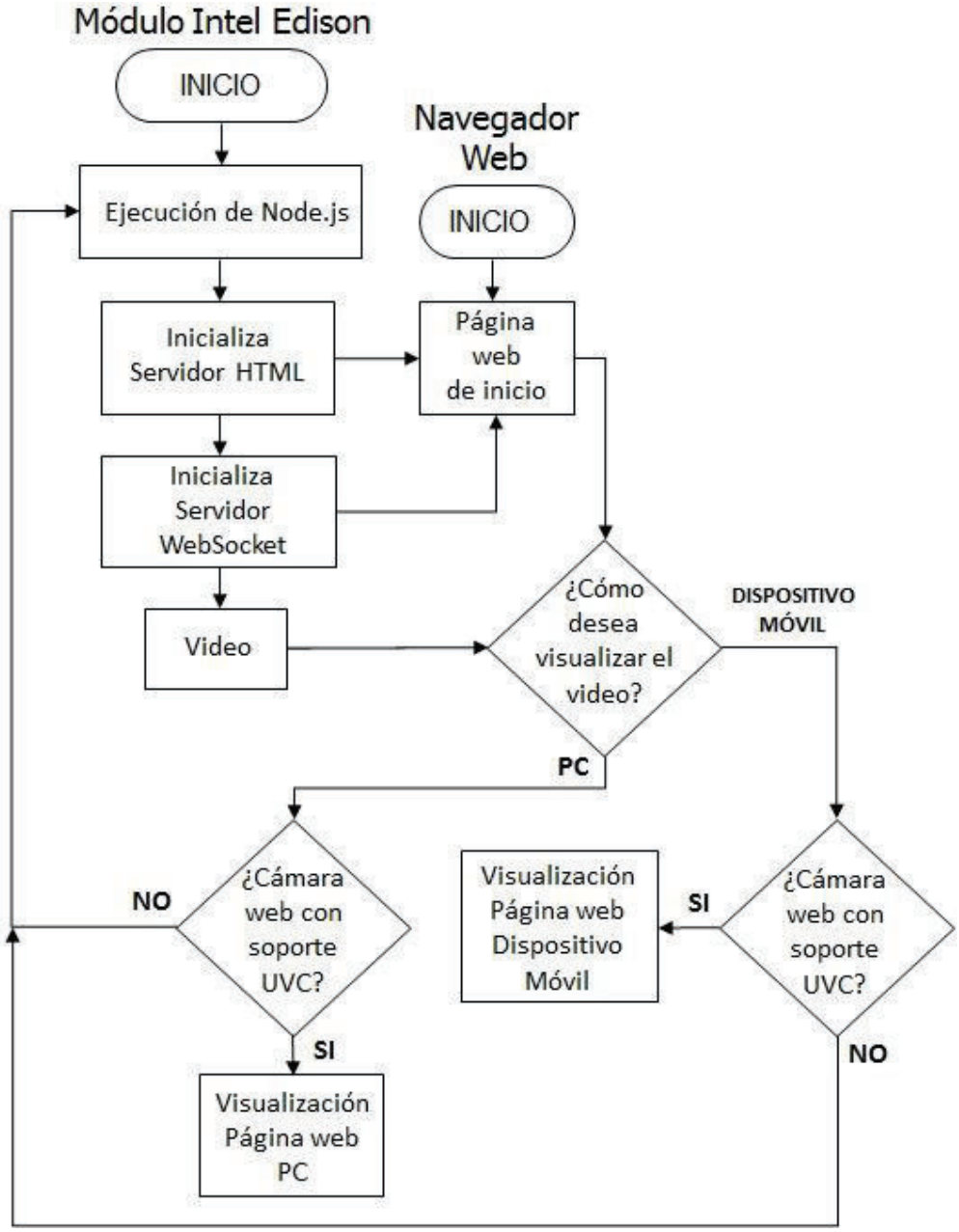


Figura 2.37. Flujograma de la aplicación.

2.4.2.1 Archivo de configuración

El proyecto edi-cam hace uso de la plataforma Node.js revisado en la sección 2.4.1.2. Todo proyecto con node.js requiere de un archivo de configuración denominado *package.json*. Este archivo contiene datos importantes en cuanto a paquetes y versiones a utilizarse en el desarrollo de la aplicación. El Código 2.6., muestra el *package.json* modificado utilizado en esta aplicación.

```
{
  "name": "intel-edison-video-ipv6",
  "private": true,
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "express": "4.10.1",
    "morgan": "1.4.1",
    "ws": "0.4.32"
  }
}
```

Código 2.6. Archivo de configuración.

La parte más relevante del archivo *package.json* es *dependencies*, donde se especifican los módulos que se utilizan en la aplicación. Entre ellos se tiene:

- *Express*: Es el framework[46] web para Node.js. Se utiliza para servir páginas web de manera más robusta.
- *Morgan:* Es un registrador. Se utiliza para registrar solicitudes automáticamente en la consola.
- *Ws*: Es una librería de WebSocket. Se utiliza para establecer una conexión continua entre el cliente y el servidor.

Este archivo se ejecuta mediante el comando `npm install` para instalar todos los paquetes utilizados en esta aplicación.

2.4.2.2 Servidor web

Un servidor web sirve contenido a un navegador. El proceso consiste en enviar peticiones desde la conexión del cliente hasta el sistema encargado de administrar cada petición (servidor), de este modo se transmite la respuesta correspondiente

[46] Framework: Es un esquema para el desarrollo o implementación de una aplicación.

de nuevo al cliente. El cliente y el servidor utilizan el protocolo HTTP[47] para comunicarse a través de una conexión TCP[48] [47].

El proyecto edi-cam se basa en un servidor web con node.js. Hace uso de las librerías *express*, *morgan* y *ws* revisados en la sección 2.4.2.1. Además añade la librería *http* que permite utilizar los protocolos HTTP para transferir datos web y *child_process* para ejecutar procesos de larga duración y liberar el hilo principal para que se siga procesando peticiones. El Código 2.7, muestra el servidor web.

```
//Módulos a utilizarse
var childProcess = require('child_process')
  , express = require('express')
  , http = require('http')
  , morgan = require('morgan')
  , ws = require('ws');

//Configuración de los archivos
var configServer = require('./lib/config/server');

//Parámetros de la aplicación
var app = express();

app.set('port', configServer.httpPort);
//Se indica en que puerto se va a estar escuchando

app.use(express.static(configServer.staticFolder));
//Esta línea se utiliza para enlazar la página index.html a la aplicación

app.use(morgan('dev'));
//Permite registrar cada solicitud en la consola

//Servidor index
require('./lib/routes').serveIndex(app, configServer.staticFolder);
//Se indica la ruta que va seguir el index.html para enlazarse a la aplicación

http.createServer(app).listen(app.get('port'), function () {
//Se establece la conexión http en el puerto.
  console.log('Servidor HTTP escuchando en el puerto ' + app.get('port'));
});
```

Código 2.7. Servidor web.

Mediante la variable *configServer* se carga el script que contiene las direcciones de los puertos a ser utilizados en el desarrollo de la aplicación los cuales se detallan en el Código 2.8.

[47] HTTP: Protocolo de comunicación que permite la transferencia de información en la web.
[48] TCP: Es un protocolo fundamental de control de transmisión para Internet.

A continuación se inicializa *express* y se configuran los puertos, el servidor de archivos estáticos donde se encuentra la página *index.html* y el comando `morgan` para registrar las solicitudes en consola.

```
httpPort: 8080,
//Puerto por donde se accede a la página web
streamPort: 8082,
//Puerto utilizado por ffmpeg
wsPort: 8084,
//Puerto utilizado por la librería ws
staticFolder: path.join(__dirname + '/../../../client')
//Dirección que apunta a la página index.html
```

Código 2.8. Puertos.

Donde:

- *httpPort:* Variable utilizada para el acceso a la página web.
- *streamPort:* Variable utilizada por *ffmpeg* para la codificación de video.
- *wsPort:* Variable utilizada para la librería WebSocket.
- *staticFolder:* Variable de acceso a la página index.html.

En modo de resumen se presenta a continuación en la Figura 2.38 un diagrama de flujo referencial de los procesos realizados en el servidor web.

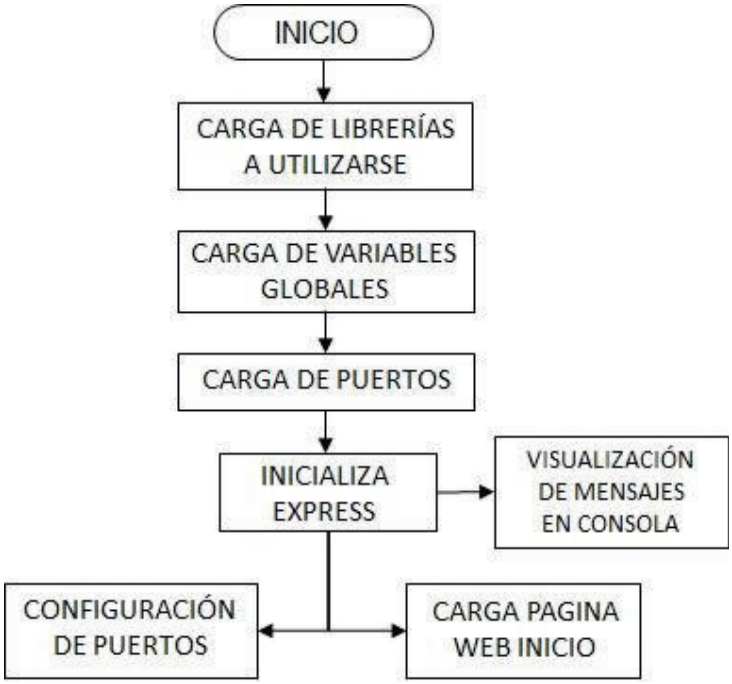


Figura 2.38. Flujograma del servidor web.

2.4.2.3 Servidor WebSocket

WebSocket es un protocolo para la web que permite una comunicación bidireccional entre cliente y servidor sobre un único socket TCP [48].

El proceso WebSocket inicia con un handshake HTTP y debe ser soportado tanto en el servidor como en el cliente [49]. En el lado del cliente, los navegadores actuales ya están en la capacidad de soportar WebSockets tales como Mozilla Firefox 8, Google Chrome 4, Safari 5 en iOs 4.2 y en Internet Explorer 10.

La conexión con el protocolo websocket se realiza de la siguiente manera:

- El cliente le pide al servidor que quiere iniciar una conexión, para ello se intercambian handshake para la validación [49].
- El servidor una vez que verifica, admite la conexión y desde ese momento dejan de utilizar HTTP y pasan a usar el protocolo WebSocket [49].

La Figura 2.39, muestra el procedimiento que se sigue en una conexión WebSocket.

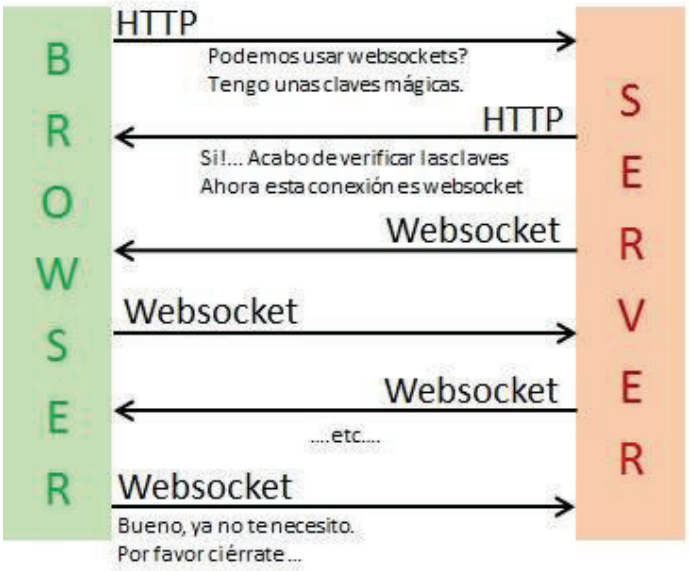


Figura 2.39. Conexión Websocket [49].

El proyecto edi-cam implementa un servidor WebSocket que hace uso de la librería *ws* revisado en la sección 2.4.2.1. Primero se añaden las variables que se utilizaran para la resolución del video. El Código 2.9, muestra las variables de video.

```
var STREAM_MAGIC_BYTES = 'jsmp';
//Enlace al archivo de video jsmpg
var width = 640;
//Resolución de ancho de video
var height = 480;
//Resolución de altura de video
```

Código 2.9. Variables de video.

En el Código 2.10, se especifica la variable *wsServer*, encargada de inicializar el servidor WebSocket para distribuir el flujo MPEG[49]. Se define la función *socket* para realizar funciones cada vez que se detecte una nueva conexión WebSocket.

```
var wsServer = new (ws.Server)({ port: configServer.wsPort });
//Se inicializa el servidor websocket en el puerto específico
console.log('Servidor WebSocket escuchando en el puerto ' + configServer.wsPort);

wsServer.on('connection', function(socket) {
//Se establece una función cada vez que se detecte una nueva conexión

  var streamHeader = new Buffer(8);

  streamHeader.write(STREAM_MAGIC_BYTES);
  streamHeader.writeUInt16BE(width, 4);
  streamHeader.writeUInt16BE(height, 6);
  socket.send(streamHeader, { binary: true });
  //Datos que se utilizan para el formato de video de la nueva conexión

  console.log('Nuevo WebSocket Conectado (' + wsServer.clients.length + '
total)');

 socket.on('close', function(code, message){
  //Se emite un mensaje cuando se cierra una conexión WebSocket
  console.log('Desconectado WebSocket (' + wsServer.clients.length + ' total)');
  });
});
```

Código 2.10. Servidor WebSocket.

La variable *streamHeader* define parámetros en cuanto al tamaño del video. Cada vez que se conecte un nuevo cliente, la función *socket* se encargara de procesar el tamaño del video y enviar el flujo al navegador. Además se registran todas las conexiones y desconexiones de los Websockets en la consola.

[49] MPEG (*Moving Picture Experts Group*): Estándar de compresión y de formatos de archivo de video digital.

2.4.2.4 Implementación de video

Para la implementación de video, el proyecto edi-cam hace uso del script *jsmpg* y de la herramienta *ffmpeg*.

El script *jsmpg* es un decodificador MPEG1[50] escrito en JavaScript y que es compatible para la transmisión de video a través de WebSockets [50]. Es de acceso libre. El código del script *jsmpg* se encuentra en el ANEXO C.

La herramienta *ffmpeg* es un programa que permite codificar y decodificar video y audio. Es un software libre y puede trabajar en la mayoría de sistemas operativos [51].

La codificación de video se realiza mediante la herramienta *ffmpeg*. Para instalar *ffmpeg* en la tarjeta Intel Edison se sigue los siguientes pasos desde el terminal de Edison.

- Se crea un nuevo directorio con el comando `mkdir` que contendrá el programa *ffmpeg*.
- Se procede a descargar la última versión de *ffmpeg* desde su página oficial mediante el comando `wget`.
- Se descomprime el archivo con el comando `tar –xf` para obtener *ffmpeg*.
- Finalmente, se limpia el directorio eliminando el archivo comprimido mediante el comando `rm`.

Para la ejecución de *ffmpeg* se hará uso del módulo *child_process* cargado previamente en el servidor web. Este módulo ejecuta el archivo *do_ffmpeg.sh* mediante el comando `childProcess.exec('../../bin/do_ffmpeg.sh')`.

El archivo *do_ffmpeg.sh* se modificó de tal manera que el video codificado se envié a través de HTTP a la dirección IPv6 de la tarjeta Intel Edison y al puerto especificado en el servidor web. El Código 2.11 ilustra el código utilizado para la codificación de video.

[50] MPEG1: Nombre de un grupo de estándares de codificación de audio y video.

```
/home/root/bin/ffmpeg/ffmpeg -s 640x480 -f video4linux2 -i /dev/video0 -f
mpeg1video \ -b 800k -r 30 http://[2000:1::7a4b:87ff:fea2:7d93]:8082
```

Código 2.11. Archivo ffmpeg.

Donde:

- *-s 640x480*: Indica la resolución (ancho x alto) del video a obtener.
- *-f video4linux2:* Es una API de captura de video para Linux.
- *-i /dev/video0*: Indica que la cámara web acoplada a la tarjeta Intel Edison es la fuente desde donde se captura el video.
- *-f mpeg1video*: Indica el formato de salida para el video.
- *-b 800k*: Indica el bitrate del vídeo.
- *-r 30*: Indica el número de imágenes por segundo (fps).
- *http://[2000:1::7a4b:87ff:fea2:7d93]:*8082: Indica la dirección IPv6 de la tarjeta Intel Edison y el puerto especificado destino del video codificado.

A continuación se explica el proceso para la implementación de video.

El proceso inicia con la captura de video de la cámara, en este caso de 640x480, luego se codifica el video con la herramienta *ffmpeg* en MPEG1 con 30 cuadros por segundo y un flujo de datos de 800 Kbit/s. Posteriormente, el video codificado se envía al host (tarjeta Intel Edison) y al puerto especificado en el servidor web a través del protocolo HTTP.

En el servidor web, mediante la función *connection* especificado en la sección 2.4.2.3, el flujo MPEG1 se distribuye a través de WebSockets a todos los navegadores conectados.

Finalmente, el navegador decodifica el flujo MPEG1 de JavaScript y transforma las imágenes decodificadas para implementar el video con Canvas[51] en la dirección IPv6 y puerto especificado en la ejecución del archivo *do_ffmpeg*.

En modo de resumen y para mejor comprensión en el proceso de implementación de video, se presenta a continuación el diagrama de flujo en la Figura 2.40.

[51] Canvas: Elemento de HTML utilizado para realizar composición de imágenes y animaciones.

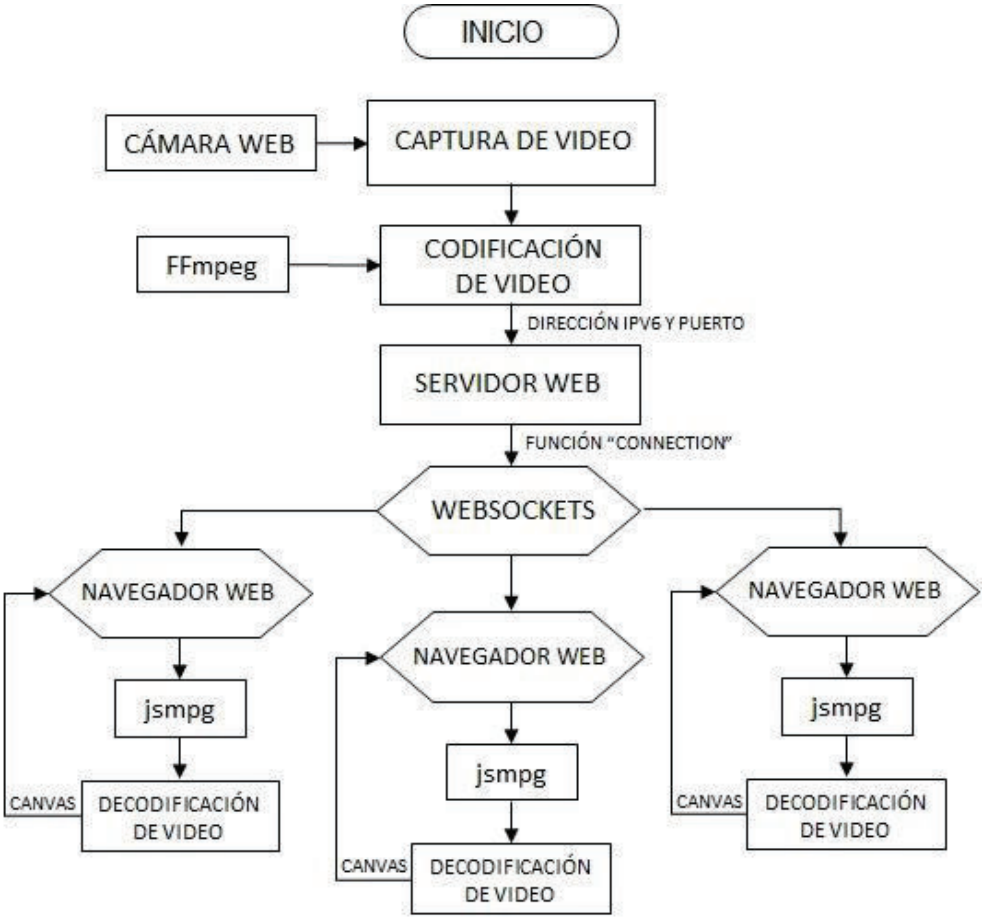


Figura 2.40. Flujograma de implementación de video.

2.4.2.5 Página web

En la página web del proyecto edi-cam se realizaron modificaciones con el fin de incorporar IPv6 a través de los websockets. Además, para observar el video transmitido desde la tarjeta Intel Edison, se desarrolla una página de inicio y dos páginas compatibles con las dimensiones de pantalla de una laptop y del dispositivo móvil. El código completo de las páginas diseñadas se presenta en el ANEXO D.

2.4.2.5.1 Página de inicio

La página de inicio (index.html) contiene información del proyecto. Para su ejecución se utiliza el servicio de archivos estáticos de *express* como se ilustra en Código 2.12.

```
app.use(express.static(configServer.staticFolder));
```

Código 2.12. Servicio de archivos estáticos.

La Figura 2.41, muestra la interfaz gráfica de la página de inicio para la PC y el dispositivo móvil.

a) Página de inicio en la PC

b) Página de inicio en el dispositivo móvil

Figura 2.41. Página de inicio.

Además se proporciona los vínculos de enlace a la página del computador y del dispositivo móvil para observar el video transmitido. El Código 2.13, muestra los comandos utilizados para ejecutar los enlaces.

```
<a href="computador.html"> <img src="computadora.png" " width=150 height=150
style="margin: 10px 0px 0px 0px" /> </a>
<!--Acceso a la página del computador-->

<a href="movil.html"> <img src="movil.png" align="center" width=100 height=90
style="margin: 10px 0px 0px 100px" /> </a>
<!--Acceso a la página del dispositivo móvil-->
```

Código 2.13. Enlace de páginas.

2.4.2.5.2 Página del computador

La página del computador (computador.html) se diseña con un tamaño de ventana óptimo, de tal manera que se pueda visualizar el video en la pantalla de la PC como se muestra en la Figura 2.42.

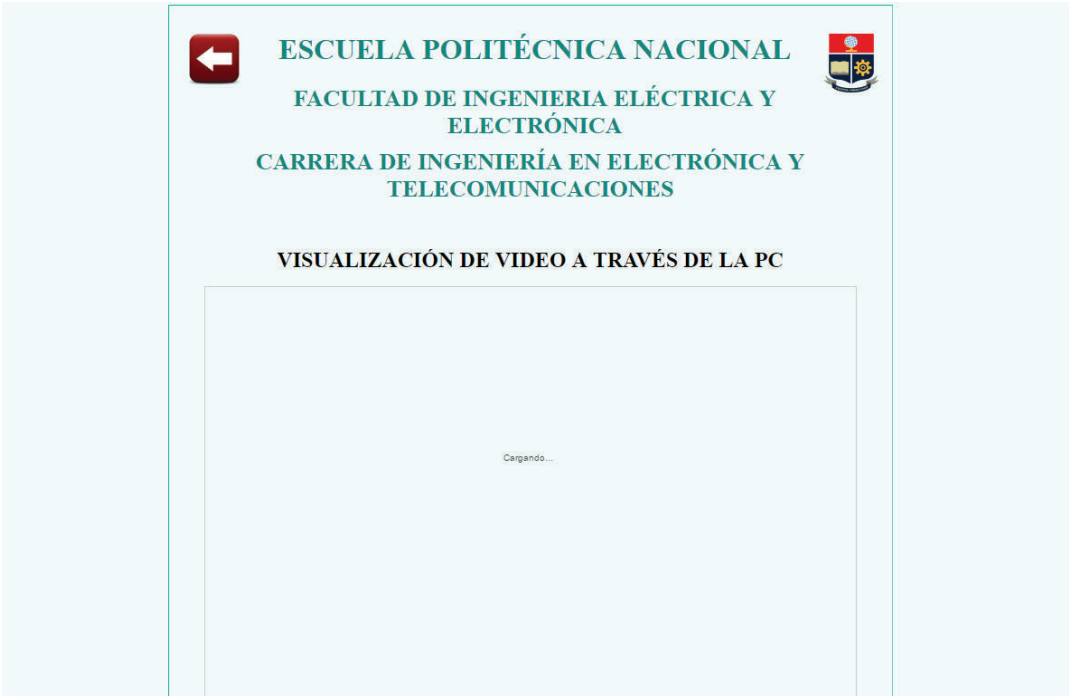


Figura 2.42. Página del computador.

Para visualizar el video en el navegador web se utiliza el script *jsmpg* revisado en la sección 2.4.2.4 y el elemento *canvas*.

Mediante el script *jsmpg*, el video codificado y distribuido por los websocket, se decodifica en el navegador web con soporte websocket y se despliega en la pantalla mediante el elemento canvas. El Código 2.14, muestra el código utilizado en el proyecto edi-cam para implementar el video en el navegador web.

```
<canvas id="canvas-video" width="640" height="480"></canvas>
<!--Se carga el id canvas-video con la resolución de pantalla para PC-->

<script src="/jsmpg.js"></script>

<script type="text/javascript">
<!--Se carga el archivo jsmpg escrito en JavaScript-->

var canvas = document.getElementById('canvas-video');

var ctx = canvas.getContext('2d');
<!--Se utiliza para acceder al objeto canvas-video y a la página index.html-->

ctx.fillStyle = '#333';

ctx.fillText('Cargando...', canvas.width/2-30, canvas.height/3);
<!--Se configura el tamaño de la ventana donde se podrá observar el video -->
```

Código 2.14. Decodificación de video.

Para que la página web acepte peticiones IPv6 se agrega la variable *client1* para abrir la conexión WebSocket, donde el elemento *ws* es el nuevo esquema URL[52] para las conexiones WebSocket. Además, se debe especificar una dirección IPv6 para establecer la conexión. En este caso la dirección IPv6 de la tarjeta Intel Edison se utiliza como URL y el puerto 8084 es el puerto utilizado por la librería *ws*.

```
var client1 = new WebSocket('ws://[2000:1::7a4b:87ff:fea2:7d93]:8084');
<!--Se especifica la dirección para abrir la conexión WebSocket-->

var player1 = new jsmpeg(client1, { canvas:canvas });
<!--Se acepta la URL para el streaming-->
```

Código 2.15. Conexión WebSocket.

Como se detalla en el Código 2.15, la variable *player1* acepta la dirección URL para la conexión WebSocket y se procede a pasar el flujo MPEG al elemento *canvas* para que este a su vez despliegue el video en el navegador web.

[52] URL (*Uniform Resource Locator*): Permite denominar recursos dentro del entorno de Internet.

2.4.2.5.3 Página del dispositivo móvil

La página del dispositivo móvil (móvil.html) se muestra en la Figura 2.43 y se implementa de la misma manera que el expuesto en la sección 2.4.2.5.2.

Figura 2.43. Página del dispositivo móvil.

Los cambios en esta página se efectúan en el tamaño de la ventana donde se observa el video. En este caso se utiliza la resolución de 640x480 píxeles.

2.5 IMPLEMENTACIÓN DEL CÓDIGO EN EL DISPOSITIVO INTEL EDISON

Para implementar los códigos expuestos en la sección 2.4 en el dispositivo Intel Edison se hará uso de la plataforma GitHub revisado en la sección 2.4.2. Para ello, se creó un proyecto con el nombre *intel-edison-video-ipv6* con las modificaciones realizadas al proyecto edi-cam.

Se procede a subir los archivos modificados al nuevo proyecto en la plataforma GitHub. La Figura 2.44, muestra el código distribuido en carpetas en la plataforma GitHub.

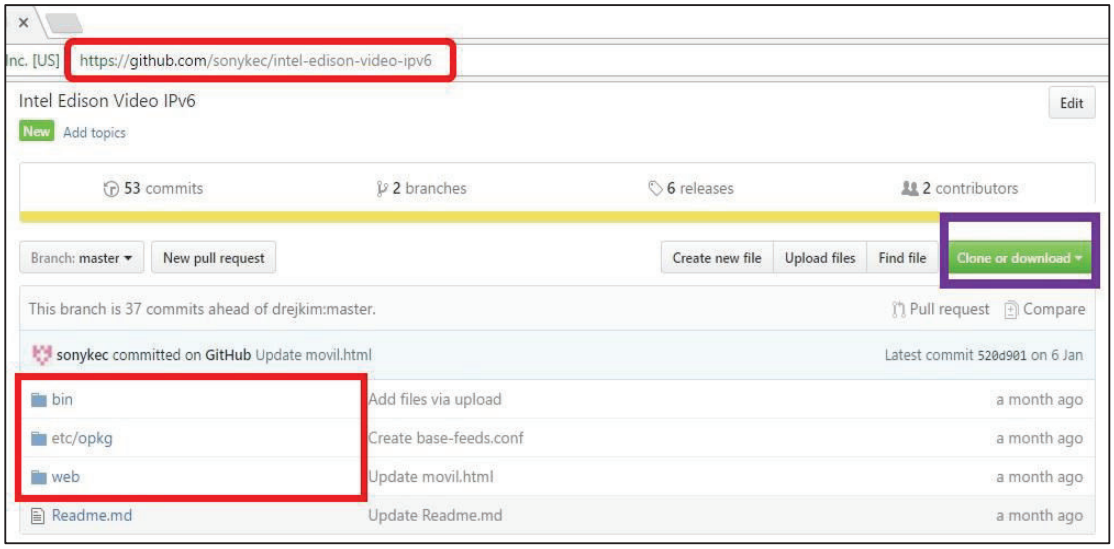


Figura 2.44. Código subido a la plataforma GitHub.

A continuación, en el dispositivo Intel Edison, se verifica si se tiene instalado el sistema de control de versiones Git. En las versiones actuales de Yocto Linux, Git está incluido en el sistema, sin embargo se puede descargar digitando desde la terminal de Edison el comando `opkg install git`.

Luego se procede a clonar los archivos del proyecto *intel-edison-video-ipv6* en la tarjeta Intel Edison ejecutando el comando:

```
git clone https://github.com/sonykec/intel-edison-video-ipv6
```

La Figura 2.45, muestra el proceso de clonación en la tarjeta Intel Edison.

```
root@edison:~# git clone https://github.com/sonykec/intel-edison-video-ipv6.git
Cloning into 'intel-edison-video-ipv6'...
remote: Counting objects: 236, done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 236 (delta 3), reused 0 (delta 0), pack-reused 226
Receiving objects: 100% (236/236), 462.10 KiB | 97.00 KiB/s, done.
Resolving deltas: 100% (80/80), done.
Checking connectivity... done.
root@edison:~#
```

Figura 2.45. Ejecución del comando git clone.

Para que el código del proyecto pueda ejecutarse en el dispositivo Intel Edison, se debe instalar los paquetes Node.js especificados en la sección 2.4.2.1. Estos paquetes se deben instalar junto al script *server.js*.

Dentro de la terminal de Edison se ejecuta los comandos necesarios para ubicarse en la carpeta del script *server.js*. Una vez dentro, se instala los paquetes con el comando `npm install`. El procedimiento anterior se muestra en la Figura 2.46.

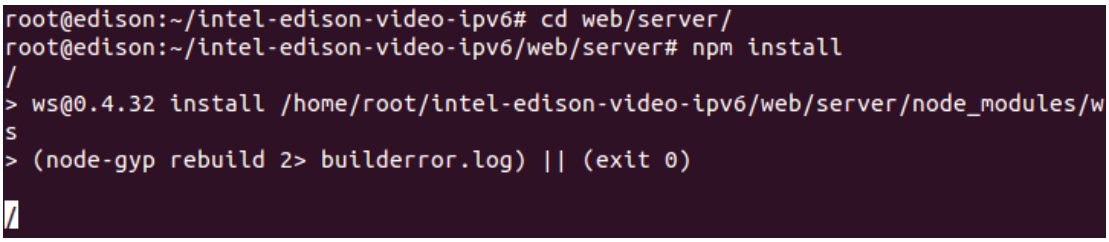


Figura 2.46. Instalación de paquetes npm.

De esta manera, el código queda implementado en el dispositivo Intel Edison.

Para inicializar la aplicación se debe ejecutar desde la terminal de Edison el comando `node server.js`. Este procedimiento se debe realizar en la carpeta donde se encuentra el script *server.js*. Una vez hecho esto, en la terminal de Edison se observara el mensaje *Stream Conectado* confirmando que la aplicación esta lista para transmitir el video como se ilustra en la Figura 2.47.

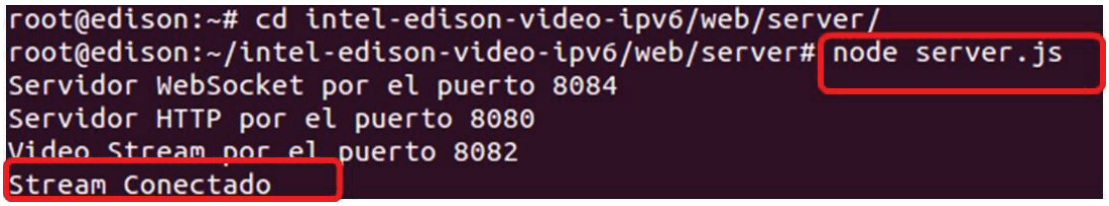


Figura 2.47. Ejecución del script server.js.

Con la ayuda del navegador web, se ingresa a la página con la dirección IPv6 del dispositivo Intel Edison, en este caso `2000:1::7a4b:87ff:fea2:7d93`, junto con el puerto 8080 para proceder a visualizar el video.

CAPÍTULO III

3 PRUEBAS DE FUNCIONAMIENTO Y MEDICIÓN DEL CONSUMO DE ENERGÍA

En el presente capítulo se presentan los resultados de las pruebas de funcionamiento realizadas en la transmisión de video desde la tarjeta Intel Edison hacia los dispositivos finales. También se realiza la medición del consumo de energía en función de la carga de procesamiento de la plataforma Intel Edison.

3.1 PRUEBAS DE FUNCIONAMIENTO

Como se mencionó en la sección 2, el dispositivo Intel Edison a través de la cámara web, transmitirá video sobre la red IPv6 hacia los dispositivos finales tales como el celular, laptop y computadora de escritorio. La visualización del video se hará mediante una página en el navegador web especificado en la sección 2.4.2.5.

A continuación se detallan las pruebas realizadas para demostrar el funcionamiento en la PC, la laptop y el dispositivo móvil.

3.1.1 PRUEBA 1 DE FUNCIONAMIENTO EN LA PC

Esta prueba fue realizada con el objetivo de demostrar el funcionamiento de la aplicación en un navegador web de una PC, en este caso se utiliza una PC con el OS Windows 7 y el navegador web Mozilla Firefox 8.

El ambiente de prueba estuvo conformado por los siguientes componentes:

- Tarjeta Intel Edison donde se ejecutó la aplicación.
- Una cámara web.
- Una batería recargable de 9V Energizer.
- PC donde se visualizó el video a través del navegador web.
- Cuatro routers que simularon la red de transporte IPv6 configurados en la sección 2.1.3.
- Dos gateways como puntos de acceso a la red IPv6 configurados en la sección 2.1.3.3.

En la Figura 3.1 se ilustra los componentes involucrados en la Prueba 1.

Figura 3.1. Componentes de la Prueba 1.

En la Figura 3.2 se observa la página de visualización de video a través de la PC.

Figura 3.2. Prueba de funcionamiento en la PC.

Los resultados obtenidos de la Prueba 1 se mencionan a continuación:

- La transmisión de video en general se observa de manera continua en el navegador Mozilla Firefox 8 de la computadora.

- En ciertos momentos se detecta un leve retraso en la visualización del video en el navegador web.
- La resolución de video mostrado en la página del navegador es buena considerando que la calidad para la transmisión de video de la cámara web es de 640x480 píxeles.

3.1.2 PRUEBA 2 DE FUNCIONAMIENTO EN LA LAPTOP

Esta prueba fue realizada con el objetivo de demostrar el funcionamiento de la aplicación en un navegador web de una computadora portátil, en este caso se utiliza una laptop con el OS Windows 10 y el navegador web Mozilla Firefox 8.

El ambiente de prueba estuvo conformado por los siguientes componentes:

- Tarjeta Intel Edison donde se ejecutó la aplicación.
- Una cámara web.
- Una batería recargable de 9V Energizer.
- Una laptop donde se visualizó el video a través del navegador web.
- Cuatro routers que simularon la red de transporte IPv6 configurados en la sección 2.1.3.
- Dos gateways como puntos de acceso a la red IPv6 configurados en la sección 2.1.3.3.

En la Figura 3.3 se ilustra los componentes involucrados en la Prueba 2.

Figura 3.3. Componentes de la Prueba 2.

En la Figura 3.4 se observa la página de visualización de video a través de la laptop.

Figura 3.4. Prueba de funcionamiento en la laptop.

Los resultados obtenidos de la Prueba 2 se mencionan a continuación:

- La transmisión de video en general se observa de manera continua en el navegador web de la laptop.
- Existen momentos en los cuales se detecta un retraso en la visualización del video en el navegador web de la laptop.
- La resolución de video mostrado en la página del navegador de la laptop es buena.

3.1.3 PRUEBA 3 DE FUNCIONAMIENTO EN EL DISPOSITIVO MÓVIL

Esta prueba fue realizada con el objetivo de demostrar el funcionamiento de la aplicación en un navegador web de un dispositivo móvil, en este caso se utiliza el

celular Samsung S4 mini con el sistema operativo Android 6.0.1 y el navegador web Google Chrome 4.

El ambiente de prueba estuvo conformado por los siguientes componentes:

- Tarjeta Intel Edison donde se ejecutó la aplicación para la transmisión de video.
- Una cámara web.
- Una batería recargable de 9V Energizer.
- Celular Samsung S4 mini donde se visualizó el video a través del navegador web.
- Cuatro routers que simularon la red de transporte IPv6 configurados en la sección 2.1.3.
- Dos gateways como puntos de acceso a la red IPv6 configurados en la sección 2.1.3.3.

En la Figura 3.5 se ilustra los componentes involucrados en la Prueba 3.

Figura 3.5. Componentes de la Prueba 3.

En la Figura 3.6 se observa la página de visualización de video a través del celular.

Figura 3.6. Prueba de funcionamiento en el dispositivo móvil.

Los resultados obtenidos de la Prueba 3 se mencionan a continuación:

- La transmisión de video en general se observa de manera continua en el navegador web del celular.
- En ciertos momentos se detecta un retraso en la visualización del video en el navegador web del celular.
- La resolución de video mostrada en la página del navegador web es buena considerando el modelo del celular.

3.2 MEDICIÓN DEL CONSUMO DE ENERGÍA

3.2.1 REQUERIMIENTOS PARA LA MEDICIÓN

Para calcular el consumo de energía se realizó la medición de voltaje y corriente cuando el dispositivo Intel Edison está transmitiendo video hacia los dispositivos finales. La medición se hace entre la batería y la tarjeta Intel Edison considerando los siguientes puntos:

- Diferentes tiempos de operación.
- Diferentes resoluciones en la transmisión de video.
- Diferentes distancias entre el dispositivo Intel Edison y el nodo Gateway 1.

Además se realizó la medición para calcular la energía cuando la tarjeta se encuentra en reposo y cuando se inicia la aplicación pero aún no se está transmitiendo el video.

3.2.1.1 Equipos de medición

Para la medición de voltaje y corriente se utilizó dos multímetros digitales. La Figura 3.7, muestra los equipos utilizados.

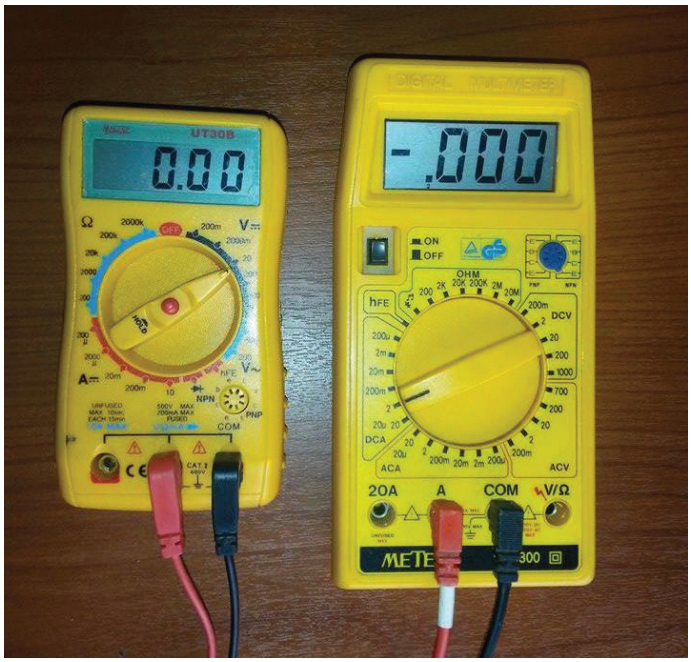


Figura 3.7. Equipos de medición.

3.2.1.2 Tiempos de operación

Los tiempos de operación que se emplearon para la medición cuando el dispositivo Intel Edison está transmitiendo video hacia el celular sobre la red IPv6 se presentan en la Tabla 3.1.

Tabla 3.1. Tiempos de Operación.

Tiempos de Operación (minutos)	1	3	5

3.2.1.3 Resoluciones de trasmisión

Las resoluciones que se consideraron para la transmisión de video en el dispositivo Intel Edison se presentan en la Tabla 3.2.

Tabla 3.2. Resoluciones.

Resoluciones (píxeles)	320x240	640x480

La resolución de 640×480 píxeles, es la resolución máxima para video de la cámara web utilizada en este proyecto, la cual se especificó en la sección 2.3.2.8.

3.2.1.4 Distancia al nodo gateway

Se emplearon diferentes distancias para las mediciones de voltaje y corriente. No se consideraron obstáculos que obstruyan la señal inalámbrica al momento de la medición. La Figura 3.8, muestra el esquema de los dispositivos involucrados en este punto.

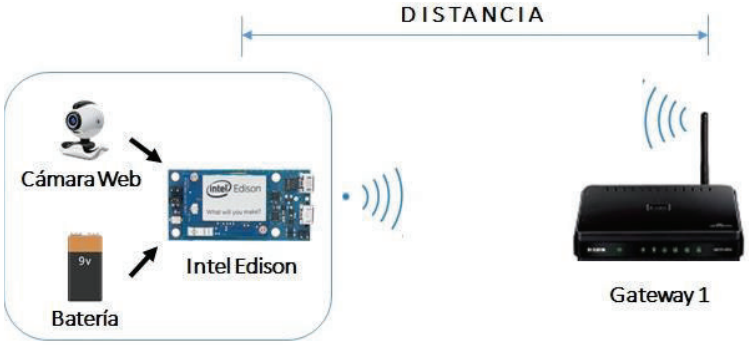


Figura 3.8. Distancia entre dispositivos.

Las distancias consideradas en la medición se presentan en la Tabla 3.3.

Tabla 3.3. Distancias al nodo Gateway.

Distancia al nodo Gateway (metros)	2	8

3.2.2 ESCENARIO DE MEDICIÓN

La Figura 3.9, muestra el escenario y los componentes involucrados en la medición de voltaje y corriente en los puntos especificados en la sección 3.2.1.

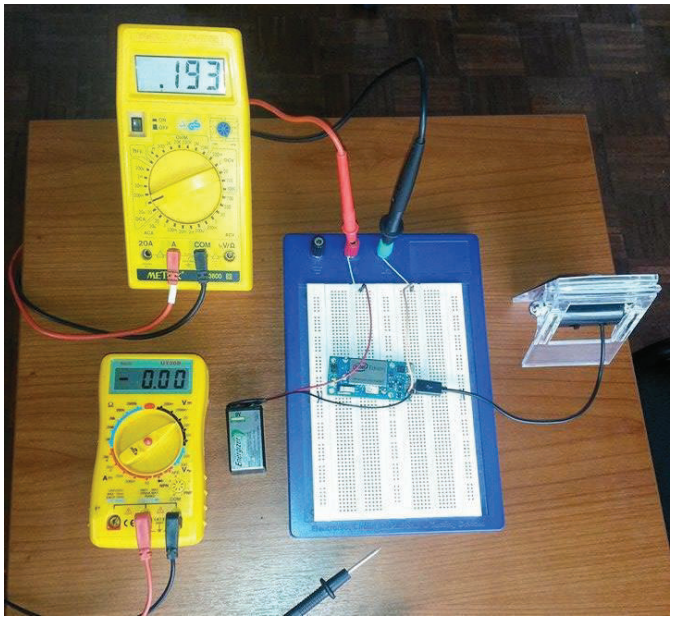


Figura 3.9. Escenario y componentes para la medición.

3.2.3 RESULTADOS OBTENIDOS

A continuación se presenta los valores obtenidos de las mediciones.

3.2.3.1 Estado de reposo

El valor medido de voltaje y corriente del dispositivo Intel Edison cuando se estabiliza y entra en estado de reposo es de 8.72V y 0.065A.

3.2.3.2 Aplicación inicializada

El valor medido de voltaje y corriente cuando se estabiliza los procesos luego de inicializar la aplicación es de 8.26V y 0.165A.

3.2.3.3 Transmisión de video hacia la PC

Los valores medidos en la transmisión de video hacia la PC se presentan en la Tabla 3.4.

Tabla 3.4. Valores de medición en transmisión de video hacia la PC.

Tiempo de Operación [min]	Resolución [px]	Distancia al nodo Gateway 1 [m]	Voltaje [V]	Corriente [A]
1	320x240	2	8.45	0.190
		6	8.43	0.192
	640x480	2	8.19	0.209
		6	8.37	0.205
3	320x240	2	8.44	0.192
		6	8.39	0.194
	640x480	2	8.08	0.214
		6	8.25	0.211
5	320x240	2	8.43	0.193
		6	8.37	0.196
	640x480	2	7.97	0.218
		6	8.20	0.213

3.2.3.4 Transmisión de video hacia la Laptop

Los valores medidos en la transmisión de video hacia la laptop se presentan en la Tabla 3.5.

Tabla 3.5. Valores de medición en transmisión de video hacia la laptop.

Tiempo de Operación [min]	Resolución [px]	Distancia al nodo Gateway 1 [m]	Voltaje [V]	Corriente [A]
1	320x240	2	8.47	0.192
		6	8.10	0.202
	640x480	2	8.42	0.206
		6	8.35	0.209
3	320x240	2	8.30	0.197
		6	8.05	0.206
	640x480	2	8.35	0.208
		6	8.21	0.215
5	320x240	2	8.15	0.202
		6	7.95	0.211
	640x480	2	8.20	0.213
		6	8.11	0.221

3.2.3.5 Transmisión de video hacia el Celular

Los valores medidos en la transmisión de video hacia el celular se presentan en la Tabla 3.6.

Tabla 3.6. Valores de medición en transmisión de video hacia el celular.

Tiempo de Operación [min]	Resolución [px]	Distancia al nodo Gateway 1 [m]	Voltaje [V]	Corriente [A]
1	320x240	2	8.35	0.195
		6	8.31	0.197
	640x480	2	8.30	0.209
		6	8.29	0.210
3	320x240	2	8.22	0.199
		6	8.21	0.202
	640x480	2	8.19	0.212
		6	8.15	0.217
5	320x240	2	8.11	0.204
		6	8.10	0.207
	640x480	2	8.05	0.219
		6	8.03	0.223

3.2.4 ANÁLISIS DE RESULTADOS

A continuación se presenta el análisis de los resultados obtenidos en la sección 3.2.3. Además se evaluó el consumo de potencia en función de la carga de procesamiento de la plataforma Intel Edison.

3.2.4.1 Estado de reposo

La potencia consumida por la tarjeta Intel Edison cuando se estabiliza y entra en estado de reposo (sin procesamiento) se calcula a partir de la Ecuación (3.1).

$$P = V * I \tag{3.1}$$

$$P = 8.72V * 0.065A$$

$$P = 0.57W$$

Se analizó y se verifico que esta potencia permanece aproximadamente constante durante la etapa de reposo.

El consumo de energía del dispositivo Intel Edison en una hora es de 0.00057Kwh.

3.2.4.1.1 Tiempo de descarga de la batería

La batería utilizada en esta etapa posee una carga eléctrica de 175mAh. Para el cálculo de la duración de la batería se consideró un consumo de corriente de 65mA obtenido en la medición de corriente y considerando que la batería se encontraba completamente cargada.

Para calcular el tiempo de descarga de la batería se utilizó la Ecuación (3.2) [52].

$$tiempo\ de\ descarga = \frac{carga\ eléctrica\ de\ la\ bateria}{consumo\ de\ corriente} \tag{3.2}$$

$$tiempo\ de\ descarga = \frac{175mAh}{65mA}$$

$$tiempo\ de\ descarga = 2.7\ horas$$

$$tiempo\ de\ descarga = 2\ horas\ y\ 42\ minutos$$

En teoría, el tiempo de descarga de la batería es de 2.7 horas, sin embargo, en la práctica se comprobó que el tiempo en el cual la tarjeta Intel Edison deja de funcionar es de aproximadamente 2 horas. Esto se debe a que la batería no se descarga completamente cuando la tarjeta Intel Edison deja de trabajar.

3.2.4.2 Aplicación inicializada

La potencia consumida por la tarjeta Intel Edison cuando se estabiliza los procesos luego de inicializar la aplicación se calcula a partir de la Ecuación (3.1).

$$P = 8.26V * 0.165A$$

$$P = 1.36W$$

Se analizó y se verifico que esta potencia permanece aproximadamente constante durante esta etapa. El consumo de energía del dispositivo Intel Edison en una hora es de 0.00136Kwh.

3.2.4.2.1 Tiempo de descarga de la batería

La batería utilizada en esta etapa posee una carga eléctrica de 175mAh. Para el cálculo de la duración de la batería se consideró un consumo de corriente de 165mA

obtenido en la medición de corriente y considerando que la batería se encontraba completamente cargada.

Para calcular el tiempo de descarga de la batería se utilizó la Ecuación (3.2).

$$tiempo\ de\ descarga = \frac{175mAh}{165mA}$$

$$tiempo\ de\ descarga = 1.1\ horas$$

$$tiempo\ de\ descarga = 1\ hora\ y\ 6\ minutos$$

En teoría, en esta etapa el tiempo de descarga de la batería es de 1.1 horas, sin embargo, en la práctica se comprobó que el tiempo en el cual la tarjeta Intel Edison deja de funcionar es de aproximadamente 55 minutos. Esto se debe a que la batería no se descarga completamente cuando la tarjeta Intel Edison deja de trabajar.

3.2.4.3 Transmisión de video hacia la PC

La potencia consumida por la tarjeta Intel Edison cuando se encuentra transmitiendo video hacia la PC se calcula a partir de la Ecuación (3.1).

La Tabla 3.7, muestra los resultados obtenidos.

Tabla 3.7. Consumo de potencia hacia la PC.

Tiempo de Operación [min]	Resolución [px]	Distancia al nodo Gateway 1 [m]	Potencia [W]
1	320x240	2	1.61
		6	1.62
	640x480	2	1.71
		6	1.72
3	320x240	2	1.62
		6	1.63
	640x480	2	1.73
		6	1.74
5	320x240	2	1.63
		6	1.64
	640x480	2	1.74
		6	1.75

Los resultados de consumo se presentan en la Tabla 3.8.

Tabla 3.8. Resultados de consumo.

Resolución [px]	Distancia al nodo Gateway 1 [m]	Potencia Promedio Consumida [W]	Consumo de Energía [Kwh]
320x240	2	1.62	0.00162
	6	1.63	0.00163
640x480	2	1.73	0.00173
	6	1.74	0.00174

3.2.4.3.1 Tiempo de descarga de la batería

La batería utilizada en este punto posee una carga eléctrica de 175mAh. Para el cálculo de la duración de la batería se consideró el consumo de corriente de 190mA para la resolución de 320x240 y de 205mA para la resolución de 640x480. Para calcular el tiempo de descarga de la batería se utilizó la Ecuación (3.2).

- Resolución de 320x240

$$tiempo\ de\ descarga = \frac{175mAh}{190mA}$$

$$tiempo\ de\ descarga = 0.92\ horas$$

$$tiempo\ de\ descarga = 55\ min$$

- Resolución de 640x480

$$tiempo\ de\ descarga = \frac{175mAh}{205mA}$$

$$tiempo\ de\ descarga = 0.85\ horas$$

$$tiempo\ de\ descarga = 51\ min$$

En teoría, el tiempo de descarga de la batería es de 55 minutos para la resolución de 320x240 y de 51 minutos para la resolución de 640x480, sin embargo, en la práctica se comprobó que el tiempo en el cual la tarjeta Intel Edison deja de funcionar es de aproximadamente 47 minutos para la resolución de 320x240 y de 42 minutos para la resolución de 640x480. Esto se debe a que la batería no se descarga completamente cuando la tarjeta Intel Edison deja de funcionar.

3.2.4.4 Transmisión de video hacia la laptop

La potencia consumida por la tarjeta Intel Edison cuando se encuentra transmitiendo video hacia la laptop se calcula a partir de la Ecuación (3.1). La Tabla 3.9, muestra los resultados obtenidos.

Tabla 3.9. Consumo de potencia hacia la laptop.

Tiempo de Operación [min]	Resolución [px]	Distancia al nodo Gateway 1 [m]	Potencia [W]
1	320 x 240	2	1.63
		6	1.64
	640 x 480	2	1.73
		6	1.75
3	320 x 240	2	1.64
		6	1.66
	640 x 480	2	1.74
		6	1.77
5	320 x 240	2	1.65
		6	1.68
	640 x 480	2	1.75
		6	1.79

Los resultados de consumo se presentan en la Tabla 3.10.

Tabla 3.10. Resultados de consumo.

Resolución [px]	Distancia al nodo Gateway 1 [m]	Potencia Promedio Consumida [W]	Consumo de Energía [Kwh]
320x240	2	1.64	0.00164
	6	1.66	0.00166
640x480	2	1.74	0.00174
	6	1.77	0.00177

3.2.4.4.1 Tiempo de descarga de la batería

La batería utilizada en este punto posee una carga eléctrica de 175mAh. Para el cálculo de la duración de la batería se consideró el consumo de corriente de 192mA para la resolución de 320x240 y de 206mA para la resolución de 640x480. Para calcular el tiempo de descarga de la batería se utilizó la Ecuación (3.2).

- Resolución de 320x240

$$tiempo\ de\ descarga = \frac{175mAh}{192mA} = 0.91\ horas = 54\ min$$

- Resolución de 640x480

$$tiempo\ de\ descarga = \frac{175mAh}{206mA} = 0.84\ horas\ 50\ min$$

En teoría, el tiempo de descarga de la batería es de 54 minutos para la resolución de 320x240 y de 50 minutos para la resolución de 640x480, sin embargo, en la práctica se comprobó que el tiempo en el cual la tarjeta Intel Edison deja de funcionar es de aproximadamente 45 minutos para la resolución de 320x240 y de 40 minutos para la resolución de 640x480. Esto se debe a que la batería no se descarga completamente cuando la tarjeta Intel Edison deja de funcionar.

3.2.4.5 Transmisión de video hacia el celular

La potencia consumida por la tarjeta Intel Edison cuando se encuentra transmitiendo video hacia el celular se calcula a partir de la Ecuación (3.1). La Tabla 3.11, muestra los resultados obtenidos.

Tabla 3.11. Consumo de potencia hacia el celular.

Tiempo de Operación [min]	Resolución [px]	Distancia al nodo Gateway 1 [m]	Potencia [W]
1	320 x 240	2	1.63
		6	1.64
	640 x 480	2	1.73
		6	1.74
3	320 x 240	2	1.64
		6	1.66
	640 x 480	2	1.74
		6	1.77
5	320 x 240	2	1.65
		6	1.68
	640 x 480	2	1.76
		6	1.79

Los resultados de consumo se presentan en la Tabla 3.12.

Tabla 3.12. Resultados de consumo.

Resolución [px]	Distancia al nodo Gateway 1 [m]	Potencia Promedio Consumida [W]	Consumo de Energía [Kwh]
320x240	2	1.64	0.00164
	6	1.66	0.00166
640x480	2	1.74	0.00173
	6	1.77	0.00177

3.2.4.5.1 Tiempo de descarga de la batería

La batería utilizada en este punto posee una carga eléctrica de 175mAh. Para el cálculo de la duración de la batería se consideró el consumo de corriente de 195mA para la resolución de 320x240 y de 209mA para la resolución de 640x480. Para calcular el tiempo de descarga de la batería se utilizó la Ecuación (3.2).

- Resolución de 320x240

$$tiempo\ de\ descarga = \frac{175mAh}{195mA} = 0.89\ horas = 53\ min$$

- Resolución de 640x480

$$tiempo\ de\ descarga = \frac{175mAh}{209mA} = 0.83\ horas = 49\ min$$

En teoría, el tiempo de descarga de la batería es de 53 minutos para la resolución de 320x240 y de 49 minutos para la resolución de 640x480, sin embargo, en la práctica se comprobó que el tiempo en el cual la tarjeta Intel Edison deja de funcionar es de aproximadamente 45 minutos para la resolución de 320x240 y de 40 minutos para la resolución de 640x480. Esto se debe a que la batería no se descarga completamente cuando la tarjeta Intel Edison deja de funcionar.

CAPÍTULO IV

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Se realizó las mediciones de voltaje y corriente en diferentes situaciones con el fin de obtener el consumo de energía cuando el dispositivo Intel Edison está transmitiendo video sobre la red IPv6 implementada en el proyecto. Las mediciones determinaron que:
    - El consumo de energía es mayor cuando el dispositivo Intel Edison transmite video con una resolución de 640x480.
    - El consumo de energía es menor cuando el dispositivo Intel Edison transmite video con una resolución de 320x240.

- El módulo Intel Edison es una nueva plataforma de computación compacta que se adapta a varias placas con el fin de poder interactuar con el usuario final. Además permite el uso de diferentes entornos de programación para el desarrollo de prototipos en el Internet de las cosas (IoT).

- Se implementó un prototipo con un nodo (módulo Intel Edison) alimentado por una batería de 9V, para la transmisión de video sobre una red IPv6.

- Se realizó la evaluación del consumo de potencia en función de la carga de procesamiento de la plataforma Intel Edison y se determinó que:
    - La potencia consumida cuando la tarjeta Intel Edison está en reposo (sin procesamiento) es menor que la potencia cuando el dispositivo se encuentra en uso (aplicación iniciada).
    - La potencia consumida cuando la tarjeta Intel Edison ha iniciado la aplicación (con procesamiento) es menor que la potencia cuando la aplicación está transmitiendo video hacia los dispositivos finales.

- Se constató que la potencia consumida por el sistema permanecía aproximadamente constante dentro de los diferentes escenarios propuestos en este proyecto. Tanto el voltaje como la corriente fueron variando en función del procesamiento de la tarjeta Intel Edison para mantener la potencia constante.

- Se calculó el tiempo aproximado de descarga de la batería de 9V y se determinó que:
  - La batería cuando la tarjeta Intel Edison está en reposo (sin procesamiento), se descarga en aproximadamente dos horas.
  - La batería cuando la tarjeta Intel Edison está en funcionamiento con la aplicación iniciada, se descarga en aproximadamente 55 minutos.
  - La batería cuando la tarjeta Intel Edison está transmitiendo video hacia los dispositivos finales, se descarga en aproximadamente 45 minutos, dependiendo de las condiciones de la transmisión.

- Las pruebas de funcionamiento y los resultados de las mediciones del consumo de energía en diferentes condiciones establecieron que la transmisión de video en la plataforma Intel Edison es más eficiente cuando se trabaja con una resolución de 320x240. De esta manera, se podrá visualizar el video de manera continua sin retardos y aprovechando al máximo la energía entregada por la batería de 9V.

4.2 RECOMENDACIONES

- En el caso de utilizar el módulo Intel Edison con el direccionamiento IPv6, se recomienda instalar el firmware más actual, con el fin de evitar posibles inconvenientes en cuanto al funcionamiento de la aplicación desarrollada en la red.

- Se recomienda utilizar la tarjeta Intel Edison en modo dispositivo cuando no se disponga de una fuente de alimentación externa (batería), de esta manera se podrá ahorrar energía cuando se esté programando o instalando aplicaciones en el dispositivo.

- Al momento de conectar la cámara web en la tarjeta Intel Edison, se recomienda utilizar el comando `dmesg -c`, para verificar su compatibilidad con el sistema. De esta manera se podrá ahorrar tiempo al momento de probar nuevas cámaras web con el dispositivo.

- Se recomienda utilizar un editor de texto externo al momento de realizar cualquier programación debido a que la tarjeta Intel Edison no dispone de una interfaz amigable al usuario al momento de editar algún archivo.

- Para subir los códigos desarrollados a la tarjeta Intel Edison, se recomienda utilizar el sistema de control de versiones Git que viene instalado por defecto en el sistema. De esta manera, los códigos creados para cualquier proyecto en un editor de texto externo, se pueden subir de manera sencilla a la tarjeta Intel Edison mediante el uso del Git.

- Cuando se esté transmitiendo video, se recomienda desconectar la batería de la tarjeta Intel Edison a fin de resetear todo el proceso que se esté dando en la tarjeta. Existen momentos en los cuales al querer volver iniciar la aplicación, la cámara web no responde. Esto se soluciona volviendo a conectar la batería.

REFERENCIAS BIBLIOGRÁFICAS

[1] Asociación Española de Domótica e Inmótica, «CEDOM,» [En línea]. Available: http://www.cedom.es/sobre-domotica/que-es-domotica. [Último acceso: 18 Noviembre 2016].

[2] IECOR, «Aplicaciones de la Dómotica,» [En línea]. Available: http://www.iecor.com/aplicaciones-de-la-domotica/. [Último acceso: 18 Noviembre 2016].

[3] F. J. Moliner Lopez, Grupos A y B de Informática Bloque Especifico, Sevilla: Mad, S.L., 2005.

[4] M. A. Mora, «Redes Inalámbricas,» [En línea]. Available: http://publicaciones.urbe.edu/index.php/telematique/article/view/781/1883. [Último acceso: 2 Diciembre 2016].

[5] L. Perla, «Manejo de redes 603,» [En línea]. Available: http://plgarcia.blogspot.com/2011/06/definicion.html. [Último acceso: 2 Diciembre 2016].

[6] T. Cooklev, Wireless Communication Standards A Study of IEEE 802.11, 802.15, and 802.16, IEEE Standards Association, 2004.

[7] M. Gast, 802.11 Wireless Networks: The Definitive Guide Creating and Administering Wireless Networks, O'Reilly Media, 2002.

[8] A. Escudero Pascual, «Itrainonline,» [En línea]. Available: http://www.itrainonline.org/itrainonline/mmtk/wireless_es/files/02_es_estandares-inalambricos_guia_v02.pdf. [Último acceso: 20 Noviembre 2016].

[9] BUFFALO TECHONOLOGY, «Tecnología estándar - Tecnologías 802.11 inalámbricas,» [En línea]. Available: http://www.buffalo-technology.com/es/tecnologia/software-asociado/wireless-80211-technologies/. [Último acceso: 25 Noviembre 2016].

[10] F. Milagro Lardiés y A. Los Santos Aransay, «Alvertolsa,» [En línea]. Available: http://www.albertolsa.com/wp-content/uploads/2009/07/ria-comparativa-de-ieee-80211-e-ieee80216-francisco-y-alberto.pdf. [Último acceso: 26 Noviembre 2016].

[11] B. Mitchell, «LifeWire,» 01 Octubre 2016. [En línea]. Available: https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553. [Último acceso: 25 Noviembre 2016].

[12] I. v. Beijnum, Running IPv6, Apress, 2005.

[13] IPv6Mx, «Fundamentos de IPv6,» [En línea]. Available: http://www.ipv6.mx/index.php/informacion/fundamentos/ipv6. [Último acceso: 5 Diciembre 2016].

[14] J. C. Aguirre Córdova y E. R. Pérez Vaca, «ANÁLISIS DE LA TRANSMISIÓN DE VOZ, DATOS Y VÍDEO SOBRE REDES MULTISERVICIOS UTILIZANDO EL PROTOCOLO IPv6,» 2002.

[15] M. Hachman, «PC World,» 7 Enero 2014. [En línea]. Available: http://www.pcworld.com/article/2085003/intel-ceo-points-toward-wearable-future-with-smart-earbud-smartwatch.html. [Último acceso: 29 Noviembre 2016].

[16] Software Intel, «Developer Zone,» [En línea]. Available: https://software.intel.com/en-us/iot/hardware/edison. [Último acceso: 20 Diciembre 2016].

[17] INTEL, «Intel Edison Compute Module,» Junio 2016. [En línea]. Available: http://www.intel.com/content/dam/support/us/en/documents/edison/sb/edison-module_HG_331189.pdf. [Último acceso: 27 Diciembre 2016].

[18] A. Muñoz, «¿Qué es Firmware?,» 29 Octubre 2016. [En línea]. Available: http://computerhoy.com/noticias/software/que-es-firmware-53182. [Último acceso: 15 Enero 2017].

[19] Software Intel, «Zona para desarrolladores Intel,» [En línea]. Available: https://software.intel.com/es-es/get-started-edison-windows-step2. [Último acceso: 1 Enero 2017].

[20] LINUX FOUNDATION, «Yocto Project,» [En línea]. Available: https://www.yoctoproject.org/about. [Último acceso: 1 Enero 2017].

[21] Desarrollo Web, «Manual de Nodejs,» [En línea]. Available: http://www.desarrolloweb.com/manuales/manual-nodejs.html. [Último acceso: 3 Enero 2017].

[22] Desarrollo Web, «Qué es Python,» 19 Noviembre 2003. [En línea]. Available: http://www.desarrolloweb.com/articulos/1325.php. [Último acceso: 3 Enero 2017].

[23] INTEL, «Intel Edison Breakout Board,» Enero 2015. [En línea]. Available: http://download.intel.com/support/edison/sb/edisonbreakout_hg_331190006.pdf. [Último acceso: 27 Diciembre 2016].

[24] ARDUINO, «Intel Edison,» [En línea]. Available: https://www.arduino.cc/en/ArduinoCertified/IntelEdison. [Último acceso: 30 Diciembre 2016].

[25] KIIR, «TURNING ON INTEL EDISON WITH MINI BREAKOUT BOARD,» [En línea]. Available: https://kiir.net/blog/turning-intel-edison-mini-breakout-board/. [Último acceso: 30 Diciembre 2016].

[26] INFORMATICA HOY, «Que es el sistema operativo?,» [En línea]. Available: http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-el-sistema-operativo.php. [Último acceso: 30 Diciembre 2016].

[27] «Definición de Linux,» [En línea]. Available: http://conceptodefinicion.de/linux/. [Último acceso: 31 Diciembre 2016].

[28] «Microsoft Windows,» [En línea]. Available: https://www.ecured.cu/Microsoft_Windows. [Último acceso: 31 Diciembre 2016].

[29] «Android: Principales características del sistema operativo de Google,» [En línea]. Available: http://culturacion.com/android-principales-caracteristicas-del-sistema-operativo-de-google/. [Último acceso: 30 Diciembre 2016].

[30] A. Castillo, «LA HISTORIA DE ANDROID: TODAS SUS VERSIONES,» 24 Septiembre 2015. [En línea]. Available: http://www.poderpda.com/editorial/la-historia-de-android-todas-sus-versiones/. [Último acceso: 30 Diciembre 2016].

[31] «Support connecting to IPv6-only wireless networks,» [En línea]. Available: https://code.google.com/p/android/issues/detail?id=32630. [Último acceso: 30 Diciembre 2016].

[32] «Historia de la webcam,» [En línea]. Available: http://www.escuelapedia.com/historia-de-la-webcam/. [Último acceso: 31 Diciembre 2016].

[33] «Linux UVC driver and tools,» [En línea]. Available: http://www.ideasonboard.org/uvc/. [Último acceso: 31 Diciembre 2016].

[34] IDS Imaging Development Systems GmbH, «CÁMARAS INDUSTRIALES IDS CON UVC,» [En línea]. Available: https://es.ids-imaging.com/usb-video-class.html. [Último acceso: 25 Enero 2017].

[35] DocSharing, «Cisco serie 1800, routers de servicios integrados: Comparación de modelos,» [En línea]. Available: https://docsharing.wordpress.com/2007/12/08/cisco-serie-1800-routers-de-servicios-integrados-comparacion-de-modelos/. [Último acceso: 1 Enero 2017].

[36] Anonimo, «PuTTY: a free SSH and Telnet client,» [En línea]. Available: http://www.chiark.greenend.org.uk/~sgtatham/putty/. [Último acceso: 1 Febrero 2017].

[37] DLINK, «Wireless N Home Router,» [En línea]. Available: http://www.dlink.com/es/es/home-solutions/connect/routers/dir-600-wireless-n-150-home-router. [Último acceso: 1 Enero 2017].

[38] «Ping: Cómo funciona y por qué es útil para diagnosticar problemas de red,» [En línea]. Available: https://www.xatakamovil.com/conectividad/ping-como-funciona-y-por-que-es-util-para-diagnosticar-problemas-de-red. [Último acceso: 4 Enero 2017].

[39] F.Manuel, «Ubuntu 14.04 LTS, lanzamiento oficial. Principales novedades de la familia Trusty Tahr,» [En línea]. Available: https://www.genbeta.com/linux/ubuntu-14-04-lts-lanzamiento-oficial-principales-novedades-de-la-familia-trusty-tahr. [Último acceso: 2 Febrero 2017].

[40] S. Intel, «Flash Tool Lite User Manual,» 6 Mayo 2015. [En línea]. Available: https://software.intel.com/en-us/articles/flash-tool-lite-user-manual. [Último acceso: 3 Febrero 2017].

[41] «Protocolo SSH,» [En línea]. Available: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html. [Último acceso: 10 Enero 2017].

[42] DEBIAN, «¿Qué es un gestor de paquetes?,» [En línea]. Available: https://www.debian.org/doc/manuals/aptitude/pr01s02.es.html. [Último acceso: 12 Enero 2017].

[43] «Sublime Text 3,» [En línea]. Available: https://www.sublimetext.com/3. [Último acceso: 15 Enero 2017].

[44] HIPERTEXTUAL, «Un vistazo a Node.js,» 30 Abril 2014. [En línea]. Available: https://hipertextual.com/archivo/2014/04/nodejs-javascript-servidor/. [Último acceso: 15 Enero 2017].

[45] E. J. Kim, «GitHub,» [En línea]. Available: https://github.com/drejkim/edi-cam. [Último acceso: 10 Diciembre 2016].

[46] L. Castillo, «¿Qué es GitHub?,» [En línea]. Available: http://conociendogithub.readthedocs.io/en/latest/data/introduccion/. [Último acceso: 3 Enero 2017].

[47] «¿Cómo funciona exactamente un servidor Web?,» [En línea]. Available: http://www.internetlab.es/post/628/como-funciona-exactamente-un-servidor-web/. [Último acceso: 17 Enero 2017].

[48] «WebSockets: Cómo utilizar Socket.io en tus aplicaciones web,» 24 Septiembre 2015. [En línea]. Available: https://carlosazaustre.es/blog/websockets-como-utilizar-socket-io-en-tu-aplicacion-web/. [Último acceso: 18 Enero 2017].

[49] P. Novas, «WebSockets y SocketIO,» 22 11 2012. [En línea]. Available: https://fernetjs.com/2012/11/websockets-y-socketio/. [Último acceso: 15 Marzo 2017].

[50] D. Szablewski, «JSMpg - MPEG1 Video,» [En línea]. Available: https://github.com/phoboslab/jsmpg. [Último acceso: 27 Noviembre 2016].

[51] «FFmpeg,» [En línea]. Available: https://ffmpeg.org/. [Último acceso: 5 Diciembre 2016].

[52] NCalculators, «Web Based Math, Engineering, Personal Finance Assessment Calculators & Converters,» [En línea]. Available: http://ncalculators.com/electrical/battery-life-calculator.htm. [Último acceso: 20 Enero 2017].

ANEXOS

ANEXO A: CONFIGURACIÓN DE LA RED DE TRANSPORTE IPv6

El archivo se encuentra almacenado en el CD adjunto.

ANEXO B: CÓDIGO DE LA APLICACIÓN

El código se encuentra almacenado en el CD adjunto.

ANEXO C: CÓDIGO DEL SCRIPT JSMPG

El código se encuentra almacenado en el CD adjunto.

ANEXO D: CÓDIGO DESARROLLADO PARA LA PAGINA WEB

El código se encuentra almacenado en el CD adjunto.